SAS/QC® 13.2 User's Guide

The SHEWHART Procedure

This document is an individual chapter from *SAS/QC® 13.2 User's Guide*.

The correct bibliographic citation for the complete manual is as follows: SAS Institute Inc. 2014. *SAS/QC® 13.2 User's Guide*. Cary, NC: SAS Institute Inc.

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

August 2014

Chapter 17
The SHEWHART Procedure

Contents

Introduction: SHEWHART Procedure

The Shewhart control chart is a graphical and analytical tool for deciding whether a process is in a state of statistical control. You can use the SHEWHART procedure to display many different types of control charts, including all commonly used charts for variables and attributes. In addition, you can use the SHEWHART procedure to

- create charts from either raw data (actual measurements) or summarized data
- analyze multiple process variables
- specify control limits in terms of a multiple of the standard error of the plotted summary statistic or as probability limits
- adjust control limits to compensate for unequal subgroup sizes
- estimate control limits from the data, compute control limits from specified values for population parameters (known standards), or read limits from an input data set
- create historical control charts that display distinct sets of control limits for multiple time phases
- perform tests for special causes based on runs patterns (Western Electric rules)
- estimate the process standard deviation using various methods (variable charts only)
- accept numeric-valued or character-valued subgroup variables
- display subgroups with date and time formats
- save chart statistics and control limits in output data sets
- tabulate chart statistics and control limits
- produce charts as traditional graphics, ODS Graphics output, or legacy line printer charts. Line printer charts can use special formatting characters that improve the appearance of the chart. Traditional graphics can be annotated, saved, and replayed.

Uses of Shewhart Charts

The Shewhart chart is named after Walter A. Shewhart (1891-1967), a physicist at the Bell Telephone Laboratories, who introduced the method in 1924 and elaborated upon it in his book *Economic Control of Quality of Manufactured Product*, (1931). The concepts underlying the control chart are that the natural variability in any manufacturing process can be quantified with a set of control limits and that the variation exceeding these limits signals a change in the process.

In industry, the Shewhart chart is the most commonly applied statistical quality control method for studying the variation in output from a manufacturing process. Shewhart charts are typically used to distinguish variation due to *special causes* from variation due to *common causes*. Special causes, also referred to as *assignable causes*, are local, sporadic problems such as the failure of a particular machine or a mistakenly recorded measurement. Common causes are problems inherent in the manufacturing system as a whole. Examples of common causes are inadequate product design, inherited defective material, and excessive humidity.

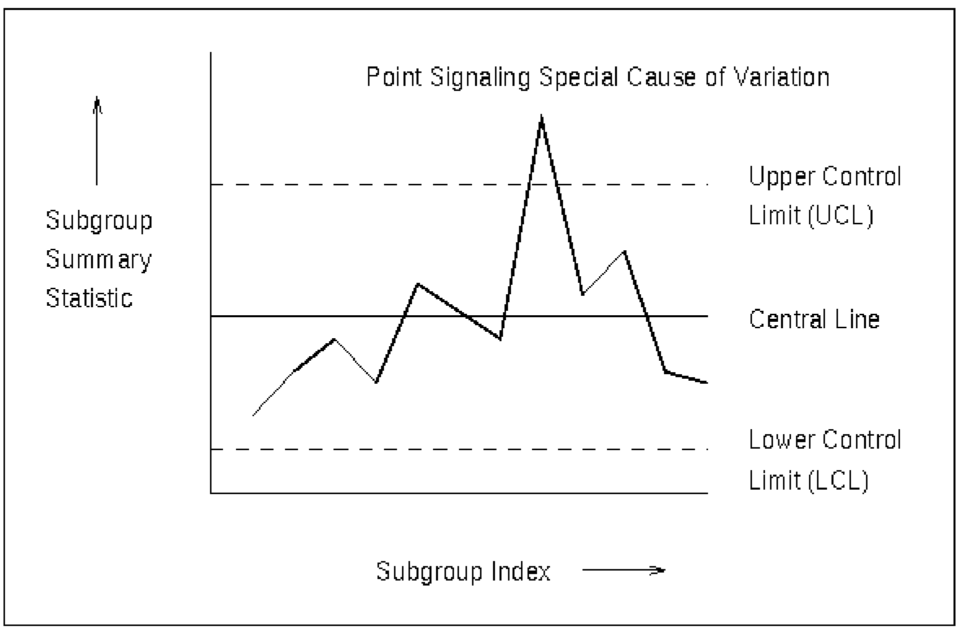
When the special causes have been identified and eliminated, the process is said to be in *statistical control*. Once statistical control has been established, Shewhart charts can be used to monitor the process for the occurrence of future special causes and to measure and reduce the effects of common causes.

Deming (1982) emphasized that the improvement of a process can begin only after statistical control has been established. Deming also noted that control chart techniques are applicable to quality improvement in service industries as well as manufacturing industries.

Characteristics of Shewhart Charts

Figure 17.1 illustrates a typical Shewhart chart.

Figure 17.1 A Shewhart Control Chart



All Shewhart charts have the following characteristics:

- Each point represents a *summary statistic* computed from a sample of measurements of a quality characteristic. For example, the summary statistic might be the average value of a critical dimension of five items selected at random, or it might be the proportion of nonconforming items in a sample of 100 items.
- The *vertical axis* of a Shewhart chart is scaled in the same units as the summary statistic.
- The samples from which the summary statistics are computed are referred to as *rational subgroups* or *subgroup samples*. The organization of the data into subgroups is critical to the interpretation of a Shewhart chart. Shewhart (1931) advocated selecting rational subgroups so that variation within

subgroups is minimized and variation among subgroups is maximized; this makes the chart more sensitive to shifts in the process level. Various approaches to subgrouping are discussed by Grant and Leavenworth (1988), Montgomery (1996), and Kume (1985).

- The *horizontal axis* of a Shewhart chart identifies the subgroup samples. Frequently, the samples are indexed according to the order in which they were taken or the time at which they were taken. Subgroup samples can also be assigned labels that indicate some other type of classification (for example, lot number).

- The *central line* on a Shewhart chart indicates the average (expected value) of the summary statistic when the process is in statistical control.

- The *upper and lower control limits*, labeled UCL and LCL, respectively, indicate the range of variation to be expected in the summary statistic when the process is in statistical control. The control limits are commonly computed as 3σ limits[1] representing three standard errors[2] of variation in the summary statistic above and below the central line. However, the limits can also be determined using a multiple of the standard error other than three, or from a specified probability (α) that a single summary statistic will exceed the limits when the process is in statistical control. Limits determined by the latter method are referred to as *probability limits*.

  The control limits are also determined by the subgroup sample size because the standard error of the summary statistic is a function of sample size. If the sample size is constant across subgroups, the control limits are typically horizontal lines, as in Figure 17.1. However, if the sample size varies from subgroup to subgroup, the limits are usually adjusted to compensate for the effect of sample size, resulting in step-like boundaries.

  Control limits can be estimated from the data being analyzed, or they can be standard, previously determined values. Estimated limits are often used when statistical control is being established, and standard limits are often used when statistical control is being maintained.

- *A point outside the control limits* signals the presence of a special cause of variation. Additionally, *tests for special causes* (also referred to as *Western Electric rules* and *runs tests*) can signal an out-of-control condition if a statistically unusual pattern of points is observed in the control chart. For example, one pattern used to diagnose the existence of a trend is seven consecutive steadily increasing points.

  When the process is in statistical control, a point may fall outside the control limits purely by chance, resulting in a false out-of-control signal. However, when the Shewhart chart correctly signals the presence of a special cause, additional action is needed to determine the nature of the problem and eliminate it.

Classification of Shewhart Charts

Shewhart charts are broadly classified according to the type of data analyzed.

[1] In this context, the symbol σ always stands for the standard error of the subgroup summary statistic that is plotted on the chart. Elsewhere in this section, σ is also used to denote the standard deviation of a process, also referred to as the population standard deviation. This dual usage is standard practice.

[2] The term *standard deviation* is also used by some authors to refer to this quantity; see, for example, Montgomery (1996). This section uses the term *standard error* for the dispersion of the distribution of a statistic and the term *standard deviation* for the dispersion of a distribution of individual measurements.

- Shewhart charts for *variables* are used when the quality characteristic of a process is measured on a continuous scale.
- Shewhart charts for *attributes* are used when the quality characteristic of a process is measured by counting the number of nonconformities (defects) in an item or the number of nonconforming (defective) items in a sample.

Shewhart charts for variables are further classified according to the subgroup summary statistic plotted on the chart.

- $\bar{X}$ and R charts display subgroup means (averages) and ranges. Typically the two charts are presented on the same page, with the $\bar{X}$ chart aligned above the R chart to facilitate the simultaneous analysis of the central tendency and variability of the process.
- $\bar{X}$ and s charts display subgroup means (averages) and standard deviations. Typically the two charts are presented on the same page, with the $\bar{X}$ chart aligned above the s chart.
- Median and range charts display subgroup medians and ranges. Typically the two charts are presented on the same page, with the median chart aligned above the R chart.
- Charts for individual measurements and moving ranges display individual measurements and moving ranges of two of more successive measurements. In this case the subgroup sample consists of a single observation.

Likewise, Shewhart charts for attributes are classified according to the subgroup summary statistic plotted on the chart:

- A p chart displays the proportion of nonconforming (defective) items in a subgroup sample.
- An np chart displays the number of nonconforming (defective) items in a subgroup sample.
- A u chart displays the number of nonconformities (defects) per unit in a subgroup sample consisting of an arbitrary number of units.
- A c chart displays the number of nonconformities (defects) in a unit (here, a subgroup sample typically consists of one unit).

You can create all of the preceding types of Shewhart charts with the SHEWHART procedure. In addition, you can create a wide variety of nonstandard Shewhart charts, including

- a trend chart displaying a time trend plot and an $\bar{X}$ chart (or median chart) that has been created removing the time trend from the data. The trend chart and $\bar{X}$ chart are presented on the same page, with the $\bar{X}$ aligned above the trend chart, to facilitate the detection of special causes after accounting for the time trend effect. Trend charts are applicable when a time trend (for instance, due to tool wear) is observed in a preliminary $\bar{X}$ chart of the original data.
- a box chart displaying a box plot (box-and-whisker plot) for each subgroup and control limits for the subgroup means. This chart facilitates detailed analysis of the subgroup distributions and is applicable with large subgroup sample sizes (ten or more).

Learning to Use the SHEWHART Procedure

Although the SHEWHART procedure provides a large number of options, you can use the procedure to create a basic Shewhart chart with as few as two SAS statements:

- the PROC SHEWHART statement, which starts the procedure and specifies the input SAS data set
- a chart statement, which specifies the type of Shewhart chart you want to create and the variables in the input data set that you want to analyze

For example, you can use the following statements to create $\bar{X}$ and R charts with 3σ limits for measurements read from a SAS data set named Drums:

```
proc shewhart data=Drums;
   xrchart Flangewidth * Hour;
run;
```

The keyword XRCHART in the chart statement specifies that $\bar{X}$ and R charts are to be created. The following SAS variables are specified in the XRCHART statement:

- A SAS variable (Flangewidth), whose values are the process measurements, is specified before the asterisk. This variable is referred to as the *process*.
- A SAS variable (Hour), whose values classify the measurements into subgroups, is specified after the asterisk. This variable is referred to as a *subgroup-variable*.

The same form of specification is used with other chart statements to create different types of Shewhart charts. The following table lists the 13 chart statements that are available with the SHEWHART procedure:

Table 17.1 Chart Statements in the SHEWHART Procedure

Statement	Chart(s) Displayed	"Getting Started" Section
BOXCHART	box chart with optional trend chart	"Getting Started: BOXCHART Statement" on page 1371
CCHART	*c* chart	"Getting Started: CCHART Statement" on page 1437
IRCHART	individual and moving range charts	"Getting Started: IRCHART Statement" on page 1474
MCHART	median chart with optional trend chart	"Getting Started: MCHART Statement" on page 1514
MRCHART	median and R charts	"Getting Started: MRCHART Statement" on page 1557
NPCHART	np chart	"Getting Started: NPCHART Statement" on page 1601
PCHART	p chart	"Getting Started: PCHART Statement" on page 1640
RCHART	R chart	"Getting Started: RCHART Statement" on page 1684
SCHART	s chart	"Getting Started: SCHART Statement" on page 1722
UCHART	u chart	"Getting Started: UCHART Statement" on page 1756
XCHART	$\bar{X}$ chart with optional trend chart	"Getting Started: XCHART Statement" on page 1795
XRCHART	$\bar{X}$ and R charts	"Getting Started: XRCHART Statement" on page 1838
XSCHART	$\bar{X}$ and s charts	"Getting Started: XSCHART Statement" on page 1887

If you are using the SHEWHART procedure for the first time, you should do the following:

- Read "PROC SHEWHART and General Statements" on page 1362.
- Read the "Getting Started" subsection in the section for the chart statement you need to create your chart. Table 17.1 provides links to these sections.

Once you have learned to use a particular chart statement, you will find it straightforward to use the remaining chart statements since their syntax is nearly the same. A separate, self-contained section is provided for each chart statement.

PROC SHEWHART and General Statements

Overview: SHEWHART Procedure

The PROC SHEWHART statement starts the SHEWHART procedure and it optionally identifies various data sets.

To create a Shewhart chart, you specify a chart statement (after the PROC SHEWHART statement) that specifies the type of Shewhart chart you want to create and the variables in the input data set that you want to analyze. For example, the following statements request $\bar{X}$ and R charts:

```
proc shewhart data=Values;
   xrchart Weight*Lot;
run;
```

Here, the DATA= option specifies an input data set (Values) with the *process* measurement variable (Weight) and the *subgroup-variable* (Lot).

You can use options in the PROC SHEWHART statement to

- specify input data sets containing variables to be analyzed, control limit information, or annotation information
- specify a graphics catalog for saving traditional graphics
- specify whether charts are to be produced as traditional graphics or line printer charts
- define characters used for features on line printer charts

See Chapter 3, "SAS/QC Graphics," for a detailed discussion of the alternatives available for producing charts with SAS/QC procedures.

NOTE: If you are learning to use the SHEWHART procedure, you should read both this section and the "Getting Started" subsection in the section for the chart statement that corresponds to the chart you want to create.

Syntax: SHEWHART Procedure

The following are the primary statements that control the SHEWHART procedure:

PROC SHEWHART < *options* > ;
BOXCHART (*processes*) * *subgroup-variable* < (*block-variables*) > < =*symbol-variable* | =*'character'* > < / *options* > ;
CCHART (*processes*) * *subgroup-variable* < (*block-variables*) > < =*symbol-variable* | =*'character'* > < / *options* > ;
IRCHART (*processes*) * *subgroup-variable* < (*block-variables*) > < =*symbol-variable* | =*'character'* > < / *options* > ;
MCHART (*processes*) * *subgroup-variable* < (*block-variables*) > < =*symbol-variable* | =*'character'* > < / *options* > ;
MRCHART (*processes*) * *subgroup-variable* < (*block-variables*) > < =*symbol-variable* | =*'character'* > < / *options* > ;
NPCHART (*processes*) * *subgroup-variable* < (*block-variables*) > < =*symbol-variable* | =*'character'* > < / *options* > ;
PCHART (*processes*) * *subgroup-variable* < (*block-variables*) > < =*symbol-variable* | =*'character'* > < / *options* > ;
RCHART (*processes*) * *subgroup-variable* < (*block-variables*) > < =*symbol-variable* | =*'character'* > < / *options* > ;
SCHART (*processes*) * *subgroup-variable* < (*block-variables*) > < =*symbol-variable* | =*'character'* > < / *options* > ;
UCHART (*processes*) * *subgroup-variable* < (*block-variables*) > < =*symbol-variable* | =*'character'* > < / *options* > ;
XCHART (*processes*) * *subgroup-variable* < (*block-variables*) > < =*symbol-variable* | =*'character'* > < / *options* > ;
XRCHART (*processes*) * *subgroup-variable* < (*block-variables*) > < =*symbol-variable* | =*'character'* > < / *options* > ;
XSCHART (*processes*) * *subgroup-variable* < (*block-variables*) > < =*symbol-variable* | =*'character'* > < / *options* > ;
INSET *keyword-list* < / *options* > ;
INSET2 *keyword-list* < / *options* > ;

The PROC SHEWHART statement invokes the procedure and specifies the input data set. The chart statements create different types of control charts. You can specify one or more of each of the chart statements. For details, read the section on the chart statement that corresponds to the type of control chart you want to produce.

In addition, you can optionally specify one of each of the following statements:

BY Statement

BY *variables* ;

You can specify a BY statement with PROC SHEWHART to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the SHEWHART procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

ID Statement

ID *variables* **;**

The ID statement specifies variables used to identify observations. The ID variables must be variables in the DATA= or HISTORY= input data sets.

The ID variables are used in the following ways:

- If you create an OUTHISTORY= or OUTTABLE= data set, the ID variables are included. If the input data set is a DATA= data set, only the values of the ID variables from the first observation in each subgroup are passed to the output data set.
- If you specify the TABLEID or TABLEALL options in a chart statement, the table produced is augmented by a column for each of the ID variables. Only the values of the ID variables from the first observation in each subgroup are tabulated. See the entry for the TABLEID option in “Dictionary of Options: SHEWHART Procedure” on page 1946.
- If you specify the BOXSTYLE=SCHEMATICID option or the BOXSTYLE= SCHEMATICIDFAR option in the BOXCHART statement, the value of the first variable listed in the ID statement is used to label each extreme observation. See Output 17.2.3 and Output 17.2.4.

Graphical Enhancement Statements

You can use TITLE, FOOTNOTE, and NOTE statements to enhance graphical and printed output. If you are creating traditional graphics, you can also use AXIS, LEGEND, and SYMBOL statements to enhance your charts. For details, refer to *SAS/GRAPH: Reference* and see the section for the control chart statement that you are using.

PROC SHEWHART Statement

The syntax for the PROC SHEWHART statement is as follows:

PROC SHEWHART < *options* > **;**

The PROC SHEWHART statement starts the SHEWHART procedure, and it optionally identifies various data sets and requests graphics output. The following section lists all *options*. See “Dictionary of Options” on page 1365 below for detailed information.

Summary of Options

The following table lists the PROC SHEWHART *options* by function:

Table 17.2 PROC SHEWHART Statement Options

Option	Description
Input Data Sets Options	
ANNOTATE=	specifies input data set containing annotation information for primary chart
ANNOTATE2=	specifies input data set containing annotation information for secondary chart
BOX=	specifies input data set containing summary statistics, control limits, and box chart outlier values
DATA=	specifies input data set containing raw data
HISTORY=	specifies input data set containing summary statistics
LIMITS=	specifies input data set containing control limits
TABLE=	specifies input data set containing summary statistics and control limits
TESTHTML=	specifies input data set defining links to be associated with subgroups with positive tests for special causes
TESTURLS=	specifies input data set containing URLs associated with subgroups with positive tests for special causes
Plotting and Graphics Options	
FORMCHAR(*index*)=	defines characters used for features on charts
GOUT=	specifies catalog for saving traditional graphics output
LINEPRINTER	requests line printer charts be produced

Dictionary of Options

The following entries provide detailed descriptions of options in the PROC SHEWHART statement.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an input data set containing Annotate variables as described in *SAS/GRAPH: Reference*. You can use this data set to add features to traditional graphics. Use this data set only when creating traditional graphics. This option is ignored if ODS Graphics is enabled or if you specify the LINEPRINTER option. Features provided in this data set are displayed on every chart produced in the current run of PROC SHEWHART.

ANNOTATE2=*SAS-data-set*

ANNO2=*SAS-data-set*

specifies an input data set that contains annotate variables. You can use this data set to add features to the secondary chart in statements that produce two charts (the IRCHART, MRCHART, XRCHART, and XSCHART statements and, when you specify the TRENDVAR= option, the BOXCHART, MCHART, and XCHART statements). The restrictions and features are the same as those for the ANNOTATE= option.

BOX=*SAS-data-set*

names an input data set that contains subgroup summary statistics, control limits, and outlier values in "strung out" form, with more than one observation per subgroup. Each observation corresponds to one feature of one subgroup's box-and-whisker plot. Typically, this data set is created as an OUTBOX= data set in a previous run of PROC SHEWHART with a BOXCHART statement. The BOX= data set is the only kind of summary data set you can use to produce schematic box-and-whisker plots. The BOXCHART statement is the only chart statement you can use with a BOX= input data set.

You cannot use a BOX= data set together with a DATA=, HISTORY=, or TABLE= data set. If you do not specify one of these four input data sets, PROC SHEWHART uses the most recently created data set as a DATA= data set.

DATA=*SAS-data-set*

names an input data set that contains raw data as observations. Note that the DATA= data set may need sorting. If the values of the *subgroup-variable* are numeric, you must sort the data set so that these values are in increasing order (within BY groups). Use PROC SORT if the data are not already sorted.

The DATA= data set may contain more than one observation for each value of the *subgroup-variable*. This happens, for example, when you produce a control chart for means and ranges with the XRCHART statement.

You cannot use a DATA= data set together with a BOX=, HISTORY=, or TABLE= data set. If you do not specify one of these four input data sets, PROC SHEWHART uses the most recently created data set as a DATA= data set. For more information, see the "DATA= Data Set" subsection in the section for the chart statement you are using.

FORMCHAR(*index*)='*string*'

defines characters used for features on legacy line printer charts, where *index* is a list of numbers ranging from 1 to 17, and *string* is a character or hexadecimal string. The *index* identifies which features are controlled with the *string* characters, as discussed in the following table. If you specify the FORMCHAR= option and omit the *index*, the *string* controls all 17 features.

Value of *index*	Description of Character	Chart Feature
1	vertical bar	frame
2	horizontal bar	frame, central line
3	box character (upper left)	frame
4	box character (upper middle)	serifs, tick (horizontal axis)
5	box character (upper right)	frame
6	box character (middle left)	not used
7	box character (middle middle)	serifs
8	box character (middle right)	tick (vertical axis)
9	box character (lower left)	frame
10	box character (lower middle)	serifs
11	box character (lower right)	frame
12	vertical bar	control limits
13	horizontal bar	control limits
14	box character (upper right)	control limits
15	box character (lower left)	control limits
16	box character (lower right)	control limits
17	box character (upper left)	control limits

Not all printers can produce all the characters in the preceding list. By default, the form character list specified with the SAS system FORMCHAR= option is used; otherwise, the default is FORMCHAR='|—-|+|—|====='. If you print to a PC screen or if your device supports the ASCII symbol set (1 or 2), the following is recommended:

```
formchar='B3,C4,DA,C2,BF,C3,C5,B4,C0,C1,D9,BA,CD,BB,C8,BC,D9'X
```

Note that the FORMCHAR= option in the PROC SHEWHART statement enables you to override temporarily the values of the SAS system option of the same name. The values of the SAS system option are not altered by using the FORMCHAR= option in the PROC SHEWHART statement.

GOUT=*graphics-catalog*

specifies the graphics catalog for traditional graphics output from PROC SHEWHART. This is useful if you want to save the output. The GOUT= option is used only when creating traditional graphics. This option is ignored if ODS Graphics is enabled or if you specify the LINEPRINTER option.

HISTORY=*SAS-data-set*

HIST=*SAS-data-set*

names an input data set that contains subgroup summary statistics. For example, you can read sample sizes, means, and ranges for the subgroups to create $\bar{X}$ and R charts. Typically, this data set is created as an OUTHISTORY= data set in a previous run of PROC SHEWHART, but it can also be created using a SAS summarization procedure such as the MEANS procedure.

Note that the HISTORY= data sets may need sorting. If the values of the *subgroup-variable* are numeric, you need to sort the data set so that these values are in increasing order (within BY groups). Use PROC SORT if the data are not already sorted. The HISTORY= data set can contain only one observation for each value for the *subgroup-variable*.

You cannot use a HISTORY= data set together with a BOX=, DATA=, or TABLE= data set. If you do not specify one of these four input data sets, PROC SHEWHART uses the most recently created data set as a DATA= data set. For more information, see the "HISTORY= Data Set" subsection in the section for the chart statement you are using.

LIMITS=*SAS-data-set*

names an input data set that contains preestablished control limits or the parameters from which control limits can be computed. Each observation in a LIMITS= data set provides control limit information for a *process*. Typically, this data set is created as an OUTLIMITS= data set in a previous run of PROC SHEWHART.

If you omit the LIMITS= option, then control limits are computed from the data in the DATA= or HISTORY= input data sets or read from the BOX= or TABLE= input data sets. For details about the variables needed in a LIMITS= data set, see the "LIMITS= Data Set" subsection in the section for the chart statement you are using.

LINEPRINTER

requests that legacy line printer charts be produced. By default, PROC SHEWHART produces ODS Graphics output if ODS Graphics is enabled and traditional graphics output if ODS Graphics is disabled and SAS/GRAPH is licensed.

TABLE=*SAS-data-set*

names an input data set that contains subgroup summary statistics and control limits. Each observation in a TABLE= data set provides information for a particular subgroup and *process*. Typically, this data set is created as an OUTTABLE= data set in a previous run of PROC SHEWHART.

You cannot use a TABLE= data set together with a BOX=, DATA=, or HISTORY= data set. If you do not specify one of these four input data sets, PROC SHEWHART uses the most recently created data set as a DATA= data set. For more information, see the "TABLE= Data Set" subsection in the section for the chart statement that you are using.

TESTHTML=*SAS-data-set*

names an input data set for creating links associated with tests for special causes when traditional graphics output is directed into HTML. A TESTHTML= data set contains variables _TEST_, _CHART_, and _URL_. _TEST_ and _CHART_ are numeric variables identifying a test for special causes (1-8) and the primary or secondary chart (1 or 2). _URL_ is a character variable containing the HTML syntax to create links associated with subgroups for which the given test on the given chart is positive. This option is ignored if you are not producing traditional graphics. See the section "Interactive Control Charts: SHEWHART Procedure" on page 2136 for more information.

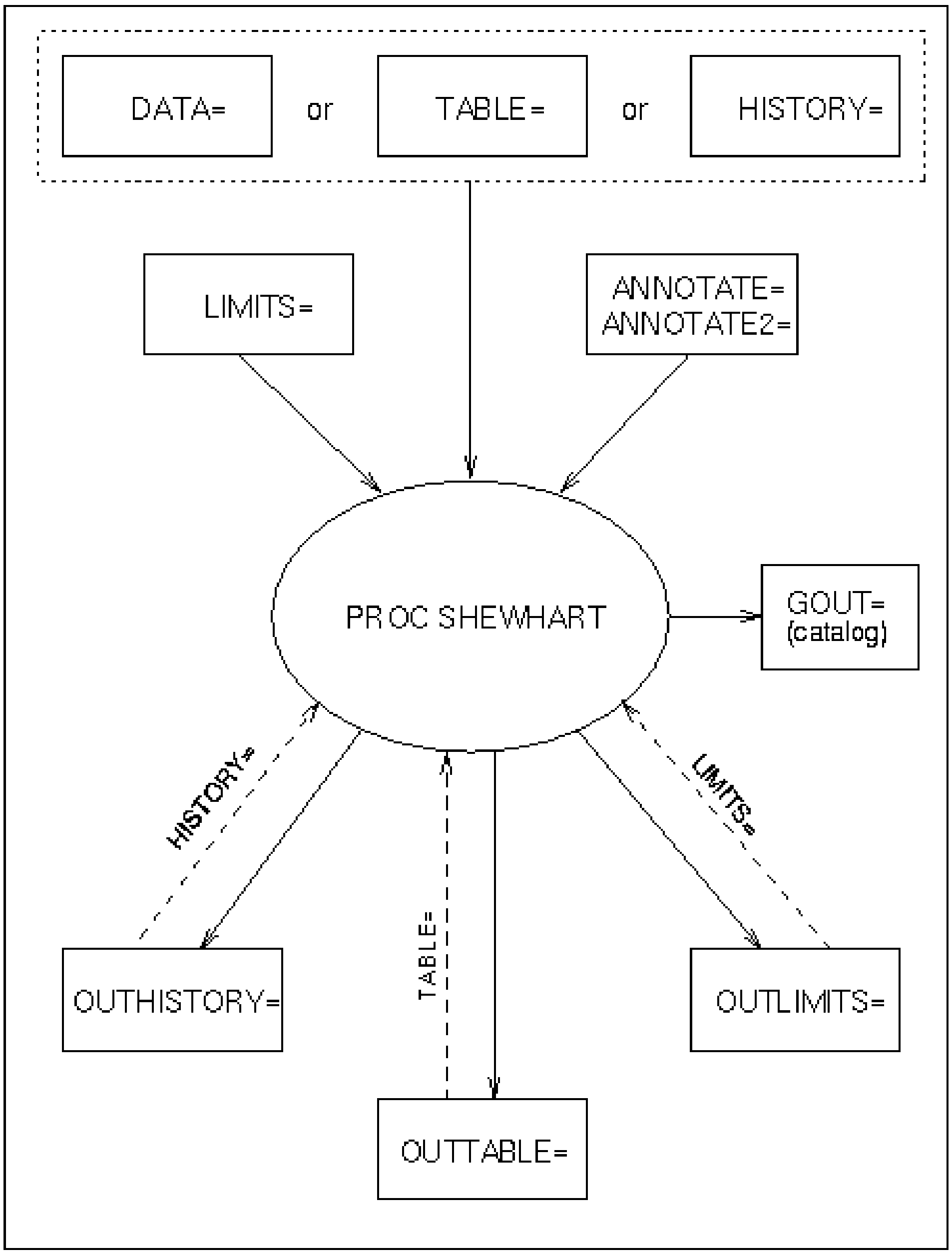
TESTURLS=*SAS-data-set*

names an input data set for associating URLs with tests for special causes when ODS Graphics output is directed into HTML. A TESTURLS= data set contains variables _TEST_, _CHART_, and _URL_. _TEST_ and _CHART_ are numeric variables identifying a test for special causes (1-8) and the primary or secondary chart (1 or 2). _URL_ is a character variable containing the URL to be associated with subgroups for which the given test on the given chart is positive. This option is ignored when ODS Graphics is disabled. See the section "Interactive Control Charts: SHEWHART Procedure" on page 2136 for more information.

Input and Output Data Sets: SHEWHART Procedure

Figure 17.2 summarizes the input and output data sets used with the SHEWHART procedure.

Figure 17.2 Input and Output Data Sets in the SHEWHART Procedure



BOXCHART Statement: SHEWHART Procedure

Overview: BOXCHART Statement

The BOXCHART statement creates an $\bar{X}$ chart for subgroup means superimposed with box-and-whisker plots of the measurements in each subgroup. Throughout this chapter, a chart of this type is referred to as a *box chart*. This chart is recommended for large subgroup sample sizes (typically greater than ten). You can also use the BOXCHART statement to create standard side-by-side box-and-whisker plots (see Example 17.2 and Example 17.3).

You can use options in the BOXCHART statement to

- specify control limits for subgroup means or medians
- compute control limits from the data based on a multiple of the standard error of the means (or medians) or as probability limits
- tabulate subgroup summary statistics and control limits
- save control limits in an output data set
- save subgroup summary statistics in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify one of several methods for estimating the process standard deviation
- specify whether subgroup standard deviations or subgroup ranges are used to estimate the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- create a secondary chart that displays a time trend removed from the data (see “Displaying Trends in Process Data” on page 2054)
- specify one of several methods for calculating quantile statistics (percentiles)
- control the style of the box-and-whisker plots
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing box charts with the BOXCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 3, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: BOXCHART Statement

This section introduces the BOXCHART statement with simple examples that illustrate commonly used options. Complete syntax for the BOXCHART statement is presented in the section "Syntax: BOXCHART Statement" on page 1385, and advanced examples are given in the section "Examples: BOXCHART Statement" on page 1416.

Creating Box Charts from Raw Data

NOTE: See *Box Chart Examples* in the SAS/QC Sample Library.

A petroleum company uses a turbine to heat water into steam that is pumped into the ground to make oil less viscous and easier to extract. This process occurs 20 times daily, and the amount of power (in kilowatts) used to heat the water to the desired temperature is recorded. The following statements create a SAS data set that contains the power output measurements for 20 days:

```
data Turbine;
   informat Day date7.;
   format Day date5.;
   label KWatts='Average Power Output';
   input Day @;
   do i=1 to 10;
      input KWatts @;
      output;
   end;
   drop i;
   datalines;
04JUL94 3196 3507 4050 3215 3583 3617 3789 3180 3505 3454
04JUL94 3417 3199 3613 3384 3475 3316 3556 3607 3364 3721
05JUL94 3390 3562 3413 3193 3635 3179 3348 3199 3413 3562
05JUL94 3428 3320 3745 3426 3849 3256 3841 3575 3752 3347
06JUL94 3478 3465 3445 3383 3684 3304 3398 3578 3348 3369
06JUL94 3670 3614 3307 3595 3448 3304 3385 3499 3781 3711
07JUL94 3448 3045 3446 3620 3466 3533 3590 3070 3499 3457
07JUL94 3411 3350 3417 3629 3400 3381 3309 3608 3438 3567
08JUL94 3568 2968 3514 3465 3175 3358 3460 3851 3845 2983
08JUL94 3410 3274 3590 3527 3509 3284 3457 3729 3916 3633
09JUL94 3153 3408 3741 3203 3047 3580 3571 3579 3602 3335
```

```
09JUL94 3494 3662 3586 3628 3881 3443 3456 3593 3827 3573
10JUL94 3594 3711 3369 3341 3611 3496 3554 3400 3295 3002
10JUL94 3495 3368 3726 3738 3250 3632 3415 3591 3787 3478
11JUL94 3482 3546 3196 3379 3559 3235 3549 3445 3413 3859
11JUL94 3330 3465 3994 3362 3309 3781 3211 3550 3637 3626
12JUL94 3152 3269 3431 3438 3575 3476 3115 3146 3731 3171
12JUL94 3206 3140 3562 3592 3722 3421 3471 3621 3361 3370
13JUL94 3421 3381 4040 3467 3475 3285 3619 3325 3317 3472
13JUL94 3296 3501 3366 3492 3367 3619 3550 3263 3355 3510
14JUL94 3795 3872 3559 3432 3322 3587 3336 3732 3451 3215
14JUL94 3594 3410 3335 3216 3336 3638 3419 3515 3399 3709
15JUL94 3850 3431 3460 3623 3516 3810 3671 3602 3480 3388
15JUL94 3365 3845 3520 3708 3202 3365 3731 3840 3182 3677
16JUL94 3711 3648 3212 3664 3281 3371 3416 3636 3701 3385
16JUL94 3769 3586 3540 3703 3320 3323 3480 3750 3490 3395
17JUL94 3596 3436 3757 3288 3417 3331 3475 3600 3690 3534
17JUL94 3306 3077 3357 3528 3530 3327 3113 3812 3711 3599
18JUL94 3428 3760 3641 3393 3182 3381 3425 3467 3451 3189
18JUL94 3588 3484 3759 3292 3063 3442 3712 3061 3815 3339
19JUL94 3746 3426 3320 3819 3584 3877 3779 3506 3787 3676
19JUL94 3727 3366 3288 3684 3500 3501 3427 3508 3392 3814
20JUL94 3676 3475 3595 3122 3429 3474 3125 3307 3467 3832
20JUL94 3383 3114 3431 3693 3363 3486 3928 3753 3552 3524
21JUL94 3349 3422 3674 3501 3639 3682 3354 3595 3407 3400
21JUL94 3401 3359 3167 3524 3561 3801 3496 3476 3480 3570
22JUL94 3618 3324 3475 3621 3376 3540 3585 3320 3256 3443
22JUL94 3415 3445 3561 3494 3140 3090 3561 3800 3056 3536
23JUL94 3421 3787 3454 3699 3307 3917 3292 3310 3283 3536
23JUL94 3756 3145 3571 3331 3725 3605 3547 3421 3257 3574
;
```

A partial listing of Turbine is shown in Figure 17.3. This data set is said to be in "strung-out" form since each observation contains the day and power output for a single heating. The first 20 observations contain the outputs for the first day, the second 20 observations contain the outputs for the second day, and so on. Because the variable Day classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable KWatts contains the output measurements and is referred to as the *process variable* (or *process* for short).

Figure 17.3 Partial Listing of the Data Set Turbine

Kilowatt Power Output Data

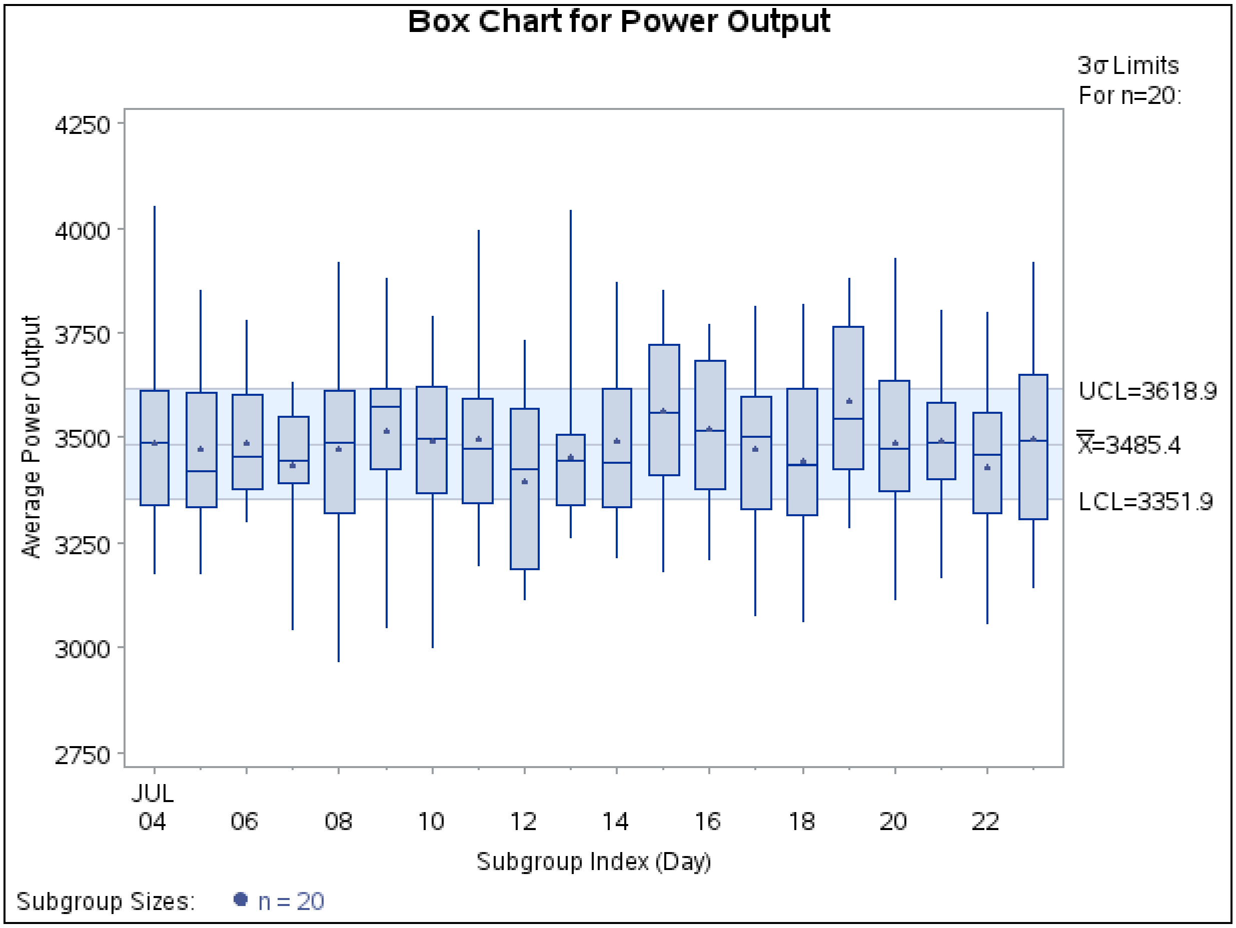
Obs	Day	KWatts
1	04JUL	3196
2	04JUL	3507
3	04JUL	4050
4	04JUL	3215
5	04JUL	3583
6	04JUL	3617
7	04JUL	3789
8	04JUL	3180
9	04JUL	3505
10	04JUL	3454
11	04JUL	3417
12	04JUL	3199
13	04JUL	3613
14	04JUL	3384
15	04JUL	3475
16	04JUL	3316
17	04JUL	3556
18	04JUL	3607
19	04JUL	3364
20	04JUL	3721
21	05JUL	3390
22	05JUL	3562
23	05JUL	3413
24	05JUL	3193
25	05JUL	3635

You can use a box chart to examine the distribution of power output for each day and to determine whether the mean level of the heating process is in control. The following statements create the box chart shown in Figure 17.4:

```
ods graphics off;
title 'Box Chart for Power Output';
symbol v=dot;
proc shewhart data=Turbine;
   boxchart KWatts*Day;
run;
```

This example illustrates the basic form of the BOXCHART statement. After the keyword BOXCHART, you specify the *process* to analyze (in this case, KWatts), followed by an asterisk and the *subgroup-variable* (Day).

Figure 17.4 Box Chart for Power Output Data (Traditional Graphics)



The input data set is specified with the DATA= option in the PROC SHEWHART statement.

By default, the BOXCHART statement requests an $\bar{X}$ chart superimposed with box-and-whisker plots for each subgroup. Table 17.3 lists the summary statistics represented by each plot. For details on the computation of percentiles, see "Percentile Definitions" on page 1414.

Table 17.3 Summary Statistics Represented by Box-and-Whisker Plots

Subgroup Summary Statistic	Feature of Box-and-Whisker Plot
Maximum	Endpoint of upper whisker
Third quartile (75th percentile)	Upper edge of box
Median (50th percentile)	Line inside box
Mean	Symbol marker (in this example, a dot)
First quartile (25th percentile)	Lower edge of box
Minimum	Endpoint of lower whisker

The within-subgroup variation in power output is stable, as indicated in Figure 17.4 by the edges of the boxes and the endpoints of the whiskers. Since the subgroup means, indicated by the dots, lie within the control limits, you can conclude that the heating process is in statistical control.

The skeletal style of the box-and-whisker plots shown in Figure 17.4 is the default. You can request different styles, as illustrated in Example 17.2. By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 17.5 and Table 17.6.

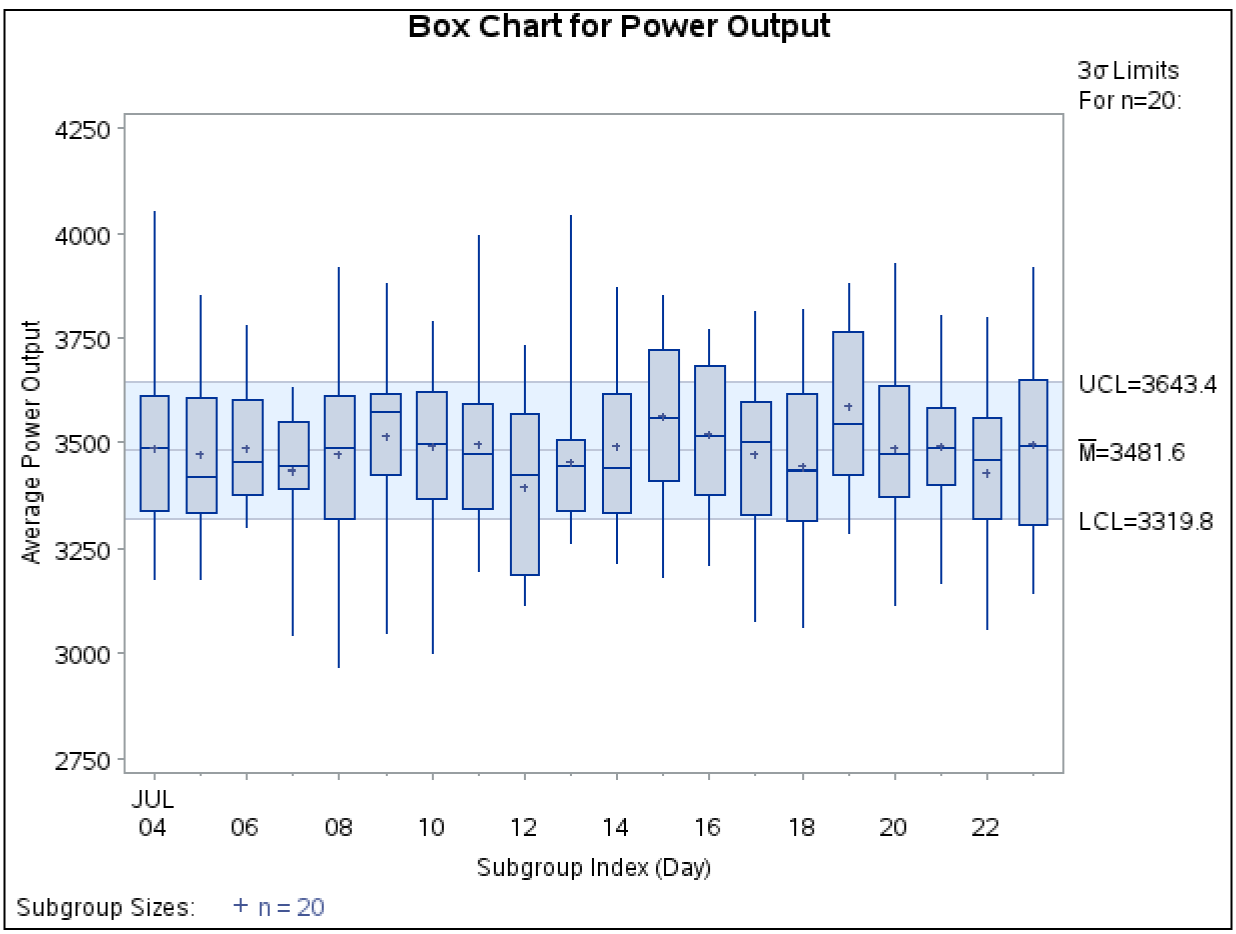
You can also create box charts in which the control limits apply to the subgroup medians. For example, the following statements create the chart shown in Figure 17.5:

```
title 'Box Chart for Power Output';
proc shewhart data=Turbine;
   boxchart KWatts*Day / controlstat = median;
run;
```

The CONTROLSTAT=MEDIAN option requests control limits that apply to the medians. Alternatively, you can specify the NOLIMITS option to suppress the display of control limits and create ordinary side-by-side box-and-whisker plots. See Example 17.2.

Options such as CONTROLSTAT= and NOLIMITS are specified after the slash (/) in the BOXCHART statement. A complete list of options is presented in the section "Syntax: BOXCHART Statement" on page 1385.

Figure 17.5 Box Chart for Power Output Data (Traditional Graphics)



Creating Box Charts from Subgroup Summary Data

NOTE: See *Box Chart Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create box charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the BOXCHART statement with data of this type.

The following data set (Oilsum) provides the data from the preceding example in summarized form. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Day).

```
data Oilsum;
   input Day KWattsL KWatts1 KWattsX KWattsM
             KWatts3 KWattsH KWattsR KWattsN;
   informat Day date7. ;
   format Day date5. ;
   label Day    ='Date of Measurement'
         KWattsL='Minimum Power Output'
         KWatts1='25th Percentile'
         KWattsX='Average Power Output'
         KWattsM='Median Power Output'
         KWatts3='75th Percentile'
         KWattsH='Maximum Power Output'
         KWattsR='Range of Power Output'
         KWattsN='Subgroup Sample Size';
   datalines;
04JUL94 3180 3340.0 3487.40 3490.0 3610.0 4050 870 20
05JUL94 3179 3333.5 3471.65 3419.5 3605.0 3849 670 20
06JUL94 3304 3376.0 3488.30 3456.5 3604.5 3781 477 20
07JUL94 3045 3390.5 3434.20 3447.0 3550.0 3629 584 20
08JUL94 2968 3321.0 3475.80 3487.0 3611.5 3916 948 20
09JUL94 3047 3425.5 3518.10 3576.0 3615.0 3881 834 20
10JUL94 3002 3368.5 3492.65 3495.5 3621.5 3787 785 20
11JUL94 3196 3346.0 3496.40 3473.5 3592.5 3994 798 20
12JUL94 3115 3188.5 3398.50 3426.0 3568.5 3731 616 20
13JUL94 3263 3340.0 3456.05 3444.0 3505.5 4040 777 20
14JUL94 3215 3336.0 3493.60 3441.5 3616.0 3872 657 20
15JUL94 3182 3409.5 3563.30 3561.0 3719.5 3850 668 20
16JUL94 3212 3378.0 3519.05 3515.0 3682.5 3769 557 20
17JUL94 3077 3329.0 3474.20 3501.5 3599.5 3812 735 20
18JUL94 3061 3315.5 3443.60 3435.0 3614.5 3815 754 20
19JUL94 3288 3426.5 3586.35 3546.0 3762.5 3877 589 20
20JUL94 3114 3373.0 3486.45 3474.5 3635.5 3928 814 20
21JUL94 3167 3400.5 3492.90 3488.0 3582.5 3801 634 20
22JUL94 3056 3322.0 3432.80 3460.0 3561.0 3800 744 20
23JUL94 3145 3308.5 3496.90 3495.0 3652.0 3917 772 20
;
```

A partial listing of Oilsum is shown in Figure 17.6.

Figure 17.6 The Summary Data Set Oilsum

Summary Data Set for Power Outputs

Day	KWattsL	KWatts1	KWattsX	KWattsM	KWatts3	KWattsH	KWattsR	KWattsN
04JUL	3180	3340.0	3487.40	3490.0	3610.0	4050	870	20
05JUL	3179	3333.5	3471.65	3419.5	3605.0	3849	670	20
06JUL	3304	3376.0	3488.30	3456.5	3604.5	3781	477	20
07JUL	3045	3390.5	3434.20	3447.0	3550.0	3629	584	20
08JUL	2968	3321.0	3475.80	3487.0	3611.5	3916	948	20

There are eight summary variables in Oilsum.

- KWattsL contains the subgroup minimums (low values).
- KWatts1 contains the 25th percentile (first quartile) for each subgroup.
- KWattsX contains the subgroup means.
- KWattsM contains the subgroup medians.
- KWatts3 contains the 75th percentile (third quartile) for each subgroup.
- KWattsH contains the subgroup maximums (high values).
- KWattsR contains the subgroup ranges.
- KWattsN contains the subgroup sample sizes.

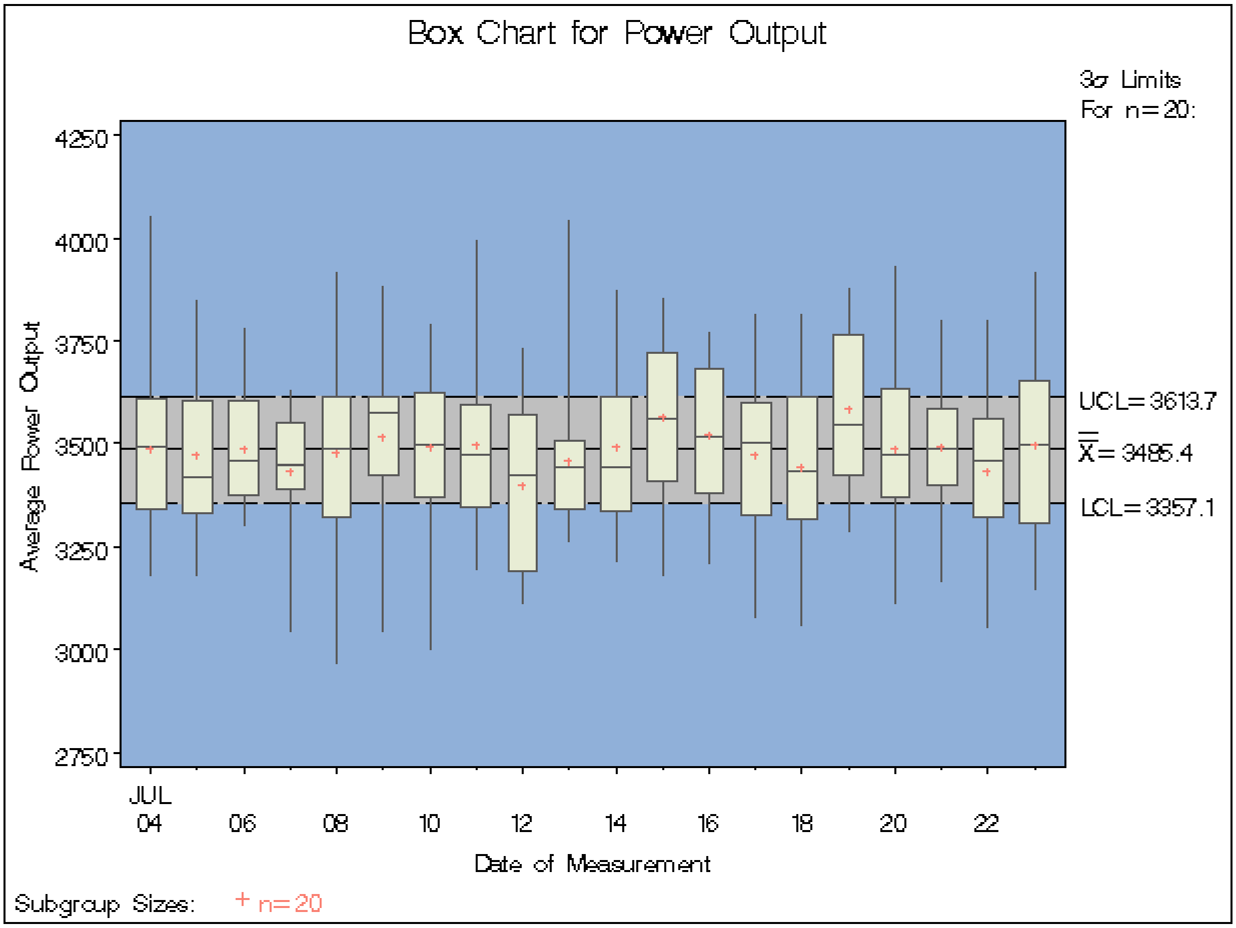
You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as illustrated by the following statements, which create the box chart shown in Figure 17.7:

```
options nogstyle;
goptions ftext=swiss;
symbol color = salmon h = .8;
title 'Box Chart for Power Output';
proc shewhart history=Oilsum;
   boxchart KWatts*Day / cinfill  = ligr
                         cboxfill = ywh
                         cboxes   = dagr
                         cframe   = vligb
                         ranges;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement and BOXCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently.

Note that the *process* KWatts is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the eight summary variables. The suffix characters *L*, *1*, *X*, *M*, *3*, *H*, *R*, and *N* indicate the contents of the variable. For example, the suffix characters *1* and *3* indicate first and third quartiles. The name Day specified after the asterisk is the name of the *subgroup-variable*.

Figure 17.7 Box Chart for Power Output Data (Traditional Graphics with NOGSTYLE)



In general, a HISTORY= input data set used with the BOXCHART statement must contain the following variables:

- subgroup variable
- subgroup minimum variable
- subgroup first quartile variable
- subgroup mean variable
- subgroup median variable
- subgroup third quartile variable
- subgroup maximum variable
- subgroup sample size variable
- either a subgroup standard deviation variable or a subgroup range variable

Furthermore, the names of the summary variables must begin with the *process* name specified in the BOXCHART statement and end with the appropriate suffix character. If the names do not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see "Creating Charts for Means and Ranges from Summary Data" on page 1841).

If you specify the RANGES option in the BOXCHART statement, the HISTORY= data set must contain a subgroup range variable; otherwise, the HISTORY= data set must contain a subgroup standard deviation variable. The RANGES option specifies that the estimate of the process standard deviation σ is to be calculated from subgroup ranges rather than subgroup standard deviations. For example, in the following statements, the data set Oilsum2 must contain a subgroup standard deviation variable named KWattsS, because the RANGES option not specified:

```
title 'Box Chart for Power Output';
proc shewhart history=Oilsum2;
   boxchart KWatts*Day;
run;
```

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1408.

Saving Summary Statistics

NOTE: See *Box Chart Examples* in the SAS/QC Sample Library.

In this example, the BOXCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Turbine and create a summary data set named Turbhist:

```
title 'Summary Data Set for Power Output';
proc shewhart data=Turbine;
   boxchart KWatts*Day / outhistory = Turbhist
                         nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 17.4.

Figure 17.8 contains a partial listing of Turbhist.

Figure 17.8 The Summary Data Set Turbhist

Summary Data Set for Power Output

Obs	Day	KWattsL	KWatts1	KWattsX	KWattsM	KWatts3	KWattsH	KWattsS	KWattsN
1	04JUL	3180	3340.0	3487.40	3490.0	3610.0	4050	220.260	20
2	05JUL	3179	3333.5	3471.65	3419.5	3605.0	3849	210.427	20
3	06JUL	3304	3376.0	3488.30	3456.5	3604.5	3781	147.025	20
4	07JUL	3045	3390.5	3434.20	3447.0	3550.0	3629	157.637	20
5	08JUL	2968	3321.0	3475.80	3487.0	3611.5	3916	258.949	20

There are nine variables in the data set Turbhist.

- Day is the subgroup variable.
- KWattsL contains the subgroup minimums.
- KWatts1 contains the first quartiles for each subgroup.
- KWattsX contains the subgroup means.
- KWattsM contains the subgroup medians.
- KWatts3 contains the third quartiles for each subgroup.
- KWattsH contains the subgroup maximums.
- KWattsS contains the subgroup standard deviations.
- KWattsN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *L*, *1*, *X*, *M*, *3*, *H*, *S*, and *N* to the *process* KWatts specified in the BOXCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

If you specify the RANGES option, the OUTHISTORY= data set includes a subgroup range variable, rather than a subgroup standard deviation variable, as demonstrated by the following statements:

```
proc shewhart data=Turbine;
   boxchart KWatts*Day / outhistory = Turbhist2
                         ranges
                         nochart;
run;
```

Figure 17.9 contains a partial listing of Turbhist2. The variable KWattsR contains the subgroup ranges.

The RANGES option is not recommended when the subgroup sample sizes are greater than 10, nor when you use the NOLIMITS option to create standard side-by-side box-and-whisker plots.

For more information, see “OUTHISTORY= Data Set” on page 1403.

Figure 17.9 The Summary Data Set Turbhist2

Summary Data Set for Power Output

Day	KWattsL	KWatts1	KWattsX	KWattsM	KWatts3	KWattsH	KWattsR	KWattsN
04JUL	3180	3340.0	3487.40	3490.0	3610.0	4050	870	20
05JUL	3179	3333.5	3471.65	3419.5	3605.0	3849	670	20
06JUL	3304	3376.0	3488.30	3456.5	3604.5	3781	477	20
07JUL	3045	3390.5	3434.20	3447.0	3550.0	3629	584	20
08JUL	2968	3321.0	3475.80	3487.0	3611.5	3916	948	20

Saving Control Limits

NOTE: See *Box Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for a box chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1384) or modify the limits with a DATA step program.

The following statements read measurements from the data set Turbine (see "Creating Box Charts from Raw Data" on page 1371) and save the control limits displayed in Figure 17.4 in a data set named TURBLIM:

```
proc shewhart data=Turbine;
   boxchart KWatts*Day / outlimits=Turblim
                         nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Turblim is listed in Figure 17.10.

Figure 17.10 The Data Set Turblim Containing Control Limit Information

Control Limits for Power Output Data

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
KWatts	Day	ESTIMATE	20	.002699796	3	3351.92	3485.41	3618.90

LCLS	_S_	_UCLS_	_STDDEV_
100.207	196.396	292.584	198.996

The data set Turblim contains one observation with the limits for *process* KWatts. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the means, and the variable _MEAN_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values.

The variables _LCLS_, _S_, and _UCLS_ are not used to create box charts, but they are included so that the data set Turblim can be used to create an *s* chart; see "XSCHART Statement: SHEWHART Procedure" on page 1886. If you specify the RANGES option in the BOXCHART statement, the variables _LCLR_, _R_, and _UCLR_, rather than the variables _LCLS_, _S_, and _UCLS_, are included in the OUTLIMITS= data set. These variables can be used to create an *R* chart; see "XRCHART Statement: SHEWHART Procedure" on page 1837.

If you specify CONTROLSTAT=MEDIAN to request control limits for medians, the variables _LCLM_ and _UCLM_, rather than the variables _LCLX_ and _UCLX_, are included in the OUTLIMITS= data set as demonstrated by the following statements:

```
proc shewhart data=Turbine;
   boxchart KWatts*Day / outlimits   = Turblim2
                         controlstat = median
                         nochart;
run;
```

Turblim2 is listed in Figure 17.11. For more information, see "OUTLIMITS= Data Set" on page 1400.

Figure 17.11 The Data Set Turblim2 Containing Control Limit Information

Control Limits for Power Output Data

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLM_	_MEAN_	_UCLM_
KWatts	Day	ESTIMATE	20	.002776264	3	3319.85	3481.63	3643.40

LCLS	_S_	_UCLS_	_STDDEV_
100.207	196.396	292.584	198.996

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
title 'Summary Statistics and Control Limit Information';
proc shewhart data=Turbine;
   boxchart KWatts*Day / outtable=Turbtab
                         nochart;
run;
```

The data set Turbtab is partially listed in Figure 17.12.

Figure 17.12 The OUTTABLE= Data Set Turbtab

Summary Statistics and Control Limit Information

VAR	Day	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_STDDEV_
KWatts	04JUL	3	20	20	3351.92	3487.40	3485.41	3618.90	198.996
KWatts	05JUL	3	20	20	3351.92	3471.65	3485.41	3618.90	198.996
KWatts	06JUL	3	20	20	3351.92	3488.30	3485.41	3618.90	198.996
KWatts	07JUL	3	20	20	3351.92	3434.20	3485.41	3618.90	198.996
KWatts	08JUL	3	20	20	3351.92	3475.80	3485.41	3618.90	198.996
KWatts	09JUL	3	20	20	3351.92	3518.10	3485.41	3618.90	198.996
KWatts	10JUL	3	20	20	3351.92	3492.65	3485.41	3618.90	198.996
KWatts	11JUL	3	20	20	3351.92	3496.40	3485.41	3618.90	198.996
KWatts	12JUL	3	20	20	3351.92	3398.50	3485.41	3618.90	198.996
KWatts	13JUL	3	20	20	3351.92	3456.05	3485.41	3618.90	198.996
KWatts	14JUL	3	20	20	3351.92	3493.60	3485.41	3618.90	198.996
KWatts	15JUL	3	20	20	3351.92	3563.30	3485.41	3618.90	198.996
KWatts	16JUL	3	20	20	3351.92	3519.05	3485.41	3618.90	198.996
KWatts	17JUL	3	20	20	3351.92	3474.20	3485.41	3618.90	198.996
KWatts	18JUL	3	20	20	3351.92	3443.60	3485.41	3618.90	198.996
KWatts	19JUL	3	20	20	3351.92	3586.35	3485.41	3618.90	198.996
KWatts	20JUL	3	20	20	3351.92	3486.45	3485.41	3618.90	198.996
KWatts	21JUL	3	20	20	3351.92	3492.90	3485.41	3618.90	198.996
KWatts	22JUL	3	20	20	3351.92	3432.80	3485.41	3618.90	198.996
KWatts	23JUL	3	20	20	3351.92	3496.90	3485.41	3618.90	198.996

EXLIM	_SUBMIN_	_SUBQ1_	_SUBMED_	_SUBQ3_	_SUBMAX_
	3180	3340.0	3490.0	3610.0	4050
	3179	3333.5	3419.5	3605.0	3849
	3304	3376.0	3456.5	3604.5	3781
	3045	3390.5	3447.0	3550.0	3629
	2968	3321.0	3487.0	3611.5	3916
	3047	3425.5	3576.0	3615.0	3881
	3002	3368.5	3495.5	3621.5	3787
	3196	3346.0	3473.5	3592.5	3994
	3115	3188.5	3426.0	3568.5	3731
	3263	3340.0	3444.0	3505.5	4040
	3215	3336.0	3441.5	3616.0	3872
	3182	3409.5	3561.0	3719.5	3850
	3212	3378.0	3515.0	3682.5	3769
	3077	3329.0	3501.5	3599.5	3812
	3061	3315.5	3435.0	3614.5	3815
	3288	3426.5	3546.0	3762.5	3877
	3114	3373.0	3474.5	3635.5	3928
	3167	3400.5	3488.0	3582.5	3801
	3056	3322.0	3460.0	3561.0	3800
	3145	3308.5	3495.0	3652.0	3917

This data set contains one observation for each subgroup sample. The variable _SUBMIN_ contains the subgroup minimums, and the variable _SUBQ1_ contains the first quartile for each subgroup. The variable _SUBX_ contains the subgroup means, and the variable _SUBMED_ contains the subgroup medians. The variable _SUBQ3_ contains the third quartiles, and the variable _SUBMAX_ contains the subgroup maximums. The variable _SUBN_ contains the subgroup sample sizes. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the means. The variable _MEAN_ contains the central line. The variables _VAR_ and Day contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1405.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Turbtab and display a box chart (not shown here) that is identical to the chart in Figure 17.4:

```
title 'Box Chart for Power Output';
proc shewhart table=Turbtab;
   boxchart KWatts*Day;
label _SUBX_ = 'Average Power Output';
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2096).

For more information, see "TABLE= Data Set" on page 1409.

Reading Preestablished Control Limits

NOTE: See *Box Chart Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set TURBLIM saved control limits computed from the measurements in Turbine. This example shows how these limits can be applied to new data. The following statements create the box chart in Figure 17.13 using new measurements in a data set named Turbine2 (not listed here) and the control limits in TURBLIM:

```
title 'Box Chart for Power Output';
ods graphics on;
proc shewhart data=Turbine2 limits=Turblim;
   boxchart KWatts*Day / odstitle=title;
run;
```

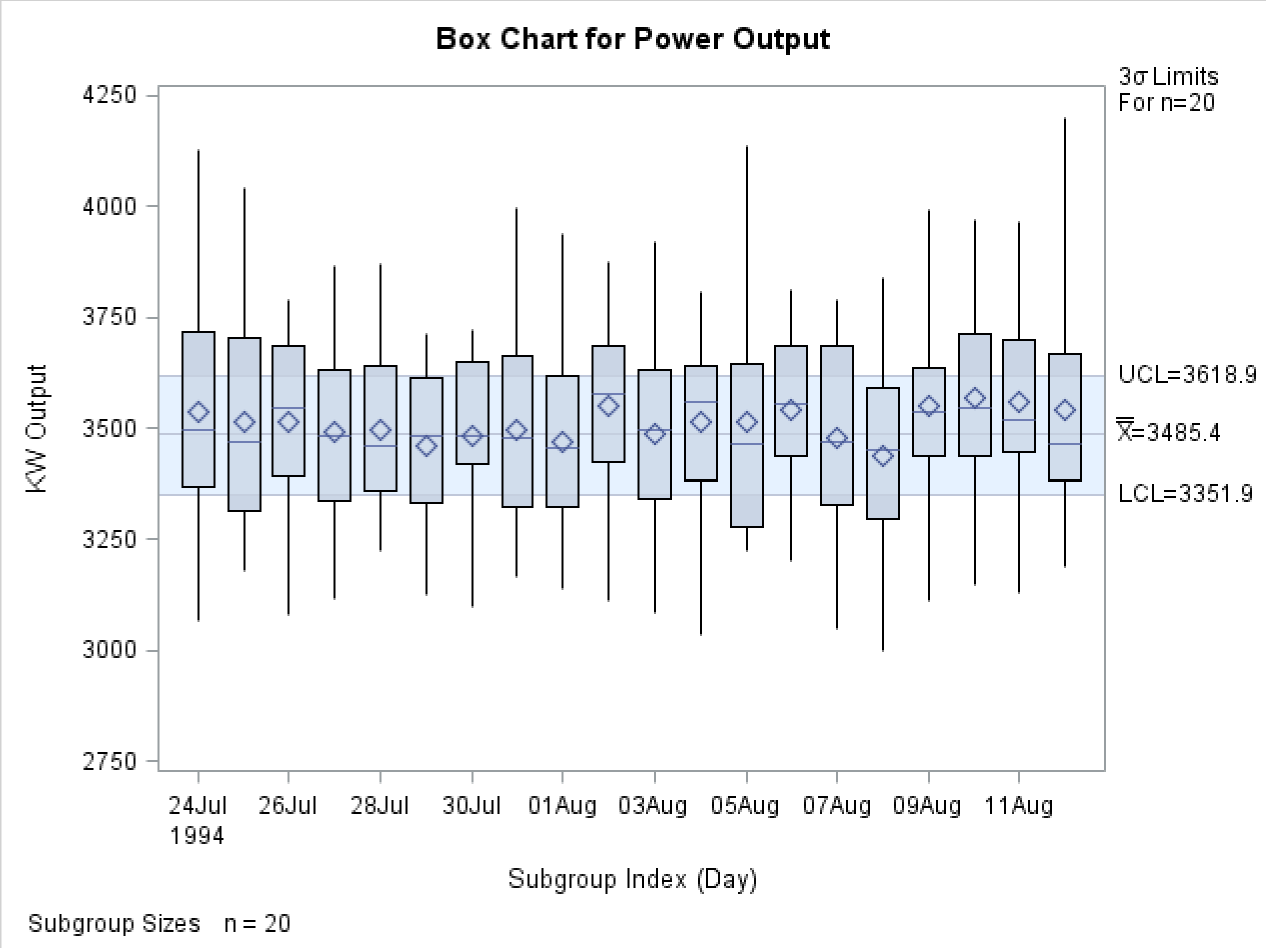
The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the box chart is created by using ODS Graphics instead of traditional graphics.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name KWatts
- the value of _SUBGRP_ matches the *subgroup-variable* name Day

The chart reveals an increase in variability beginning on August 1.

Figure 17.13 Box Chart for Second Set of Power Outputs (ODS Graphics)



In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1407 for details concerning the variables that you must provide.

Syntax: BOXCHART Statement

The basic syntax for the BOXCHART statement is as follows:

BOXCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

BOXCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=**'*character*' > / < *options* > **;**

You can use any number of BOXCHART statements in the SHEWHART procedure. The components of the BOXCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Box Charts from Raw Data" on page 1371.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Box Charts from Subgroup Summary Data" on page 1376.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1381.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct box charts for Weight, Length, and Width:

```
proc shewhart data=summary;
   boxchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding BOXCHART statement, Day is the subgroup variable. For details, see "Subgroup Variables" on page 1871.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. These blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2026 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the means.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2025 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create a box chart using an asterisk (*) to plot the means:

```
proc shewhart data=values lineprinter;
   boxchart weight*day='*';
run;
```

options

enhance the appearance of the box chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" on page 1387 section, which follows, lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1946 describes each option in detail.

Summary of Options

The following tables list the BOXCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1946.

Table 17.4 BOXCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	requests probability limits for control charts
CONTROLSTAT=	specifies whether control limits are computed for subgroup means or subgroup medians
LIMITN=	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than from a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	specifies width of control limits in terms of multiple *k* of standard error of plotted statistic
Options for Displaying Control Limits	
CINFILL=	specifies color for area inside control limits
CLIMITS=	specifies color of control limits, central line, and related labels
LCLLABEL=	specifies label for lower control limit on box chart
LIMLABSUBCHAR=	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	specifies line type for control limits
NDECIMAL=	specifies number of digits to right of decimal place in default labels for control limits and central line in box chart
NOCTL	suppresses display of central line in box chart

Table 17.4 *continued*

Option	Description
NOLCL	suppresses display of lower control limit in box chart
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit in box chart
UCLLABEL=	specifies label for upper control limit in box chart
WLIMITS=	specifies width for control limits and central line
XSYMBOL=	specifies label for central line in box chart
Process Mean and Standard Deviation Options	
MEDCENTRAL=	specifies method for estimating process mean μ
MU0=	specifies known value of μ_0 for process mean μ
RANGES	specifies that estimate of process standard deviation σ is to be calculated from subgroup ranges
SIGMA0=	specifies known value σ_0 for process standard deviation σ
SMETHOD=	specifies method for estimating process standard deviation σ
TYPE=	identifies whether parameters are estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Controlling Box Appearance	
BOXCONNECT=	connects subgroup means, medians, maximum values, minimum values, or quartiles in box-and-whisker plots
BOXSTYLE=	specifies style of box-and-whisker plots
BOXWIDTH=	specifies width of box-and-whisker plots
BOXWIDTHSCALE=	specifies that widths of box-and-whisker plots vary proportionately to subgroup sample size
CBOXES=	specifies color for outlines of box-and-whisker plots
CBOXFILL=	specifies fill color for interior of box-and-whisker plots
IDCOLOR=	specifies outlier symbol color in schematic box-and-whisker plots
IDCTEXT=	specifies text color to label outliers or process variable values
IDFONT=	specifies text font to label outliers or process variable values
IDHEIGHT=	specifies text height to label outliers or process variable values
IDSYMBOL=	specifies outlier symbol in schematic box-and-whisker plots
IDSYMBOLHEIGHT=	specifies outlier symbol height in schematic box-and-whisker plots

Table 17.4 *continued*

Option	Description
LBOXES=	specifies line types for outlines of box-and-whisker plots
NOTCHES	specifies that box-and-whisker plots are to be notched
PCTLDEF=	specifies percentile definition used for box-and-whisker plots
SERIFS	adds serifs to the whiskers of skeletal box-and-whisker plots
WHISKERPERCENTILE=	specifies that whiskers be drawn to percentile values
Options for Plotting and Labeling Points	
ALLLABEL=	labels every point on box chart
ALLLABEL2=	labels every point on trend chart
CCONNECT=	specifies color for line segments that connect points on chart
CFRAMELAB=	specifies fill color for frame around labeled points
CLABEL=	specifies color for labels
COUT=	specifies color for portions of line segments that connect points outside control limits
LABELANGLE=	specifies angle at which labels are drawn
LABELFONT=	specifies software font for labels (alias for the TEST-FONT= option)
LABELHEIGHT=	specifies height of labels (alias for the TESTHEIGHT= option)
NOCONNECT	suppresses line segments that connect points on chart
NOTRENDCONNECT	suppresses line segments that connect points on trend chart
OUTLABEL=	labels points outside control limits on box chart
SYMBOLLEGEND=	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	computes zone widths independently above and below center line
NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	suppresses tests across *phase* boundaries
TESTS=	specifies tests for special causes for the box chart
TEST2RUN=	specifies length of pattern for Test 2
TEST3RUN=	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL=	provides labels for points where test is positive
TESTLABEL*n*=	specifies label for *n*th test for special causes
TESTNMETHOD=	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points

Table 17.4 *continued*

Option	Description
TESTRESET=	enables tests for special causes to be reset for the box chart
WESTGARD=	requests that Westgard rules be applied to the box chart
ZONELABELS	adds labels A, B, and C to zone lines
ZONES	adds lines to box chart delineating zones A, B, and C
ZONEVALPOS=	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	specifies color for labels indicating points where test is positive
CTESTSYMBOL=	specifies color for symbol used to plot points where test is positive
CZONES=	specifies color for lines and labels delineating zones A, B, and C
LTESTS=	specifies type of line connecting points where test is positive
LZONES=	specifies line type for lines delineating zones A, B, and C
TESTFONT=	specifies software font for labels at points where test is positive
TESTHEIGHT=	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	specifies plot symbol for points where test is positive
TESTSYMBOLHT=	specifies symbol height for points where test is positive
WTESTS=	specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill colors for frame for plot area
CTEXT=	specifies color for tick mark values and axis labels
DISCRETE	produces horizontal axis for discrete numeric group values
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis

Table 17.4 *continued*

Option	Description
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis of box chart
VAXIS2=	specifies major tick mark values for vertical axis of trend chart
VFORMAT=	specifies format for primary vertical axis tick mark labels
VFORMAT2=	specifies format for secondary vertical axis tick mark labels
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=	specifies width of axis lines
Plot Layout Options	
ALLN	plots summary statistics for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process variable only when exceptions occur
INTERVAL=	specifies natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	specifies maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of box chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	specifies number of pages or screens to be used to display chart

Table 17.4 *continued*

Option	Description
TRENDVAR=	specifies list of trend variables
YPCT1=	specifies length of vertical axis on box chart as a percentage of sum of lengths of vertical axes for box and trend charts
ZEROSTD	displays box chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= and HREF2= options
CVREF=	specifies color for lines requested by VREF= and VREF2= options
HREF=	specifies position of reference lines perpendicular to horizontal axis on box chart
HREF2=	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFDATA=	specifies position of reference lines perpendicular to horizontal axis on box chart
HREF2DATA=	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS=	specifies labels for HREF= lines
HREF2LABELS=	specifies labels for HREF2= lines
HREFLABPOS=	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	specifies line type for HREF= and HREF2= lines
LVREF=	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to charts created for all BY groups
VREF=	specifies position of reference lines perpendicular to vertical axis on box chart
VREF2=	specifies position of reference lines perpendicular to vertical axis on trend chart
VREFLABELS=	specifies labels for VREF= lines
VREF2LABELS=	specifies labels for VREF2= lines
VREFLABPOS=	specifies position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=	specifies line type for grid requested with the ENDGRID option
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines

Table 17.4 *continued*

Option	Description
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme points are clipped
CLIPLEGEND=	specifies text for clipping legend
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features to box chart
ANNOTATE2=	specifies annotate data set that adds features to trend chart
DESCRIPTION=	specifies description of box chart's GRSEG catalog entry
FONT=	specifies software font for labels and legends on charts
NAME=	specifies name of box chart's GRSEG catalog entry
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option
WTREND=	specifies width of line segments connecting points on trend chart
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	specifies variables whose values define colors for filling background of *block-variable* legend
BOXES=	specifies variables whose values define colors box outlines
BOXFILL=	specifies variables whose values define colors for filling boxes
CFRAMELAB	draws a frame around labeled points
CPHASEBOX	requests boxes enclosing all plotted points for a phase
CPHASEBOXCONNECT	requests lines connecting adjacent enclosing boxes
CPHASEBOXFILL	fills boxes enclosing all plotted points for a phase
CPHASEMEANCONNECT	requests lines connecting phase average value points
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
BOXTRANSPARENCY=	specifies the box fill transparency for box-and-whisker charts
INFILLTRANSPARENCY=	specifies the control limit infill transparency
NOBLOCKREF	suppresses block and phase reference lines
NOBLOCKREFFILL	suppresses block and phase wall fills
NOBOXFILLLEGEND	suppresses legend for levels of a BOXFILL= variable
NOFILLLEGEND	suppresses legend for levels of a BOXFILL= variable
NOPHASEREF	suppresses block and phase reference lines
NOPHASEREFFILL	suppresses block and phase wall fills

Table 17.4 *continued*

Option	Description
NOREF	suppresses block and phase reference lines
NOREFFILL	suppresses block and phase wall fills
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	specifies a graph footnote
ODSFOOTNOTE2=	specifies a secondary graph footnote
ODSLEGENDEXPAND	specifies that legend entries contain all levels observed in the data
ODSTITLE=	specifies a graph title
ODSTITLE2=	specifies a secondary graph title
OUTHIGHURL=	specifies variable whose values are URLs to be associated with outliers above the upper fence on a schematic box chart
OUTLOWURL=	specifies variable whose values are URLs to be associated with outliers below the lower fence on a schematic box chart
OVERLAYURL=	specifies URLs to associate with overlay points
OVERLAY2URL=	specifies URLs to associate with overlay points on secondary chart
PHASEBOXLABELS	draws phase labels as titles along the top of phase boxes
PHASEPOS=	specifies vertical position of phase legend
PHASEREFLEVEL=	associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
POINTSURL=	specifies variable whose values are URLs to be associated with points representing individual observations
REFFILLTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	draws central line labels using a simulated software font
URL=	specifies a variable whose values are URLs to be associated with subgroups
URL2=	specifies a variable whose values are URLs to be associated with subgroups on secondary chart
WBOXES=	specifies width of box outlines for box-and-whisker charts
Input Data Set Options	
MISSBREAK	specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTBOX=	creates output data set containing subgroup summary statistics, control limits, and outlier values for box chart
OUTHISTORY=	creates output data set containing subgroup summary statistics
OUTINDEX=	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	creates output data set containing control limits

Table 17.4 *continued*

Option	Description
OUTTABLE=	creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	creates a basic table of subgroup values, subgroup sample sizes, subgroup summary statistics, and control limits
TABLEALL	is equivalent to the options TABLE, TABLEBOX, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUT, and TABLETEST
TABLEBOX	augments basic table with columns for minimum, 25th percentile, median, 75th percentile, and maximum of observations in a subgroup
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	specifies α value and type for computing capability index confidence limits
LSL=	specifies list of lower specification limits
TARGET=	specifies list of target values
USL=	specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for *block-variable* legend
BLOCKLABTYPE=	specifies text size of *block-variable* legend
BLOCKPOS=	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASEBOX=	specifies color for box enclosing all plotted points for a phase
CPHASEBOXCONNECT=	specifies color for line segments connecting adjacent enclosing boxes
CPHASEBOXFILL=	specifies fill color for box enclosing all plotted points for a phase
CPHASELEG=	specifies text color for *phase* legend

Table 17.4 *continued*

Option	Description
CPHASEMEANCONNECT=	specifies color for line segments connecting average value points within a phase
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE=	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEMEANSYMBOL=	specifies symbol marker for average of values within a phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	specifies *phases* to be read from an input data set
Overlay Options	
CCOVERLAY=	specifies colors for primary chart overlay line segments
CCOVERLAY2=	specifies colors for secondary chart overlay line segments
COVERLAY=	specifies colors for primary chart overlay plots
COVERLAY2=	specifies colors for secondary chart overlay plots
COVERLAYCLIP=	specifies color for clipped points on overlays
LOVERLAY=	specifies line types for primary chart overlay line segments
LOVERLAY2=	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=	specifies variables to overlay on primary chart
OVERLAY2=	specifies variables to overlay on secondary chart
OVERLAY2HTML=	specifies links to associate with secondary chart overlay points
OVERLAY2ID=	specifies labels for secondary chart overlay points
OVERLAY2SYM=	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
OVERLAYHTML=	specifies links to associate with primary chart overlay points

Table 17.4 *continued*

Option	Description
OVERLAYID=	specifies labels for primary chart overlay points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols for primary chart overlays
OVERLAYSYMHT=	specifies symbol heights for primary chart overlays
WOVERLAY=	specifies widths of primary chart overlay line segments
WOVERLAY2=	specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	specifies variable whose values create links to be associated with subgroups on primary chart
HTML2=	specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	specifies variable whose values create links to be associated with symbols in the symbol legend
OUTHIGHHTML=	specifies variable whose values create links to be associated with outliers above the upper fence on a schematic box chart
OUTLOWHTML=	specifies variable whose values create links to be associated with outliers below the lower fence on a schematic box chart
POINTSHTML=	specifies variable whose values create links to be associated with points representing individual observations
WEBOUT=	creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	specifies plot character for clipped points
CONNECTCHAR=	specifies character used to form line segments that connect points on chart
HREFCHAR=	specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	specifies characters indicating *symbol-variable*
TESTCHAR=	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	specifies line character for VREF= and VREF2= lines
ZONECHAR=	specifies character for lines that delineate zones for tests for special causes

Details: BOXCHART Statement

Constructing Box Charts

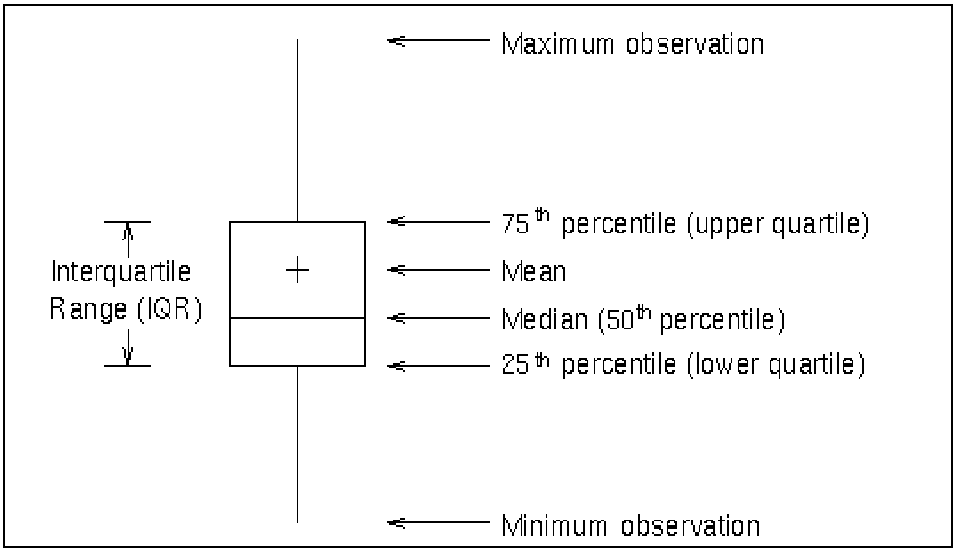
The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	mean of measurements in *i*th subgroup
n_i	sample size of *i*th subgroup
N	the number of subgroups
x_{ij}	*j*th measurement in the *i*th subgroup, $j = 1, 2, 3 \ldots, n_i$
$x_{i(j)}$	*j*th largest measurement in the *i*th subgroup: $$x_{i(1)} \le x_{i(2)} \le \ldots \le x_{i(n_i)}$$
$\overline{\overline{X}}$	weighted average of subgroup means
M_i	median of the measurements in the *i*th subgroup: $$M_i = \begin{cases} x_{i((n_i+1)/2)} & \text{if } n_i \text{ is odd} \\ (x_{i(n_i/2)} + x_{i((n_i/2)+1)})/2 & \text{if } n_i \text{ is even} \end{cases}$$
$\bar{M}$	average of the subgroup medians: $$\bar{M} = (n_1 M_1 + \ldots + n_N M_N)/(n_1 + \ldots + n_N)$$
$\tilde{M}$	median of the subgroup medians. Denote the *j*th largest median by $M_{(j)}$ so that $M_{(1)} \le M_{(2)} \le \ldots \le M_{(N)}$. $$\tilde{M} = \begin{cases} M_{((N+1)/2)} & \text{if } N \text{ is odd} \\ (M_{(N/2)} + M_{(N/2)+1})/2 & \text{if } N \text{ is even} \end{cases}$$
$e_M(n)$	standard error of the median of *n* independent, normally distributed variables with unit standard deviation (the value of $e_M(n)$ can be calculated with the STDMED function in a DATA step)
$Q_p(n)$	$100 \times p$th percentile ($0 < p < 1$) of the distribution of the median of *n* independent observations from a normal population with unit standard deviation
z_p	$100 \times p$th percentile of the standard normal distribution
$D_p(n)$	$100 \times p$th percentile of the distribution of the range of *n* independent observations from a normal population with unit standard deviation

Elements of Box-and-Whisker Plots

A box-and-whisker plot is displayed for the measurements in each subgroup on the box chart. Figure 17.14 illustrates the elements of each plot.

Figure 17.14 Box-and-Whisker Plot



The skeletal style of the box-and-whisker plot shown in Figure 17.14 is the default. You can specify alternative styles with the BOXSTYLE= option; see Example 17.2 or the entry for BOXSTYLE= in "Dictionary of Options: SHEWHART Procedure" on page 1946.

Control Limits and Central Line

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of $\bar{X}_i$ (or M_i) above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that $\bar{X}_i$ (or M_i) exceeds the limits

The CONTROLSTAT= option specifies whether control limits are computed for subgroup means (the default) or subgroup medians. The following tables provide the formulas for the limits:

Table 17.5 Control Limits and Central Line for Box Charts

CONTROLSTAT=MEAN	CONTROLSTAT=MEDIAN
LCLX = lower limit = $\overline{\overline{X}} - k\hat{\sigma}/\sqrt{n_i}$	LCLM = lower limit = $\bar{M} - k\hat{\sigma}e_M(n_i)$
Central Line = $\overline{\overline{X}}$	Central Line = $\bar{M}$
UCLX = upper limit = $\overline{\overline{X}} + k\hat{\sigma}/\sqrt{n_i}$	UCLM = upper limit = $\bar{M} + k\hat{\sigma}e_M(n_i)$

Table 17.6 Probability Limits and Central Line for Box Charts

CONTROLSTAT=MEAN	CONTROLSTAT=MEDIAN
LCLX = lower limit = $\overline{\overline{X}} - z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$	LCLM = lower limit = $\bar{M} - Q_{\alpha/2}(n_i)\hat{\sigma}$
Central Line = $\overline{\overline{X}}$	Central Line = $\bar{M}$
UCLX = upper limit = $\overline{\overline{X}} + z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$	UCLM = upper limit = $\bar{M} + Q_{1-\alpha/2}(n_i)\hat{\sigma}$

In the preceding tables, replace $\bar{M}$ with $\overline{\overline{X}}$ if you specify MEDCENTRAL=AVGMEAN in addition to CONTROLSTAT=MEDIAN. Likewise, replace $\bar{M}$ with $\tilde{M}$ if you specify MEDCENTRAL=MEDMED in addition to CONTROLSTAT=MEDIAN. If standard values μ_0 and σ_0 are available for μ and σ, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 17.5 and Table 17.6.

Note that the limits vary with n_i. The formulas for median limits assume that the data are normally distributed.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

NOTE: You can suppress the display of the control limits with the NOLIMITS option. This is useful for creating standard side-by-side box-and-whisker plots.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 17.7 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLM	lower control limit for subgroup median
LCLR	lower control limit for subgroup range
LCLS	lower control limit for subgroup standard deviation
LCLX	lower control limit for subgroup mean
LIMITN	nominal sample size associated with the control limits
LSL	lower specification limit
MEAN	process mean (value of central line on box chart)
_R_	value of central line on R chart
_S_	value of central line on s chart
SIGMAS	multiple (k) of standard error of $\bar{X}_i$ or M_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the BOXCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLM	upper control limit for subgroup median
UCLR	upper control limit for subgroup range
UCLS	upper control limit for subgroup standard deviation
UCLX	upper control limit for subgroup mean
USL	upper specification limit
VAR	*process* specified in the BOXCHART statement

Notes:

1. The variables _LCLM_ and _UCLM_ are included if you specify CONTROLSTAT=MEDIAN; otherwise, the variables _LCLX_ and _UCLX_ are included.

2. The variables _LCLR_, _R_, and _UCLR_ are included if you specify the RANGES option; otherwise, the variables _LCLS_, _S_, and _UCLS_ are included. These variables are not used to create box charts, but they enable the OUTLIMITS= data set to be used as a LIMITS= data set with the XRCHART, XSCHART, MRCHART, SCHART, and RCHART statements.

3. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLM_, _UCLM_, _LCLR_, _R_, _UCLR_, _LCLS_, _S_, and _UCLS_.

4. If the limits are defined in terms of a multiple k of the standard error of $\bar{X}_i$, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function. If the limits are defined in terms of a multiple k of the standard error of M_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - F_{med}(k, n))$, where $F_{med}(\cdot, n)$ is the cumulative distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _ALPHA_.

5. If the limits for means are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1-\alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function. If the limits for medians are probability limits, the value of _SIGMAS_ is computed as $k = F_{med}^{-1}(1-\alpha/2, n)$, where $F_{med}^{-1}(\cdot, n)$ is the inverse distribution function of the median of a random sample of n standard normally distributed observations, and n is the value _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _SIGMAS_.

6. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1874 for computational details.

7. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the BOXCHART statement. For an example, see "Saving Control Limits" on page 1381.

OUTBOX= Data Set

The OUTBOX= data set saves subgroup summary statistics, control limits, and outlier values. The following variables can be saved:

- the *subgroup-variable*
- the variable _VAR_, containing the process variable name
- the variable _TYPE_, identifying features of box-and-whisker plots
- the variable _VALUE_, containing values of box-and-whisker plot features
- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing links associated with box-and-whisker plot features

_ID_ is included in the OUTBOX= data set only if one of the keywords SCHEMATICID or SCHEMATICIDFAR is specified with the BOXSTYLE= option. _HTML_ is present only if one or more of the HTML=, OUTHIGHHTML=, OUTLOWHTML=, or POINTSHTML= options are specified.

Each observation in an OUTBOX= data set records the value of a single feature of one subgroup's box-and-whisker plot, such as its mean. The _TYPE_ variable identifies the feature whose value is recorded in _VALUE_. The following table lists valid _TYPE_ variable values:

Table 17.8 Valid _TYPE_

Value	Description
N	subgroup size
SIGMAS	multiple (k) of standard error of $\bar{X}_i$ or M_i
ALPHA	probability (α) of exceeding limits
LIMITN	nominal sample size associated with control limits
LCLM	lower control limit for subgroup median
LCLX	lower control limit for subgroup mean
UCLM	upper control limit for subgroup median
UCLX	upper control limit for subgroup mean
PROCMED	process median
PROCMEAN	process mean
EXLIM	control limit exceeded on box chart
TREND	trend variable value
MIN	minimum subgroup value
Q1	subgroup first quartile
MEDIAN	subgroup median
MEAN	subgroup mean
Q3	subgroup third quartile
MAX	subgroup maximum value
LOW	low outlier value
HIGH	high outlier value
LOWHISKR	low whisker value, if different from MIN
HIWHISKR	high whisker value, if different from MAX
FARLOW	low far outlier value
FARHIGH	high far outlier value

Additionally, the following variables, if specified, are included:

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables can be saved:

- the *subgroup-variable*
- a subgroup minimum variable named by the prefix *process* suffixed with *L*
- a subgroup first-quartile variable named by the prefix *process* suffixed with *1*

- a subgroup mean variable named by the prefix *process* suffixed with *X*
- a subgroup median variable named by the prefix *process* suffixed with *M*
- a subgroup third-quartile variable named by the prefix *process* suffixed with *3*
- a subgroup maximum variable named by the prefix *process* suffixed with *H*
- a subgroup sample size variable named by the prefix *process* suffixed with *N*
- a subgroup range variable named by the prefix *process* suffixed with *R* or a subgroup standard deviation variable named by *process* suffixed with *S*

A subgroup range variable is included if you specify the RANGES option; otherwise, a subgroup standard deviation variable is included.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the BOXCHART statement. For example, consider the following statements:

```
proc shewhart data=steel;
   boxchart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthL, Width1, WidthM, WidthX, Width3, WidthH, WidthS, WidthN, DiameterL, Diameter1, DiameterM, DiameterX, Diameter3, DiameterH, DiameterS, and DiameterN.

The variables WidthS and DiameterS are included since the RANGES option is not specified. If you specified the RANGES option, the data set Summary would contain the variables WidthR and DiameterR rather than WidthS and DiameterS.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see “Saving Summary Statistics” on page 1379.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables can be saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on box chart
LCLM	lower control limit for median
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBMAX	subgroup maximum
SUBMED	subgroup median
SUBMIN	subgroup minimum
SUBN	subgroup sample size
SUBQ1	subgroup first quartile (25th percentile)
SUBQ3	subgroup third quartile (75th percentile)
SUBX	subgroup mean
TESTS	tests for special causes signaled on box chart
UCLM	upper control limit for median
UCLX	upper control limit for mean
VAR	*process* specified in the BOXCHART statement

The variables _LCLM_ and _UCLM_ are included if you specify CONTROLSTAT=MEDIAN; otherwise, the variables _LCLX_ and _UCLX_ are included. In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- _TREND_ (if the TRENDVAR= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The kth character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.

3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1381.

ODS Tables

The following table summarizes the ODS tables that you can request with the BOXCHART statement.

Table 17.9 ODS Tables Produced with the BOXCHART Statement

Table Name	Description	Options
BOXCHART	box plot summary statistics	TABLE, TABLEALL, TABLEBOX, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*). **NOTE:** In SAS/QC 13.1 the SHEWHART procedure does not support the creation of graphs that are editable with the ODS Graphics Editor.

The appearance of a graph produced using ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. BOXCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1946.

When ODS Graphics is in effect, the BOXCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 17.10.

Table 17.10 ODS Graphics Produced by the BOXCHART Statement

ODS Graph Name	Plot Description
BoxChart	box chart

See Chapter 3, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the BOXCHART statement must be a SAS variable in the data set. This variable provides measurements which must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, specified in the BOXCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the *i*th subgroup contains n_i measurements, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the *i*th subgroup. For example, if each subgroup contains 20 items and there are 30 subgroup samples, the DATA= data set should contain 600 observations. Other variables that can be read from a DATA= data set include

- _PHASE_ (if READPHASES= is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option for an example, see "Displaying Stratification in Phases" on page 2031.

For an example of a DATA= data set, see "Creating Box Charts from Raw Data" on page 1371.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   boxchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 17.7. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLX_, _MEAN_, and _UCLX_ or (if you specify CONTROLSTAT=MEDIAN) the variables _LCLM_, _MEAN_, and _UCLM_. These variables specify the control limits directly.
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 17.5 and Table 17.6

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1384.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedures or to read output data sets created with SAS summarization procedures, such as PROC UNIVARIATE.

A HISTORY= data set used with the BOXCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup minimum variable for each *process*
- a subgroup first-quartile variable for each *process*
- a subgroup median variable for each *process*
- a subgroup mean variable for each *process*
- a subgroup third-quartile variable for each *process*
- a subgroup maximum variable for each *process*
- a subgroup sample size variable for each *process*
- either a subgroup range variable or a subgroup standard deviation variable for each *process*

If you specify the RANGES option, the subgroup range variable must be included; otherwise, the subgroup standard deviation variable must be included.

The names of the subgroup summary statistics variables must be the *process* name concatenated with the following special suffix characters:

Subgroup Summary Statistic	Suffix Character
subgroup minimum	L
subgroup first-quartile	1
subgroup median	M
subgroup mean	X
subgroup third-quartile	3
subgroup maximum	H
subgroup sample size	N
subgroup range	R
subgroup standard deviation	S

For example, consider the following statements:

```
proc shewhart history=summary;
   boxchart (weight Yieldstrength)*batch;
run;
```

The data set Summary must include the variables Batch, WeightL, Weight1, WeightM, WeightX, Weight3, WeightH, WeightS, WeightN, YieldstrengthL, Yieldstrength1, YieldstrengthM, YieldstrengthX, Yieldstrength3, YieldstrengthH, YieldstrengthS, and YieldstrengthN.

If the RANGES option were specified in the preceding BOXCHART statement, it would be necessary for Summary to include the variables WeightR and YieldstrengthR rather than WeightS and YieldstrengthS.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if READPHASES= is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (see "Displaying Stratification in Phases" on page 2031 for an example).

For an example of a HISTORY= data set, see "Creating Box Charts from Subgroup Summary Data" on page 1376.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the

SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2096.

The following table lists the variables required in a TABLE= data set used with the BOXCHART statement:

Table 17.11 Variables Required in a TABLE= Data Set

Variable	Description
LCLM	lower control limit for median
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
subgroup-variable	values of the *subgroup-variable*
SUBMAX	subgroup maximum
SUBMIN	subgroup minimum
SUBMED	subgroup median
SUBN	subgroup sample size
SUBQ1	subgroup first quartile (25th percentile)
SUBQ3	subgroup third quartile (75th percentile)
SUBX	subgroup mean
UCLM	upper control limit for median
UCLX	upper control limit for mean

Note that if you specify CONTROLSTAT=MEDIAN, the variables _LCLM_, _SUBMED_, and _UCLM_ are required; otherwise, the variables _LCLX_, _SUBX_, and _UCLX_ are required.

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1381.

BOX= Data Set

You can read summary statistics, control limits, and outlier values from a BOX= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTBOX= data set created in a previous run of the SHEWHART procedure to display a box chart.

A BOX= data set must contain the following variables:

- the group variable
- _VAR_, containing the process variable name
- _TYPE_, identifying features of box-and-whisker plots
- _VALUE_, containing values of those features

Each observation in a BOX= data set records the value of a single feature of one subgroup's box-and-whisker plot, such as its mean. The _TYPE_ variable identifies the feature whose value is recorded in a given observation. The following table lists valid _TYPE_ variable values:

Table 17.12 Valid _TYPE_ Values in a BOX= Data Set

Value	Description
N	subgroup size
SIGMAS	multiple (k) of standard error of $\bar{X}_i$ or M_i
ALPHA	probability (α) of exceeding limits
LIMITN	nominal sample size associated with control limits
LCLM	lower control limit for subgroup median
LCLX	lower control limit for subgroup mean
UCLM	upper control limit for subgroup median
UCLX	upper control limit for subgroup mean
PROCMED	process median
PROCMEAN	process mean
EXLIM	control limit exceeded on box chart
TREND	trend variable value
MIN	minimum subgroup value
Q1	subgroup first quartile
MEDIAN	subgroup median
MEAN	subgroup mean
Q3	subgroup third quartile
MAX	subgroup maximum value
LOW	low outlier value
HIGH	high outlier value
LOWHISKR	low whisker value, if different from MIN
HIWHISKR	high whisker value, if different from MAX
FARLOW	low far outlier value
FARHIGH	high far outlier value

The features identified by the _TYPE_ values N, LCLM or LCLX, UCLM or UCLX, PROCMED or PROCMEAN, MIN, Q1, MEDIAN, MEAN, Q3, and MAX are required for each subgroup.

Other variables that can be read from a BOX= data set include:

- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing links to be associated with features on box plots
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

When you specify one of the keywords SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option, values of _ID_ are used as outlier labels. If _ID_ does not exist in the BOX= data set, the values of the first variable listed in the ID statement are used.

Methods for Estimating the Standard Deviation

When control limits are computed from the input data, three methods (referred to as default, MVLUE and RMSDF) are available for estimating the process standard deviation σ. The method depends on whether you specify the RANGES option. If you specify this option, σ is estimated using subgroup ranges; otherwise, σ is estimated using subgroup standard deviations.

Default Method Based on Subgroup Standard Deviations

If you do not specify the RANGES option, the default estimate for σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \cdots + s_N/c_4(n_N)}{N}$$

where *N* is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the *i*th subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}(x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the *i*th subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of *N* unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

Default Method Based on Subgroup Ranges

If you specify the RANGES option, the default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the ith subgroup.

$$R_i = \max_{1 \leq j \leq n_i}(x_{ij}) - \min_{1 \leq j \leq n_i}(x_{ij})$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method Based on Subgroup Standard Deviations

If you do not specify the RANGES option and specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). This estimate is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

MVLUE Method Based on Subgroup Ranges

If you specify the RANGES option and SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The unbiasing factor $d_3(n_i)$ is defined so that, if the observations are normally distributed, the expected value of σ_{R_i} is $d_3(n_i)\sigma$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

RMSDF Method Based on Subgroup Standard Deviations

If you do not specify the RANGES option and specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

where $n = n_1 + \cdots + n_N - (N - 1)$. The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications, it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Percentile Definitions

You can use the PCTLDEF= option to specify one of five definitions for computing quantile statistics (percentiles). Let n equal the number of nonmissing values for a variable, and let $x_1, x_2, \ldots, x_n$ represent the ordered values of the process variable. For the tth percentile, set $p = t/100$, and express np as

$$np = j + g$$

where j is the integer part of np, and g is the fractional part of np.

The tth percentile (call it y) can be defined in five ways, as described in the next five sections.

PCTLDEF=1

This uses the weighted average at x_{np}

$$y = (1 - g)x_j + gx_{j+1}$$

where x_0 is taken to be x_1.

PCTLDEF=2

This uses the observation numbered closest to np

$$y = x_i$$

where i is the integer part of $np + 1/2$.

PCTLDEF=3

This uses the empirical distribution function

$$\begin{array}{ll} y = x_j & \text{if } g = 0 \\ y = x_{j+1} & \text{if } g > 0 \end{array}$$

PCTLDEF=4

This uses the weighted average aimed at $x_{p(n+1)}$

$$y = (1 - g)x_j + gx_{j+1}$$

where $(n + 1)p = j + g$, and where x_{n+1} is taken to be x_n.

PCTLDEF=5

This uses the empirical distribution function with averaging

$$\begin{array}{ll} y = (x_j + x_{j+1})/2 & \text{if } g = 0 \\ y = x_{j+1} & \text{if } g > 0 \end{array}$$

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical (box chart)	DATA=	*process*
Vertical (box chart)	HISTORY=	subgroup mean variable
Vertical (box chart)	TABLE=	_SUBX_

Note that if you specify the CONTROLSTAT=MEDIAN option, you should assign the label to the subgroup median variable in a HISTORY= data set or to the variable _SUBMED_ in an TABLE= data set.

If you specify the TRENDVAR= option, you can provide distinct labels for the vertical axes of the box and trend charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the box chart, and the second part labels the vertical axis of the trend chart.

For an example, see “Labeling Axes” on page 2063.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: BOXCHART Statement

This section provides advanced examples of the BOXCHART statement.

Example 17.1: Using Box Charts to Compare Subgroups

NOTE: See *Using Box Charts to Compare Subgroups* in the SAS/QC Sample Library.

In this example, a box chart is used to compare the delay times for airline flights during the Christmas holidays with the delay times prior to the holiday period. The following statements create a data set named Times with the delay times in minutes for 25 flights each day. When a flight is cancelled, the delay is recorded as a missing value.

```
data Times;
   informat Day date7. ;
   format   Day date7. ;
   input Day @ ;
   do Flight=1 to 25;
      input Delay @ ;
      output;
   end;
   datalines;
16DEC88   4  12   2   2  18   5   6  21   0   0
          0  14   3   .   2   3   5   0   6  19
          7   4   9   5  10
17DEC88   1  10   3   3   0   1   5   0   .   .
          1   5   7   1   7   2   2  16   2   1
          3   1  31   5   0
18DEC88   7   8   4   2   3   2   7   6  11   3
          2   7   0   1  10   2   3  12   8   6
          2   7   2   4   5
19DEC88  15   6   9   0  15   7   1   1   0   2
          5   6   5  14   7  20   8   1  14   3
         10   0   1  11   7
20DEC88   2   1   0   4   4   6   2   2   1   4
          1  11   .   1   0   6   5   5   4   2
          2   6   6   4   0
21DEC88   2   6   6   2   7   7   5   2   5   0
          9   2   4   2   5   1   4   7   5   6
          5   0   4  36  28
22DEC88   3   7  22   1  11  11  39  46   7  33
         19  21   1   3  43  23   9   0  17  35
         50   0   2   1   0
23DEC88   6  11   8  35  36  19  21   .   .   4
          6  63  35   3  12  34   9   0  46   0
          0  36   3   0  14
24DEC88  13   2  10   4   5  22  21  44  66  13
          8   3   4  27   2  12  17  22  19  36
          9  72   2   4   4
25DEC88   4  33  35   0  11  11  10  28  34   3
         24   6  17   0   8   5   7  19   9   7
```

```
         21  17  17   2   6
26DEC88   3   8   8   2   7   7   8   2   5   9
          2   8   2  10  16   9   5  14  15   1
         12   2   2  14  18
;
```

First, the MEANS procedure is used to count the number of cancelled flights for each day. This information is then added to the data set Times.

```
proc means data=Times noprint;
   var Delay;
   by Day ;
   output out=Cancel nmiss=Ncancel;

data Times;
   merge Times cancel;
   by Day;
run;
```

The following statements create a data set named Weather that contains information about possible causes for delays. This data set is merged with the data set Times.

```
data Weather;
   informat Day date7. ;
   format   Day date7. ;
   length Reason $ 16 ;
   input Day Flight Reason & ;
   datalines;
16DEC88  8   Fog
17DEC88  18  Snow Storm
17DEC88  23  Sleet
21DEC88  24  Rain
21DEC88  25  Rain
22DEC88  7   Mechanical
22DEC88  15  Late Arrival
24DEC88  9   Late Arrival
24DEC88  22  Late Arrival
;

data Times;
   merge Times Weather;
   by Day Flight;
run;
```

Next, control limits are established using the delays prior to the holiday period.

```
   proc shewhart data=Times;
      where Day <= '21DEC88'D;
      boxchart Delay * Day /
         nochart
         outlimits=Timelim;
   run;
```

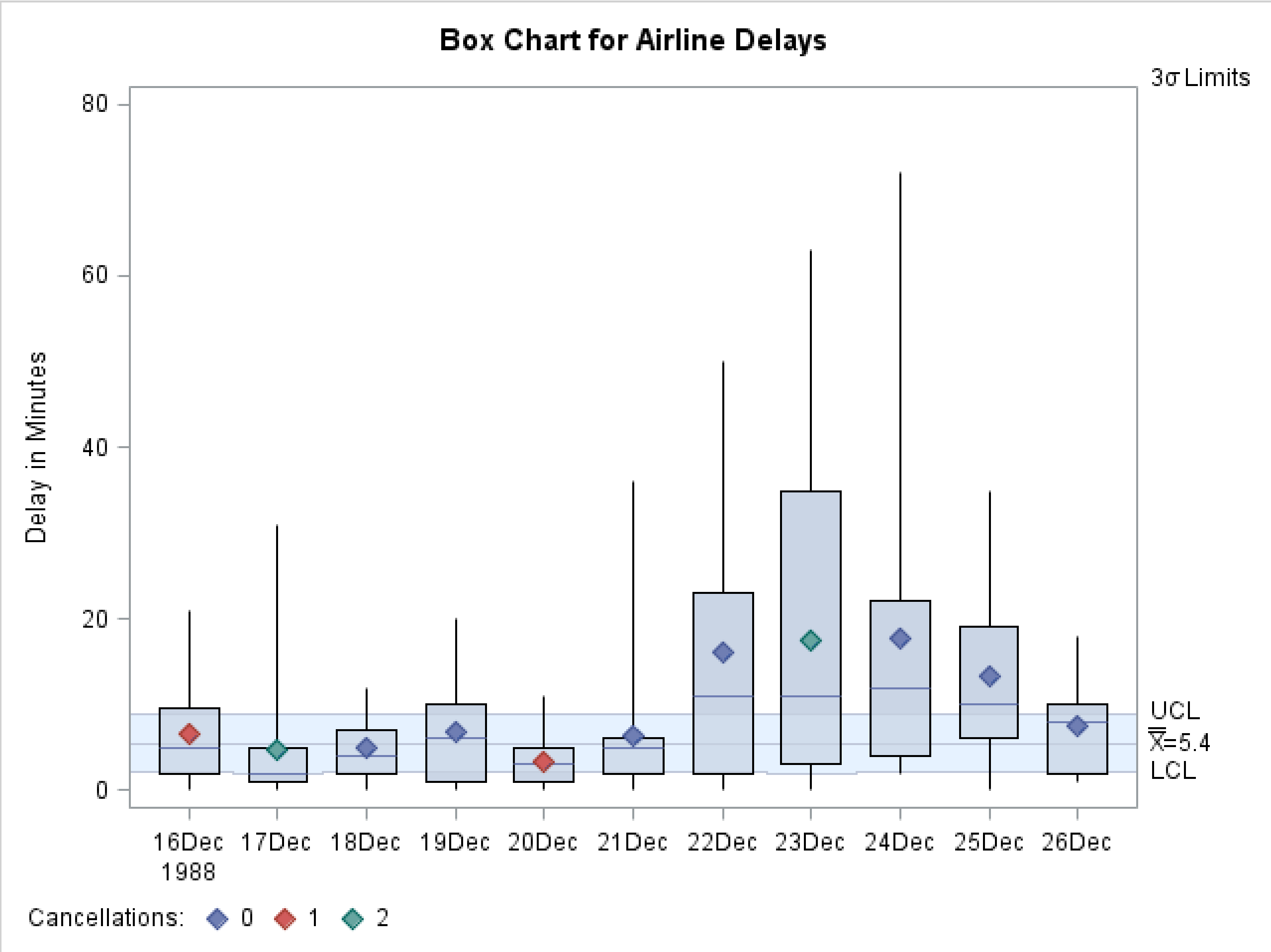
The OUTLIMITS= option names a data set (Timelim) that saves the control limits. The NOCHART option suppresses the display of the chart.

The following statements create a box chart for the complete set of data using the control limits in Timelim:

```
ods graphics on;
title 'Box Chart for Airline Delays';
proc shewhart data=Times limits=Timelim ;
   boxchart Delay * Day = Ncancel /
      readlimits
      nohlabel
      nolegend
      odstitle = title;
   label Delay   = 'Delay in Minutes'
         Ncancel = 'Cancellations:';
run;
```

The box chart is shown in Output 17.1.1. The level of the *symbol-variable* Ncancel determines the symbol marker for each subgroup mean, and the SYMBOLLEGEND= option controls the appearance of the legend for the symbols. The NOHLABEL option suppresses the label for the horizontal axis, and the NOLEGEND option suppresses the default legend for subgroup sample sizes.

Output 17.1.1 Box Chart for Airline Data



The delay distributions from December 22 through December 25 are drastically different from the delay distributions during the pre-holiday period. Both the mean delay and the variability of the delays are much greater during the holiday period.

Example 17.2: Creating Various Styles of Box-and-Whisker Plots

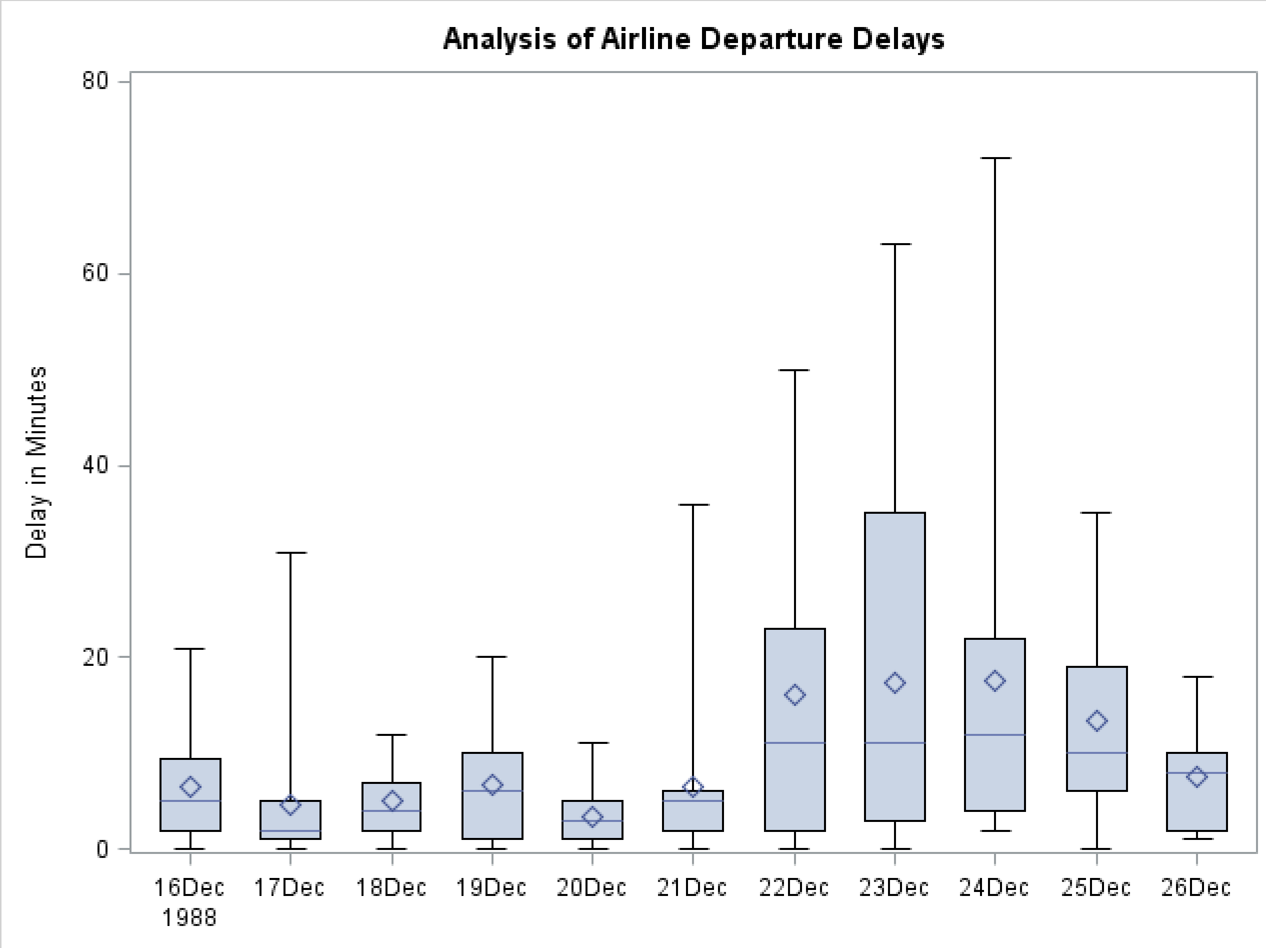
NOTE: See *Creating Various Styles of Box Charts* in the SAS/QC Sample Library.

This example uses the flight delay data of the preceding example to illustrate how you can create box charts with various styles of box-and-whisker plots. For simplicity, the control limits are suppressed. The following statements create a chart, shown in Output 17.2.1, that displays *skeletal box-and-whisker plots*:

```
ods graphics on;
title 'Analysis of Airline Departure Delays';
proc shewhart data=Times limits=Timelim ;
   boxchart Delay * Day /
      odstitle = title
      boxstyle = skeletal
      serifs
      nolimits
      nohlabel
      nolegend;
   label Delay = 'Delay in Minutes';
run;
```

In a skeletal box-and-whisker plot, the whiskers are drawn from the quartiles to the extreme values of the subgroup sample. You can also request this style by omitting the BOXSTYLE= option, since this style is the default. The SERIFS option adds serifs to the whiskers (by default, serifs are omitted with the skeletal style). The NOLIMITS option suppresses the display of the control limits.

Output 17.2.1 BOXSTYLE=SKELETAL with Serifs

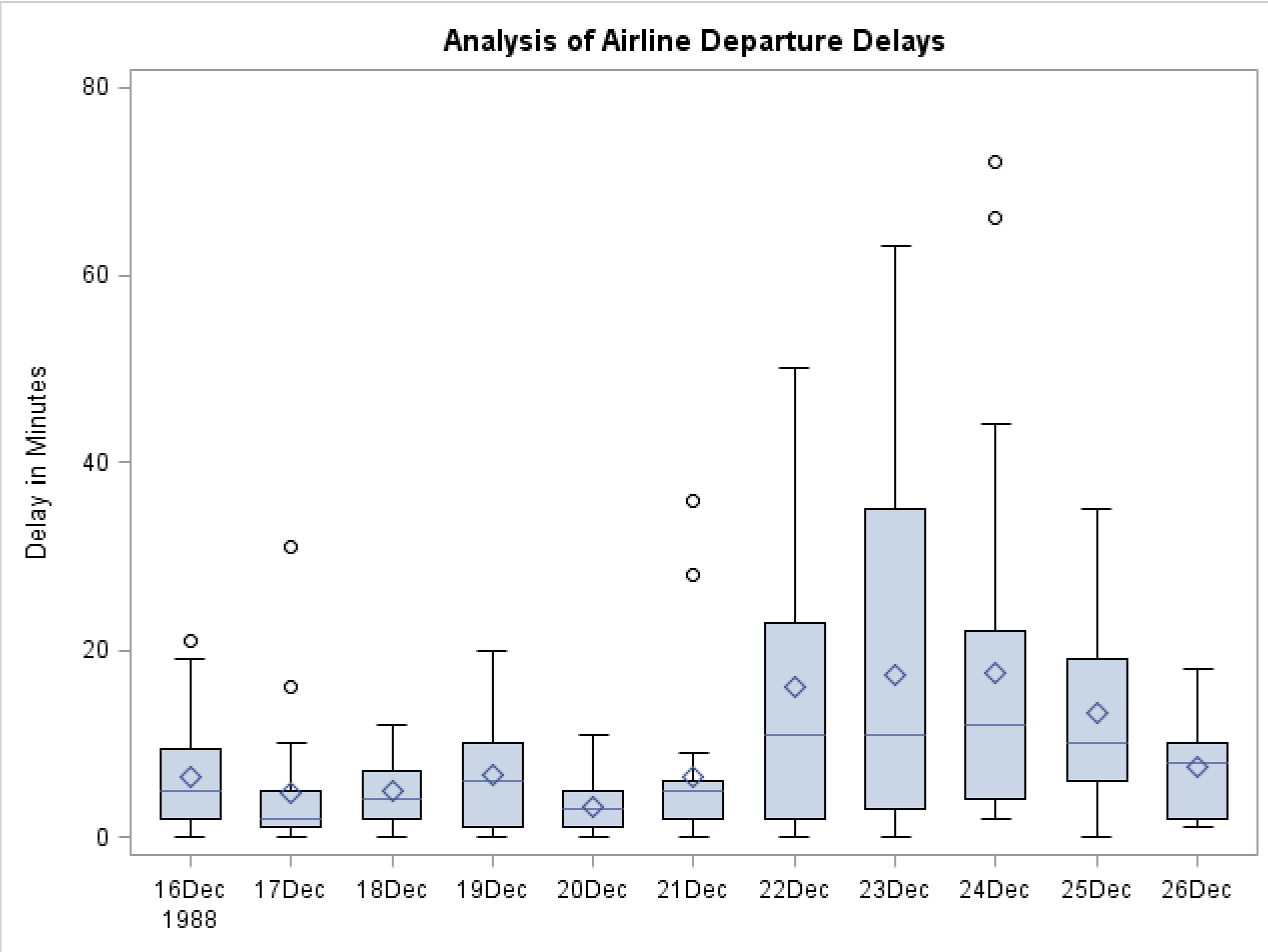


The following statements request a box chart with *schematic box-and-whisker plots*:

```
title 'Analysis of Airline Departure Delays';
proc shewhart data=Times limits=Timelim ;
   boxchart Delay * Day /
      odstitle = title
      boxstyle = schematic
      nolimits
      nohlabel
      nolegend;
   label Delay = 'Delay in Minutes';
run;
```

The chart is shown in Output 17.2.2. When BOXSTYLE=SCHEMATIC is specified, the whiskers are drawn to the most extreme points in the subgroup sample that lie within or equal to so-called “fences.” The *upper fence* is defined as the third quartile (represented by the upper edge of the box) plus 1.5 times the interquartile range (IQR). The *lower fence* is defined as the first quartile (represented by the lower edge of the box) minus 1.5 times the interquartile range. Observations outside the fences are identified with a special symbol. The default symbol is a square, and you can specify the shape and color for this symbol with the IDSYMBOL= and IDCOLOR= options. Serifs are added to the whiskers by default. For further details, see the entry for BOXSTYLE= in “Dictionary of Options: SHEWHART Procedure” on page 1946.

Output 17.2.2 BOXSTYLE=SCHEMATIC

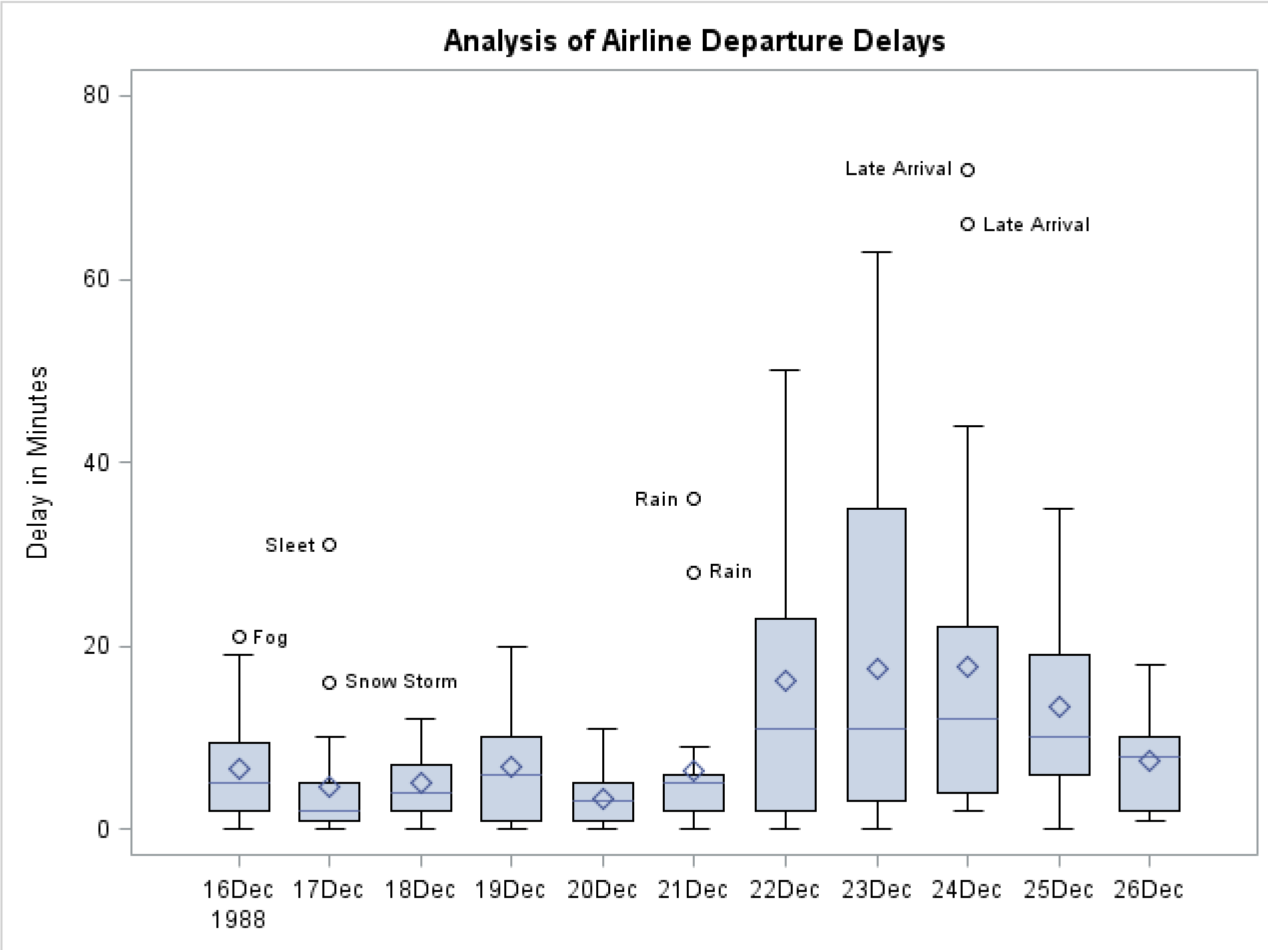


The following statements create a box chart with schematic box-and-whisker plots in which the observations outside the fences are labeled:

```
title 'Analysis of Airline Departure Delays';
proc shewhart data=Times limits=Timelim ;
   boxchart Delay * Day /
      odstitle = title
      boxstyle = schematicid
      llimits  = 20
      nolimits
      nohlabel
      nolegend;
   id Reason;
   label Delay = 'Delay in Minutes';
run;
```

The chart is shown in Output 17.2.3. If you specify BOXSTYLE=SCHEMATICID, schematic box-and-whisker plots are displayed in which the value of the first ID variable (in this case, Reason) is used to label each observation outside the fences.

Output 17.2.3 BOXSTYLE=SCHEMATICID

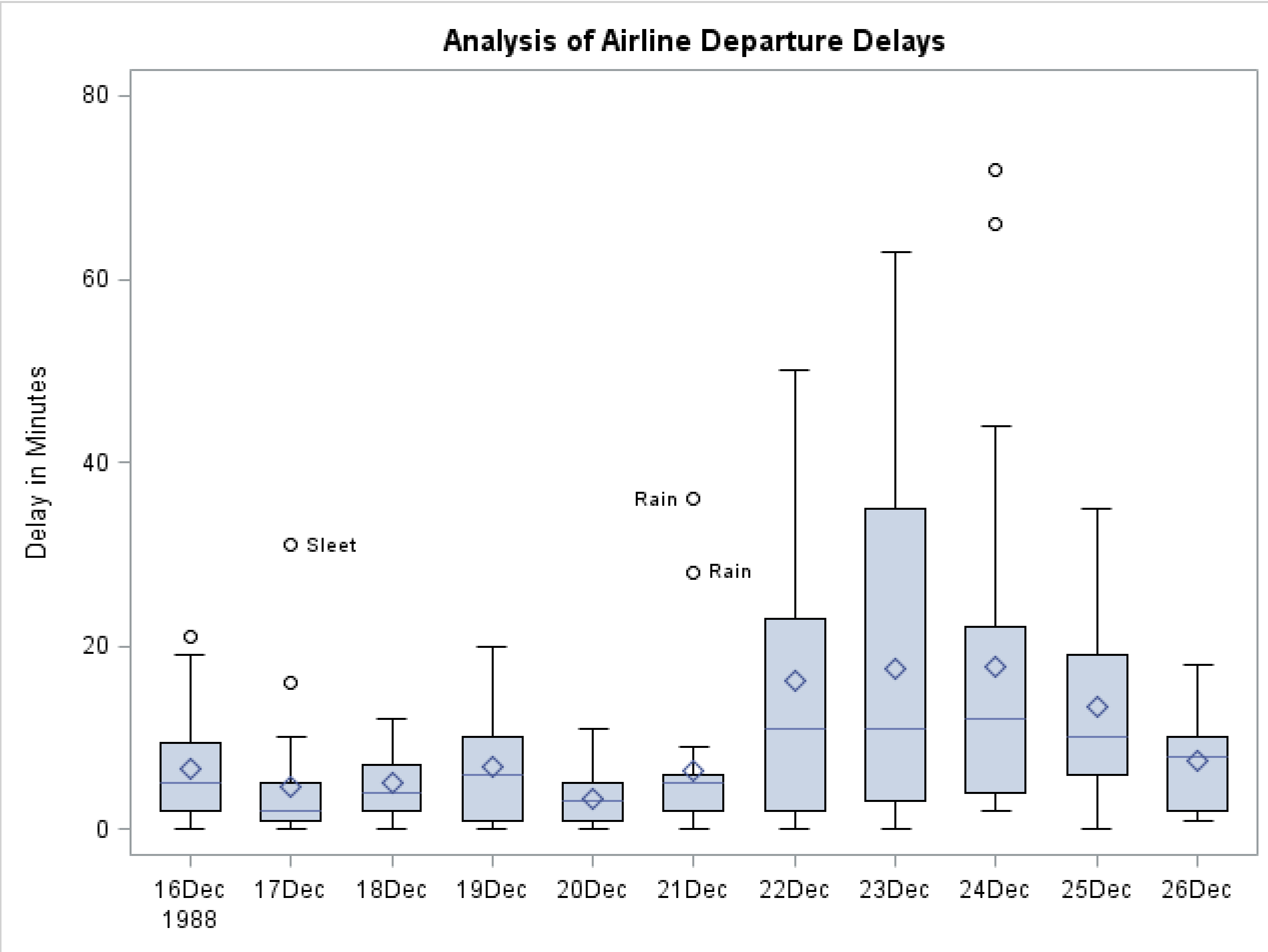


The following statements create a box chart with schematic box-and-whisker plots in which only the extreme observations outside the fences are labeled:

```
title 'Analysis of Airline Departure Delays';
proc shewhart data=Times limits=Timelim ;
   boxchart Delay * Day /
      odstitle = title
      boxstyle = schematicidfar
      nolimits
      nohlabel
      nolegend;
   id Reason;
   label Delay = 'Delay in Minutes';
run;
```

The chart is shown in Output 17.2.4. If you specify BOXSTYLE=SCHEMATICIDFAR, schematic box-and-whisker plots are displayed in which the value of the first ID variable is used to label each observation outside the *lower* and *upper far fences*. The *lower* and *upper far fences* are located 3×IQR below the 25th percentile and above the 75th percentile, respectively. Observations between the fences and the far fences are identified with a symbol but are not labeled.

Output 17.2.4 BOXSTYLE=SCHEMATICIDFAR



Other options for controlling the display of box-and-whisker plots include the BOXWIDTH=, BOXWIDTHSCALE=, CBOXES=, CBOXFILL=, and LBOXES= options. For details, see the corresponding entries in "Dictionary of Options: SHEWHART Procedure" on page 1946.

Example 17.3: Creating Notched Box-and-Whisker Plots

Note: See *Using Box Charts to Compare Subgroups* in the SAS/QC Sample Library.

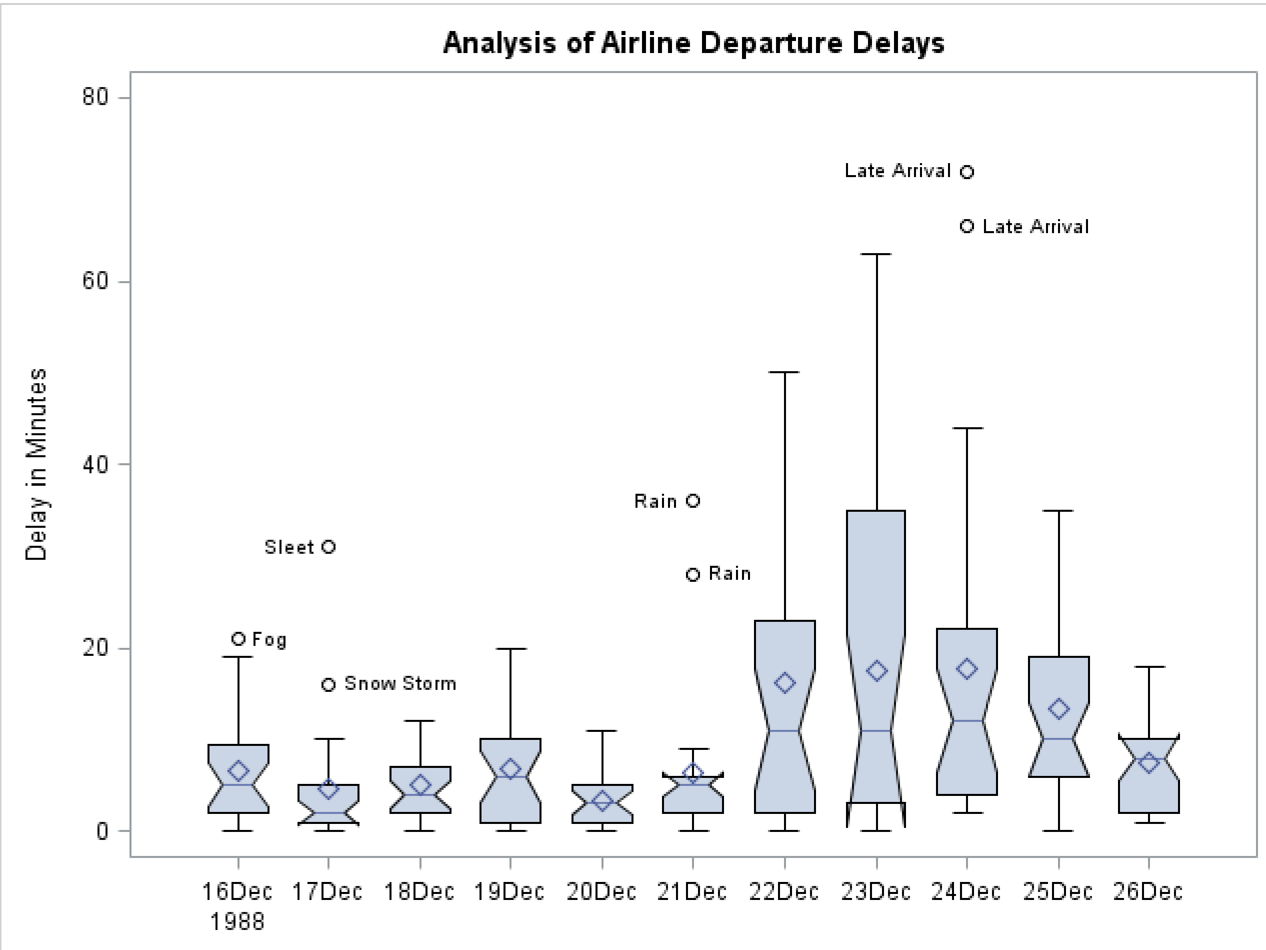
The following statements use the flight delay data of Example 17.1 to illustrate how to create side-by-side box-and-whisker plots with notches:

```
title 'Analysis of Airline Departure Delays';
proc shewhart data=Times limits=Timelim ;
   boxchart Delay * Day /
      odstitle = title
      boxstyle = schematicid
      nolimits
      nohlabel
```

```
         nolegend
         notches;
      id Reason;
      label Delay = 'Delay in Minutes';
   run;
```

The control limits are suppressed with the NOLIMITS option. The notches, requested with the NOTCHES option, measure the significance of the difference between two medians. The medians are significantly different at approximately the 95% level if the notches do not overlap. For details, see the entry for NOTCHES in "Dictionary of Options: SHEWHART Procedure" on page 1946.

Output 17.3.1 Notched Side-by-Side Box-and-Whisker Plots



Example 17.4: Creating Box-and-Whisker Plots with Varying Widths

NOTE: See *Varying Width Box-and-Whisker Plots* in the SAS/QC Sample Library.

This example shows how to create a box chart with box-and-whisker plots whose widths vary proportionately with the subgroup sample size. The following statements create a SAS data set named Times2 that contains flight departure delays (in minutes) recorded daily for eight consecutive days:

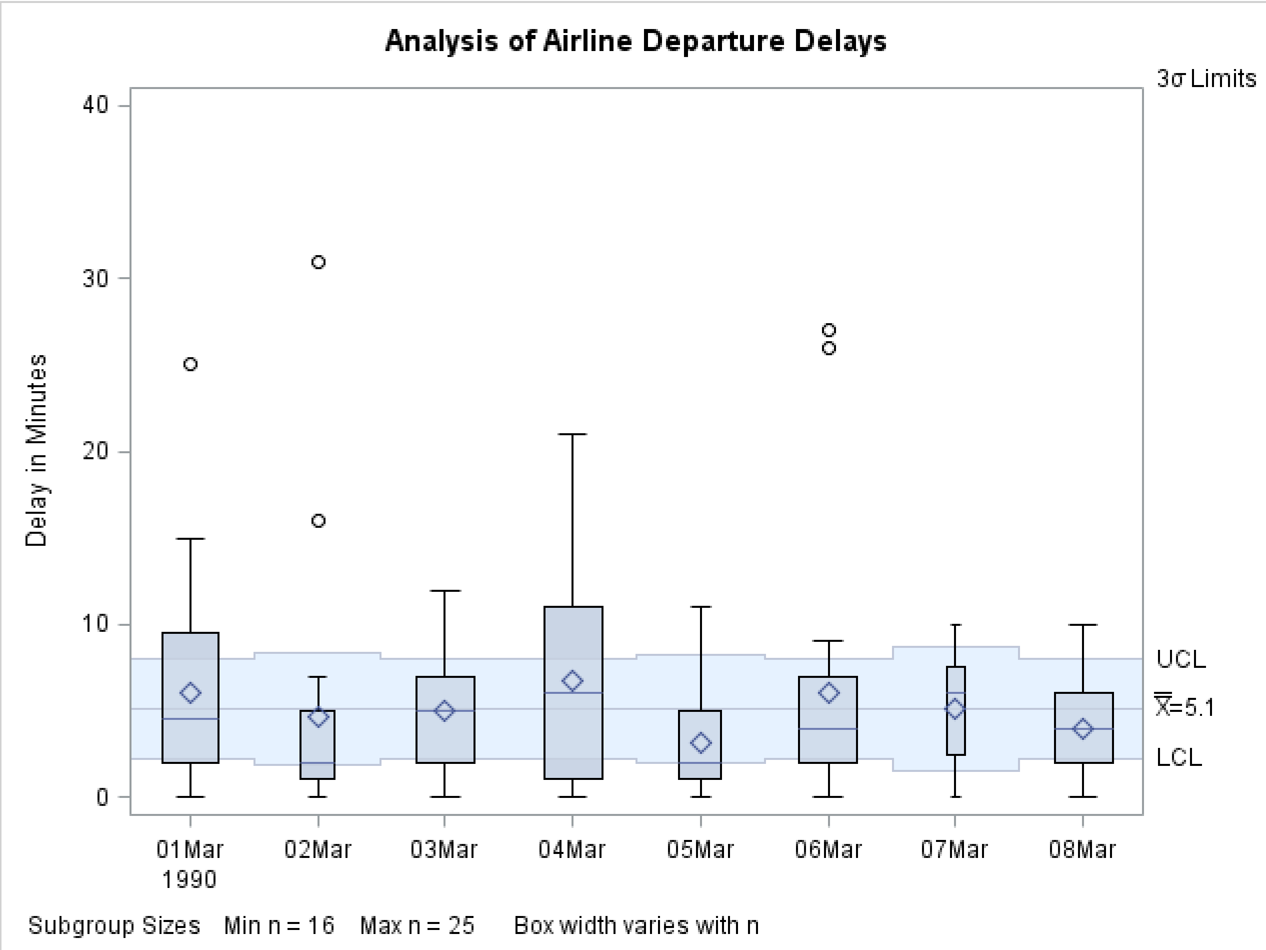
```
data Times2;
   label Delay = 'Delay in Minutes';
   informat Day date7. ;
   format   Day date7. ;
   input Day @ ;
   do Flight=1 to 25;
      input Delay @ ;
      output;
   end;
   datalines;
01MAR90   12   4   2   2  15   8   0  11   0   0
           0  12   3   .   2   3   5   0   6  25
           7   4   9   5  10
02MAR90    1   .   3   .   0   1   5   0   .   .
           1   5   7   .   7   2   2  16   2   1
           3   1  31   .   0
03MAR90    6   8   4   2   3   2   7   6  11   3
           2   7   0   1  10   2   5  12   8   6
           2   7   2   4   5
04MAR90   12   6   9   0  15   7   1   1   0   2
           5   6   5  14   7  21   8   1  14   3
          11   0   1  11   7
05MAR90    2   1   0   4   .   6   2   2   1   4
           1  11   .   1   0   .   5   5   .   2
           3   6   6   4   0
06MAR90    8   6   5   2   9   7   4   2   5   1
           2   2   4   2   5   1   3   9   7   8
           1   0   4  26  27
07MAR90    9   6   6   2   7   8   .   .  10   8
           0   2   4   3   .   .   .   7   .   6
           4   0   .   .   .
08MAR90    1   6   6   2   8   8   5   3   5   0
           8   2   4   2   5   1   6   4   5  10
           2   0   4   1   1
;
```

The following statements create the box chart shown in Output 17.4.1:

```
ods graphics on;
title 'Analysis of Airline Departure Delays';
proc shewhart data=Times2;
   boxchart Delay * Day /
      nohlabel
      boxstyle      = schematic
      odstitle      = title
      boxwidthscale = 1 ;
run;
```

The BOXWIDTHSCALE=1 option specifies that the widths of the box-and-whisker plots are to vary proportionately to the subgroup sample size n. This option is useful in situations where the sample size varies widely across subgroups. For further details, see the entry for BOXWIDTHSCALE= in "Dictionary of Options: SHEWHART Procedure" on page 1946.

Output 17.4.1 Box Chart with Box-and-Whisker Plots of Varying Widths



Example 17.5: Creating Box-and-Whisker Plots with Different Line Styles and Colors

NOTE: See *Varying Width Box-and-Whisker Plots* in the SAS/QC Sample Library.

The control limits in Output 17.4.1 apply to the subgroup means. This example illustrates how you can modify the chart to indicate whether the variability of the process is in control. The following statements create a box chart for Delay in which a dashed outline and a light gray fill color are used for a box-and-whisker plot if the corresponding subgroup standard deviation exceeds its 3σ limits.

First, the SHEWHART procedure is used to create an OUTTABLE= data set (Delaytab) that contains a variable (_EXLIMS_) that records which standard deviations exceed their 3σ limits.

```
proc shewhart data=Times2;
   xschart Delay * Day / nochart
                         outtable = Delaytab;
run;
```

Then, this information is used to set the line styles and fill colors as follows:

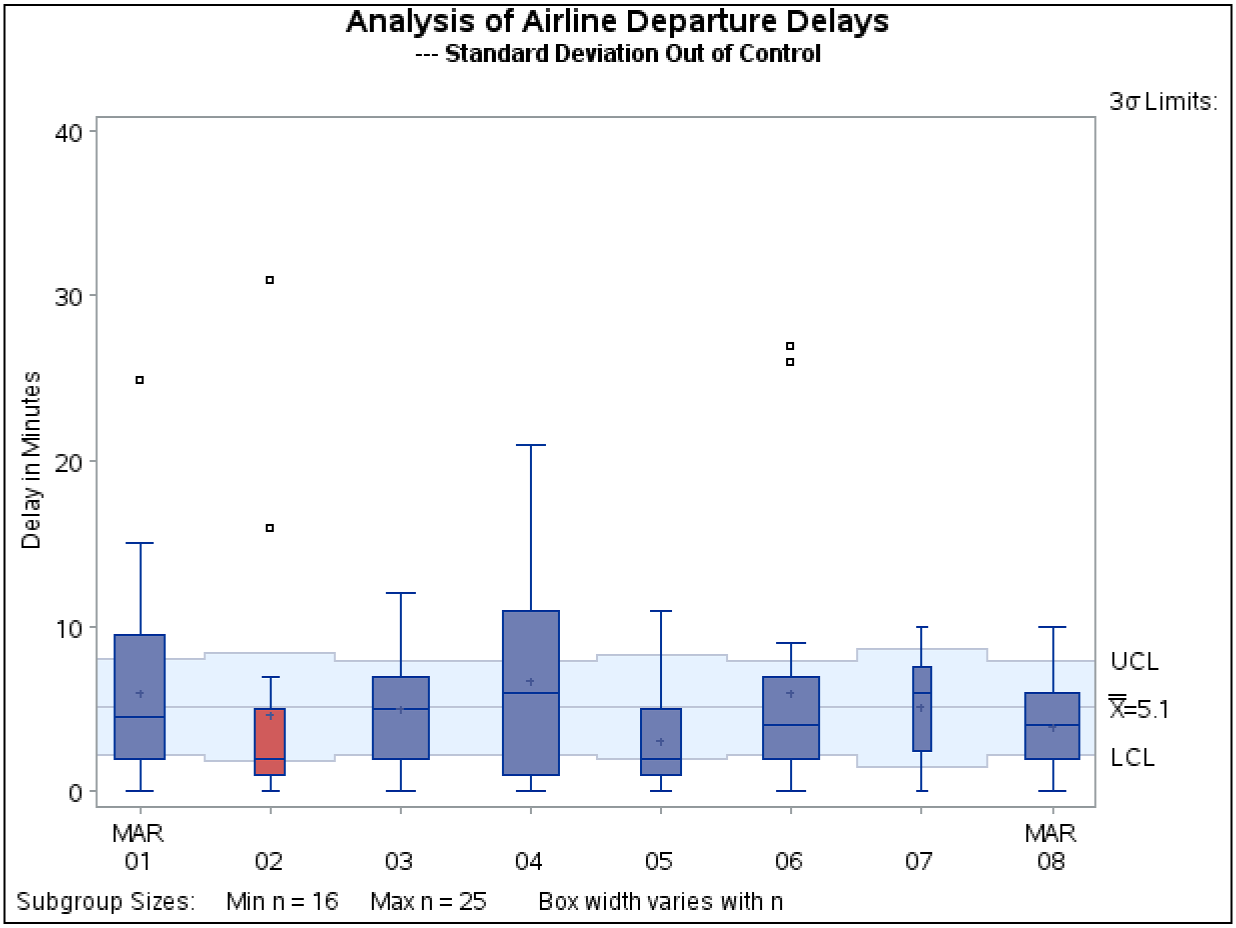
```
data Delaytab;
   length Boxcolor $ 8 ;
   set Delaytab;
   keep Day Boxcolor;
   if _exlims_ = 'UPPER' or _exlims_ = 'LOWER' then do;
      Boxcolor = 'Outside' ;
   end;
   else do;
      Boxcolor = 'Inside' ;
   end;
run;

data Times2;
   merge Times2 Delaytab;
   by Day;
run;
```

The following statements create the modified box chart:

```
ods graphics off;
title  'Analysis of Airline Departure Delays' ;
title2 '--- Standard Deviation Out of Control';
proc shewhart data=Times2;
   boxchart Delay * Day /
        nohlabel
        boxstyle      = schematic
        boxfill       = ( Boxcolor )
        boxwidthscale = 1
        odstitle      = title;
run;
```

The chart is shown in Output 17.5.1. The values of the variable Linestyle specified with the LBOXES= option determine the outline styles for the box-and-whisker plots. The values of the variable Boxcolor specified with the CBOXFILL= option determines the fill colors. For further details, see the entries for these options in "Dictionary of Options: SHEWHART Procedure" on page 1946. The chart indicates that the large variability for March 2 should be checked.

Output 17.5.1 Box Chart Displaying Out-of-Control Subgroup Standard Deviations



Example 17.6: Computing the Control Limits for Subgroup Maximums

NOTE: See *Control Chart for the Subgroup Maximum* in the SAS/QC Sample Library.

This example illustrates how to compute and display control limits for the *maximum* of a subgroup sample. Subgroup samples of 20 metal braces are collected daily, and the lengths of the braces are measured in centimeters. These data are analyzed extensively in Example 17.43. The box chart for LogLength (the log of length) shown in Output 17.43.3 indicates that the subgroup mean is in control and that the subgroup distributions of LogLength are approximately normal. The following statements save the control limits for the mean of the LogLength in a data set named Logllims:

```
data LengthData;
   set LengthData;
   LogLength=log(Length-105);
run;

proc shewhart data=LengthData;
   xchart LogLength*Day /
      nochart
      outlimits=Logllims;
run;
```

The next statements replace the control limits for the mean of LogLength with control limits for the maximum of LogLength:

```
data Maxlim;
   set LengthData;
   set Logllims;
   drop expmax stdmax;
   label _lclx_ = 'Lower Limit for Maximum of 20'
         _uclx_ = 'Upper Limit for Maximum of 20'
         _mean_ = 'Central Line for Maximum of 20';
   expmax = _stddev_*1.86748 + _mean_;
   stdmax = _stddev_*0.52507;
   _lclx_ = expmax - _sigmas_*stdmax;
   _uclx_ = expmax + _sigmas_*stdmax;
   _mean_ = expmax;
   call symput('avgmax',left(put(expmax,8.1)));
run;
```

The control limits are computed using the fact that the maximum of a sample of size 20 from a normal population with zero mean and unit standard deviation has an expected value of 1.86747 and a standard deviation of 0.52509; refer to Teichroew (1962) and see Table 17.13. Finally, the following statements create a box chart for LogLength that displays control limits for the subgroup maximum:

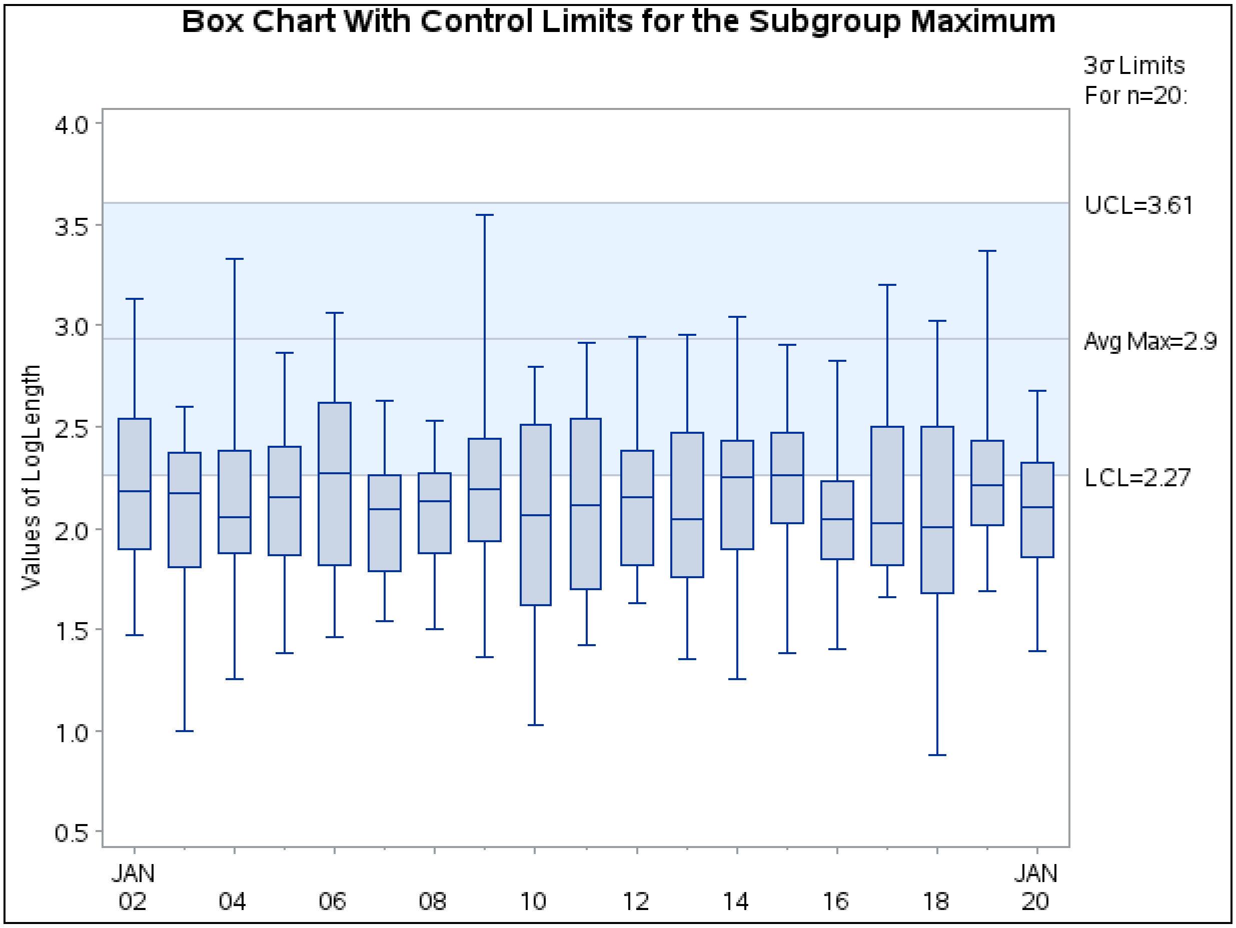
```
ods graphics off;
title 'Box Chart With Control Limits for the Subgroup Maximum';
symbol v=none;
proc shewhart data=LengthData limits=Maxlim;
   boxchart LogLength*Day /
      ranges
      serifs
      nohlabel
      nolegend
      xsymbol="Avg Max=&AVGMAX" ;
   label LogLength='Values of LogLength';
run;
```

The box chart, shown in Output 17.6.1, indicates that the maximum is in control since the tips of the upper whiskers fall within the control limits.

The SYMPUT call is used to pass the value of _MEAN_ in a macro variable to the SHEWHART procedure so that this value can be used to label the central line.

You can apply the variable replacement method shown here to data with sample sizes other than 20 by replacing the constants 1.86747 and 0.52509 with the appropriate values from Table 17.13. Austin (1973) describes a method for approximating these values. You can also use the preceding statements to display control limits for the subgroup minimum by changing the sign of the expected values in Table 17.13.

Output 17.6.1 Box Chart for Subgroup Maximum



The variable replacement method can also be used to create a variety of box charts, including the modifications suggested by Iglewicz and Hoaglin (1987) and Rocke (1989).

Table 17.13 Expected Values and Standard Deviations of Maximum of a Normal Sample

n	Expected Value	Standard Deviation
2	0.56418	0.82565
3	0.84628	0.74798
4	1.02937	0.70123
5	1.16296	0.66899
6	1.26720	0.64494
7	1.35217	0.62605
8	1.42360	0.61065
9	1.48501	0.59780
10	1.53875	0.58681
11	1.58643	0.57730
12	1.62922	0.56891
13	1.66799	0.56144
14	1.70338	0.55474
15	1.73591	0.54869
16	1.76599	0.54316
17	1.79394	0.53809
18	1.82003	0.53342
19	1.84448	0.52910
20	1.86747	0.52509

Example 17.7: Constructing Multi-Vari Charts

"Multi-vari" charts[3] are used in a variety of industries to analyze process data with nested (hierarchical) patterns of variation

- within-sample variation (for example, position within wafer)
- sample-to-sample variation within batches of samples (for example, wafer within lot)
- batch-to-batch variation (for example, across lots)

This example illustrates the construction of a "multi-vari" display. The following statements create a SAS data set named Parm that contains the value of a measured parameter (Measure) recorded at each of five positions on wafers produced in lots.

[3]Multi-vari charts should not be confused with multivariate control charts.

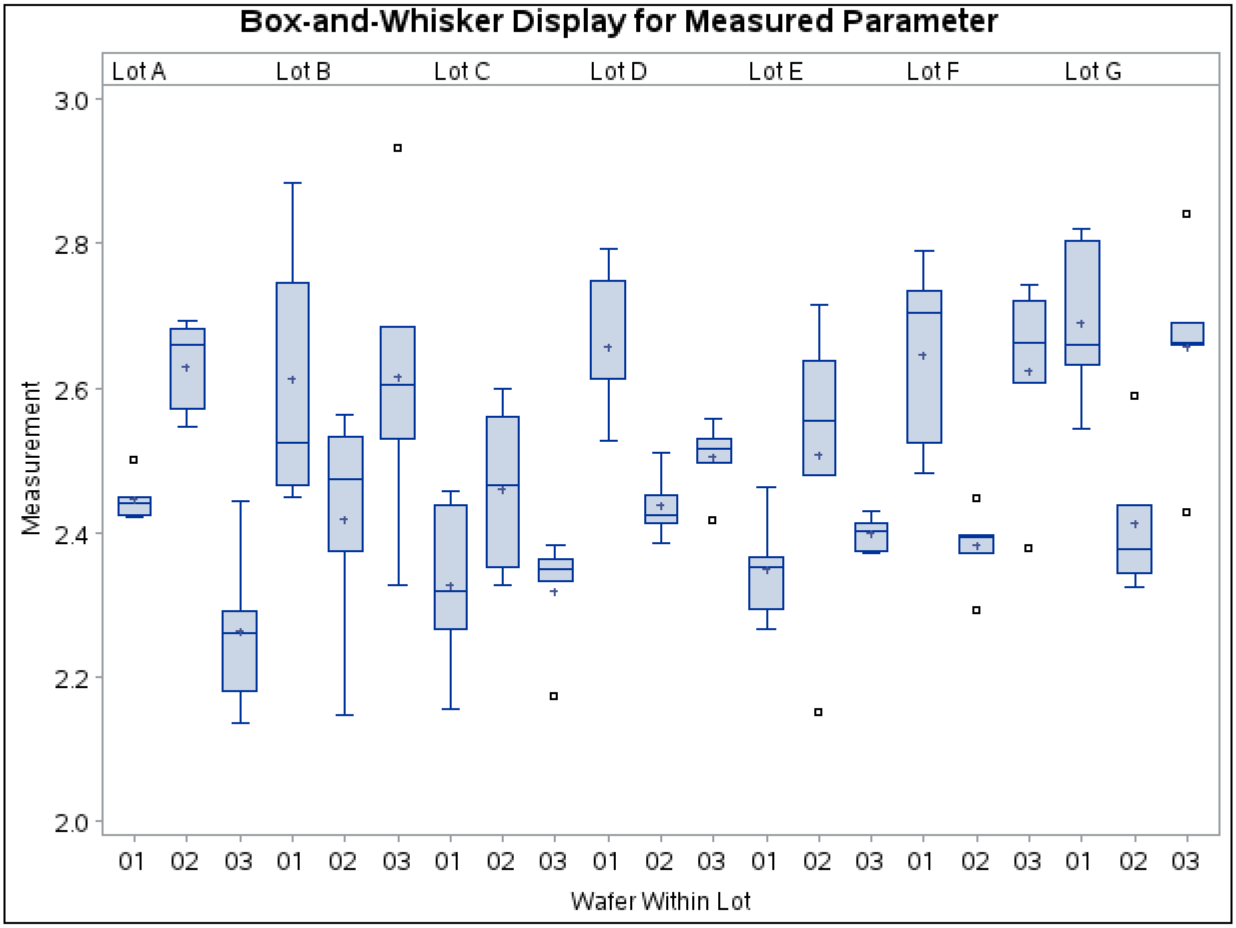
```
data Parm;
   length _phase_ $ 5 Wafer $ 2 position $ 1;
   input  _phase_ $  &  Wafer $ &  position $  Measure ;
   datalines;
   Lot A      01         L        2.42435
   Lot A      01         B        2.44150
   Lot A      01         C        2.42143
   Lot A      01         T        2.44960
   Lot A      01         R        2.50050
   Lot A      02         L        2.68188
   Lot A      02         B        2.57195
   Lot A      02         C        2.54678
   Lot A      02         T        2.65978
   Lot A      02         R        2.69208
   Lot A      03         L        2.18005
   Lot A      03         B        2.13593
   Lot A      03         C        2.44303
   Lot A      03         T        2.29052
   Lot A      03         R        2.25963
   Lot B      01         L        2.46573
   Lot B      01         B        2.44898
   Lot B      01         C        2.52365

   ... more lines ...

   Lot G      03         C        2.66303
   Lot G      03         T        2.65913
   Lot G      03         R        2.84378
;
```

The following statements create an ordinary side-by-side box-and-whisker display for the measurements.

```
ods graphics off;
title 'Box-and-Whisker Display for Measured Parameter';
proc shewhart data=Parm;
   boxchart Measure*Wafer /
         nolimits
         boxstyle  = schematic
         idsymbol  = square
         readphase = all
         phaselegend
         nolegend;
label Measure = 'Measurement'
      Wafer   = 'Wafer Within Lot';
run;
```

The display is shown in Output 17.7.1. Here, the *subgroup-variable* is Wafer, and the option BOXSTYLE=SCHEMATIC is specified to request schematic box-and-whisker plots for the measurements in each subgroup (wafer) sample. The lot values are provided as the values of the special variable _PHASE_, which is read when the option READPHASE=ALL is specified. The option PHASELEGEND requests the legend for phase (lot) values at the top of the chart, and the NOLEGEND option suppresses the default legend for sample sizes. The NOLIMITS option suppresses the display of control limits. This option is recommended whenever you are using the BOXCHART statement to create side-by-side box-and-whisker plots.

Output 17.7.1 Box-and-Whisker Plot Using BOXSTYLE=SCHEMATIC

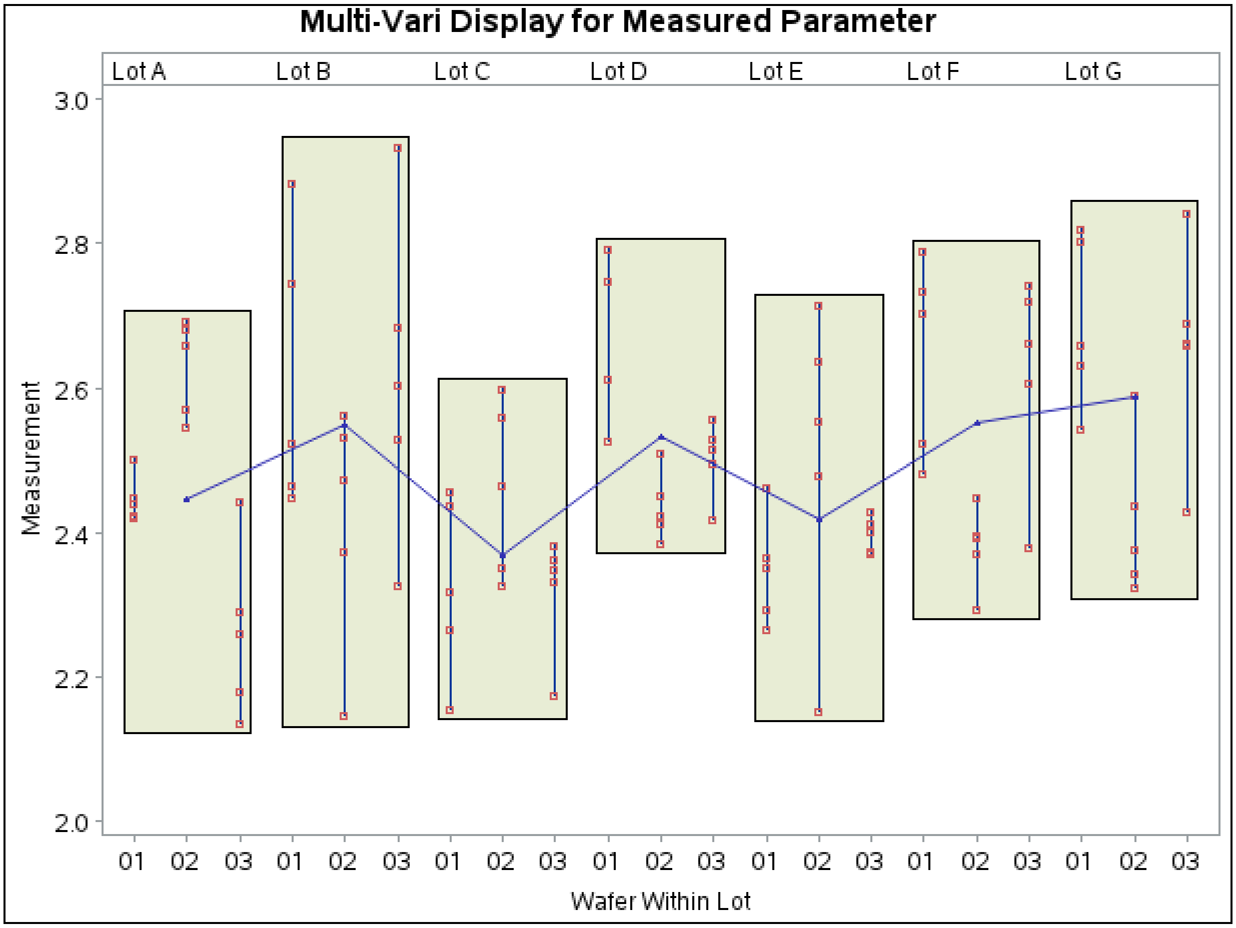


The box-and-whisker display in Output 17.7.1 is not particularly appropriate for these data since there are only five measurements in each wafer and since the variation within each wafer may depend on the position, which is not indicated. The next statements use the BOXCHART statement to produce a multi-vari chart for the same data.

```
symbol v=none;
title 'Multi-Vari Display for Measured Parameter';
proc shewhart data=Parm;
   boxchart Measure*Wafer /
      nolimits
      boxstyle           = pointsjoin
      idsymbol           = square
      cphaseboxfill      = ywh
      cphasebox          = black
      cphasemeanconnect  = bib
      phasemeansymbol    = dot
      readphase          = all
      phaselegend
      nolegend;
label Measure = 'Measurement'
      Wafer   = 'Wafer Within Lot';
run;
```

The display is shown in Output 17.7.2.

Output 17.7.2 Multi-Vari Chart Using BOXSTYLE=POINTSJOIN

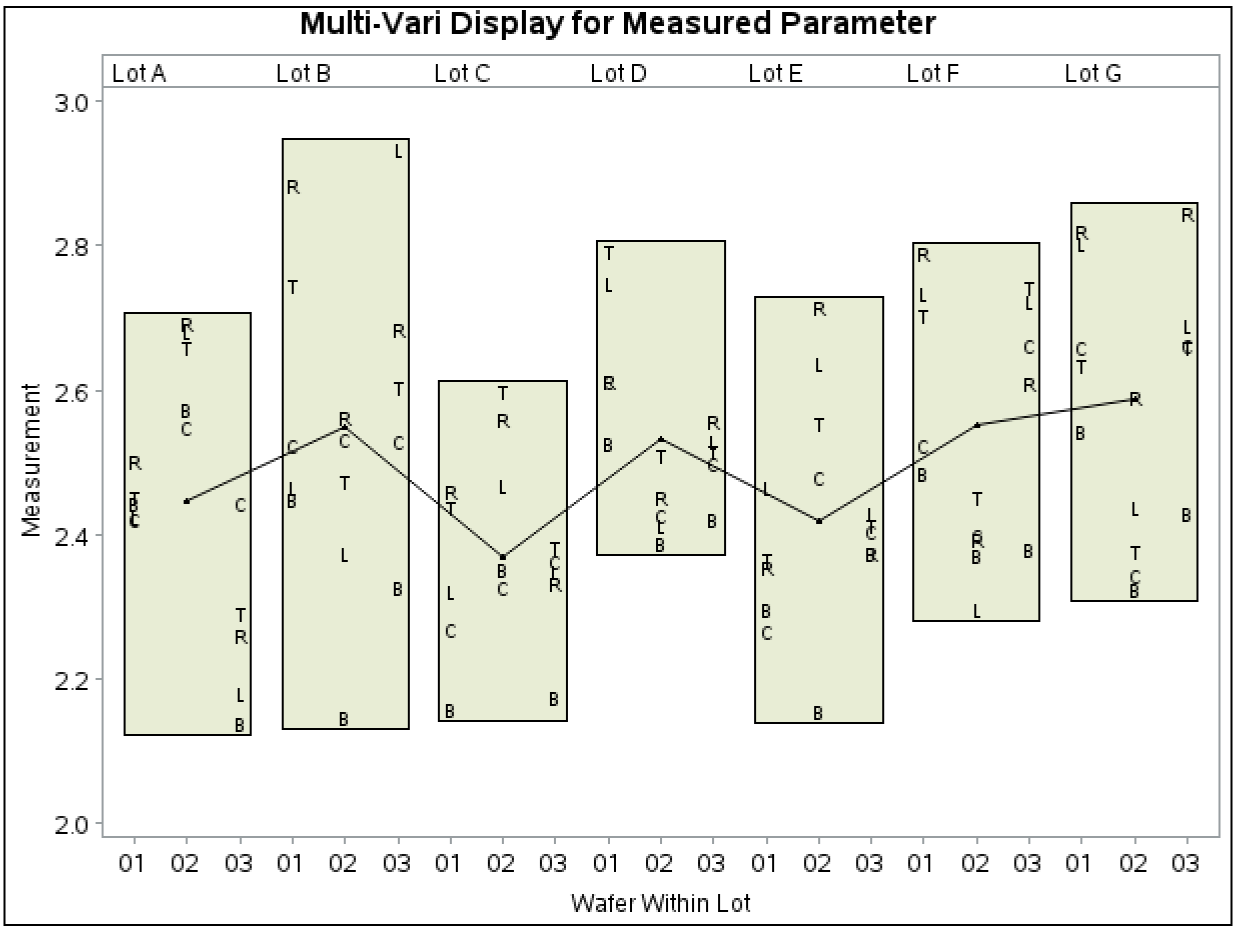


The option BOXSTYLE=POINTSJOIN specifies that the values for each wafer are to be displayed as points joined by a vertical line. The IDSYMBOL= option specifies the symbol marker for the points. The option V=NONE in the SYMBOL statement is specified to suppress the symbol for the wafer averages shown in Output 17.7.1. The option CPHASEBOX=BLACK specifies that the points for each lot are to be enclosed in a black box, and the CPHASEBOXFILL= option specifies the fill color for the box. The option CPHASEMEANCONNECT=BLACK specifies that the means of the lots are to be connected with black lines, and the PHASEMEANSYMBOL= option specifies the symbol marker for the lot means.

The following statements create a slightly different multi-vari chart using the values of the variable POSITION to identify the measurements for each wafer. Note that the option BOXSTYLE=POINTSID is specified and that POSITION is specified as the ID variable. The display is shown in Output 17.7.3.

```
symbol v=none;
title 'Multi-Vari Display for Measured Parameter';
proc shewhart data=Parm;
   boxchart Measure*Wafer /
      nolimits
      cphaseboxfill      = ywh
      cphasemeanconnect = black
      boxstyle           = pointsid
      phasemeansymbol    = dot
      readphase          = all
      phaselegend
      nolegend;
   label Measure = 'Measurement'
         Wafer   = 'Wafer Within Lot';
   id position;
run;
```

Output 17.7.3 Multi-Vari Chart Using BOXSTYLE=POINTSID



CCHART Statement: SHEWHART Procedure

Overview: CCHART Statement

The CCHART statement creates *c* charts for the numbers of nonconformities (defects) in subgroup samples.

You can use options in the CCHART statement to

- specify the number of inspection units per subgroup. Typically (but not necessarily), each subgroup consists of a single unit.
- compute control limits from the data based on a multiple of the standard error of the counts or as probability limits
- tabulate subgroup summary statistics and control limits
- save control limits in an output data set
- save subgroup summary statistics in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) value for the average number of nonconformities per inspection unit
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control the layout and appearance of the chart

You have three alternatives for producing *c* charts with the CCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 3, “SAS/QC Graphics,” for more information about producing these different kinds of graphs.

Getting Started: CCHART Statement

This section introduces the CCHART statement with simple examples that illustrate commonly used options. Complete syntax for the CCHART statement is presented in the section "Syntax: CCHART Statement" on page 1446, and advanced examples are given in the section "Examples: CCHART Statement" on page 1466.

Creating c Charts from Defect Count Data

NOTE: See *c Chart Examples* in the SAS/QC Sample Library.

A *c* chart is used to monitor the number of paint defects on new trucks. Twenty trucks of the same model are inspected, and the number of paint defects per truck is recorded. The following statements create a SAS data set named Trucks, which contains the defect counts:

```
data Trucks;
   input TruckID $ Defects @@;
   label TruckID='Truck Identification Number'
         Defects='Number of Paint Defects';
   datalines;
C1   5   C2   4   C3   4   C4   8   C5  7
C6  12   C7   3   C8  11   E4   8   E9  4
E7   9   E6  13   A3   5   A4   4   A7  9
Q1  15   Q2   8   Q3   9   Q9  10   Q4  8
;
```

A partial listing of Trucks is shown in Figure 17.15.

Figure 17.15 The Data Set Trucks

Paint Defects on New Trucks

TruckID	Defects
C1	5
C2	4
C3	4
C4	8
C5	7

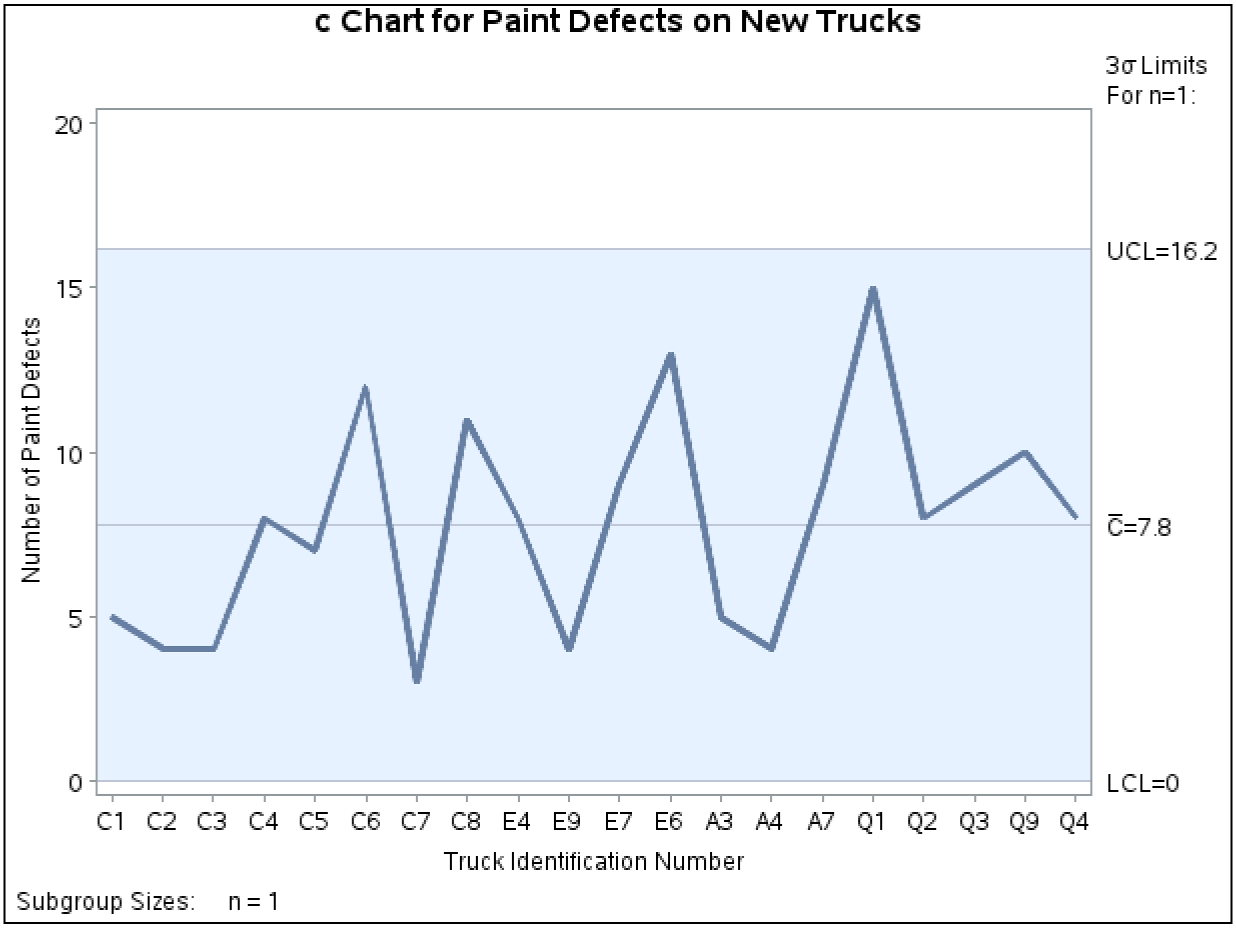
There is a single observation per truck. The variable TruckID identifies the subgroup sample and is referred to as the *subgroup-variable*. The variable Defects contains the number of nonconformities in each subgroup sample and is referred to as the *process variable* (or *process* for short).

The following statements create the *c* chart shown in Figure 17.16:

```
ods graphics off;
title 'c Chart for Paint Defects on New Trucks';
proc shewhart data=Trucks;
   cchart Defects*TruckID;
run;
```

This example illustrates the basic form of the CCHART statement. After the keyword CCHART, you specify the *process* to analyze (in this case, Defects) followed by an asterisk and the *subgroup-variable* (TruckID).

Figure 17.16 *c* Chart of Paint Defects (Traditional Graphics)



Each point on the *c* chart represents the number of nonconformities for a particular subgroup. For instance, the value plotted for the first subgroup is 5 (since there are five paint defects on the first truck). By default, the control limits shown are 3σ limits estimated from the data; the formulas are given in "Control Limits" on page 1457. Since none of the points exceed the 3σ limits, the *c* chart indicates that the painting process is in statistical control.

See "Constructing Charts for Numbers of Nonconformities (c Charts)" on page 1456 for details concerning *c* charts. For more details on reading raw data, see "DATA= Data Set" on page 1462.

Saving Control Limits

NOTE: See *c Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for a *c* chart in a SAS data set; this enables you to apply the control limits to future data (see the section "Reading Preestablished Control Limits" on page 1441) or subsequently modify the limits with a DATA step program.

The following statements read the data set Trucks introduced in "Creating c Charts from Defect Count Data" on page 1437 and saves the control limit information displayed in Figure 17.16 in a data set named Deflim:

```
proc shewhart data=Trucks;
   cchart Defects*TruckID / outlimits=Deflim
                            nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. Options such as OUTLIMITS= and NOCHART are specified after the slash (/) in the CCHART statement. A complete list of options is presented in the section "Syntax: CCHART Statement" on page 1446. The data set Deflim is listed in Figure 17.17.

Figure 17.17 The Data Set Deflim Containing Control Limit Information

Control Limits Data Set Deflim

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_U_	_LCLC_	_C_	_UCLC_
Defects	TruckID	ESTIMATE	1	.002492887	3	7.8	0	7.8	16.1785

The data set Deflim contains one observation with the limits for the *process* Defects. The variables _LCLC_, and _UCLC_ contain the lower and upper control limits. The variable _C_ contains the central line, and the variable _U_ contains the average number of nonconformities per inspection unit. Since all the subgroups contain a single inspection unit, the values of _C_ and _U_ are the same. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _U_ is an estimate or standard value. For more information, see the section "OUTLIMITS= Data Set" on page 1459.

Alternatively, you can use the OUTTABLE= option to create an output data set that saves both the control limits and the subgroup statistics, as illustrated by the following statements:

```
title 'Number of Nonconformities and Control Limit Information';
proc shewhart data=Trucks;
   cchart Defects*TruckID / outtable=Trucktab
                            nochart;
run;
```

The OUTTABLE= data set Trucktab is listed in Figure 17.18.

Figure 17.18 The Data Set Trucktab

Number of Nonconformities and Control Limit Information

VAR	TruckID	_SIGMAS_	_LIMITN_	_SUBN_	_LCLC_	_SUBC_	_C_	_UCLC_	_EXLIM_
Defects	C1	3	1	1	0	5	7.8	16.1785	
Defects	C2	3	1	1	0	4	7.8	16.1785	
Defects	C3	3	1	1	0	4	7.8	16.1785	
Defects	C4	3	1	1	0	8	7.8	16.1785	
Defects	C5	3	1	1	0	7	7.8	16.1785	
Defects	C6	3	1	1	0	12	7.8	16.1785	
Defects	C7	3	1	1	0	3	7.8	16.1785	
Defects	C8	3	1	1	0	11	7.8	16.1785	
Defects	E4	3	1	1	0	8	7.8	16.1785	
Defects	E9	3	1	1	0	4	7.8	16.1785	
Defects	E7	3	1	1	0	9	7.8	16.1785	
Defects	E6	3	1	1	0	13	7.8	16.1785	
Defects	A3	3	1	1	0	5	7.8	16.1785	
Defects	A4	3	1	1	0	4	7.8	16.1785	
Defects	A7	3	1	1	0	9	7.8	16.1785	
Defects	Q1	3	1	1	0	15	7.8	16.1785	
Defects	Q2	3	1	1	0	8	7.8	16.1785	
Defects	Q3	3	1	1	0	9	7.8	16.1785	
Defects	Q9	3	1	1	0	10	7.8	16.1785	
Defects	Q4	3	1	1	0	8	7.8	16.1785	

This data set contains one observation for each subgroup sample. The variables _SUBC_ and _SUBN_ contain the number of nonconformities per subgroup and the number of inspection units per subgroup. The variables _LCLC_ and _UCLC_ contain the lower and upper control limits, and the variable _C_ contains the central line. The variables _VAR_ and TruckID contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1460.

An OUTTABLE= data set can be read later as a TABLE= data set in the SHEWHART procedure. For example, the following statements read Trucktab and display a *c* chart (not shown here) identical to the chart in Figure 17.16:

```
title 'c Chart for Paint Defects in New Trucks';
proc shewhart table=Trucktab;
   cchart Defects*Truckid;
label _SUBC_ = 'Number of Paint Defects';
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2096). For more information, see "TABLE= Data Set" on page 1464.

Reading Preestablished Control Limits

NOTE: See *c Chart Examples* in the SAS/QC Sample Library.

In the previous example, control limits were saved in a SAS data set named Deflim. This example shows how these limits can be applied to defect data for a second group of trucks, which are provided in the following data set:

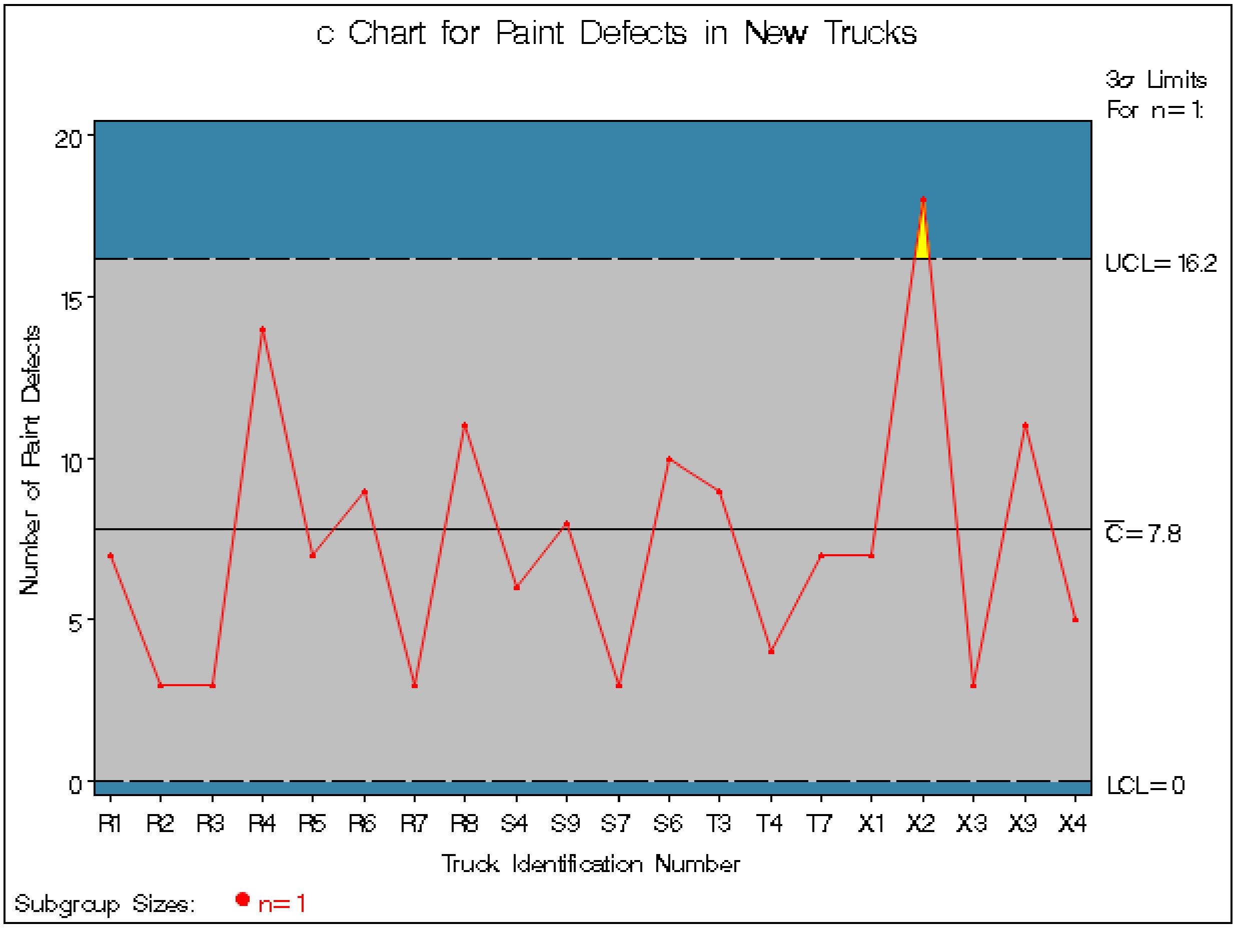
```
data Trucks2;
   input TruckID $ Defects @@;
   label TruckID='Truck Identification Number'
         Defects='Number of Paint Defects';
   datalines;
R1   7   R2    3   R3    3   R4   14   R5   7
R6   9   R7    3   R8   11   S4    6   S9   8
S7   3   S6   10   T3    9   T4    4   T7   7
X1   7   X2   18   X3    3   X9   11   X4   5
;
```

The following statements plot the number of paint defects for the second group of trucks on a *c* chart using the control limits in Deflim. The chart is shown in Figure 17.19.

```
options nogstyle;
goptions ftext=swiss;
symbol v=dot color=red height=.8;
title 'c Chart for Paint Defects in New Trucks';
proc shewhart data=Trucks2 limits=Deflim;
   cchart Defects*TruckID / cframe   = steel
                            cconnect = red
                            cinfill  = ligr
                            coutfill = yellow ;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement and CCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently.

Figure 17.19 *c* Chart for Second Set of Trucks (Traditional Graphics with NOGSTYLE)



Note that the number of defects on the truck with identification number X2 exceeds the upper control limit, indicating that the process is out-of-control. The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Defects
- the value of _SUBGRP_ matches the *subgroup-variable* name TruckID

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1463 for details concerning the variables that you must provide.

Creating c Charts from Nonconformities per Unit

NOTE: See *c Chart Examples* in the SAS/QC Sample Library.

In the previous example, the input data set provided the number of nonconformities per subgroup sample. However, in some applications, as illustrated here, the data may be provided as the number of nonconformities *per inspection unit* for each subgroup.

A clothing manufacturer ships shirts in boxes of ten. Prior to shipment, each shirt is inspected for flaws. Since the manufacturer is interested in the average number of flaws per shirt, the number of flaws found in each box is divided by ten and then recorded. The following statements create a SAS data set named Shirts, which contains the average number of flaws per shirt for 25 boxes:

```
data Shirts;
   input Box avgdefu @@;
   avgdefn=10;
   datalines;
 1  0.4    2  0.7    3  0.5    4  1.0    5  0.3
 6  0.2    7  0.0    8  0.4    9  0.4   10  0.6
11  0.2   12  0.7   13  0.3   14  0.1   15  0.3
16  0.6   17  0.6   18  0.3   19  0.7   20  0.3
21  0.0   22  0.1   23  0.5   24  0.6   25  0.4
;
```

A partial listing of Shirts is shown in Figure 17.20.

Figure 17.20 The Data Set Shirts

Average Number of Shirt Flaws

Box	avgdefu	avgdefn
1	0.4	10
2	0.7	10
3	0.5	10
4	1.0	10
5	0.3	10

The data set Shirts contains three variables: the box number (Box), the average number of flaws per shirt (avgdefu), and the number of shirts per box (avgdefn). Here, a *subgroup* is a box of shirts, and an *inspection unit* is an individual shirt. Note that each subgroup consists of ten inspection units.

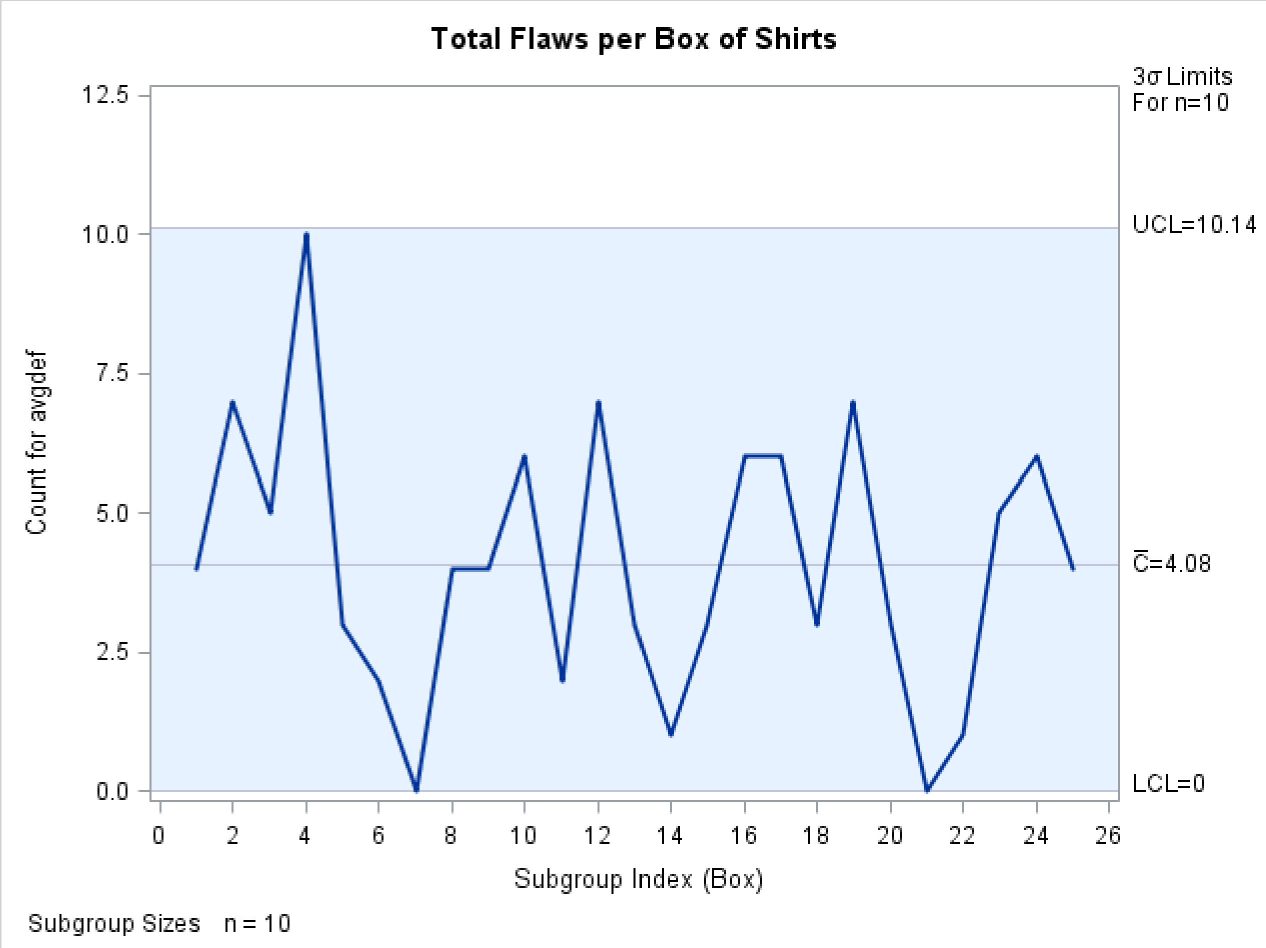
To create a *c* chart plotting the total number of flaws per box (instead of per shirt), you can specify Shirts as a HISTORY= data set.

```
ods graphics on;
title 'Total Flaws per Box of Shirts';
proc shewhart history=Shirts;
   cchart avgdef*Box / odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the *c* chart is created using ODS Graphics instead of traditional graphics.

Note that avgdef is *not* the name of a SAS variable in the data set but is instead the common prefix for the SAS variable names avgdefu and avgdefn. The suffix characters *U* and *N* indicate *number of nonconformities per unit* and *sample size*, respectively. This naming convention enables you to specify two variables in the HISTORY= data set with a single name referred to as the *process*. The name Box specified after the asterisk is the name of the *subgroup-variable*. The *c* chart is shown in Figure 17.21.

Figure 17.21 *c* Chart for Boxes of Shirts (ODS Graphics)



In general, a HISTORY= input data set used with the CCHART statement must contain the following variables:

- subgroup variable
- subgroup number of nonconformities per unit variable
- subgroup sample size variable

Furthermore, the names of the nonconformities per unit and sample size variables must begin with the *process* name specified in the CCHART statement and end with the special suffix characters *U* and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. Suppose that, instead of the variables avgdefu and avgdefn, the data set Shirts contained the variables Shirtdef and Sizes. The following statements would temporarily rename Shirtdef and Sizes to avgdefu and avgdefn:

```
proc shewhart
   history=Shirts (rename=(Shirtdef = AvgdefU
                           Sizes    = AvgdefN ));
   cchart Avgdef*Box;
run;
```

For more information, see "HISTORY= Data Set" on page 1463.

Saving Nonconformities per Unit

NOTE: See *c Chart Examples* in the SAS/QC Sample Library.

A department store receives boxes of shirts containing 10, 25, or 50 shirts. Each box is inspected, and the total number of defects per box is recorded. The following statements create a SAS data set named Shirts2, which contains the total defects per box for 20 boxes:

```
data Shirts2;
   input Box Flaws Nshirts @@;
   datalines;
 1  3  10     2  8  10     3 15  25     4 20  25
 5  9  25     6  1  10     7  1  10     8 21  50
 9  3  10    10  7  10    11  1  10    12 21  25
13  9  25    14  3  25    15 12  50    16 18  50
17  7  10    18  4  10    19  8  10    20  4  10
;
```

A partial listing of Shirts2 is shown in Figure 17.22.

Figure 17.22 The Data Set Shirts2

Number of Shirt Flaws per Box

Box	Flaws	Nshirts
1	3	10
2	8	10
3	15	25
4	20	25
5	9	25

The variable Box contains the box number, the variable Flaws contains the number of flaws in each box, and the variable Nshirts contains the number of shirts in each box. To evaluate the quality of the shirts, you should report the average number of defects per shirt. The following statements create a data set containing the number of flaws per shirt and the number of shirts per box:

```
proc shewhart data=Shirts2;
   cchart Flaws*Box / subgroupn   = Nshirts
                      outhistory = shirthist
                      nochart ;
run;
```

The SUBGROUPN= option names the variable in the DATA= data set whose values specify the number of inspection units per subgroup. The OUTHISTORY= option names an output data set containing the number of nonconformities per inspection unit and the number of inspection units per subgroup. A partial listing of Shirthist is shown in Figure 17.23.

Figure 17.23 The Data Set Shirthist

Average Defects Per Shirt

Box	FlawsU	FlawsN
1	0.30	10
2	0.80	10
3	0.60	25
4	0.80	25
5	0.36	25

There are three variables in the data set Shirthist.

- Box contains the subgroup index.
- FlawsU contains the numbers of nonconformities per inspection unit.
- FlawsN contains the subgroup sample sizes.

Note that the variables containing the numbers of nonconformities per inspection unit and subgroup sample sizes are named by adding the suffix characters *U* and *N* to the *process* Defects specified in the CCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1459.

Syntax: CCHART Statement

The basic syntax for the CCHART statement is as follows:

CCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

CCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=**'*character*' > / < *options* > **;**

You can use any number of CCHART statements in the SHEWHART procedure. The components of the CCHART statement are described as follows:

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If numbers of nonconformities per subgroup are read from a DATA= data set, *process* must be the name of the variable containing the numbers of nonconformities.

  For an example, see "Creating c Charts from Defect Count Data" on page 1437.

- If numbers of nonconformities per unit and numbers of inspection units per subgroup are read from a HISTORY= data set, *process* must be the common prefix of the appropriate variables in the HISTORY= data set.

  For an example, see "Creating c Charts from Nonconformities per Unit" on page 1442.

- If numbers of nonconformities per subgroup, numbers of inspection units per subgroup, and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set.

  For an example, see "Saving Control Limits" on page 1439.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct *c* charts for Defects and Flaws:

```
proc shewhart data=Info;
   cchart (Defects Flaws)*Sample;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding CCHART statement, SAMPLE is the subgroup variable. For details, see "Subgroup Variables" on page 1871.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. These blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2026 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the number of nonconformities.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2025 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create a *c* chart using an asterisk (*) to plot the points:

```
proc shewhart data=Info lineprinter;
   cchart Defects*Sample='*';
run;
```

options

enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options", which follows, lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1946 describes each option in detail.

Summary of Options

The following tables list the CCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1946.

Table 17.14 CCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	requests probability limits for chart
LIMITN=	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
PROBLIMITS=	requests probability limits at discrete values
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
ACTUALALPHA	displays the actual probability of a point being outside the control limits in the control limits legend
CINFILL=	specifies color for area inside control limits
CLIMITS=	specifies color of control limits, central line, and related labels
CSYMBOL=	specifies label for central line
LCLLABEL=	specifies label for lower control limit
LIMLABSUBCHAR=	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	specifies line type for control limits
NDECIMAL=	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits

Table 17.14 *continued*

Option	Description
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit
UCLLABEL=	specifies label for upper control limit
WLIMITS=	specifies width for control limits and central line
Standard Value Options	
TYPE=	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
U0=	specifies known average number of nonconformities per unit
Options for Plotting and Labeling Points	
ALLLABEL=	labels every point on *c* chart
CLABEL=	specifies color for labels
CCONNECT=	specifies color for line segments that connect points on chart
CFRAMELAB=	specifies fill color for frame around labeled points
CNEEDLES=	specifies color for needles that connect points to central line
COUT=	specifies color for portions of line segments that connect points outside control limits
COUTFILL=	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	specifies angle at which labels are drawn
LABELFONT=	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=	labels points outside control limits
SYMBOLLEGEND=	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=	specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	computes zone widths independently above and below center line
NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	suppresses tests across *phase* boundaries

Table 17.14 *continued*

Option	Description
TESTS=	specifies tests for special causes
TEST2RUN=	specifies length of pattern for Test 2
TEST3RUN=	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL=	provides labels for points where test is positive
TESTLABEL*n*=	specifies label for *n*th test for special causes
TESTNMETHOD=	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=	enables tests for special causes to be reset
WESTGARD=	requests that Westgard rules be applied
ZONELABELS	adds labels A, B, and C to zone lines
ZONES	adds lines delineating zones A, B, and C
ZONEVALPOS=	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	specifies color for labels indicating points where test is positive
CTESTSYMBOL=	specifies color for symbol used to plot points where test is positive
CZONES=	specifies color for lines and labels delineating zones A, B, and C
LTESTS=	specifies type of line connecting points where test is positive
LZONES=	specifies line type for lines delineating zones A, B, and C
TESTFONT=	specifies software font for labels at points where test is positive
TESTHEIGHT=	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	specifies plot symbol for points where test is positive
TESTSYMBOLHT=	specifies symbol height for points where test is positive
WTESTS=	specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill colors for frame for plot area
CTEXT=	specifies color for tick mark values and axis labels
DISCRETE	produces horizontal axis for discrete numeric group values
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text

Table 17.14 *continued*

Option	Description
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis
VFORMAT=	specifies format for vertical axis tick mark labels
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=	specifies width of axis lines
Plot Layout Options	
ALLN	plots means for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	specifies number of pages or screens to be used to display chart

Table 17.14 *continued*

Option	Description
ZEROSTD	displays *c* chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= options
CVREF=	specifies color for lines requested by VREF= options
HREF=	specifies position of reference lines perpendicular to horizontal axis
HREFDATA=	specifies position of reference lines perpendicular to horizontal axis
HREFLABELS=	specifies labels for HREF= lines
HREFLABPOS=	specifies position of HREFLABELS= labels
LHREF=	specifies line type for HREF= lines
LVREF=	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	specifies position of reference lines perpendicular to vertical axis
VREFLABELS=	specifies labels for VREF= lines
VREFLABPOS=	position of VREFLABELS= labels
Grid Options	
CGRID=	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=	specifies line type for grid requested with the ENDGRID option
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme points are clipped
CLIPLEGEND=	specifies text for clipping legend
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features chart
DESCRIPTION=	specifies description of *c* chart's GRSEG catalog entry
FONT=	specifies software font for labels and legends on charts
NAME=	specifies name of *c* chart's GRSEG catalog entry
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option

Table 17.14 *continued*

Option	Description
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	shades areas between control limits and connected points lying outside the limits
STARFILL=	specifies a variable identfying groups of stars filled with different colors
STARS=	specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	specifies the control limit infill transparency
MARKERS	plots subgroup points with markers
NOBLOCKREF	suppresses block and phase reference lines
NOBLOCKREFFILL	suppresses block and phase wall fills
NOFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOPHASEREF	suppresses block and phase reference lines
NOPHASEREFFILL	suppresses block and phase wall fills
NOREF	suppresses block and phase reference lines
NOREFFILL	suppresses block and phase wall fills
NOSTARFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	specifies a graph footnote
ODSFOOTNOTE2=	specifies a secondary graph footnote
ODSLEGENDEXPAND	specifies that legend entries contain all levels observed in the data
ODSTITLE=	specifies a graph title
ODSTITLE2=	specifies a secondary graph title
OUTFILLTRANSPARENCY=	specifies control limit outfill transparency
OVERLAYURL=	specifies URLs to associate with overlay points
PHASEPOS=	specifies vertical position of phase legend
PHASEREFLEVEL=	associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	draws central line labels using a simulated software font
STARTRANSPARENCY=	specifies star fill transparency

Table 17.14 *continued*

Option	Description
URL=	specifies a variable whose values are URLs to be associated with subgroups
Input Data Set Options	
MISSBREAK	specifies that observations with missing values are not to be processed
SUBGROUPN	specifies subgroup sample sizes as constant number *n* or as values of variable in a DATA= data set
Output Data Set Options	
OUTHISTORY=	creates output data set containing subgroup summary statistics
OUTINDEX=	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	creates output data set containing control limits
OUTTABLE=	creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for *block-variable* legend
BLOCKLABTYPE=	specifies text size of *block-variable* legend
BLOCKPOS=	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE=	specifies value of _PHASE_ in the OUTHISTORY= data set

Table 17.14 *continued*

Option	Description
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	specifies color for STARCIRCLES= circles
CSTARFILL=	specifies color for filling stars
CSTAROUT=	specifies outline color for stars exceeding inner or outer circles
CSTARS=	specifies color for outlines of stars
LSTARCIRCLES=	specifies line types for STARCIRCLES= circles
LSTARS=	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	specifies reference circles for stars
STARINRADIUS=	specifies inner radius of stars
STARLABEL=	specifies vertices to be labeled
STARLEGEND=	specifies style of legend for star vertices
STARLEGENDLAB=	specifies label for STARLEGEND= legend
STAROUTRADIUS=	specifies outer radius of stars
STARSPECS=	specifies method used to standardize vertex variables
STARSTART=	specifies angle for first vertex
STARTYPE=	specifies graphical style of star
STARVERTICES=	superimposes star at each point on chart
WSTARCIRCLES=	specifies width of STARCIRCLES= circles
WSTARS=	specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	specifies colors for overlay line segments
COVERLAY=	specifies colors for overlay plots
COVERLAYCLIP=	specifies color for clipped points on overlays
LOVERLAY=	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=	specifies variables to overlay on chart
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
OVERLAYHTML=	specifies links to associate with overlay points
OVERLAYID=	specifies labels for overlay points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols for overlays
OVERLAYSYMHT=	specifies symbol heights for overlays

Table 17.14 *continued*

Option	Description
WOVERLAY=	specifies widths of overlay line segments
Options for Interactive Control Charts	
HTML=	specifies a variable whose values create links to be associated with subgroups
HTML_LEGEND=	specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	specifies plot character for clipped points
CONNECTCHAR=	specifies character used to form line segments that connect points on chart
HREFCHAR=	specifies line character for HREF= lines
SYMBOLCHARS=	specifies characters indicating *symbol-variable*
TESTCHAR=	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	specifies line character for VREF= lines
ZONECHAR=	specifies character for lines that delineate zones for tests for special causes

Details: CCHART Statement

Constructing Charts for Numbers of Nonconformities (c Charts)

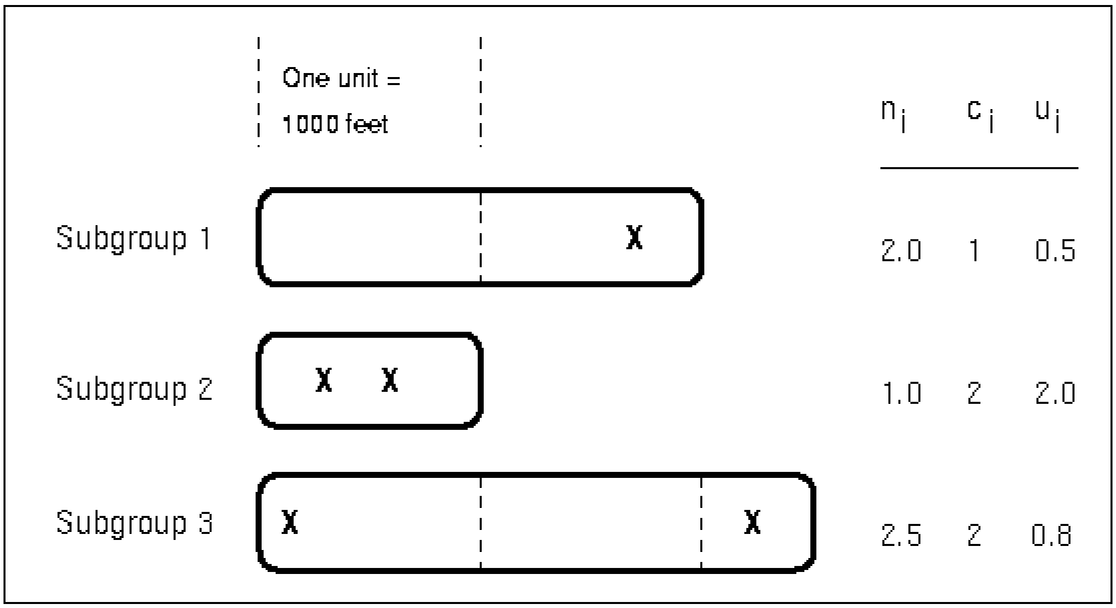
The following notation is used in this section:

u	expected number of nonconformities per unit produced by the process
u_i	number of nonconformities per unit in the ith subgroup
c_i	total number of nonconformities in the ith subgroup
n_i	number of inspection units in the ith subgroup. Typically, $n_i = 1$ and $u_i = c_i$ for c charts. In general, $u_i = c_i / n_i$.
$\bar{u}$	average number of nonconformities per unit taken across subgroups. The quantity $\bar{u}$ is computed as a weighted average: $$\bar{u} = \frac{n_1 u_1 + \cdots + n_N u_N}{n_1 + \cdots + n_N} = \frac{c_1 + \cdots + c_N}{n_1 + \cdots + n_N}$$
N	number of subgroups
χ^2_ν	has a central χ^2 distribution with ν degrees of freedom

Plotted Points

Each point on a *c* chart represents the total number of nonconformities (c_i) in a subgroup. For example, Figure 17.24 displays three sections of pipeline that are inspected for defective welds (indicated by an *X*). Each section represents a *subgroup* composed of a number of *inspection units*, which are 1000-foot-long sections. The number of units in the *i*th subgroup is denoted by n_i, which is the subgroup sample size. The value of n_i can be fractional; Figure 17.24 shows $n_3 = 2.5$ units in the third subgroup.

Figure 17.24 Terminology for *c* Charts and *u* Charts



The *number of nonconformities* in the *i*th subgroup is denoted by c_i. The *number of nonconformities per unit* in the *i*th subgroup is denoted by $u_i = c_i/n_i$. In Figure 17.24, the number of welds per inspection unit in the third subgroup is $u_3 = 2/2.5 = 0.8$.

A *u* chart created with the UCHART statement plots the quantity u_i for the *i*th subgroup (see "UCHART Statement: SHEWHART Procedure" on page 1755). An advantage of a *u* chart is that the value of the central line at the *i*th subgroup does not depend on n_i. This is not the case for a *c* chart, and consequently, a *u* chart is often preferred when the number of units n_i is not constant across subgroups.

Central Line

On a *c* chart, the central line indicates an estimate for $n_i u$, which is computed as $n_i \bar{u}$. If you specify a known value (u_0) for *u*, the central line indicates the value of $n_i u_0$.

Note that the central line varies with subgroup sample size n_i. When $n_i = 1$ for all subgroups, the central line has the constant value $\bar{c} = (c_1 + \cdots + c_N)/N$.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of c_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).

- as probability limits defined in terms of α, a specified probability that c_i exceeds the limits

The lower and upper control limits, LCLC and UCLC respectively, are given by

$$\begin{aligned} \text{LCLC} &= \max\left(n_i\bar{u} - k\sqrt{n_i\bar{u}}\ ,0\right) \\ \text{UCLC} &= n_i\bar{u} + k\sqrt{n_i\bar{u}} \end{aligned}$$

The upper and lower control limits vary with the number of inspection units per subgroup n_i. If $n_i = 1$ for all subgroups, the control limits have constant values.

$$\begin{aligned} \text{LCLC} &= \max\left(\bar{c} - k\sqrt{\bar{c}}\ ,0\right) \\ \text{UCLC} &= \bar{c} + k\sqrt{\bar{c}} \end{aligned}$$

An upper probability limit UCLC for c_i can be determined using the fact that

$$\begin{aligned} P\{c_i > \text{UCLC}\} &= 1 - P\{c_i \le \text{UCLC}\} \\ &= 1 - P\{\chi^2_{2(\text{UCLC}+1)} \ge 2n_i\bar{u}\} \end{aligned}$$

The upper probability limit UCLC is then calculated by setting

$$1 - P\{\chi^2_{2(\text{UCLC}+1)} \ge 2n_i\bar{u}\} = \alpha/2$$

and solving for UCLC.

A similar approach is used to calculate the lower probability limit LCLC, using the fact that

$$\begin{aligned} P\{c_i < \text{LCLC}\} &= P\{c_i \le \text{LCLC} - 1\} \\ &= P\{\chi^2_{2((\text{LCLC}-1)+1)} > 2n_i\bar{u}\} \\ &= P\{\chi^2_{2\text{LCLC}} > 2n_i\bar{u}\} \end{aligned}$$

The lower probability limit LCLC is then calculated by setting

$$P\{\chi^2_{2\text{LCLC}} > 2n_i\bar{u}\} = \alpha/2$$

and solving for LCLC. This assumes that the process is in statistical control and that c_i has a Poisson distribution. For more information, refer to Johnson, Kotz, and Kemp (1992). Note that the probability limits vary with the number of inspection units per subgroup (n_i) and are asymmetric about the central line.

If a standard value u_0 is available for u, replace $\bar{u}$ with u_0 in the formulas for the control limits. You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify u_0 with the U0= option or with the variable _U_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 17.15 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_C_	value of central line on c chart ($n_i \bar{u}$ or $n_i u_0$)
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLC	lower control limit for number of nonconformities
LIMITN	sample size associated with the control limits
SIGMAS	multiple (k) of standard error of c_i
SUBGRP	*subgroup-variable* specified in the CCHART statement
TYPE	type (estimate or standard value) of _U_
_U_	average number of nonconformities per unit ($\bar{u}$ or u_0)
UCLC	upper control limit for number of nonconformities
VAR	*process* specified in the CCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variables _C_, _LCLC_, _UCLC_, and _LIMITN_.

2. If the limits are defined in terms of a multiple k of the standard error of c_i, the value of _ALPHA_ is computed as $P\{c_i < \text{\_LCLC\_}\} + P\{c_i > \text{\_UCLC\_}\}$. If control limits vary with subgroup sample size and are determined in terms of k, _ALPHA_ is assigned the special missing value V.

3. If the limits are probability limits, the value of _SIGMAS_ is computed as $(\text{\_UCLC\_} - \text{\_C\_})/\sqrt{\text{\_C\_}}$. If probability limits vary with subgroup sample size, _SIGMAS_ is assigned the special missing value V.

4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the CCHART statement. For an example, see "Saving Control Limits" on page 1439.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup sample size variable named by *process* suffixed with *N*
- a subgroup number of nonconformities per unit variable named by *process* suffixed with *U*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the CCHART statement. For example, consider the following statements:

```
proc shewhart data=Fabric;
   cchart (Flaws Ndefects)*lot / outhistory=Summary;
run;
```

The data set SUMMARY contains variables named lot, FlawsU, FlawsN, NdefctsU, and NdefctsN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example that creates an OUTHISTORY= data set, see “Saving Nonconformities per Unit” on page 1445. Note that an OUTHISTORY= data set created with the CCHART statement can be used as a HISTORY= data set by either the CCHART statement or the UCHART statement.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
_C_	average number of nonconformities
EXLIM	control limit exceeded on *c* chart
LCLC	lower control limit for number of nonconformities
LIMITN	nominal sample size associated with the control limits
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBC	subgroup number of nonconformities
SUBN	subgroup sample size
TESTS	tests for special causes signaled on *c* chart
UCLC	upper control limit for number of nonconformities
VAR	*process* specified in the CCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*

- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The *k*th character of a value of _TESTS_ is *k* if Test *k* is positive at that subgroup. For example, if you request the first four tests (the ones appropriate for *c* charts) and Tests 2 and 4 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, a 4 for the fourth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1439.

ODS Tables

The following table summarizes the ODS tables that you can request with the CCHART statement.

Table 17.16 ODS Tables Produced with the CCHART Statement

Table Name	Description	Options
CCHART	*c* chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*). **NOTE:** In SAS/QC 13.1 the SHEWHART procedure does not support the creation of graphs that are editable with the ODS Graphics Editor.

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. CCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1946.

When ODS Graphics is in effect, the CCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 17.17.

Table 17.17 ODS Graphics Produced by the CCHART Statement

ODS Graph Name	Plot Description
CChart	*c* chart

See Chapter 3, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read the number of nonconformities in subgroup samples from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the CCHART statement must be a SAS variable in the data set. This variable provides the number of nonconformities in subgroup samples indexed by the *subgroup-variable*. Typically (but not necessarily), the subgroup consists of a single inspection unit. The *subgroup-variable*, specified in the CCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. The data set must contain one observation per subgroup. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2031.

For an example of a DATA= data set, see "Creating c Charts from Defect Count Data" on page 1437.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   cchart Defects*Lot;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLC_, _C_, and _UCLC_, which specify the control limits
- the variable _U_, which is used to calculate the control limits (see "Control Limits" on page 1457)

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE' and 'STANDARD'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1441.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to create your own HISTORY= data set. A HISTORY= data set used with the CCHART statement must contain the following variables:

- *subgroup-variable*
- subgroup number of nonconformities per unit variable for each *process*
- subgroup sample size variable (number of units per subgroup) for each *process*

The names of the subgroup number of nonconformities per unit and subgroup sample size variables must be the *process* name concatenated with the special suffix characters *U* and *N*, respectively. For example, consider the following statements:

```
proc shewhart history=summary;
   cchart (flaws ndefects)*lot;
run;
```

The data set summary must include the variables lot, flawsU, flawsN, ndefctsU, and ndefctsN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character. Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (see "Displaying Stratification in Phases" on page 2031 for an example).

For an example of a HISTORY= data set, see "Creating c Charts from Nonconformities per Unit" on page 1442.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure or to create your own TABLE= data set. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2096.

The following table lists the variables required in a TABLE= data set used with the CCHART statement:

Table 17.18 Variables Required in a TABLE= Data Set

Variable	Description
_C_	average number of nonconformities
LCLC	lower control limit for nonconformities
LIMITN	nominal sample size associated with the control limits
subgroup-variable	values of the *subgroup-variable*
SUBC	subgroup number of nonconformities
SUBN	subgroup sample size
UCLC	upper control limit for nonconformities

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see “Saving Control Limits” on page 1439.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup defect counts variable
Vertical	TABLE=	_SUBC_

For an example, see “Labeling Axes” on page 2063.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: CCHART Statement

This section provides advanced examples of the CCHART statement.

Example 17.8: Applying Tests for Special Causes

NOTE: See *Tests for Special Causes Applied to c Chart* in the SAS/QC Sample Library.

This example illustrates how you can apply tests for special causes to make *c* charts more sensitive to special causes of variation. Twenty trucks of the same model are inspected, and the number of paint defects per truck is recorded. The following statements create a SAS data set named Trucks3:

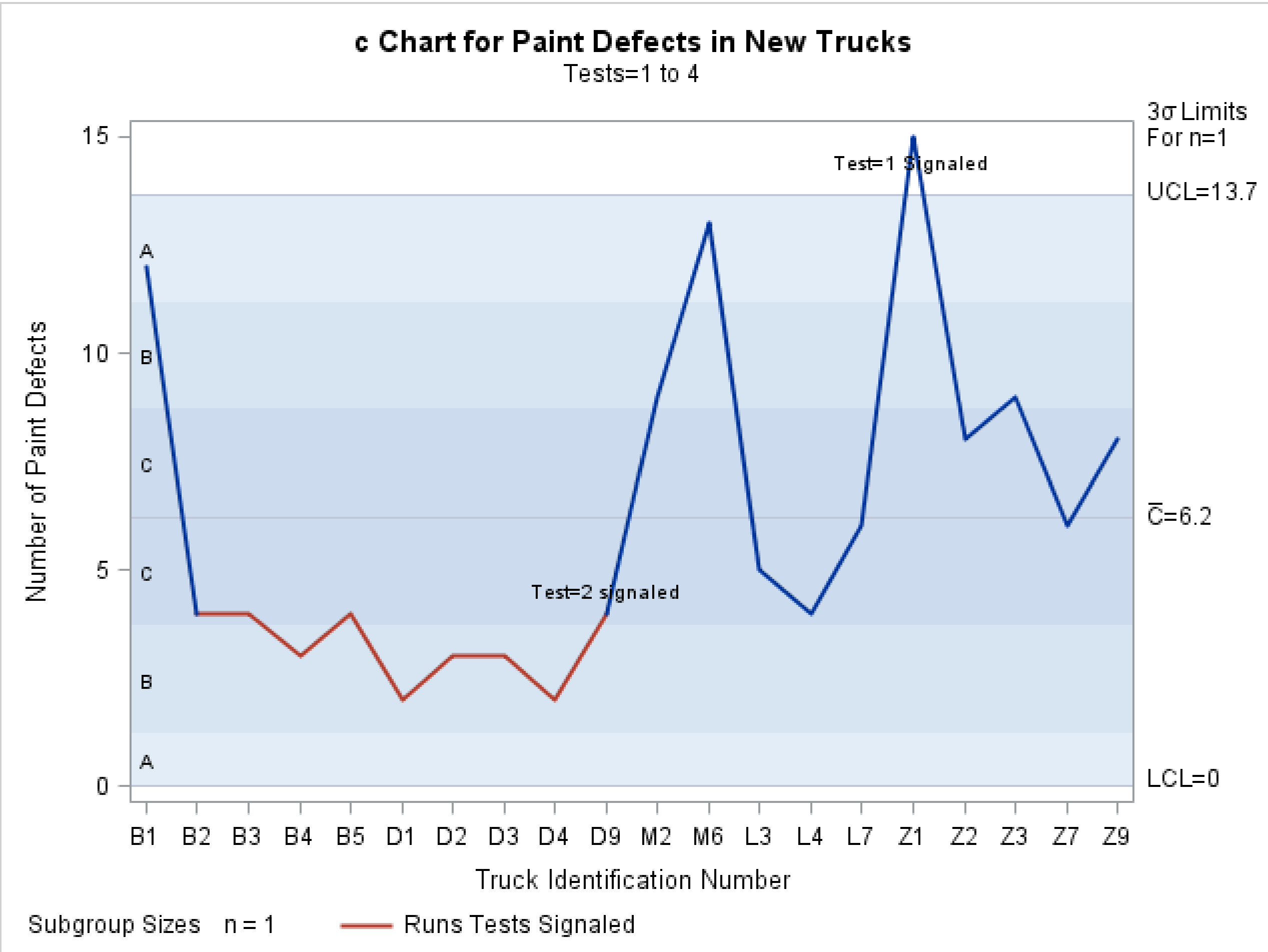
```
data Trucks3;
   input TruckID $ Defects @@;
   label TruckID='Truck Identification Number'
         Defects='Number of Paint Defects';
   datalines;
B1  12    B2  4    B3  4    B4   3
B5   4    D1  2    D2  3    D3   3
D4   2    D9  4    M2  9    M6  13
L3   5    L4  4    L7  6    Z1  15
Z2   8    Z3  9    Z7  6    Z9   8
;
```

The following statements create a *c* chart and tabulate the information on the chart. The chart and table are shown in Output 17.8.1 and Output 17.8.2.

```
ods graphics on;
title1 'c Chart for Paint Defects in New Trucks';
title2 'Tests=1 to 4';
proc shewhart data=Trucks3;
   cchart Defects*TruckID / tests      = 1 to 4
                            testlabel1 = 'Test=1 Signaled'
                            testlabel2 = 'Test=2 signaled'
                            odstitle   = title
                            odstitle2  = title2
                            zonelabels
                            tabletests
                            tablelegend;
run;
```

The TESTS= option requests Tests 1, 2, 3, and 4, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2073. Only Tests 1, 2, 3, and 4 are recommended for *c* charts. The TESTLABEL1= and TESTLABEL2= options specify the labels for points where Tests 1 and 2 are positive. The TESTFONT= option specifies the font for the labels indicating points at which the tests are positive.

Output 17.8.1 Tests for Special Causes Displayed on *c* Chart



Output 17.8.2 Tabular Form of *c* Chart

c Chart for Paint Defects in New Trucks
Tests=1 to 4

The SHEWHART Procedure

c Chart Summary for Defects					
		3 Sigma Limits with n=1 for Count			
TruckID	Subgroup Sample Size	Lower Limit	Subgroup Count	Upper Limit	Special Tests Signaled
B1	1.00000	0	12.000000	13.669940	
B2	1.00000	0	4.000000	13.669940	
B3	1.00000	0	4.000000	13.669940	
B4	1.00000	0	3.000000	13.669940	
B5	1.00000	0	4.000000	13.669940	
D1	1.00000	0	2.000000	13.669940	
D2	1.00000	0	3.000000	13.669940	
D3	1.00000	0	3.000000	13.669940	
D4	1.00000	0	2.000000	13.669940	
D9	1.00000	0	4.000000	13.669940	2
M2	1.00000	0	9.000000	13.669940	
M6	1.00000	0	13.000000	13.669940	
L3	1.00000	0	5.000000	13.669940	
L4	1.00000	0	4.000000	13.669940	
L7	1.00000	0	6.000000	13.669940	
Z1	1.00000	0	15.000000	13.669940	1
Z2	1.00000	0	8.000000	13.669940	
Z3	1.00000	0	9.000000	13.669940	
Z7	1.00000	0	6.000000	13.669940	
Z9	1.00000	0	8.000000	13.669940	

Test Descriptions	
Test 1	One point beyond Zone A (outside control limits)
Test 2	Nine points in a row on one side of center line

The ZONELABELS option requests zone lines and displays zone labels on the chart. The zones are used to define the tests. The TABLETESTS option requests a table of counts of nonconformities, subgroup sample sizes, and control limits, together with a column indicating the subgroups at which the tests are positive. The TABLELEGEND option adds a legend describing the tests that are positive.

Output 17.8.1 and Output 17.8.2 indicate that Test 1 is positive at Truck Z1 and Test 2 is positive at Truck D9.

Example 17.9: Specifying a Known Expected Number of Nonconformities

NOTE: See *c Chart Based on Known (Standard) Value* in the SAS/QC Sample Library.

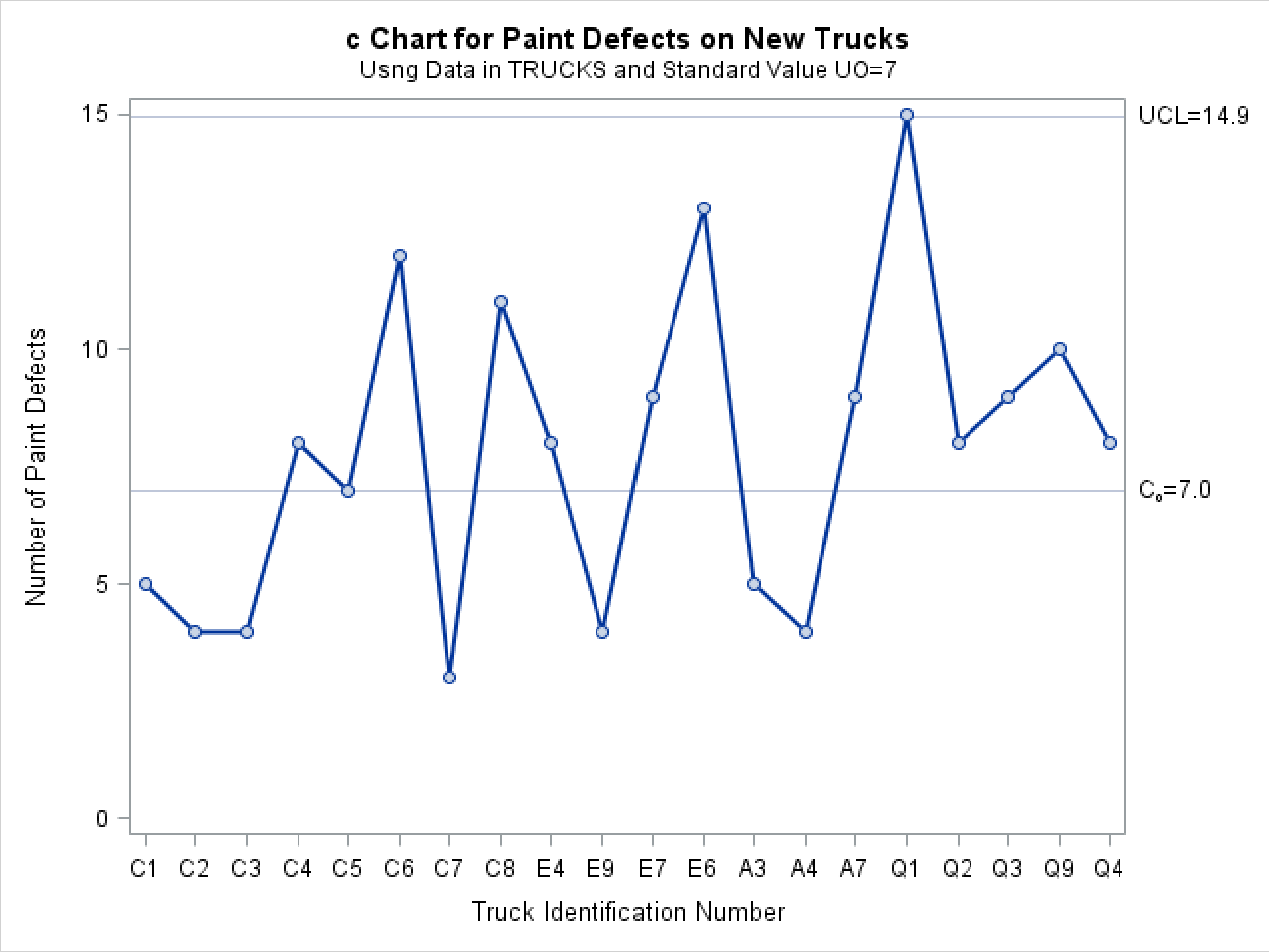
This example illustrates how you can create a c chart based on a known (standard) value u_0 for the expected number of nonconformities per unit.

A c chart is used to monitor the number of paint defects per truck. The defect counts are provided as values of the variable Defects in the data set Trucks given in "Creating c Charts from Defect Count Data" on page 1437. Based on previous testing, it is known that $u_0 = 7$. The following statements create a c chart with control limits derived from this value:

```
ods graphics on;
title 'c Chart for Paint Defects on New Trucks';
title2 'Usng Data in TRUCKS and Standard Value UO=7';
proc shewhart data=Trucks;
   cchart Defects*TruckID / u0         = 7
                            csymbol    = c0
                            odstitle   = title
                            odstitle2  = title2
                            markers
                            nolegend
                            nolimitslegend
                            nolimit0;
run;
```

The chart is shown in Output 17.9.1. The U0= option specifies u_0, and the CSYMBOL= option requests a label for the central line indicating that the line represents a standard value. The NOLEGEND option suppresses the legend for the subgroup sample size, and the NOLIMITSLEGEND option suppresses the legend for the control limits that appears by default in the upper right corner of the chart. The NOLIMIT0 option suppresses the display of the lower limit when it is equal to zero.

Output 17.9.1 A c Chart with Standard Value u_0



The number of paint defects on Truck Q1 exceeds the upper control limit, indicating that the process is out of control.

Alternatively, you can specify u_0 as the value of the variable _U_ in a LIMITS= data set, as follows:

```
data tlimits;
   length _subgrp_ _var_ _type_ $8;
   _U_      = 7;
   _subgrp_ = 'truckid';
   _var_    = 'defects';
   _limitn_ = 1;
   _type_   = 'STANDARD';

proc shewhart data=trucks limits=tlimits;
   cchart defects*truckid / csymbol=c0
                            nolegend
                            nolimitslegend
                            nolimit0;
run;
```

The chart produced by these statements is identical to the one in Output 17.9.1.

For further details, see "LIMITS= Data Set" on page 1463.

Example 17.10: Creating c Charts for Varying Numbers of Units

NOTE: See *c Chart for Varying Number of Inspection Units* in the SAS/QC Sample Library.

In applications where the number of inspection units per subgroup is not equal to one, a *u* chart is typically used to analyze the number of nonconformities *per unit* (see "UCHART Statement: SHEWHART Procedure" on page 1755). However, as shown in this example, you can use the CCHART statement to create a *c* chart for this type of data.

Figure 17.25 Difference between *c* Charts and *u* Charts

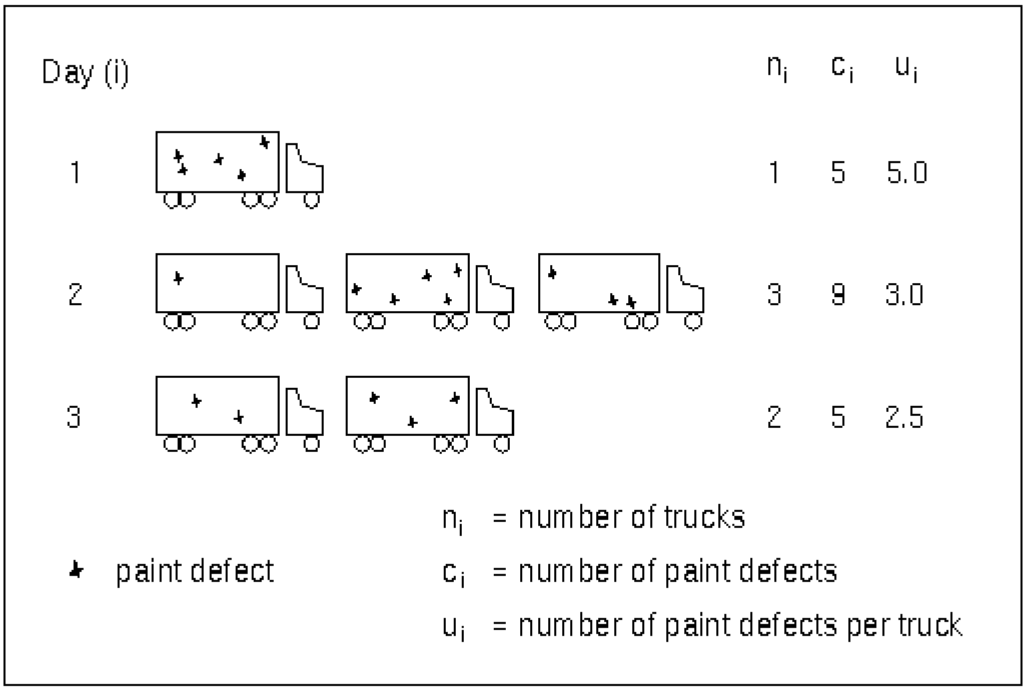


Figure 17.25 illustrates a situation in which varying numbers of trucks are painted each day. Trucks painted on the same day are regarded as *subgroups*, and each truck is regarded as an *inspection unit*. The following statements create a SAS data set named Trucks4, which contains paint defects for trucks painted on 26 days:

```
data Trucks4;
   input Day Defects Ntrucks @@;
   label Day='Day'
         Defects='Number of Paint Defects';
   datalines;
 1   5  1    2   9  3
 3   5  2    4   9  2
 5  24  4    6  10  2
 7  15  3    8  17  3
 9  16  3   10  13  2
11  28  4   12  18  5
13   8  2   14   7  2
15   5  1   16  17  3
17   2  1   18  17  3
```

```
19   15   4    20   19   5
21    6   3    22   23   5
23   27   4    24    6   2
25   12   2    26   12   3
;
```

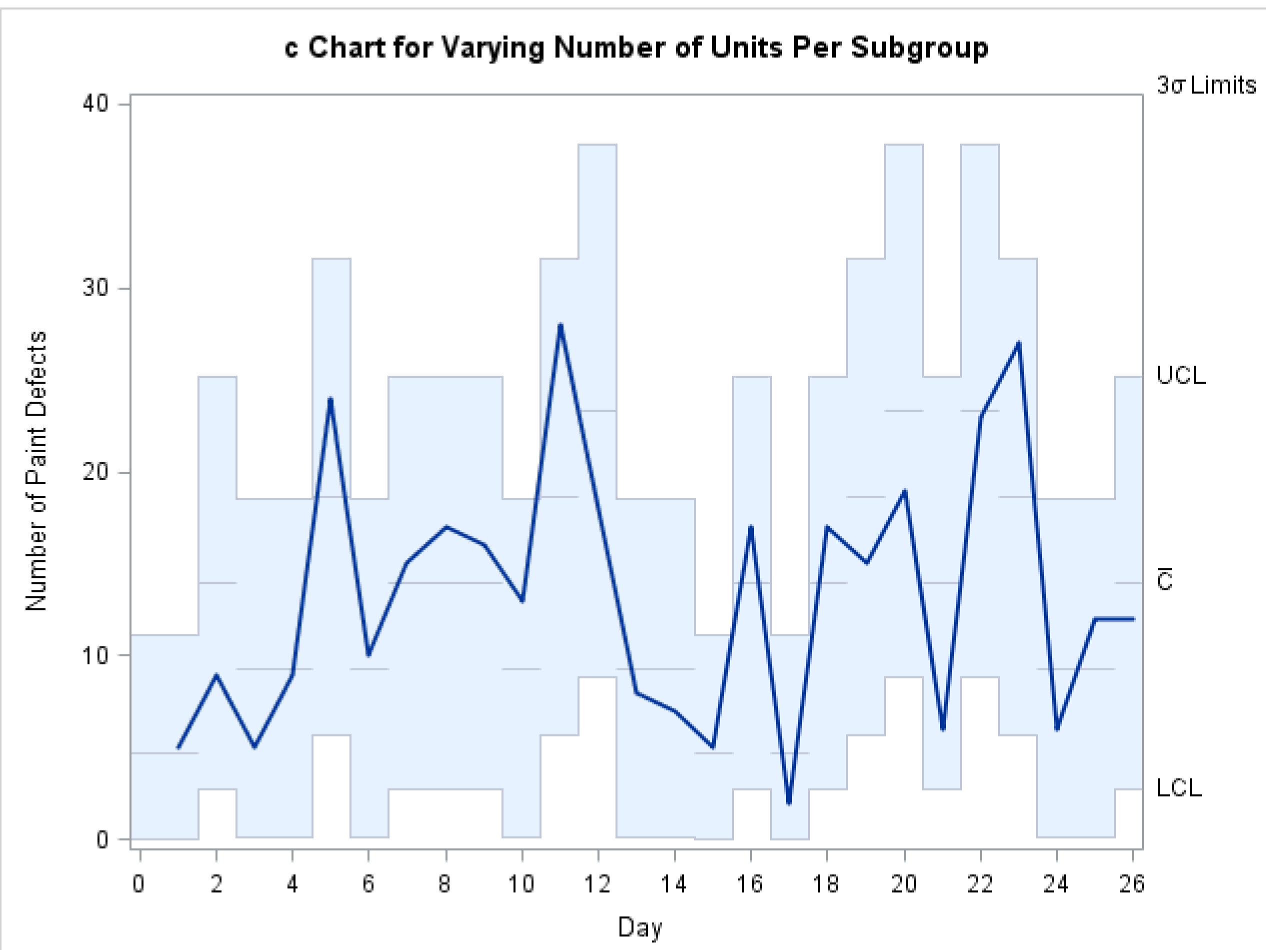
The variable Defects provides the defect count (c_i) for the ith day, and the variable Ntrucks provides the number of inspection units (n_i). The following statements create a c chart for this data:

```
ods graphics on;
title 'c Chart for Varying Number of Units Per Subgroup';
proc shewhart data=Trucks4;
   cchart Defects*Day / subgroupn = Ntrucks
                        odstitle  = title
                        nolegend;
run;
```

The SUBGROUPN= option specifies the subgroup sample size variable Ntrucks (in general, the values of this variable need not be integers). Alternatively, you can specify a fixed value with the SUBGROUPN= option. When this option is not specified, it is assumed that $n_i = 1$.

The chart is shown in Output 17.10.1. Note that the central line and the control limits vary with the number of inspection units.

Output 17.10.1 *c* Chart for Varying Number of Units



IRCHART Statement: SHEWHART Procedure

Overview: IRCHART Statement

The IRCHART statement creates control charts for individual measurements and moving ranges. These charts are appropriate when only one measurement is available for each subgroup sample and when the measurements are independently and normally distributed.

You can use options in the IRCHART statement to

- compute control limits from the data based on a multiple of the standard error of the individual measurements and moving ranges or as probability limits
- tabulate individual measurements, moving ranges, and control limits
- save control limits in an output data set
- save individual measurements and moving ranges in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) process mean and standard deviation for computing control limits
- specify the number of consecutive measurements to use when computing the moving ranges
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing charts of individual measurements and moving ranges with the IRCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 3, “SAS/QC Graphics,” for more information about producing these different kinds of graphs.

Getting Started: IRCHART Statement

This section introduces the IRCHART statement with simple examples that illustrate commonly used options. Complete syntax for the IRCHART statement is presented in the section "Syntax: IRCHART Statement" on page 1483, and advanced examples are given in the section "Examples: IRCHART Statement" on page 1506.

Creating Individual Measurements and Moving Range Charts

NOTE: See *Individual Measurement and Moving Range Charts* in the SAS/QC Sample Library.

An aeronautics company manufacturing jet engines measures the inner diameter of the forward face of each engine (in centimeters). The following statements create a SAS data set that contains the diameter measurements for 20 engines:

```
data Jets;
   input Engine Diam @@;
   label Engine = "Engine Number";
   datalines;
 1 78.4   2 80.1   3 84.4   4 79.1   5 80.4
 6 83.5   7 73.8   8 83.5   9 75.0  10 76.8
11 70.5  12 80.3  13 82.4  14 79.4  15 86.4
16 90.5  17 77.7  18 82.5  19 79.9  20 83.2
;
```

A partial listing of Jets is shown in Figure 17.26.

Figure 17.26 Partial Listing of the Data Set Jets

The Data Set JETS

Engine	Diam
1	78.4
2	80.1
3	84.4
4	79.1
5	80.4

Each observation contains the diameter measurement and identification number for a particular engine. The variable Engine identifies the sequence of engines and is referred to as the *subgroup-variable*.[4] The variable Diam contains the measurements and is referred to as the *process variable* (or *process* for short).

Since the production rate is low, individual measurements and moving range charts are used to monitor the process. The following statements create the charts shown in Figure 17.27:

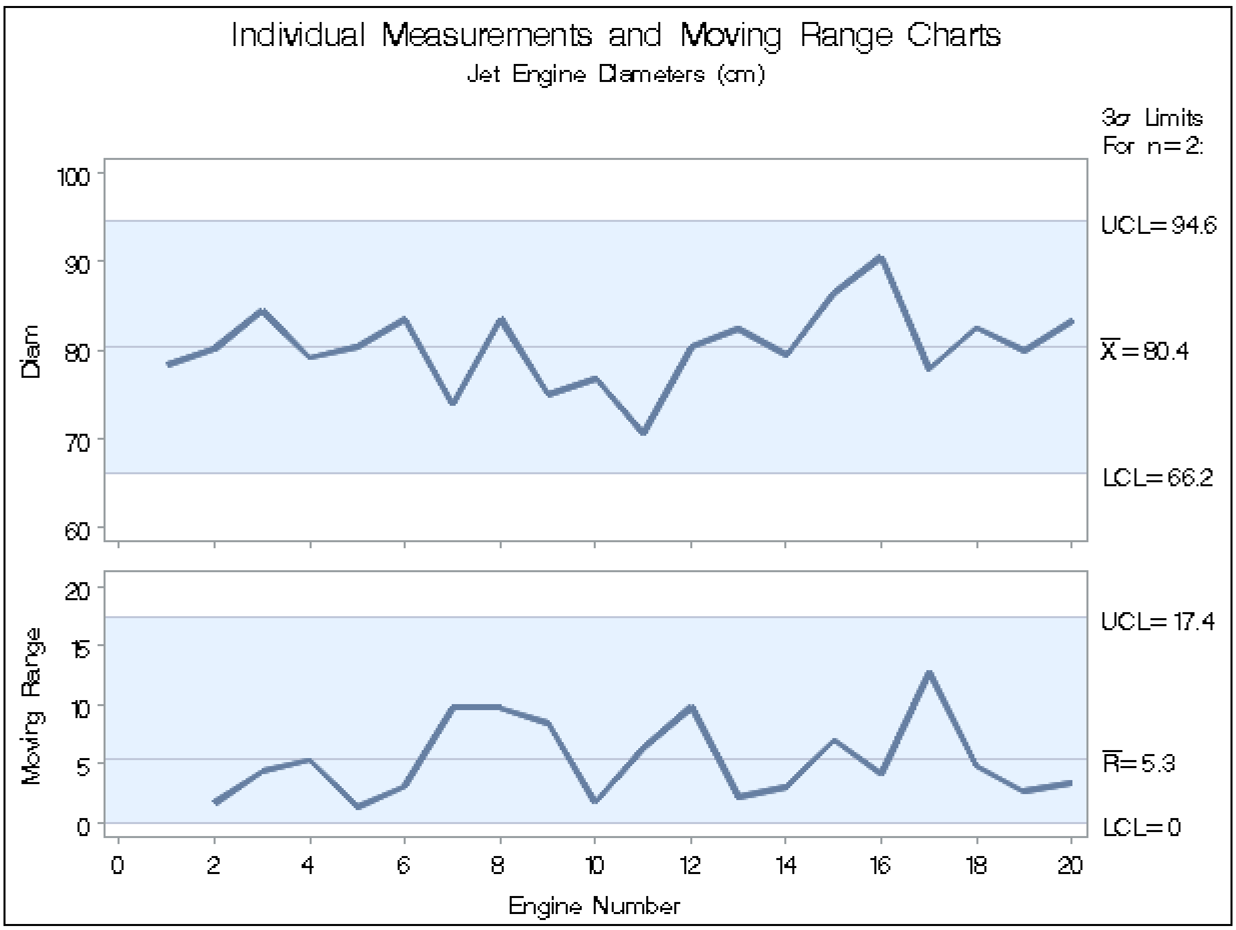
[4]Technically, the data for individual measurements and moving range charts are not arranged in rational subgroups. The term *subgroup-variable* is used for consistency with other chart statements in the SHEWHART procedure, and it is convenient to think of the "subgroups" as consisting of single measurements.

```
ods graphics off;
title 'Individual Measurements and Moving Range Charts';
title2 'Jet Engine Diameters (cm)';
proc shewhart data=Jets;
   irchart Diam*Engine;
run;
```

This example illustrates the basic form of the IRCHART statement. After the keyword IRCHART, you specify the *process* to analyze (in this case, Diam), followed by an asterisk and the *subgroup-variable* (Engine).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Figure 17.27 Individual Measurements and Moving Range Charts (Traditional Graphics)



Each point on the individual measurements chart indicates the inner diameter of a particular engine. Each point on the moving range chart indicates the range of the two most recent measurements. For instance, the moving range plotted for the second engine is $|78.4 - 80.1| = 1.7$. No moving range is plotted for the first engine. Since all of the individual measurements and moving ranges lie within the control limits, it can be concluded that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in section "Limits for Individual Measurements and Moving Range Charts" on page 1496. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1480.

Saving Individual Measurements and Moving Ranges

NOTE: See *Individual Measurement and Moving Range Charts* in the SAS/QC Sample Library.

In this example, the IRCHART statement is used to create an output data set containing individual measurements and moving ranges. The following statements read the diameter measurements from the data set Jets (see "Creating Individual Measurements and Moving Range Charts" on page 1474) and create a data set named Jetinfo:

```
proc shewhart data=Jets;
   irchart Diam*Engine / outhistory = Jetinfo
                         nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the charts, which would be identical to those in Figure 17.27. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the IRCHART statement. A complete list of options is presented in the section "Syntax: IRCHART Statement" on page 1483.

Figure 17.28 contains a partial listing of Jetinfo.

Figure 17.28 The Data Set Jetinfo

Individual Measurements and Moving Ranges for Diameters

Engine	Diam	DiamR
1	78.4	.
2	80.1	1.7
3	84.4	4.3
4	79.1	5.3
5	80.4	1.3

The data set Jetinfo contains one observation for each engine, and it includes three variables.

- Engine contains the subgroup index.
- Diam contains the individual measurements.
- DiamR contains the moving ranges.

Note that the variable containing the moving ranges is named by adding the suffix character *R* to the *process* Diam specified in the IRCHART statement.

For more information, see "OUTHISTORY= Data Set" on page 1498.

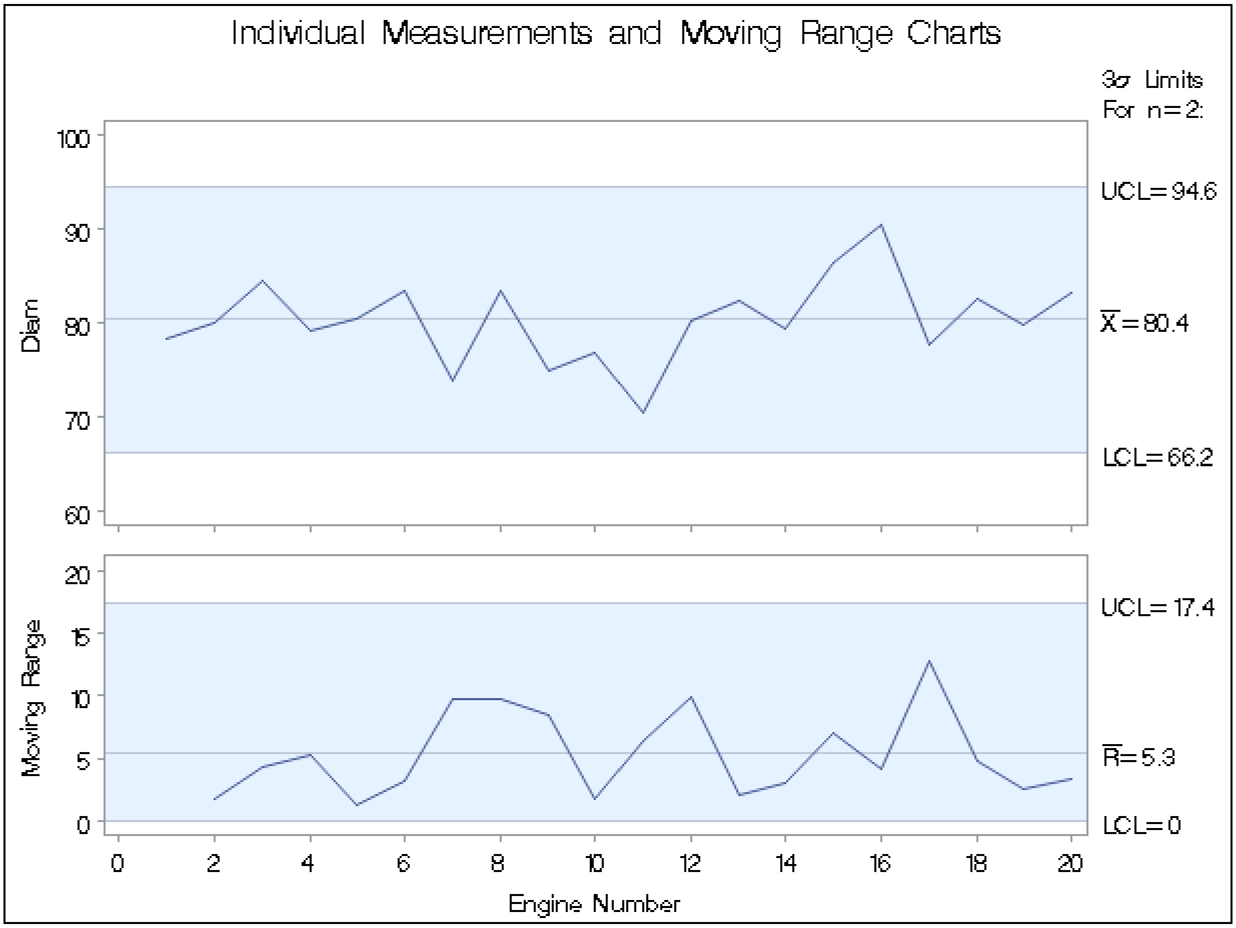
Reading Individual Measurements and Moving Ranges

NOTE: See *Individual Measurement and Moving Range Charts* in the SAS/QC Sample Library.

In some applications, both individual measurements and moving ranges may be provided. You can read this type of data set by specifying it with the HISTORY= option in the PROC SHEWHART statement. For example, the following statements read the data set Jetinfo (see Figure 17.28) and create the charts shown in Figure 17.29:

```
symbol h = .8;
title 'Individual Measurements and Moving Range Charts';
proc shewhart history=Jetinfo;
   irchart Diam*Engine;
run;
```

Figure 17.29 Charts Produced from Summary Data Set Jetinfo



A HISTORY= data set used with the IRCHART statement must contain the following variables:

- subgroup variable
- individual measurements variable
- moving range variable

Furthermore, the name of the moving range variable must begin with the *process* name specified in the IRCHART statement and end with the special suffix character *R*. If the name does not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename this variable for the duration of the procedure step (see "Creating Charts for Means and Ranges from Summary Data" on page 1841). For more information, see "HISTORY= Data Set" on page 1502.

Saving Control Limits

NOTE: See *Individual Measurement and Moving Range Charts* in the SAS/QC Sample Library.

You can save the control limits for individual measurements and moving range charts in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1480) or modify the limits with a DATA step program.

The following statements read the diameter measurements from the data set Jets (see "Creating Individual Measurements and Moving Range Charts" on page 1474) and save the control limits displayed in Figure 17.27 in a data set named Jetlim:

```
proc shewhart data=Jets;
   irchart Diam*Engine / outlimits = Jetlim
                         nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set Jetlim is listed in Figure 17.30.

Figure 17.30 The Data Set Jetlim Containing Control Limit Information

Control Limits for Diameters

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLI_	_MEAN_	_UCLI_
Diam	Engine	ESTIMATE	2	.002699796	3	66.2290	80.39	94.5510

LCLR	_R_	_UCLR_	_STDDEV_
0	5.32632	17.3986	4.72032

The data set Jetlim contains one observation with the limits for *process* Diam. The variables _LCLI_ and _UCLI_ contain the control limits for the individual measurements, and the variable _MEAN_ contains the central line. The variables _LCLR_ and _UCLR_ contain the control limits for the moving ranges, and the variable _R_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the number of consecutive measurements used to compute the moving ranges, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. For more information, see "OUTLIMITS= Data Set" on page 1497.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Jets;
   irchart Diam*Engine / outtable=Jtable
                         nochart;
run;
```

The data set Jtable is listed in Figure 17.31.

Figure 17.31 The Data Set Jtable

Summary Statistics and Control Limit Information

VAR	Engine	_SIGMAS_	_LIMITN_	_LCLI_	_SUBI_	_MEAN_	_UCLI_	_STDDEV_	_EXLIM_
Diam	1	3	2	66.2290	78.4	80.39	94.5510	4.72032	
Diam	2	3	2	66.2290	80.1	80.39	94.5510	4.72032	
Diam	3	3	2	66.2290	84.4	80.39	94.5510	4.72032	
Diam	4	3	2	66.2290	79.1	80.39	94.5510	4.72032	
Diam	5	3	2	66.2290	80.4	80.39	94.5510	4.72032	
Diam	6	3	2	66.2290	83.5	80.39	94.5510	4.72032	
Diam	7	3	2	66.2290	73.8	80.39	94.5510	4.72032	
Diam	8	3	2	66.2290	83.5	80.39	94.5510	4.72032	
Diam	9	3	2	66.2290	75.0	80.39	94.5510	4.72032	
Diam	10	3	2	66.2290	76.8	80.39	94.5510	4.72032	
Diam	11	3	2	66.2290	70.5	80.39	94.5510	4.72032	
Diam	12	3	2	66.2290	80.3	80.39	94.5510	4.72032	
Diam	13	3	2	66.2290	82.4	80.39	94.5510	4.72032	
Diam	14	3	2	66.2290	79.4	80.39	94.5510	4.72032	
Diam	15	3	2	66.2290	86.4	80.39	94.5510	4.72032	
Diam	16	3	2	66.2290	90.5	80.39	94.5510	4.72032	
Diam	17	3	2	66.2290	77.7	80.39	94.5510	4.72032	
Diam	18	3	2	66.2290	82.5	80.39	94.5510	4.72032	
Diam	19	3	2	66.2290	79.9	80.39	94.5510	4.72032	
Diam	20	3	2	66.2290	83.2	80.39	94.5510	4.72032	

LCLR	_SUBR_	_R_	_UCLR_	_EXLIMR_
0	.	5.32632	17.3986	
0	1.7	5.32632	17.3986	
0	4.3	5.32632	17.3986	
0	5.3	5.32632	17.3986	
0	1.3	5.32632	17.3986	
0	3.1	5.32632	17.3986	
0	9.7	5.32632	17.3986	
0	9.7	5.32632	17.3986	
0	8.5	5.32632	17.3986	
0	1.8	5.32632	17.3986	
0	6.3	5.32632	17.3986	
0	9.8	5.32632	17.3986	
0	2.1	5.32632	17.3986	
0	3.0	5.32632	17.3986	
0	7.0	5.32632	17.3986	
0	4.1	5.32632	17.3986	
0	12.8	5.32632	17.3986	
0	4.8	5.32632	17.3986	
0	2.6	5.32632	17.3986	
0	3.3	5.32632	17.3986	

This data set contains one observation for each subgroup. The variables _SUBI_ and _SUBR_ contain the individual measurements and moving ranges. The variables _LCLI_ and _UCLI_ contain the lower and upper control limits for the individual measurements chart, and the variables _LCLR_ and _UCLR_ contain the lower and upper control limits for the moving range chart. The variable _MEAN_ contains the central line of the individual measurements chart, and the variable _R_ contains the central line of the moving range chart. The variables _VAR_ and Engine contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1499.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Jtable and display charts (not shown here) identical to those in Figure 17.27:

```
title 'Individual Measurements and Moving Range Control Charts';
title2 'Jet Engine Diameters (cm)';
proc shewhart table=Jtable;
   irchart Diam*Engine;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2096).

For more information, see "TABLE= Data Set" on page 1503.

Reading Preestablished Control Limits

NOTE: See *Individual Measurement and Moving Range Charts* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Jetlim saved control limits computed from the measurements in Jets. This example shows how these limits can be applied to data for an additional 20 jet engines provided in the following data set:

```
data Jets2;
   input Engine Diam @@;
   label Diam   = "Inner Diameter (cm)"
         Engine = "Engine Number";
   datalines;
21 81.8  22 87.5  23 80.0  24 89.3  25 83.9
26 76.3  27 75.8  28 82.4  29 82.6  30 77.7
31 79.3  32 81.4  33 76.8  34 75.9  35 86.3
36 77.4  37 80.9  38 87.1  39 85.7  40 73.3
;
```

The following statements create individual measurements and moving range charts for the data in Jets2 using the control limits in Jetlim:

```
options nogstyle;
goptions ftext=swiss;
title 'Individual Measurements and Moving Range Control Charts';
proc shewhart data=Jets2 limits=Jetlim;
   irchart Diam*Engine / cframe = vigb
                         cconnect = yellow
                         coutfill = red
                         cinfill = vlib;
run;
options gstyle;
```

The charts are shown in Figure 17.32. The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the IRCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently.

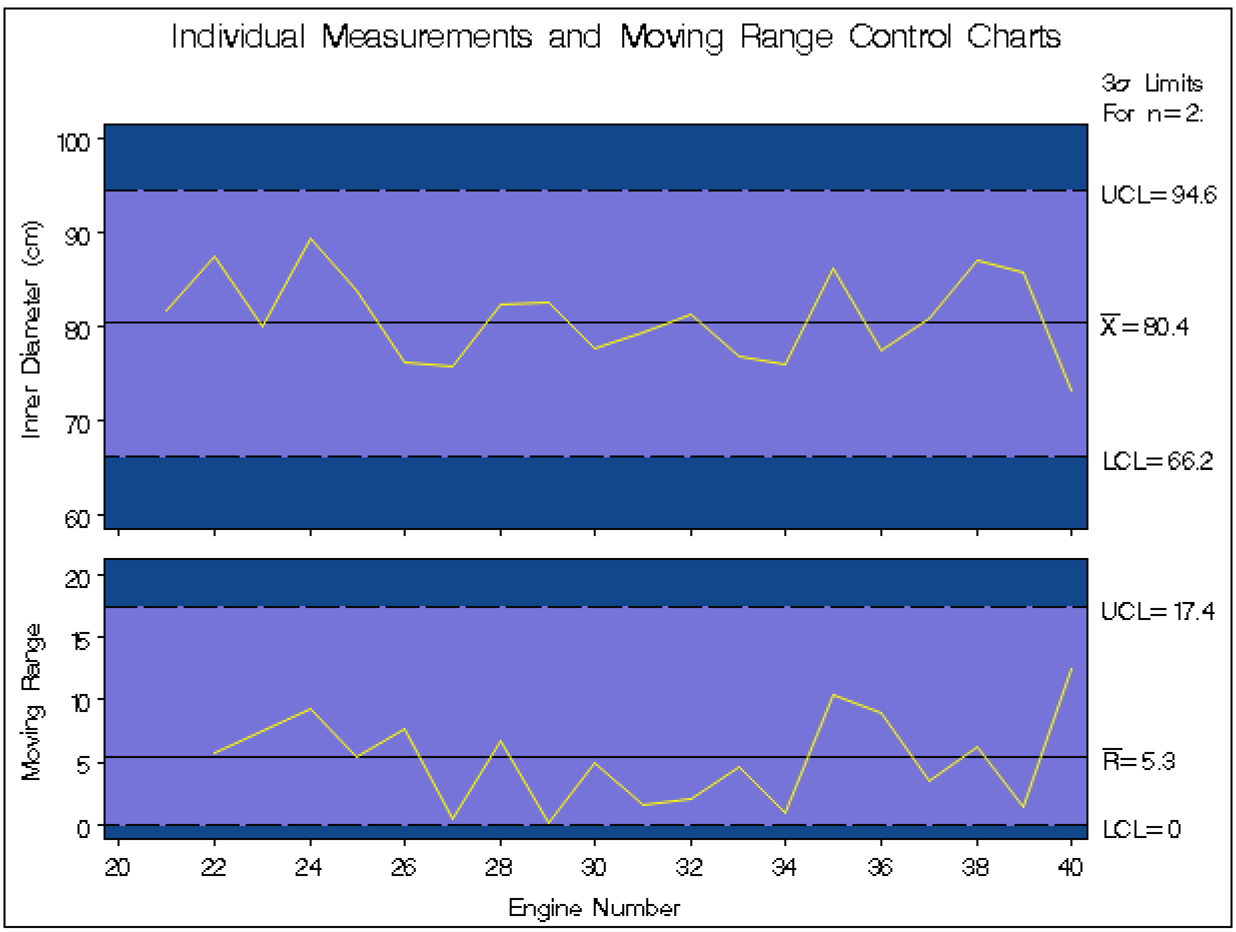
The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Diam
- the value of _SUBGRP_ matches the *subgroup-variable* name Engine

The charts indicate that the process is in control, since all the individual measurements and moving ranges lie within their respective control limits.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1501 for details concerning the variables that you must provide.

Figure 17.32 Charts for Second Set of Engine Noise Levels (Traditional Graphics with NOGSTYLE



Specifying the Computation of the Moving Range

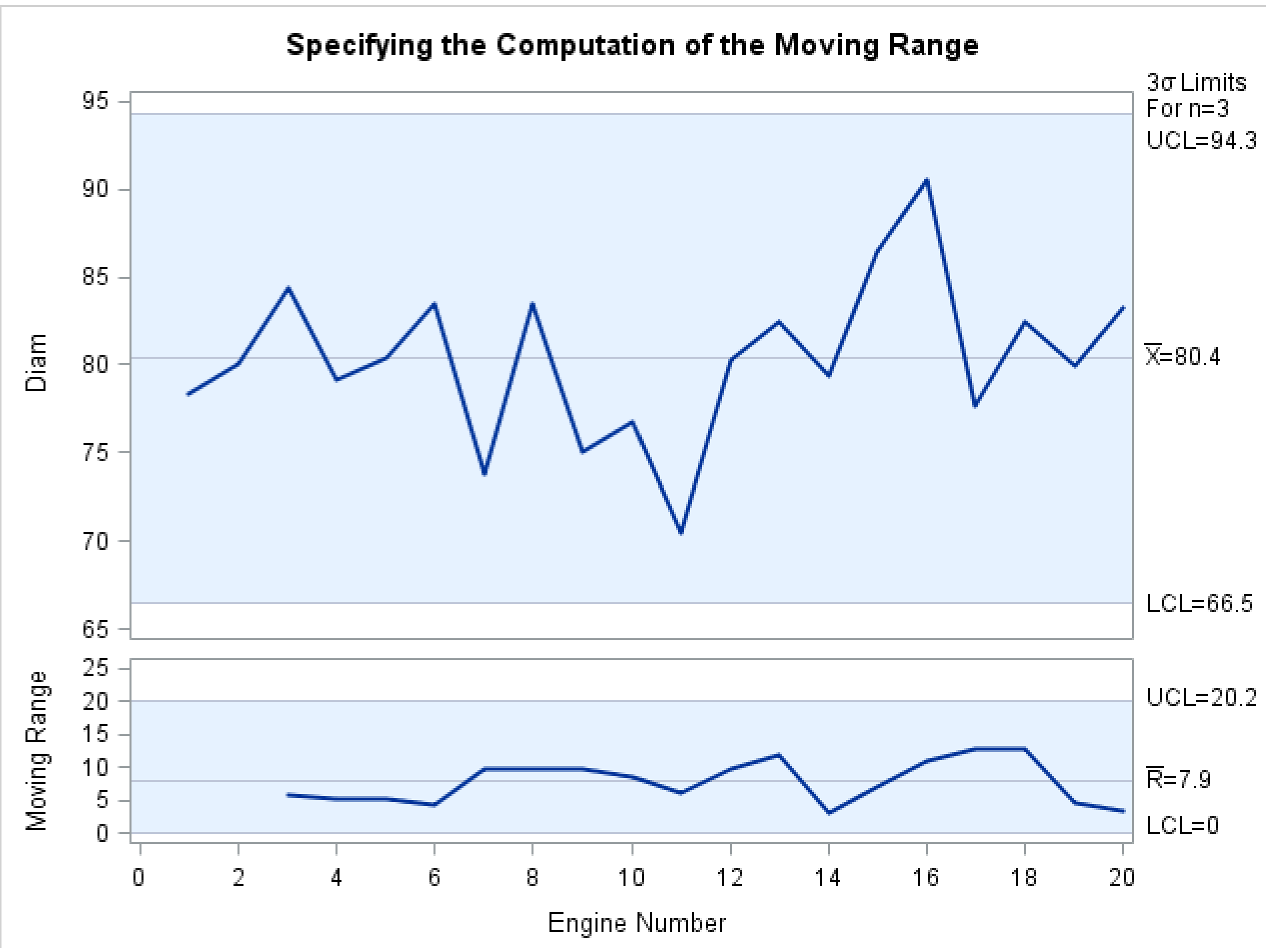
NOTE: See *Individual Measurement and Moving Range Charts* in the SAS/QC Sample Library.

By default, the IRCHART statement uses two consecutive measurements to calculate moving ranges. However, you can specify a different number of measurements to use, as illustrated by the following statements:

```
ods graphics on;
title 'Specifying the Computation of the Moving Range';
proc shewhart data=Jets;
   irchart Diam*Engine / limitn=3 odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the charts are created by using ODS Graphics instead of traditional graphics. The LIMITN= option specifies the number of consecutive measurements used to compute the moving ranges. The resulting charts are shown in Figure 17.33.

Figure 17.33 Computing Moving Ranges from Three Consecutive Measurements (ODS Graphics)



Note that the LIMITN= value is displayed in the legend above the control limit labels. The charts indicate that the process is in control, since all the points lie within the control limits.

Syntax: IRCHART Statement

The basic syntax for the IRCHART statement is as follows:

IRCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

IRCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / < *options* > **;**

You can use any number of IRCHART statements in the SHEWHART procedure. The components of the IRCHART statement are described as follows:

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the individual measurements. For an example, see "Creating Individual Measurements and Moving Range Charts" on page 1474.
- If individual measurements and moving ranges are read from a HISTORY= data set, *process* must be the name of the variable containing the individual measurements as well as the prefix of the variable containing the moving ranges in the HISTORY= data set. For an example, see "Saving Individual Measurements and Moving Ranges" on page 1476.
- If individual measurements, moving ranges, and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1478.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct individual measurements and moving range charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   irchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding IRCHART statement, Day is the subgroup variable. Note that each "subgroup" consists of a single observation. For details, see "Subgroup Variables" on page 1871.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2026 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the individual measurements and moving ranges.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2025 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create charts using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   irchart Weight*Day='*';
run;
```

options

enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The section "Summary of Options", which follows, lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1946 describes each option in detail.

Summary of Options

The following tables list the IRCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1946.

Table 17.19 IRCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	requests probability limits for chart
LIMITN=	specifies either nominal sample size for fixed control limits or varying limits
MRRESTART	restarts the moving range computation at missing values
NOREADLIMITS	computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)

Table 17.19 *continued*

Option	Description
SIGMAS=	specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	specifies color for area inside control limits
CLIMITS=	specifies color of control limits, central line, and related labels
LCLLABEL=	specifies label for lower control limit on individual measurements chart
LCLLABEL2=	specifies label for lower control limit on moving range chart
LIMLABSUBCHAR=	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	specifies line type for control limits
NDECIMAL=	specifies number of digits to right of decimal place in default labels for control limits and central line on individual measurements chart
NDECIMAL2=	specifies number of digits to right of decimal place in default labels for control limits and central line on moving range chart
NOCTL	suppresses display of central line on individual measurements chart
NOCTL2	suppresses display of central line on moving range chart
NOLCL	suppresses display of lower control limit on individual measurements chart
NOLCL2	suppresses display of lower control limit on moving range chart
NOLIMIT0	suppresses display of zero lower control limit on moving range chart
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit on individual measurements chart
NOUCL2	suppresses display of upper control limit on moving range chart
RSYMBOL=	specifies label for central line on moving range chart
UCLLABEL=	specifies label for upper control limit on individual measurements chart
UCLLABEL2=	specifies label for upper control limit on moving range chart

Table 17.19 *continued*

Option	Description
WLIMITS=	specifies width for control limits and central line
XSYMBOL=	specifies label for central line on individual measurements chart
Process Mean and Standard Deviation Options	
MU0=	specifies known value of μ_0 for process mean μ
SIGMA0=	specifies known value σ_0 for process standard deviation σ
SMETHOD=	specifies method for estimating process standard deviation σ
TYPE=	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	labels every point on individual measurements chart
ALLLABEL2=	labels every point on moving range chart
CLABEL=	specifies color for labels
CCONNECT=	specifies color for line segments that connect points on chart
CFRAMELAB=	specifies fill color for frame around labeled points
CNEEDLES=	specifies color for needles that connect points to central line
COUT=	specifies color for portions of line segments that connect points outside control limits
COUTFILL=	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	specifies angle at which labels are drawn
LABELFONT=	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=	labels points outside control limits on individual measurements chart
OUTLABEL2=	labels points outside control limits on moving range chart
SYMBOLLEGEND=	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=	specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	computes zone widths independently above and below center line

Table 17.19 *continued*

Option	Description
NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	suppresses tests across *phase* boundaries
TESTS=	specifies tests for special causes for the individual measurements chart
TEST2RUN=	specifies length of pattern for Test 2
TEST3RUN=	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL=	provides labels for points where test is positive
TESTLABEL*n*=	specifies label for *n*th test for special causes
TESTNMETHOD=	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=	enables tests for special causes to be reset
WESTGARD=	requests that Westgard rules be applied to the individual measurements chart
ZONELABELS	adds labels A, B, and C to zone lines for individual measurements chart
ZONES	adds lines to individual measurements chart delineating zones A, B, and C
ZONEVALPOS=	specifies position of ZONEVALUES labels
ZONEVALUES	labels individual measurements chart zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	specifies color for labels indicating points where test is positive
CTESTSYMBOL=	specifies color for symbol used to plot points where test is positive
CZONES=	specifies color for lines and labels delineating zones A, B, and C
LTESTS=	specifies type of line connecting points where test is positive
LZONES=	specifies line type for lines delineating zones A, B, and C
TESTFONT=	specifies software font for labels at points where test is positive
TESTHEIGHT=	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	specifies plot symbol for points where test is positive
TESTSYMBOLHT=	specifies symbol height for points where test is positive
WTESTS=	specifies width of line connecting points where test is positive

Table 17.19 *continued*

Option	Description
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill colors for frame for plot area
CTEXT=	specifies color for tick mark values and axis labels
DISCRETE	produces horizontal axis for discrete numeric group values
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default to moving range chart
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis of individual measurements chart
VAXIS2=	specifies major tick mark values for vertical axis of moving range chart
VFORMAT=	specifies format for primary vertical axis tick mark labels
VFORMAT2=	specifies format for secondary vertical axis tick mark labels
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=	specifies width of axis lines
Plot Layout Options	
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process only when exceptions occur

Table 17.19 *continued*

Option	Description
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NOCHART	suppresses creation of charts
NOCHART2	suppresses creation of moving range chart
NOFRAME	suppresses frame for plot area
NPANELPOS=	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
SEPARATE	displays individual measurements and moving range charts on separate screens or pages
TOTPANELS=	specifies number of pages or screens to be used to display chart
YPCT1=	specifies length of vertical axis on individual measurements chart as a percentage of sum of lengths of vertical axes for individual measurements and moving range charts
ZEROSTD	displays individual measurements chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= and HREF2= options
CVREF=	specifies color for lines requested by VREF= and VREF2= options
HREF=	specifies position of reference lines perpendicular to horizontal axis on individual measurements chart
HREF2=	specifies position of reference lines perpendicular to horizontal axis on moving range chart
HREFDATA=	specifies position of reference lines perpendicular to horizontal axis on individual measurements chart
HREF2DATA=	specifies position of reference lines perpendicular to horizontal axis on moving range chart
HREFLABELS=	specifies labels for HREF= lines
HREF2LABELS=	specifies labels for HREF2= lines
HREFLABPOS=	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	specifies line type for HREF= and HREF2= lines
LVREF=	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	specifies position of reference lines perpendicular to vertical axis on individual measurements chart

Table 17.19 *continued*

Option	Description
VREF2=	specifies position of reference lines perpendicular to vertical axis on moving range chart
VREFLABELS=	specifies labels for VREF= lines
VREF2LABELS=	specifies labels for VREF2= lines
VREFLABPOS=	position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=	specifies line type for grid requested with the ENDGRID option
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme points are clipped
CLIPLEGEND=	specifies text for clipping legend
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features to individual measurements chart
ANNOTATE2=	specifies annotate data set that adds features to moving range chart
DESCRIPTION=	specifies description of individual measurements chart's GRSEG catalog entry
DESCRIPTION2=	specifies description of moving range chart's GRSEG catalog entry
FONT=	specifies software font for labels and legends on charts
LTMARGIN=	specifies width of left margin area for plot requested with LTMPLOT= option
LTMPLOT=	requests univariate plot in left margin
NAME=	specifies name of individual measurements chart's GRSEG catalog entry
NAME2=	specifies name of moving range chart's GRSEG catalog entry
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option

Table 17.19 *continued*

Option	Description
RTMARGIN=	specifies width of right margin area for plot requested with LTMPLOT= option
RTMPLOT=	requests univariate plot in right margin
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	shades areas between control limits and connected points lying outside the limits
STARFILL=	specifies a variable identfying groups of stars filled with different colors
STARS=	specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	specifies the control limit infill transparency
MARKERS	plots subgroup points with markers
NOBLOCKREF	suppresses block and phase reference lines
NOBLOCKREFFILL	suppresses block and phase wall fills
NOFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOPHASEREF	suppresses block and phase reference lines
NOPHASEREFFILL	suppresses block and phase wall fills
NOREF	suppresses block and phase reference lines
NOREFFILL	suppresses block and phase wall fills
NOSTARFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	specifies a graph footnote
ODSFOOTNOTE2=	specifies a secondary graph footnote
ODSLEGENDEXPAND	specifies that legend entries contain all levels observed in the data
ODSTITLE=	specifies a graph title
ODSTITLE2=	specifies a secondary graph title
OUTFILLTRANSPARENCY=	specifies control limit outfill transparency
OVERLAYURL=	specifies URLs to associate with overlay points
OVERLAY2URL=	specifies URLs to associate with overlay points on secondary chart
PHASEPOS=	specifies vertical position of phase legend

Table 17.19 *continued*

Option	Description
PHASEREFLEVEL=	associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	draws central line labels using a simulated software font
STARTRANSPARENCY=	specifies star fill transparency
URL=	specifies a variable whose values are URLs to be associated with subgroups
URL2=	specifies a variable whose values are URLs to be associated with subgroups on secondary chart
Input Data Set Options	
MISSBREAK	specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	creates output data set containing subgroup summary statistics
OUTINDEX=	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	creates output data set containing control limits
OUTTABLE=	creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	specifies α value and type for computing capability index confidence limits
LSL=	specifies list of lower specification limits
TARGET=	specifies list of target values
USL=	specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for *block-variable* legend
BLOCKLABTYPE=	specifies text size of *block-variable* legend

Table 17.19 *continued*

Option	Description
BLOCKPOS=	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE=	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	specifies color for STARCIRCLES= circles
CSTARFILL=	specifies color for filling stars
CSTAROUT=	specifies outline color for stars exceeding inner or outer circles
CSTARS=	specifies color for outlines of stars
LSTARCIRCLES=	specifies line types for STARCIRCLES= circles
LSTARS=	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	specifies reference circles for stars
STARINRADIUS=	specifies inner radius of stars
STARLABEL=	specifies vertices to be labeled
STARLEGEND=	specifies style of legend for star vertices
STARLEGENDLAB=	specifies label for STARLEGEND= legend
STAROUTRADIUS=	specifies outer radius of stars
STARSPECS=	specifies method used to standardize vertex variables
STARSTART=	specifies angle for first vertex
STARTYPE=	specifies graphical style of star
STARVERTICES=	superimposes star at each point on individual measurements chart
WSTARCIRCLES=	specifies width of STARCIRCLES= circles
WSTARS=	specifies width of STARVERTICES= stars

Table 17.19 *continued*

Option	Description
Overlay Options	
CCOVERLAY=	specifies colors for primary chart overlay line segments
CCOVERLAY2=	specifies colors for secondary chart overlay line segments
COVERLAY=	specifies colors for primary chart overlay plots
COVERLAY2=	specifies colors for secondary chart overlay plots
COVERLAYCLIP=	specifies color for clipped points on overlays
LOVERLAY=	specifies line types for primary chart overlay line segments
LOVERLAY2=	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=	specifies variables to overlay on primary chart
OVERLAY2=	specifies variables to overlay on secondary chart
OVERLAY2HTML=	specifies links to associate with secondary chart overlay points
OVERLAY2ID=	specifies labels for secondary chart overlay points
OVERLAY2SYM=	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
OVERLAYHTML=	specifies links to associate with primary chart overlay points
OVERLAYID=	specifies labels for primary chart overlay points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols for primary chart overlays
OVERLAYSYMHT=	specifies symbol heights for primary chart overlays
WOVERLAY=	specifies widths of primary chart overlay line segments
WOVERLAY2=	specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	specifies a variable whose values create links to be associated with subgroups
HTML2=	specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	specifies plot character for clipped points
CONNECTCHAR=	specifies character used to form line segments that connect points on chart
HREFCHAR=	specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	specifies characters indicating *symbol-variable*

Table 17.19 *continued*

Option	Description
TESTCHAR=	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	specifies line character for VREF= and VREF2= lines
ZONECHAR=	specifies character for lines that delineate zones for tests for special causes

Details: IRCHART Statement

Constructing Charts for Individual Measurements and Moving Ranges

The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
X_i	the ith individual measurement
$\bar{X}$	mean of the individual measurements, computed as $(X_1 + \cdots + X_N)/N$, where N is the number of individual measurements
n	number of consecutive measurements used to calculate the moving ranges (by default, $n = 2$)
R_i	moving range computed for the ith subgroup (corresponding to the ith individual measurement). If $i < n$, then R_i is assigned a missing value. Otherwise, $$R_i = \max(X_i, X_{i-1}, \ldots, X_{i-n+1}) - \min(X_i, X_{i-1}, \ldots, X_{i-n+1})$$ This formula assumes that $X_i, X_{i-1}, \ldots, X_{i-n+1}$ are nonmissing.
$\bar{R}$	average of the nonmissing moving ranges, computed as $$\frac{R_n + R_{n+1} \cdots + R_N}{N + 1 - n}$$
$d_2(n)$	expected value of the range of n independent normally distributed variables with unit standard deviation
$d_3(n)$	standard error of the range of n independent observations from a normal population with unit standard deviation
z_p	100pth percentile ($0 < p < 1$) of the standard normal distribution
$D_p(n)$	100pth percentile ($0 < p < 1$) of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on an individual measurements chart, indicates the value of a measurement (X_i).

Each point on a moving range chart indicates the value of a moving range (R_i). With $n = 2$, for example, if the first three measurements are 3.4, 3.7, and 3.6, the first moving range is missing, the second moving range is $|3.7 - 3.4| = 0.3$, and the third moving range is $|3.6 - 3.7| = 0.1$.

Central Lines

By default, the central line on an individual measurements chart indicates an estimate for μ, which is computed as $\bar{X}$. If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

The central line on a moving range chart indicates an estimate for the expected moving range, computed as $d_2(n)\hat{\sigma}$ where $\hat{\sigma} = \bar{R}/d_2(n)$. If you specify a known value ($\hat{\sigma}_0$) for σ, the central line indicates the value of $d_2(n)\sigma_0$.

Control Limits

You can compute the limits

- as a specified multiple (k) of the standard errors of X_i and R_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that X_i or R_i exceeds the limits

The following table provides the formulas for the limits:

Table 17.20 Limits for Individual Measurements and Moving Range Charts

Control Limits	
Individual Measurements Chart	LCL = lower control limit = $\bar{X} - k\hat{\sigma}$ UCL = upper control limit = $\bar{X} + k\hat{\sigma}$
Moving Range Chart	LCL = lower control limit = $\max(d_2(n)\hat{\sigma} - kd_3(n)\hat{\sigma}, 0)$ UCL = upper control limit = $d_2(n)\hat{\sigma} + kd_3(n)\hat{\sigma}$
Probability Limits	
Individual Measurements Chart	LCL = lower control limit = $\bar{X} - z_{\alpha/2}\hat{\sigma}$ UCL = upper control limit = $\bar{X} + z_{\alpha/2}\hat{\sigma}$
Moving Range Chart	LCL = lower control limit = $D_{\alpha/2}(n)\hat{\sigma}$ UCL = upper control limit = $D_{1-\alpha/2}(n)\hat{\sigma}$

The formulas assume that the measurements are normally distributed. Note that the probability limits for the moving range are asymmetric about the central line. If standard values μ_0 and σ_0 are available for μ and σ, replace $\bar{X}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 17.20.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.

- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify n with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in the LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in the LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 17.21 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLI	lower control limit for individual measurements
LCLR	lower control limit for moving ranges
LIMITN	number of consecutive measurements used to compute moving ranges
LSL	lower specification limit
MEAN	process mean
_R_	value of central line on moving range chart
SIGMAS	multiple (k) of standard error of individual measurement or moving range
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the IRCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLI	upper control limit for individual measurements
UCLR	upper control limit for moving ranges range
USL	upper specification limit
VAR	*process* specified in the IRCHART statement

Notes:

1. If the limits are defined in terms of a multiple k of the standard errors of X_i and R_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.

2. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.

3. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1874 for computational details.

4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the IRCHART statement. For an example, see "Saving Control Limits" on page 1478.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves individual measurements and moving ranges. The following variables are saved:

- the *subgroup-variable*
- an individual measurements variable named by *process*
- a moving range variable named by *process* suffixed with *R*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

A variable containing the moving ranges is created for each *process* specified in the IRCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   irchart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, Width, WidthR, Diameter, and DiameterR.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Individual Measurements and Moving Ranges" on page 1476.

OUTTABLE= Data Set

The OUTTABLE= data set saves individual measurements, moving ranges, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on individual measurements chart
EXLIMR	control limit exceeded on moving range chart
LCLI	lower control limit for individual measurements
LCLR	lower control limit for moving range
LIMITN	number of consecutive measurements used to compute moving ranges
MEAN	process mean
_R_	average range
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBI	individual measurement
SUBR	moving range
TESTS	tests for special causes signaled on individual measurements chart
UCLI	upper control limit for individual measurements
UCLR	upper control limit for moving range
VAR	*process* specified in the IRCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).

2. The variable _TESTS_ is saved if you specify the TESTS= option. The kth character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.

3. The variables _EXLIM_, _EXLIMR_, and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1478.

ODS Tables

The following table summarizes the ODS tables that you can request with the IRCHART statement.

Table 17.22 ODS Tables Produced with the IRCHART Statement

Table Name	Description	Options
IRCHART	individual measurement and moving range chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section “Enabling and Disabling ODS Graphics” (Chapter 21, *SAS/STAT User’s Guide*). **NOTE:** In SAS/QC 13.1 the SHEWHART procedure does not support the creation of graphs that are editable with the ODS Graphics Editor.

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. IRCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in “Dictionary of Options: SHEWHART Procedure” on page 1946.

When ODS Graphics is in effect, the IRCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 17.23.

Table 17.23 ODS Graphics Produced by the IRCHART Statement

ODS Graph Name	Plot Description
IRChart	individual measurements and moving range chart

See Chapter 3, “SAS/QC Graphics,” for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read individual measurements from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the IRCHART statement must be a SAS variable in the data set. This variable provides measurements of items indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the IRCHART statement, must also be a SAS variable in the data set. Each observation in a DATA= data set must contain a measurement for each *process* and a value for the *subgroup-variable*. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option in the IRCHART statement (for an example, see "Displaying Stratification in Phases" on page 2031.

For an example of a DATA= data set, see "Creating Individual Measurements and Moving Range Charts" on page 1474.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=info limits=Conlims;
   irchart Weight*ID;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 17.20. The LIMITS= data set can also be created directly using a DATA step.

When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLI_, _MEAN_, _UCLI_, _LCLR_, _R_, and _UCLR_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 17.20

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'. See Example 17.12 for an illustration.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1480.

HISTORY= Data Set

You can read individual measurements and moving ranges from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure.

A HISTORY= data set used with the IRCHART statement must contain the following:

- the *subgroup-variable*
- an individual measurements variable for each *process*
- a moving range variable for each *process*

The name of the individual measurements variable must be the *process* specified in the IRCHART statement. The name of the moving range variable must be the prefix *process* concatenated with the special suffix character *R*. For example, consider the following statements:

```
proc shewhart history=Summary;
   irchart (Weight Yieldstrength) * ID;
run;
```

The data set Summary must include the variables ID, Weight, WeightR, YieldstrengthN, and YieldstrengthR.

Note that if you specify a *process* name that contains 32 characters, the name of the moving range variable must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with *R*.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*

- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2031 for an example).

For an example of a HISTORY= data set, see "Reading Individual Measurements and Moving Ranges" on page 1476.

TABLE= Data Set

You can read individual measurements, moving ranges, and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2096.

The following table lists the variables required in a TABLE= data set used with the IRCHART statement:

Table 17.24 Variables Required in a TABLE= Data Set

Variable	Description
LCLI	lower control limit for individual measurements
LCLR	lower control limit for moving range
LIMITN	number of consecutive measurements used to calculate moving ranges
MEAN	process mean
_R_	average moving range
subgroup-variable	values of the *subgroup-variable*
SUBI	individual measurements
SUBR	moving ranges
UCLI	upper control limit for individual measurements
UCLR	upper control limit for moving range

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.

- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1478.

Methods for Estimating the Standard Deviation

When control limits are computed from the input data, three methods (referred to as default, MAD and MMR) are available for estimating the process standard deviation σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \bar{R}/d_2(n)$$

where $\bar{R}$ is the average of the moving ranges, n is the number of consecutive individual measurements used to compute each moving range, and the unbiasing factor $d_2(n)$ is defined so that if the observations are normally distributed, the expected value of R_i is

$$E(R_i) = d_2(n_i)\sigma$$

This method is described in the American Society for Testing and Materials (1976).

MAD Method

If you specify SMETHOD=MAD, a median absolute deviation estimator is computed for σ, as described by Boyles (1997). It is computed as

$$\hat{\sigma} = \text{median}\{|X_i - \tilde{X}|, 1 \le i \le N\}/0.6745$$

where $\tilde{X}$ is the sample median.

MMR Method

If you specify SMETHOD=MMR, a median moving range estimator is computed for σ. This estimator is described by Boyles (1997). It is computed as

$$\hat{\sigma} = \tilde{R}/0.954$$

where $\tilde{R}$ is the median of the nonmissing moving ranges.

Interpreting Charts for Individual Measurements and Moving Ranges

Montgomery (1996) points out that a moving range chart should be interpreted with care because "the moving ranges are correlated, and this correlation may often induce a pattern or runs or cycles on the chart." For this reason Nelson (1982) recommends against plotting the moving ranges. Nelson notes that the assumption of normality is more critical for an individual measurements chart than for an $\bar{X}$ chart. You can use the NOCHART2 option in the IRCHART statement to specify that only the individual measurements chart is to be displayed. See Example 17.13 for an illustration. If, instead, you specify the SEPARATE option, the charts for individual measurements and moving ranges are displayed on separate screens.

An alternative method for creating an individual measurements chart is to use the XCHART statement, which uses an estimate of σ based on moving ranges of two consecutive measurements when the subgroup sample sizes are all equal to one. Note that the XCHART statement displays the control limit legend $n = 1$ to indicate the common subgroup sample size, whereas the IRCHART statement displays a legend that indicates the number of consecutive measurements used to compute the moving ranges (the "pseudo subgroup sample size").

Nelson (1982) explains that the reason for estimating the process standard deviation σ from moving ranges of two consecutive measurements rather than the sample standard deviation of the measurements is that "the moving range of two minimizes inflationary effects on the variability which are caused by trends and oscillations that may be present." Nelson suggests that any moving range that exceeds 3.5 times the average moving range should be removed from the calculation of the average moving range.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical (Individual measurements chart)	DATA=	*process*
Vertical (Individual measurements chart)	HISTORY=	subgroup measurement variable
Vertical (Individual measurements chart)	TABLE=	_SUBI_

You can specify distinct labels for the vertical axes of the individual measurements and moving range charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the individual measurements chart, and the second part labels the vertical axis of the moving range chart.

For example, the following sets of statements specify the label *Avg gap in mm* for the vertical axis of the individual measurements chart and the label *Range in mm* for the vertical axis of the moving range chart:

```
proc shewhart data=Doors;
   irchart Gap*Hour / split = '/' ;
   label Gap = 'Avg gap in mm/Range in mm';
run;

proc shewhart history=Doorhist;
   irchart Gap*Hour / split = '/' ;
   label Gap = 'Avg gap in mm/Range in mm';
run;

proc shewhart table=Doortab;
   irchart Gap*Hour / split = '/' ;
   label _SUBI_ = 'Avg gap in mm/Range in mm';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: IRCHART Statement

This section provides advanced examples of the IRCHART statement.

Example 17.11: Applying Tests for Special Causes

NOTE: See *IRCHART with Tests for Special Causes* in the SAS/QC Sample Library.

This example illustrates how you can apply tests for special causes to make an individual measurements chart more sensitive to special causes of variation. The following statements create a data set named Engines, which contains the weights for 25 jet engines:

```
data Engines;
   input ID Weight @@;
   label Weight='Engine Weight (lbs)'
         ID    ='Engine ID Number';
   datalines;
1711 1270   1712 1258   1713 1248   1714 1260
1715 1263   1716 1260   1717 1259   1718 1240
1719 1260   1720 1246   1721 1238   1722 1253
1723 1249   1724 1245   1725 1251   1726 1252
1727 1249   1728 1274   1729 1258   1730 1268
1731 1248   1732 1295   1733 1243   1734 1253
1735 1258
;
```

Individual measurements and moving range charts are used to monitor the weights. The following statements produce the tables shown in Output 17.11.1 and create the charts shown in Output 17.11.2:

```
title 'Tests for Special Causes Applied to Jet Engine Weights';
ods graphics on;
proc shewhart data=Engines;
   irchart Weight*ID /
       tests    = 1 to 8
       test2run = 7
       odstitle = title
       tabletest
       zonelabels
       markers;
run;
```

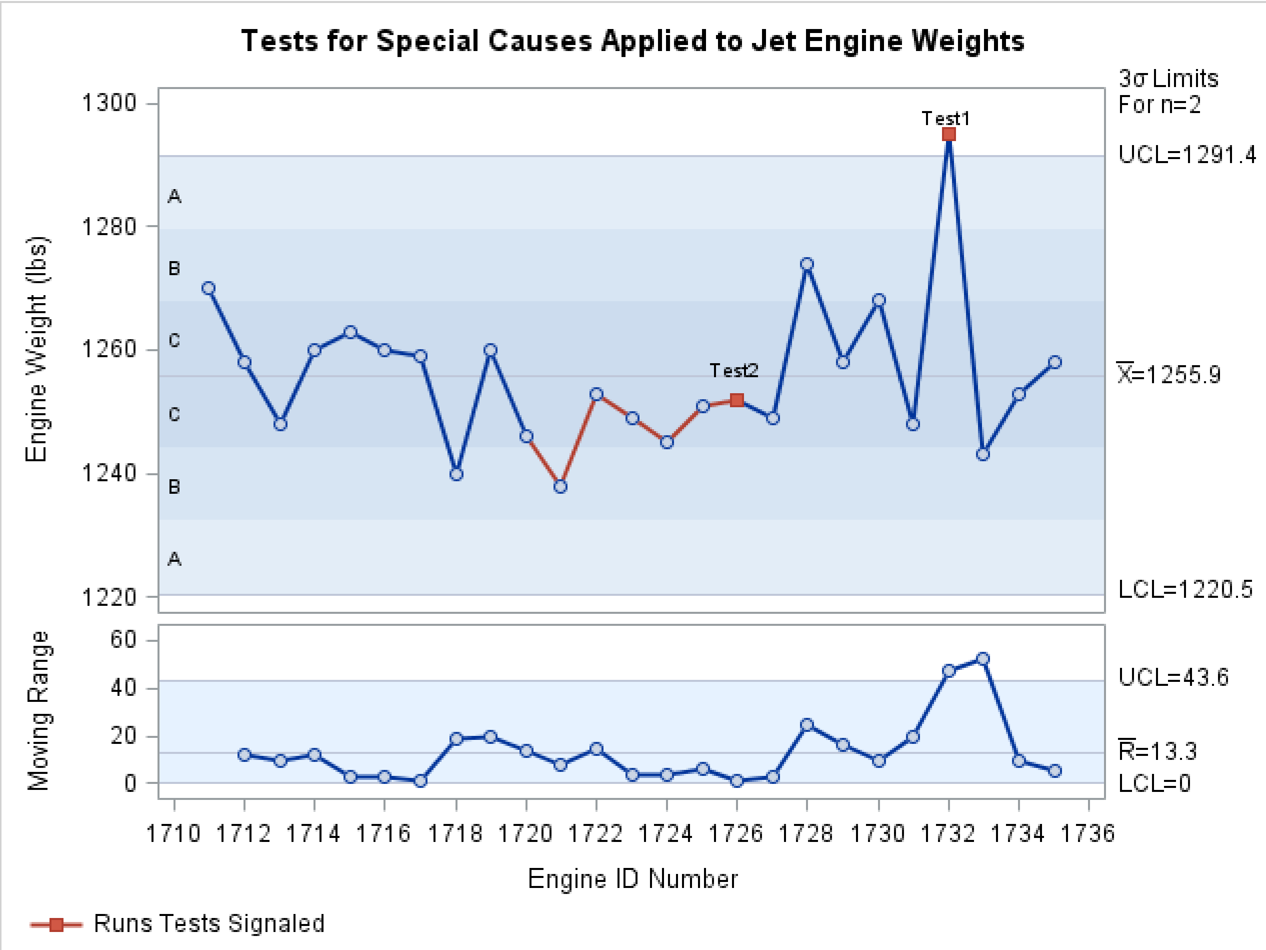
The TESTS= option applies eight tests for special causes, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2073. The TEST2RUN= option specifies the length of the pattern for Test 2. The TABLETESTS option requests a table of individual measurements, moving ranges, and control limits, and it adds a column indicating which measurements tested positive for special causes.

The ZONELABELS option displays zone lines and zone labels on the individual measurements chart. The zones are used to define the tests.

Output 17.11.1 Tabular Form of Individual Measurements and Moving Range Chart

Individual Measurements Chart Summary for Weight							
	3 Sigma Limits with n=2 for Weight				3 Sigma Limits with n=2 for Moving Range		
ID	Lower Limit	Weight	Upper Limit	Special Tests Signaled	Lower Limit	Moving Range	Upper Limit
1711	1220.4709	1270.0000	1291.3691		0	.	43.553759
1712	1220.4709	1258.0000	1291.3691		0	12.000000	43.553759
1713	1220.4709	1248.0000	1291.3691		0	10.000000	43.553759
1714	1220.4709	1260.0000	1291.3691		0	12.000000	43.553759
1715	1220.4709	1263.0000	1291.3691		0	3.000000	43.553759
1716	1220.4709	1260.0000	1291.3691		0	3.000000	43.553759
1717	1220.4709	1259.0000	1291.3691		0	1.000000	43.553759
1718	1220.4709	1240.0000	1291.3691		0	19.000000	43.553759
1719	1220.4709	1260.0000	1291.3691		0	20.000000	43.553759
1720	1220.4709	1246.0000	1291.3691		0	14.000000	43.553759
1721	1220.4709	1238.0000	1291.3691		0	8.000000	43.553759
1722	1220.4709	1253.0000	1291.3691		0	15.000000	43.553759
1723	1220.4709	1249.0000	1291.3691		0	4.000000	43.553759
1724	1220.4709	1245.0000	1291.3691		0	4.000000	43.553759
1725	1220.4709	1251.0000	1291.3691		0	6.000000	43.553759
1726	1220.4709	1252.0000	1291.3691	2	0	1.000000	43.553759
1727	1220.4709	1249.0000	1291.3691		0	3.000000	43.553759
1728	1220.4709	1274.0000	1291.3691		0	25.000000	43.553759
1729	1220.4709	1258.0000	1291.3691		0	16.000000	43.553759
1730	1220.4709	1268.0000	1291.3691		0	10.000000	43.553759
1731	1220.4709	1248.0000	1291.3691		0	20.000000	43.553759
1732	1220.4709	1295.0000	1291.3691	1	0	47.000000	43.553759
1733	1220.4709	1243.0000	1291.3691		0	52.000000	43.553759
1734	1220.4709	1253.0000	1291.3691		0	10.000000	43.553759
1735	1220.4709	1258.0000	1291.3691		0	5.000000	43.553759

Output 17.11.2 Tests for Special Causes



Output 17.11.1 and Output 17.11.2 indicate that Test 1 was positive for engine 1732 and Test 2 was positive for engine 1726. Test 1 detects one point beyond Zone A (outside the control limits) and Test 2 detects seven points (TEST2RUN=7) in a row on one side of the central line.

Example 17.12: Specifying Standard Values for the Process Mean and Standard Deviation

NOTE: See *Specifying Known Values for IRCHART* in the SAS/QC Sample Library.

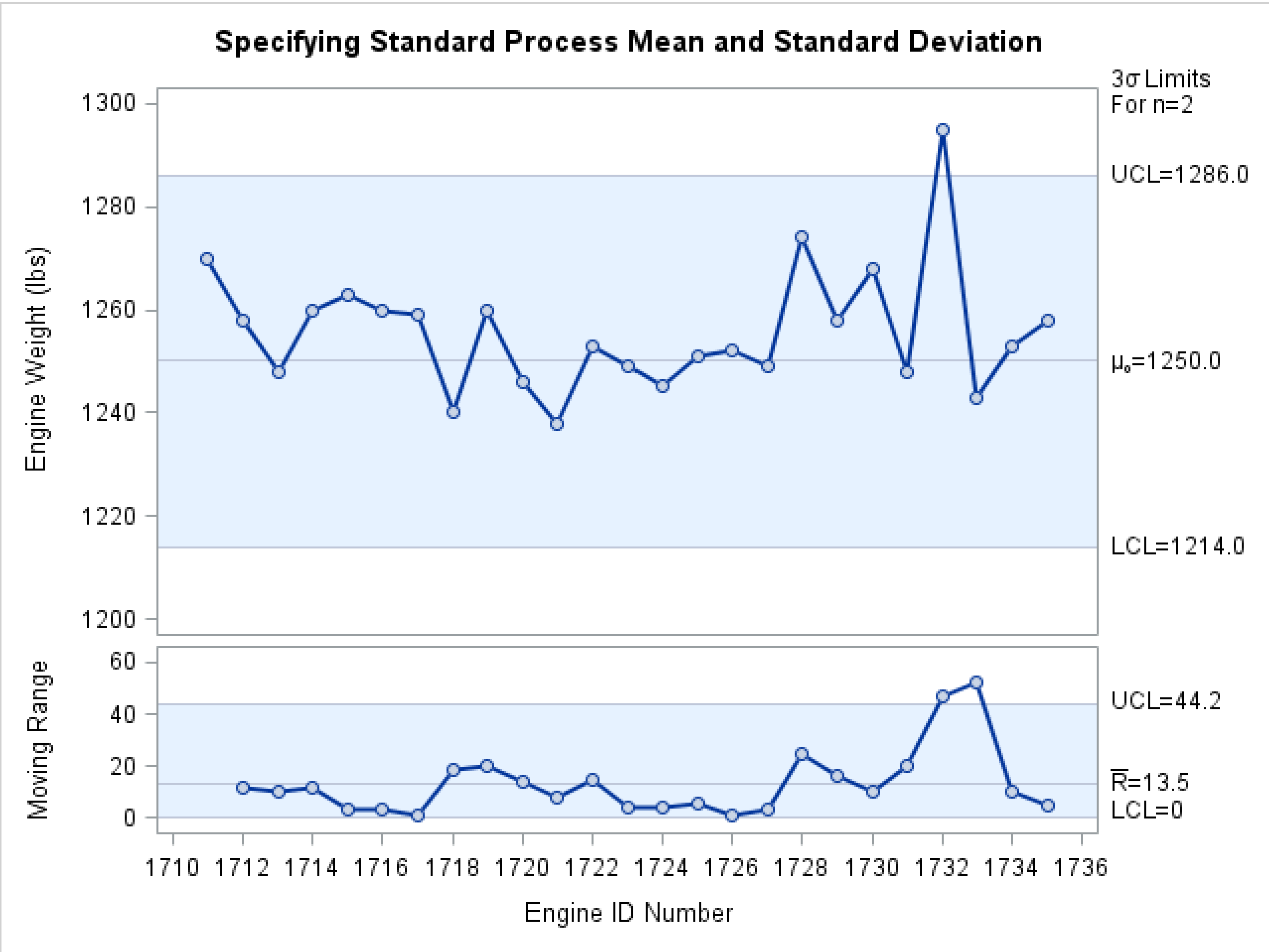
By default, the IRCHART statement estimates the process mean (μ) and standard deviation (σ) from the data, as in the previous example. However, there are applications in which known (standard) values μ_0 and σ_0 are available for these parameters based on previous experience or extensive sampling.

For example, suppose that the manufacturing process described in the previous example produces engines whose weights are normally distributed with a mean of 1250 and a standard deviation of 12. The following statements create individual measurements and moving range charts based on these values:

```
ods graphics on;
title 'Specifying Standard Process Mean and Standard Deviation';
proc shewhart data=Engines;
   irchart Weight*ID /
      odstitle = title
      mu0      = 1250
      sigma0   = 12
      xsymbol  = mu0
      markers;
run;
```

The charts are shown in Output 17.12.1. The MU0= option and SIGMA0= option specify μ_0 and σ_0. The XSYMBOL= option specifies the label for the central line on the individual measurements chart, and the keyword MU0 requests a label indicating that the central line is based on a standard value.

Output 17.12.1 Specifying Standard Values with MU0= and SIGMA0=



You can also specify μ_0 and σ_0 as the values of the variables _MEAN_ and _STDDEV_ in a LIMITS= data set. For example, the following statements create a LIMITS= data set with the standard values specified in the preceding IRCHART statement:

```
data Enginelimits;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'Weight';
   _subgrp_ = 'id';
   _limitn_ = 2;
   _type_   = 'STANDARD';
   _mean_   = 1250;
   _stddev_ = 12;
run;
```

The variables _VAR_ and _SUBGRP_ are required, and their values must match the *process* and *subgroup-variable*, respectively, specified in the IRCHART statement. The bookkeeping variable _TYPE_ is not required, but it is recommended to indicate that the variables _MEAN_ and _STDDEV_ provide standard values rather than estimated values. See "LIMITS= Data Set" on page 1501 for details.

The following statements read Enginelimits as a LIMITS= data set:

```
proc shewhart data=Engines limits=Enginelimits;
   irchart Weight*ID / xsymbol=mu0;
run;
```

The resulting charts (not shown here) are identical to those shown in Output 17.12.1.

Example 17.13: Displaying Distributional Plots in the Margin

NOTE: See *IRCHARTS with Margin Plots* in the SAS/QC Sample Library.

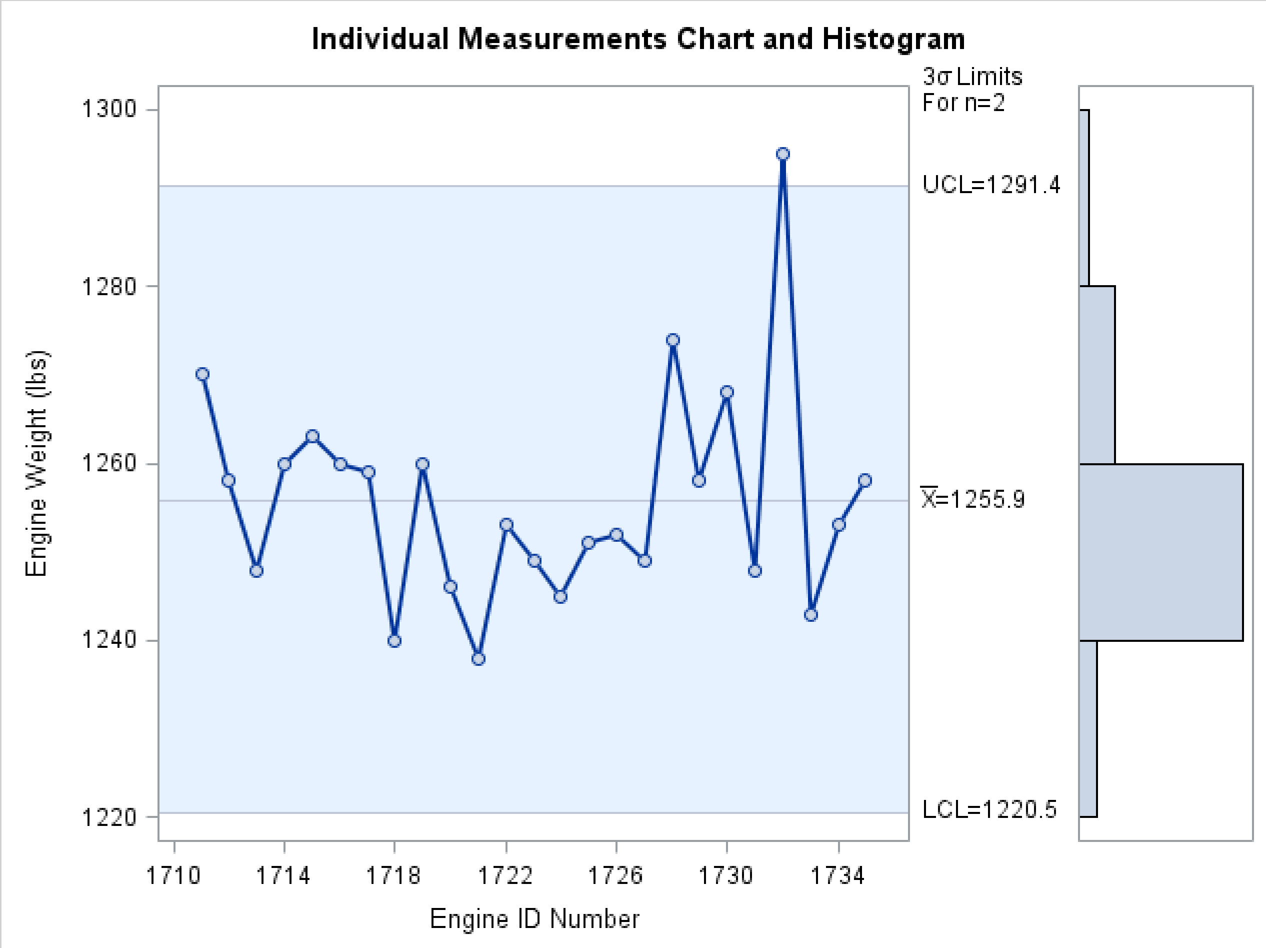
You can augment a chart for individual measurements with one of several graphical displays, such as a histogram or a box-and-whisker plot. These displays summarize the measurements plotted on the chart, and, if the process is in statistical control, they provide a view of the process distribution.

For example, the following statements create an individual measurements chart for the engine weight measurements in the data set Engines (see Example 17.11) augmented with a histogram of the weights:

```
ods graphics on;
title 'Individual Measurements Chart and Histogram';
proc shewhart data=Engines;
   irchart Weight*ID /
      odstitle = title
      rtmplot  = histogram
      markers
      nochart2;
run;
```

The chart is shown in Output 17.13.1. The RTMPLOT= option requests a histogram in the right margin. The NOCHART2 option suppresses the display of the moving range chart.

Output 17.13.1 Histogram in Right Margin



The following *keywords*, requesting different types of plots, are available with the RTMPLOT= option:

Keyword	**Marginal Plot**
HISTOGRAM	histogram
DIGIDOT	digidot plot
SKELETAL	skeletal box-and-whisker plot
SCHEMATIC	schematic box-and-whisker plot
SCHEMATICID	schematic box-and-whisker plot with outliers labeled
SCHEMATICIDFAR	schematic box-and-whisker plot with far outliers labeled

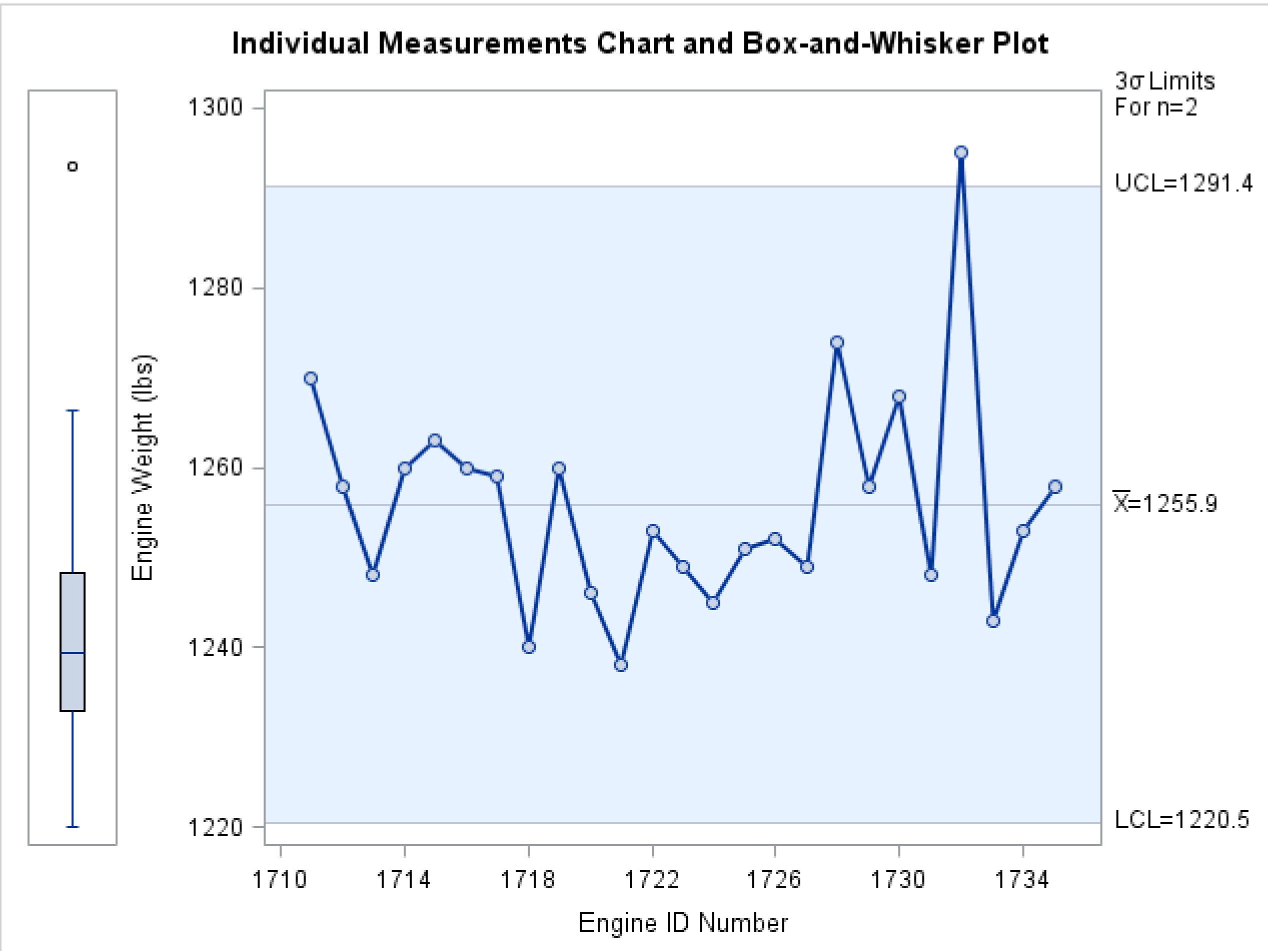
See the entry for the BOXSTYLE= option in "Dictionary of Options: SHEWHART Procedure" on page 1946 for a description of the various box-and-whisker plots.

You can also use the LTMPLOT= option to request univariate plots in the left margin. The following statements request an individual measurements chart with a box-and-whisker plot in the left margin:

```
title 'Individual Measurements Chart and Box-and-Whisker Plot';
proc shewhart data=Engines;
   irchart Weight*ID /
      odstitle = title
      ltmplot  = schematic
      ltmargin = 8
      markers
      nochart2;
run;
```

The chart is shown in Output 17.13.2. The same *keywords* that are available with the RTMPLOT= option can be specified with the LTMPLOT= option. The LTMARGIN= option specifies the width (in horizontal percent screen units) of the left margin.

Output 17.13.2 Box-and-Whisker Plot in Left Margin



MCHART Statement: SHEWHART Procedure

Overview: MCHART Statement

The MCHART statement creates a chart for subgroup medians, which is used to monitor the central tendency of a process.

You can use options in the MCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted medians or as probability limits
- tabulate subgroup sample sizes, subgroup medians, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes and subgroup medians in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify one of several methods for estimating the process standard deviation
- specify whether subgroup standard deviations or subgroup ranges are used to estimate the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- create a secondary chart that displays a time trend removed from the data (see "Displaying Trends in Process Data" on page 2054)
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the charts more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing medians charts with the MCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.

- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 3, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

NOTE: When analyzing variables data, you should examine the variability of the process as well as the mean level. You can use the MRCHART statement in the SHEWHART procedure to monitor both the mean level and variability.

Getting Started: MCHART Statement

This section introduces the MCHART statement with simple examples that illustrate commonly used options. Complete syntax for the MCHART statement is presented in the section "Syntax: MCHART Statement" on page 1527.

Creating Charts for Medians from Raw Data

NOTE: See *Median Chart Examples* in the SAS/QC Sample Library.

A consumer products company weighs detergent boxes (in pounds) to determine whether the fill process is in control. The following statements create a SAS data set named Detergent, which contains the weights for five boxes in each of 28 lots. A lot is considered a rational subgroup.

```
data Detergent;
   input Lot @;
   do i=1 to 5;
      input Weight @;
      output;
   end;
   drop i;
   datalines;
 1 17.39 26.93 19.34 22.56 24.49
 2 23.63 23.57 23.54 20.56 22.17
 3 24.35 24.58 23.79 26.20 21.55
 4 25.52 28.02 28.44 25.07 23.39
 5 23.25 21.76 29.80 23.09 23.70
 6 23.01 22.67 24.70 20.02 26.35
 7 23.86 24.19 24.61 26.05 24.18
 8 26.00 26.82 28.03 26.27 25.85
 9 21.58 22.31 25.03 20.86 26.94
10 22.64 21.05 22.66 29.26 25.02
11 26.38 27.50 23.91 26.80 22.53
12 23.01 23.71 25.26 20.21 22.38
13 23.15 23.53 22.98 21.62 26.99
14 26.83 23.14 24.73 24.57 28.09
15 26.15 26.13 20.57 25.86 24.70
16 25.81 23.22 23.99 23.91 27.57
17 25.53 22.87 25.22 24.30 20.29
18 24.88 24.15 25.29 29.02 24.46
```

```
19 22.32 25.96 29.54 25.92 23.44
20 25.63 26.83 20.95 24.80 27.25
21 21.68 21.11 26.07 25.17 27.63
22 26.72 27.05 24.90 30.08 25.22
23 31.58 22.41 23.67 23.47 24.90
24 28.06 23.44 24.92 24.64 27.42
25 21.10 22.34 24.96 26.50 24.51
26 23.80 24.03 24.75 24.82 27.21
27 25.10 26.09 27.21 24.28 22.45
28 25.53 22.79 26.26 25.85 25.64
;
```

A partial listing of Detergent is shown in Figure 17.34.

Figure 17.34 Partial Listing of the Data Set Detergent

The Data Set DETERGENT

Lot	Weight
1	17.39
1	26.93
1	19.34
1	22.56
1	24.49
2	23.63
2	23.57
2	23.54
2	20.56
2	22.17
3	24.35
3	24.58
3	23.79
3	26.20
3	21.55
4	25.52

The data set Detergent is said to be in "strung-out" form, since each observation contains the lot number and weight of a single box. The first five observations contain the weights for the first lot, the second five observations contain the weights for the second lot, and so on. Because the variable Lot classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable Weight contains the weights and is referred to as the *process variable* (or *process* for short).

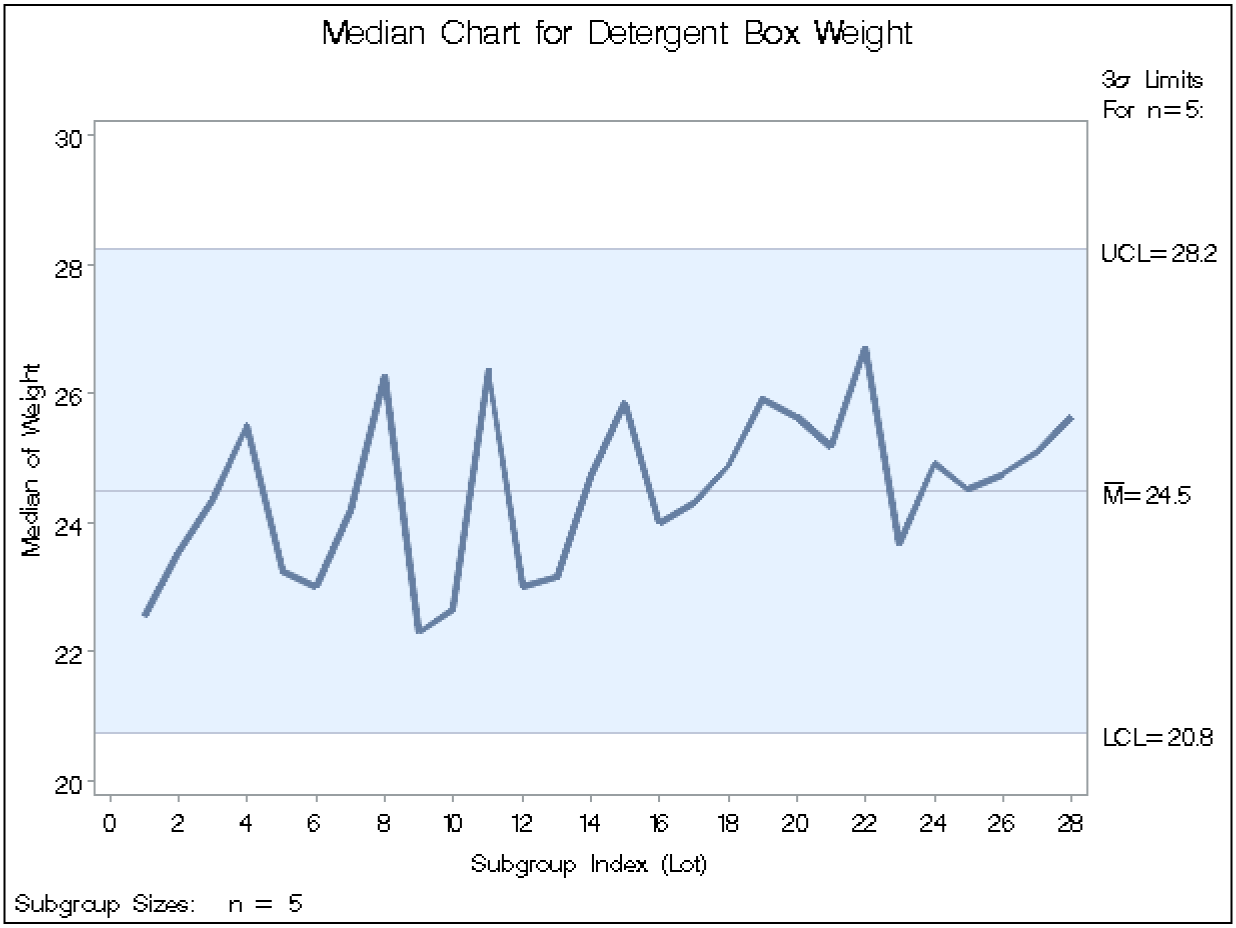
The within-subgroup variability of the weights is known to be stable. You can use a median chart to determine whether the mean level of the weights is in control. The following statements create the median chart shown in Figure 17.35:

```
ods graphics off;
title 'Median Chart for Detergent Box Weight';
proc shewhart data=Detergent;
   mchart Weight*Lot;
run;
```

This example illustrates the basic form of the MCHART statement. After the keyword MCHART, you specify the *process* to analyze (in this case, Weight) followed by an asterisk and the *subgroup-variable* (Lot).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Figure 17.35 Median Chart for Detergent Box Weight Data (Traditional Graphics)



Each point on the chart represents the median of the weights for a particular lot. For instance, the weights for the first lot are 17.39, 19.34, 22.56, 24.49, and 26.93, and consequently, the median plotted for this lot is 22.56.

Since all of the subgroup medians lie within the control limits, you can conclude that the process is in statistical control. By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 17.26. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1525.

For computational details, see "Constructing Median Charts" on page 1538. For more details on reading raw measurements, see "DATA= Data Set" on page 1544.

Creating Charts for Medians from Subgroup Summary Data

NOTE: See *Median Chart Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create median charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the MCHART statement with data of this type.

The following data set (Detsum) provides the data from the preceding example in summarized form. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Lot). The variable WeightM contains the subgroup medians, the variable WeightR contains the subgroup ranges, and the variable WeightN contains the subgroup sample sizes (these are all five).

```
data Detsum;
   input Lot WeightM WeightR;
   WeightN = 5;
   datalines;
 1  22.56  9.54
 2  23.54  3.07
 3  24.35  4.65
 4  25.52  5.05
 5  23.25  8.04
 6  23.01  6.33
 7  24.19  2.19
 8  26.27  2.18
 9  22.31  6.08
10  22.66  8.21
11  26.38  4.97
12  23.01  5.05
13  23.15  5.37
14  24.73  4.95
15  25.86  5.58
16  23.99  4.35
17  24.30  5.24
18  24.88  4.87
19  25.92  7.22
20  25.63  6.30
21  25.17  6.52
22  26.72  5.18
23  23.67  9.17
24  24.92  4.62
25  24.51  5.40
26  24.75  3.41
27  25.10  4.76
28  25.64  3.47
;
```

A partial listing of Detsum is shown in Figure 17.36.

Figure 17.36 The Summary Data Set Detsum

Summary Data Set for Detergent Box Weights

Lot	WeightM	WeightR	WeightN
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5
6	23.01	6.33	5
7	24.19	2.19	5
8	26.27	2.18	5
9	22.31	6.08	5
10	22.66	8.21	5
11	26.38	4.97	5
12	23.01	5.05	5
13	23.15	5.37	5
14	24.73	4.95	5
15	25.86	5.58	5
16	23.99	4.35	5
17	24.30	5.24	5
18	24.88	4.87	5
19	25.92	7.22	5
20	25.63	6.30	5
21	25.17	6.52	5
22	26.72	5.18	5
23	23.67	9.17	5
24	24.92	4.62	5
25	24.51	5.40	5
26	24.75	3.41	5
27	25.10	4.76	5
28	25.64	3.47	5

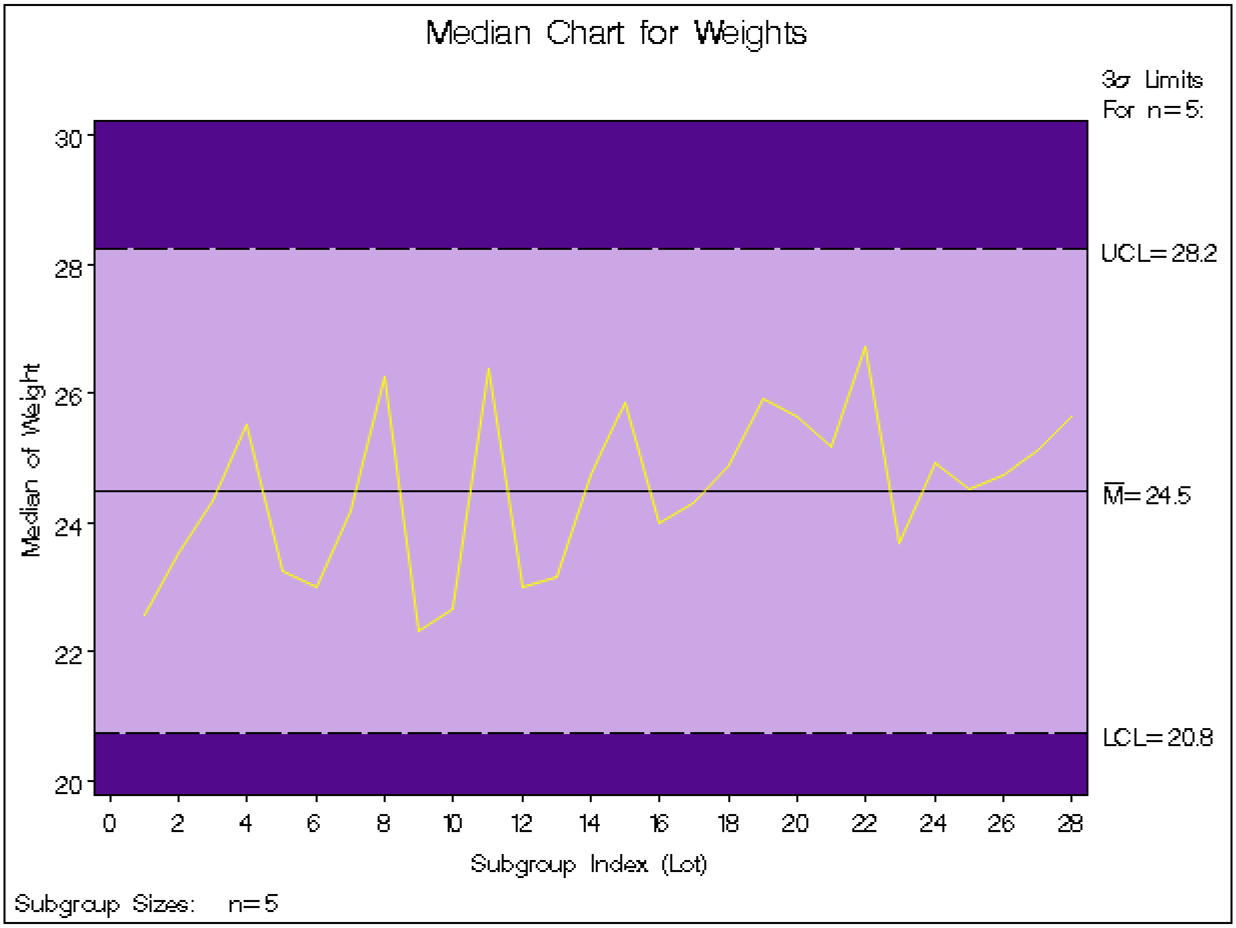
You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext=swiss;
title 'Median Chart for Weights';
proc shewhart history=Detsum;
   mchart Weight*Lot / cframe   = viv
                       cinfill = vpav
                       cconnect = yellow;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the MCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting median chart is shown in Figure 17.37.

Note that Weight is *not* the name of a SAS variable in the data set Detsum but is, instead, the common prefix for the names of the three SAS variables WeightM, WeightR, and WeightN. The suffix characters *M*, *R*, and *N* indicate *median*, *range*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in the HISTORY= data set with a single name (Weight), which is referred to as the *process*. The name Lot specified after the asterisk is the name of the *subgroup-variable*.

Figure 17.37 Median Chart from Summary Data Set Detsum (Traditional Graphics with NOGSTYLE)



In general, a HISTORY= input data set used with the MCHART statement must contain the following variables:

- subgroup variable
- subgroup median variable
- either a subgroup range variable or a subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of the subgroup median, range (or standard deviation), and sample size variables must begin with the *process* name specified in the MCHART statement and end with the special suffix characters *M*, *R* (or *S*), and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see “Creating Charts for Medians and Ranges from Summary Data” on page 1560).

If you specify the STDDEVIATIONS option in the MCHART statement, the HISTORY= data set must contain a subgroup standard deviation variable; otherwise, the HISTORY= data set must contain a subgroup range variable. The STDDEVIATIONS option specifies that the estimate of the process standard deviation σ is to be calculated from subgroup standard deviations rather than subgroup ranges. For example, in the following statements, the data set Detsum2 must contain a subgroup standard deviation variable named WeightS:

```
title 'Median Chart for Weights';
symbol v=dot;
proc shewhart history=Detsum2;
   mchart Weight*Lot / stddeviations;
run;
```

Options such as STDDEVIATIONS are specified after the slash (/) in the MCHART statement. A complete list of options is presented in the section “Syntax: MCHART Statement” on page 1527.

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see “HISTORY= Data Set” on page 1545.

Saving Summary Statistics

NOTE: See *Median Chart Examples* in the SAS/QC Sample Library.

In this example, the MCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Detergent and create a summary data set named Dethist:

```
proc shewhart data=Detergent;
   mchart Weight*Lot / outhistory = Dethist
                       nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 17.35. Figure 17.38 contains a partial listing of Dethist.

Figure 17.38 The Summary Data Set Dethist

Summary Data Set DETHIST for Detergent Box Weights

Lot	WeightM	WeightR	WeightN
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5
6	23.01	6.33	5
7	24.19	2.19	5
8	26.27	2.18	5
9	22.31	6.08	5
10	22.66	8.21	5
11	26.38	4.97	5
12	23.01	5.05	5
13	23.15	5.37	5
14	24.73	4.95	5
15	25.86	5.58	5
16	23.99	4.35	5
17	24.30	5.24	5
18	24.88	4.87	5
19	25.92	7.22	5
20	25.63	6.30	5
21	25.17	6.52	5
22	26.72	5.18	5
23	23.67	9.17	5
24	24.92	4.62	5
25	24.51	5.40	5
26	24.75	3.41	5
27	25.10	4.76	5
28	25.64	3.47	5

There are four variables in the data set Dethist.

- Lot contains the subgroup index.
- WeightM contains the subgroup medians.
- WeightR contains the subgroup ranges.
- WeightN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *M*, *R*, and *N* to the *process* Weight specified in the MCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

If you specify the STDDEVIATIONS option, the OUTHISTORY= data set includes a subgroup standard deviation variable instead of a subgroup range variable, as demonstrated by the following statements:

```
proc shewhart data=Detergent;
   mchart Weight*Lot / outhistory = Dethist2
                       stddeviations
                       nochart;
run;
```

Figure 17.39 contains a partial listing of Dethist2.

Figure 17.39 The Summary Data Set Dethist2

Summary Data Set DETHIST for Detergent Box Weights

Lot	WeightM	WeightR	WeightN
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5
6	23.01	6.33	5
7	24.19	2.19	5
8	26.27	2.18	5
9	22.31	6.08	5
10	22.66	8.21	5
11	26.38	4.97	5
12	23.01	5.05	5
13	23.15	5.37	5
14	24.73	4.95	5
15	25.86	5.58	5
16	23.99	4.35	5
17	24.30	5.24	5
18	24.88	4.87	5
19	25.92	7.22	5
20	25.63	6.30	5
21	25.17	6.52	5
22	26.72	5.18	5
23	23.67	9.17	5
24	24.92	4.62	5
25	24.51	5.40	5
26	24.75	3.41	5
27	25.10	4.76	5
28	25.64	3.47	5

The variable WeightS, which contains the subgroup standard deviations, is named by adding the suffix character *S* to the *process* Weight.

For more information, see "OUTHISTORY= Data Set" on page 1541.

Saving Control Limits

NOTE: See *Median Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for a median chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1525) or modify the limits with a DATA step program.

The following statements read measurements from the data set Detergent (see "Creating Charts for Medians from Raw Data" on page 1514) and save the control limits displayed in Figure 17.35 in a data set named Detlim:

```
proc shewhart data=Detergent;
   mchart Weight*Lot / outlimits=Detlim
                       nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set Detlim is listed in Figure 17.40.

Figure 17.40 The Data Set Detlim Containing Control Limit Information

Control Limits for Detergent Box Weights

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLM_	_MEAN_	_UCLM_
Weight	Lot	ESTIMATE	5	.002909021	3	20.7554	24.4996	28.2439

LCLR	_R_	_UCLR_	_STDDEV_
0	5.42036	11.4613	2.33041

The data set Detlim contains one observation with the limits for the *process* Weight. The variables _LCLM_ and _UCLM_ contain the lower and upper control limits for the medians, and the variable _MEAN_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values.

The variables _LCLR_, _R_, and _UCLR_ are not used to create median charts, but they are included so the data set Detlim can be used to create an *R* chart; see "MRCHART Statement: SHEWHART Procedure" on page 1556 and "RCHART Statement: SHEWHART Procedure" on page 1683. If you specify the STDDEVIATIONS option in the MCHART statement, the variables _LCLS_, _S_, and _UCLS_ are included in the OUTLIMITS= data set. These variables can be used to create an *s* chart; see "SCHART Statement: SHEWHART Procedure" on page 1721. For more information, see "OUTLIMITS= Data Set" on page 1540.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Detergent;
   mchart Weight*Lot / outtable=Dtable
                       nochart;
run;
```

The data set Dtable is listed in Figure 17.41.

Figure 17.41 The Data Set Dtable

Summary Statistics and Control Limit Information

VAR	Lot	_SIGMAS_	_LIMITN_	_SUBN_	_LCLM_	_SUBMED_	_MEAN_	_UCLM_	_STDDEV_	_EXLIM_
Weight	1	3	5	5	20.7554	22.56	24.4996	28.2439	2.33041	
Weight	2	3	5	5	20.7554	23.54	24.4996	28.2439	2.33041	
Weight	3	3	5	5	20.7554	24.35	24.4996	28.2439	2.33041	
Weight	4	3	5	5	20.7554	25.52	24.4996	28.2439	2.33041	
Weight	5	3	5	5	20.7554	23.25	24.4996	28.2439	2.33041	
Weight	6	3	5	5	20.7554	23.01	24.4996	28.2439	2.33041	
Weight	7	3	5	5	20.7554	24.19	24.4996	28.2439	2.33041	
Weight	8	3	5	5	20.7554	26.27	24.4996	28.2439	2.33041	
Weight	9	3	5	5	20.7554	22.31	24.4996	28.2439	2.33041	
Weight	10	3	5	5	20.7554	22.66	24.4996	28.2439	2.33041	
Weight	11	3	5	5	20.7554	26.38	24.4996	28.2439	2.33041	
Weight	12	3	5	5	20.7554	23.01	24.4996	28.2439	2.33041	
Weight	13	3	5	5	20.7554	23.15	24.4996	28.2439	2.33041	
Weight	14	3	5	5	20.7554	24.73	24.4996	28.2439	2.33041	
Weight	15	3	5	5	20.7554	25.86	24.4996	28.2439	2.33041	
Weight	16	3	5	5	20.7554	23.99	24.4996	28.2439	2.33041	
Weight	17	3	5	5	20.7554	24.30	24.4996	28.2439	2.33041	
Weight	18	3	5	5	20.7554	24.88	24.4996	28.2439	2.33041	
Weight	19	3	5	5	20.7554	25.92	24.4996	28.2439	2.33041	
Weight	20	3	5	5	20.7554	25.63	24.4996	28.2439	2.33041	
Weight	21	3	5	5	20.7554	25.17	24.4996	28.2439	2.33041	
Weight	22	3	5	5	20.7554	26.72	24.4996	28.2439	2.33041	
Weight	23	3	5	5	20.7554	23.67	24.4996	28.2439	2.33041	
Weight	24	3	5	5	20.7554	24.92	24.4996	28.2439	2.33041	
Weight	25	3	5	5	20.7554	24.51	24.4996	28.2439	2.33041	
Weight	26	3	5	5	20.7554	24.75	24.4996	28.2439	2.33041	
Weight	27	3	5	5	20.7554	25.10	24.4996	28.2439	2.33041	
Weight	28	3	5	5	20.7554	25.64	24.4996	28.2439	2.33041	

This data set contains one observation for each subgroup sample. The variables _SUBMED_ and _SUBN_ contain the subgroup medians and subgroup sample sizes. The variables _LCLM_ and _UCLM_ contain the lower and upper control limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and Lot contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1542.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Dtable and display a median chart (not shown here) identical to the chart in Figure 17.35:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart table=Dtable;
   mchart Weight*Lot;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2096). For more information, see "TABLE= Data Set" on page 1547.

Reading Preestablished Control Limits

NOTE: See *Median Chart Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Detlim saved control limits computed from the measurements in Detergent. This example shows how these limits can be applied to new data provided in the following data set:

```
data Detergent2;
   input Lot @;
   do i=1 to 5;
      input Weight @;
      output;
   end;
   drop i;
   datalines;
29 16.66 27.49 18.87 22.53 24.72
30 23.74 23.67 23.64 20.26 22.09
31 24.56 24.82 23.92 26.67 21.38
32 25.89 28.73 29.21 25.38 23.47
33 23.32 21.61 30.75 23.13 23.82
34 23.04 22.65 24.96 19.64 26.84
35 24.01 24.38 24.86 26.50 24.37
36 26.43 27.36 28.74 26.74 26.27
37 21.41 22.24 25.34 20.59 27.51
38 22.62 20.81 22.64 30.15 25.32
39 26.86 28.14 24.06 27.35 22.49
40 23.03 23.83 25.59 19.85 22.33
41 23.19 23.63 23.00 21.46 27.57
42 27.38 23.18 24.99 24.81 28.82
43 26.60 26.58 20.26 26.27 24.96
44 26.22 23.28 24.15 24.06 28.23
45 25.90 22.88 25.55 24.50 19.95
46 16.66 27.49 18.87 22.53 24.72
47 23.74 23.67 23.64 20.26 22.09
48 24.56 24.82 23.92 26.67 21.38
49 25.89 28.73 29.21 25.38 23.47
50 23.32 21.61 30.75 23.13 23.82
;
```

The following statements create a median chart for the data in Detergent2 using the control limits in Detlim:

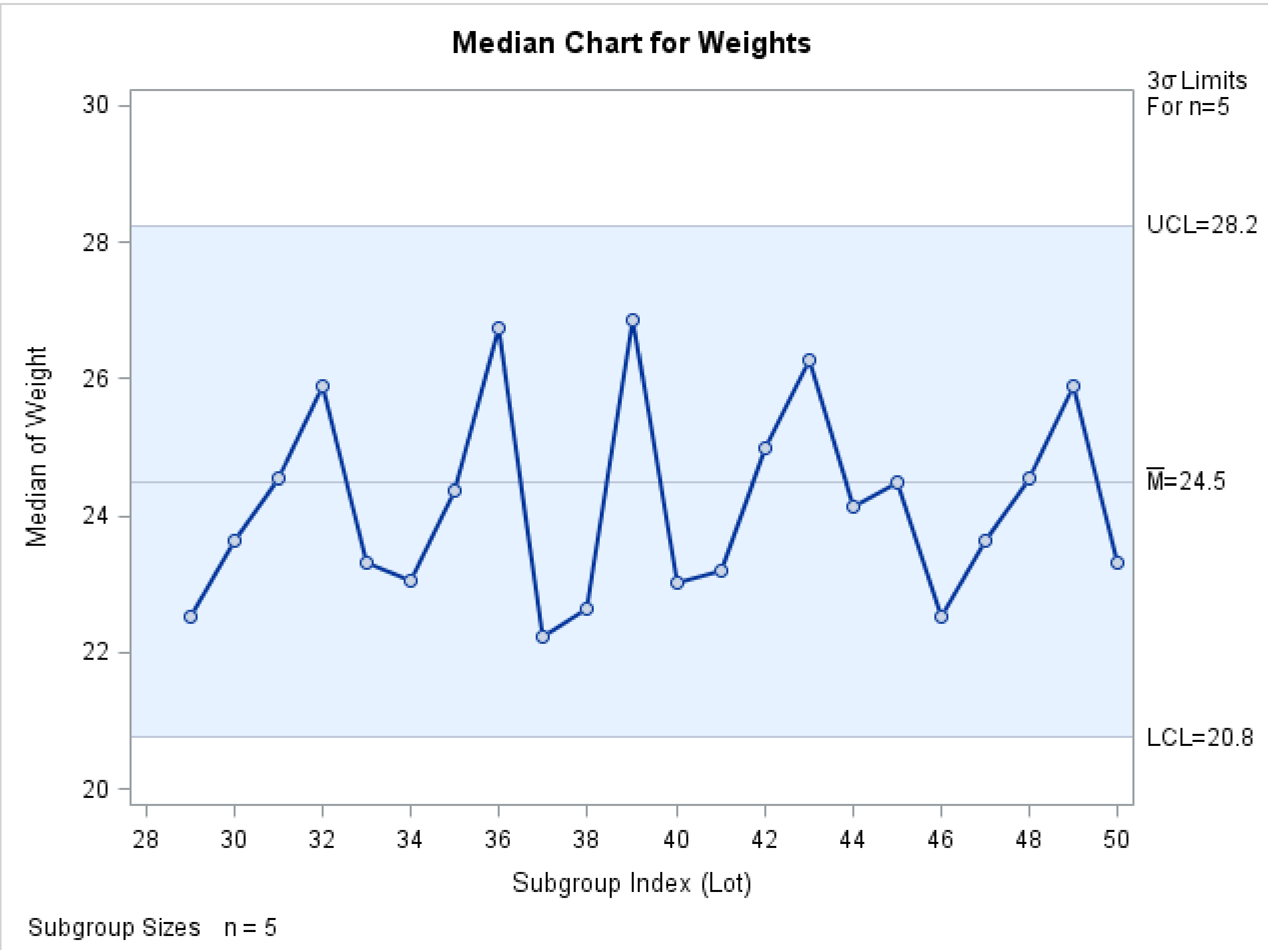
```
ods graphics on;
title 'Median Chart for Weights';
proc shewhart data=Detergent2 limits=Detlim;
   mchart Weight*Lot / odstitle=title markers;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the median chart is created using ODS Graphics instead of traditional graphics. The chart is shown in Figure 17.42.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Weight
- the value of _SUBGRP_ matches the *subgroup-variable* name Lot

Figure 17.42 Median Chart for Second Set of Detergent Box Weight Data (ODS Graphics)



The chart indicates that the process is in control, since all the medians lie within the control limits.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1545 for details concerning the variables that you must provide.

Syntax: MCHART Statement

The basic syntax for the MCHART statement is as follows:

MCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

MCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=**'*character*' > / < *options* > **;**

You can use any number of MCHART statements in the SHEWHART procedure. The components of the MCHART statement are described as follows:

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Medians from Raw Data" on page 1514.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Medians from Subgroup Summary Data" on page 1517.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1523.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct median charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   mchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding MCHART statement, Day is the subgroup variable. For details, see "Subgroup Variables" on page 1871.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2026 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the medians.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2025 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create a median chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   mchart Weight*Day='*';
run;
```

options

enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The section "Summary of Options", which follows, lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1946 describes each option in detail.

Summary of Options

The following tables list the MCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1946.

Table 17.25 MCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	requests probability limits for chart
LIMITN=	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	specifies color for area inside control limits

Table 17.25 *continued*

Option	Description
CLIMITS=	specifies color of control limits, central line, and related labels
LCLLABEL=	specifies label for lower control limit
LIMLABSUBCHAR=	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	specifies line type for control limits
NDECIMAL=	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit
UCLLABEL=	specifies label for upper control limit
WLIMITS=	specifies width for control limits and central line
XSYMBOL=	specifies label for central line
Process Mean and Standard Deviation Options	
MEDCENTRAL=	specifies method for estimating process mean μ
MU0=	specifies known value of μ_0 for process mean μ
SIGMA0=	specifies known value σ_0 for process standard deviation σ
SMETHOD=	specifies method for estimating process standard deviation σ
STDDEVIATIONS	specifies that estimate of process standard deviation σ is to be calculated from subgroup standard deviations
TYPE=	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	labels every point on median chart
ALLLABEL2=	labels every point on trend chart
CLABEL=	specifies color for labels
CCONNECT=	specifies color for line segments that connect points on chart
CFRAMELAB=	specifies fill color for frame around labeled points
CNEEDLES=	specifies color for needles that connect points to central line
COUT=	specifies color for portions of line segments that connect points outside control limits

Table 17.25 *continued*

Option	Description
COUTFILL=	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	specifies angle at which labels are drawn
LABELFONT=	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
NOTRENDCONNECT	suppresses line segments that connect points on trend chart
OUTLABEL=	labels points outside control limits
SYMBOLLEGEND=	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=	specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	computes zone widths independently above and below center line
NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	suppresses tests across *phase* boundaries
TESTS=	specifies tests for special causes
TEST2RUN=	specifies length of pattern for Test 2
TEST3RUN=	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL=	provides labels for points where test is positive
TESTLABEL*n*=	specifies label for *n*th test for special causes
TESTNMETHOD=	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=	enables tests for special causes to be reset
WESTGARD=	requests that Westgard rules be applied
ZONELABELS	adds labels A, B, and C to zone lines
ZONES	adds lines delineating zones A, B, and C
ZONEVALPOS=	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	specifies color for labels indicating points where test is positive

Table 17.25 *continued*

Option	Description
CTESTSYMBOL=	specifies color for symbol used to plot points where test is positive
CZONES=	specifies color for lines and labels delineating zones A, B, and C
LTESTS=	specifies type of line connecting points where test is positive
LZONES=	specifies line type for lines delineating zones A, B, and C
TESTFONT=	specifies software font for labels at points where test is positive
TESTHEIGHT=	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	specifies plot symbol for points where test is positive
TESTSYMBOLHT=	specifies symbol height for points where test is positive
WTESTS=	specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill colors for frame for plot area
CTEXT=	specifies color for tick mark values and axis labels
DISCRETE	produces horizontal axis for discrete numeric group values
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis of median chart
VAXIS2=	specifies major tick mark values for vertical axis of trend chart
VFORMAT=	specifies format for primary vertical axis tick mark labels

Table 17.25 *continued*

Option	Description
VFORMAT2=	specifies format for secondary vertical axis tick mark labels
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=	specifies width of axis lines
Plot Layout Options	
ALLN	plots means for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	specifies number of pages or screens to be used to display chart
TRENDVAR=	specifies list of trend variables
YPCT1=	specifies length of vertical axis on median chart as a percentage of sum of lengths of vertical axes for median and trend charts
ZEROSTD	displays median chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= and HREF2= options
CVREF=	specifies color for lines requested by VREF= and VREF2= options
HREF=	specifies position of reference lines perpendicular to horizontal axis on median chart

Table 17.25 *continued*

Option	Description
HREF2=	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFDATA=	specifies position of reference lines perpendicular to horizontal axis on median chart
HREF2DATA=	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS=	specifies labels for HREF= lines
HREF2LABELS=	specifies labels for HREF2= lines
HREFLABPOS=	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	specifies line type for HREF= and HREF2= lines
LVREF=	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	specifies position of reference lines perpendicular to vertical axis on median chart
VREF2=	specifies position of reference lines perpendicular to vertical axis on trend chart
VREFLABELS=	specifies labels for VREF= lines
VREF2LABELS=	specifies labels for VREF2= lines
VREFLABPOS=	position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=	specifies line type for grid requested with the ENDGRID option
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme points are clipped
CLIPLEGEND=	specifies text for clipping legend
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features to median chart
ANNOTATE2=	specifies annotate data set that adds features to trend chart

Table 17.25 *continued*

Option	Description
DESCRIPTION=	specifies description of median chart's GRSEG catalog entry
FONT=	specifies software font for labels and legends on charts
NAME=	specifies name of median chart's GRSEG catalog entry
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option
WTREND=	specifies width of line segments connecting points on trend chart
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	shades areas between control limits and connected points lying outside the limits
STARFILL=	specifies a variable identfying groups of stars filled with different colors
STARS=	specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	specifies the control limit infill transparency
MARKERS	plots subgroup points with markers
NOBLOCKREF	suppresses block and phase reference lines
NOBLOCKREFFILL	suppresses block and phase wall fills
NOFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOPHASEREF	suppresses block and phase reference lines
NOPHASEREFFILL	suppresses block and phase wall fills
NOREF	suppresses block and phase reference lines
NOREFFILL	suppresses block and phase wall fills
NOSTARFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	specifies a graph footnote
ODSFOOTNOTE2=	specifies a secondary graph footnote
ODSLEGENDEXPAND	specifies that legend entries contain all levels observed in the data
ODSTITLE=	specifies a graph title
ODSTITLE2=	specifies a secondary graph title

Table 17.25 *continued*

Option	Description
OUTFILLTRANSPARENCY=	specifies control limit outfill transparency
OVERLAYURL=	specifies URLs to associate with overlay points
OVERLAY2URL=	specifies URLs to associate with overlay points on secondary chart
PHASEPOS=	specifies vertical position of phase legend
PHASEREFLEVEL=	associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	draws central line labels using a simulated software font
STARTRANSPARENCY=	specifies star fill transparency
URL=	specifies a variable whose values are URLs to be associated with subgroups
URL2=	specifies a variable whose values are URLs to be associated with subgroups on secondary chart
Input Data Set Options	
MISSBREAK	specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	creates output data set containing subgroup summary statistics
OUTINDEX=	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	creates output data set containing control limits
OUTTABLE=	creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	specifies α value and type for computing capability index confidence limits
LSL=	specifies list of lower specification limits

Table 17.25 *continued*

Option	Description
TARGET=	specifies list of target values
USL=	specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for *block-variable* legend
BLOCKLABTYPE=	specifies text size of *block-variable* legend
BLOCKPOS=	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE=	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	specifies color for STARCIRCLES= circles
CSTARFILL=	specifies color for filling stars
CSTAROUT=	specifies outline color for stars exceeding inner or outer circles
CSTARS=	specifies color for outlines of stars
LSTARCIRCLES=	specifies line types for STARCIRCLES= circles
LSTARS=	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	specifies reference circles for stars
STARINRADIUS=	specifies inner radius of stars
STARLABEL=	specifies vertices to be labeled
STARLEGEND=	specifies style of legend for star vertices
STARLEGENDLAB=	specifies label for STARLEGEND= legend
STAROUTRADIUS=	specifies outer radius of stars
STARSPECS=	specifies method used to standardize vertex variables
STARSTART=	specifies angle for first vertex
STARTYPE=	specifies graphical style of star

Table 17.25 *continued*

Option	Description
STARVERTICES=	superimposes star at each point on median chart
WSTARCIRCLES=	specifies width of STARCIRCLES= circles
WSTARS=	specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	specifies colors for primary chart overlay line segments
CCOVERLAY2=	specifies colors for secondary chart overlay line segments
COVERLAY=	specifies colors for primary chart overlay plots
COVERLAY2=	specifies colors for secondary chart overlay plots
COVERLAYCLIP=	specifies color for clipped points on overlays
LOVERLAY=	specifies line types for primary chart overlay line segments
LOVERLAY2=	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=	specifies variables to overlay on primary chart
OVERLAY2=	specifies variables to overlay on secondary chart
OVERLAY2HTML=	specifies links to associate with secondary chart overlay points
OVERLAY2ID=	specifies labels for secondary chart overlay points
OVERLAY2SYM=	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
OVERLAYHTML=	specifies links to associate with primary chart overlay points
OVERLAYID=	specifies labels for primary chart overlay points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols for primary chart overlays
OVERLAYSYMHT=	specifies symbol heights for primary chart overlays
WOVERLAY=	specifies widths of primary chart overlay line segments
WOVERLAY2=	specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	specifies a variable whose values create links to be associated with subgroups
HTML2=	specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	specifies plot character for clipped points
CONNECTCHAR=	specifies character used to form line segments that connect points on chart

Table 17.25 *continued*

Option	Description
HREFCHAR=	specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	specifies characters indicating *symbol-variable*
TESTCHAR=	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	specifies line character for VREF= and VREF2= lines
ZONECHAR=	specifies character for lines that delineate zones for tests for special causes

Details: MCHART Statement

Constructing Median Charts

The following notation is used in this section:

μ process mean (expected value of the population of measurements)

σ process standard deviation (standard deviation of the population of measurements)

$\bar{X}_i$ mean of measurements in ith subgroup

n_i sample size of ith subgroup

N the number of subgroups

x_{ij} jth measurement in the ith subgroup, $j = 1, 2, 3, \ldots, n_i$

$x_{i(j)}$ jth largest measurement in the ith subgroup. Then

$$x_{i(1)} \leq x_{i(2)} \leq \ldots \leq x_{i(n_i)}$$

$\bar{\bar{X}}$ weighted average of subgroup means

M_i median of the measurements in the ith subgroup:

$$M_i = \begin{cases} x_{i((n_i+1)/2)} & \text{if } n_i \text{ is odd} \\ (x_{i(n_i/2)} + x_{i((n_i/2)+1)})/2 & \text{if } n_i \text{ is even} \end{cases}$$

$\bar{M}$ average of the subgroup medians:

$$\bar{M} = (n_1 M_1 + \ldots + n_N M_N)/(n_1 + \ldots + n_N)$$

$\tilde{M}$ median of the subgroup medians. Denote the jth largest median by $M_{(j)}$ so that $M_{(1)} \leq M_{(2)} \leq \ldots \leq M_{(N)}$. Then

$$\tilde{M} = \begin{cases} M_{((N+1)/2)} & \text{if } N \text{ is odd} \\ (M_{(N/2)} + M_{(N/2)+1})/2 & \text{if } N \text{ is even} \end{cases}$$

$e_M(n)$ standard error of the median of n independent, normally distributed variables with unit standard deviation (the value of $e_M(n)$ can be calculated with the STDMED function in a DATA step)

$Q_p(n)$	100pth percentile ($0 < p < 1$) of the distribution of the median of n independent observations from a normal population with unit standard deviation
z_p	100pth percentile of the standard normal distribution
$D_p(n)$	100pth percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on a median chart indicates the value of a subgroup median (M_i). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the value plotted for this subgroup is $M_{10} = 15$.

Central Line

The value of the central line indicates an estimate for μ, which is computed as

- $\bar{M}$ by default
- $\bar{\bar{X}}$ when you specify MEDCENTRAL=AVGMEAN
- $\tilde{M}$ when you specify MEDCENTRAL=MEDMED
- μ_0 when you specify μ_0 with the MU0= option

Control Limits

You can compute the limits

- as a specified multiple (k) of the standard error of M_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that M_i exceeds the limits

The following table provides the formulas for the limits:

Table 17.26 Limits for Median Charts

Control Limits
LCLM = lower limit = $\bar{M} - k\hat{\sigma}e_M(n_i)$ UCLM = upper limit = $\bar{M} + k\hat{\sigma}e_M(n_i)$
Probability Limits
LCLM = lower limit = $\bar{M} - Q_{\alpha/2}(n_i)\hat{\sigma}$ UCLM = upper limit = $\bar{M} + Q_{1-\alpha/2}(n_i)\hat{\sigma}$

Note that the limits vary with n_i. In Table 17.26, replace $\bar{M}$ with $\bar{\bar{X}}$ if you specify MEDCENTRAL=AVGMEAN, and replace $\bar{M}$ with $\tilde{M}$ if you specify MEDCENTRAL=MEDMED. Replace $\bar{M}$

with μ_0 if you specify μ_0 with the MU0= option, and replace $\hat{\sigma}$ with σ_0 if you specify σ_0 with the SIGMA0= option. The formulas assume that the data are normally distributed.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in the LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in the LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 17.27 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLM	lower control limit for subgroup median
LCLR	lower control limit for subgroup range
LCLS	lower control limit for subgroup standard deviation
LIMITN	sample size associated with the control limits
LSL	lower specification limit
MEAN	value of central line on median chart ($\bar{M}$, $\tilde{M}$, $\overline{\overline{X}}$, or μ_0)
_R_	value of central line on R chart
_S_	value of central line on s chart
SIGMAS	multiple (k) of standard error of M_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the MCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLM	upper control limit for subgroup median
UCLR	upper control limit for subgroup range

Table 17.27 *continued*

Option	Description
UCLS	upper control limit for subgroup standard deviation
USL	upper specification limit
VAR	*process* specified in the MCHART statement

Notes:

1. The variables _LCLS_, _S_, and _UCLS_ are included if you specify the STDDEVIATIONS option; otherwise, the variables _LCLR_, _R_, and _UCLR_ are included. These variables are not used to create median charts, but they enable the OUTLIMITS= data set to be used as a LIMITS= data set with the BOXCHART, XRCHART, XSCHART, and MRCHART statements.

2. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLM_, _UCLM_, _LCLR_, _R_, _UCLR_, _LCLS_, _S_, and _UCLS_.

3. If the limits are defined in terms of a multiple k of the standard error of M_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - F_{med}(k, n))$, where $F_{med}(\cdot, n)$ is the cumulative distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _ALPHA_.

4. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = F_{med}^{-1}(1 - \alpha/2, n)$, where $F_{med}^{-1}(\cdot, n)$ is the inverse distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _SIGMAS_.

5. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See “Capability Indices” on page 1874 for computational details.

6. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the MCHART statement. For an example, see “Saving Control Limits” on page 1523.

OUTHISTORY= Data Set

The OUTHISTORY= option saves subgroup summary statistics. The following variables can be saved:

- the *subgroup-variable*
- a subgroup median variable named by *process* suffixed with *M*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a subgroup sample size variable named by *process* suffixed with *N*

A subgroup standard deviation variable is included if you specify the STDDEVIATIONS option; otherwise, a subgroup range variable is included.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Variables containing subgroup summary statistics are created for each *process* specified in the MCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   mchart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthM, WidthR, WidthN, DiameterM, DiameterR, and DiameterN. The variables WidthR and DiameterR are included, since the STDDEVIATIONS option is not specified. If you specified the STDDEVIATIONS option, the data set Summary would contain WidthS and DiameterS rather than WidthR and DiameterR.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1520.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on median chart
LCLM	lower control limit for median
LIMITN	nominal sample size associated with the control limits
MEAN	estimate of process mean ($\bar{M}$, $\tilde{M}$, $\overline{\overline{X}}$, or μ_0)
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBMED	subgroup median
SUBN	subgroup sample size
TESTS	tests for special causes signaled on median chart
UCLM	upper control limit for median
VAR	*process* specified in the MCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- _TREND_ (if the TRENDVAR= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The *k*th character of a value of _TESTS_ is *k* if Test *k* is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1523.

ODS Tables

The following table summarizes the ODS tables that you can request with the MCHART statement.

Table 17.28 ODS Tables Produced with the MCHART Statement

Table Name	Description	Options
MCHART	median chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*). **NOTE:** In SAS/QC 13.1 the SHEWHART procedure does not support the creation of graphs that are editable with the ODS Graphics Editor.

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. MCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1946.

When ODS Graphics is in effect, the MCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 17.29.

Table 17.29 ODS Graphics Produced by the MCHART Statement

ODS Graph Name	Plot Description
MChart	median chart

See Chapter 3, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the MCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the MCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the ith subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the ith subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2031).

For an example of a DATA= data set, see "Creating Charts for Medians from Raw Data" on page 1514.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   mchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLM_, _MEAN_, and _UCLM_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 17.26

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see the section "Reading Preestablished Control Limits" on page 1525.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to read output data sets created with SAS summarization procedures, such as PROC UNIVARIATE.

A HISTORY= data set used with the MCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup median variable for each *process*
- a subgroup sample size variable for each *process*
- either a subgroup range variable or a subgroup standard deviation variable for each *process*

The names of the subgroup summary statistics variables must be the *process* name concatenated with the following special suffix characters:

Subgroup Summary Statistic	**Suffix Character**
subgroup median	M
subgroup mean	X
subgroup sample size	N
subgroup range	R
subgroup standard deviation	S

You must provide the subgroup mean variable only if you specify the MEDCENTRAL=AVGMEAN option. If you specify the STDDEVIATIONS option, the subgroup standard deviation variable must be included; otherwise, the subgroup range variable must be included.

For example, consider the following statements:

```
proc shewhart history=Summary;
   mchart (Weight Yieldstrength)*Batch / medcentral=avgmean;
run;
```

The data set Summary must include the variables Batch, WeightX, WeightM, WeightR, WeightN, YieldstrengthX, YieldstrengthM, YieldstrengthR, and YieldstrengthN. If the STDDEVIATIONS option were specified in the preceding MCHART statement, it would be necessary for Summary to include the variables Batch, WeightX, WeightM, WeightS, WeightN, YieldstrengthX, YieldstrengthM, YieldstrengthS, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2031 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Medians from Subgroup Summary Data" on page 1517.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2096.

The following table lists the variables required in a TABLE= data set used with the MCHART statement:

Table 17.30 Variables Required in a TABLE= Data Set

Variable	Description
LCLM	lower control limit for median
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
subgroup-variable	values of the *subgroup-variable*
SUBMED	subgroup median
SUBN	subgroup sample size
UCLM	upper control limit for median

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1523.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, three methods (referred to as default, MVLUE, and RMSDF) are available with the MCHART statement for estimating the process standard deviation σ. The method used to calculate σ depends on whether you specify the STDDEVIATIONS option in the MCHART statement. If this option is specified, σ is estimated using subgroup standard deviations; otherwise, σ is estimated using subgroup ranges. For further details and formulas, see "Methods for Estimating the Standard Deviation" on page 1826.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup median variable
Vertical	TABLE=	_SUBMED_

If you specify the TRENDVAR= option, you can provide distinct labels for the vertical axes of the median and trend charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the median chart, and the second part labels the vertical axis of the trend chart.

For an example, see "Labeling Axes" on page 2063.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: MCHART Statement

This section provides more advanced examples of the MCHART statement.

Example 17.14: Controlling Value of Central Line

You can specify options in the MCHART statement to request one of the following values for the central line on median charts:

- the average of the subgroup medians
- the average of the subgroup means

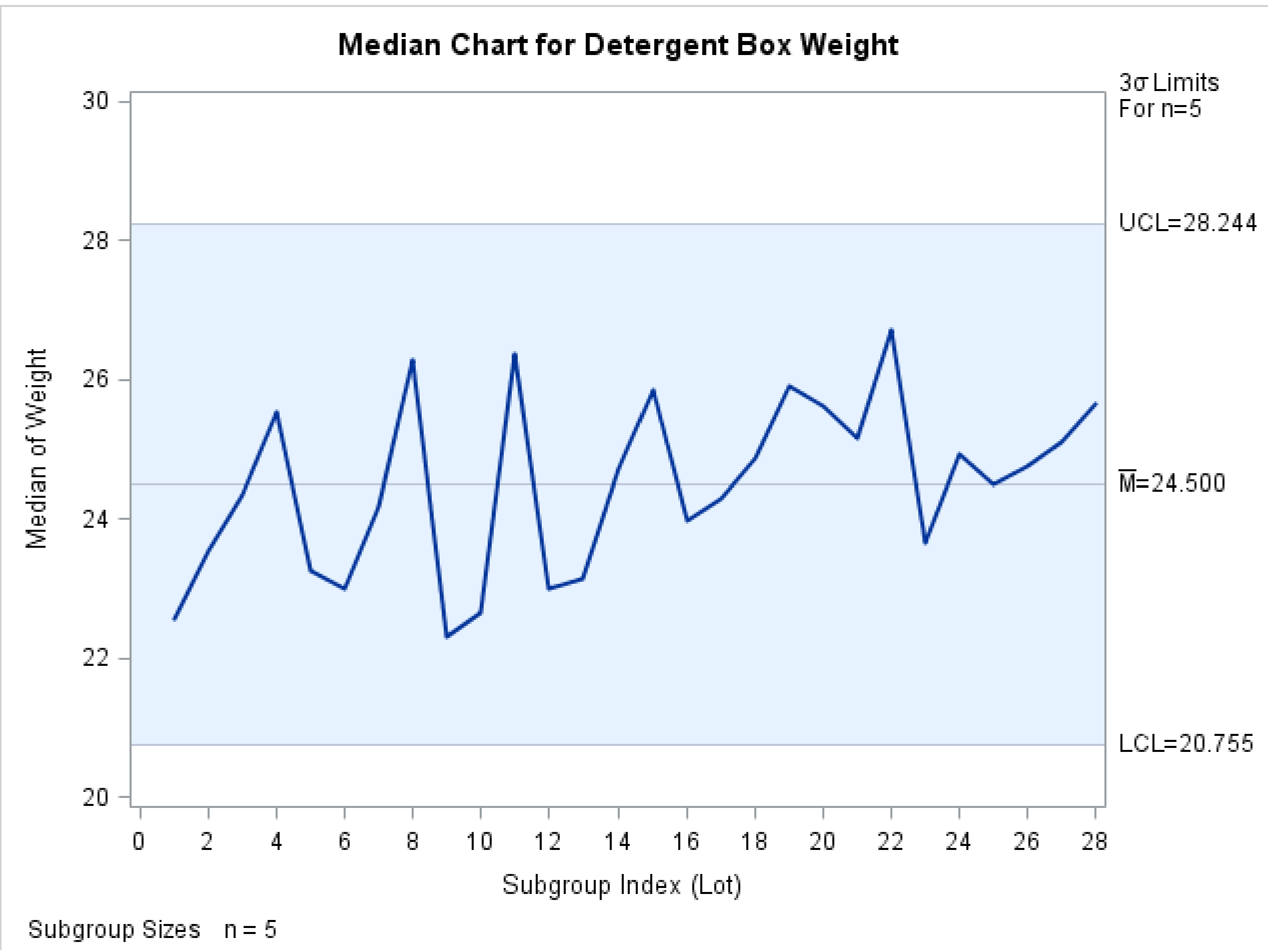
- the median of the subgroup medians
- a standard value of the process mean

By default, the value of the central line is the average of the subgroup medians. The following statements create a median chart for the detergent box weights stored in the data set Detergent (see "Creating Charts for Medians from Raw Data" on page 1514) with the average of the subgroup medians as the central line. The resulting chart is shown in Output 17.14.1.

```
ods graphics on;
title 'Median Chart for Detergent Box Weight';
proc shewhart data=Detergent;
   mchart Weight*Lot / ndecimal = 3
                       odstitle = title;
run;
```

The NDECIMAL= option specifies the number of decimal digits in the default labels for the control limits and central line.

Output 17.14.1 Central Line is Average of Subgroup Medians

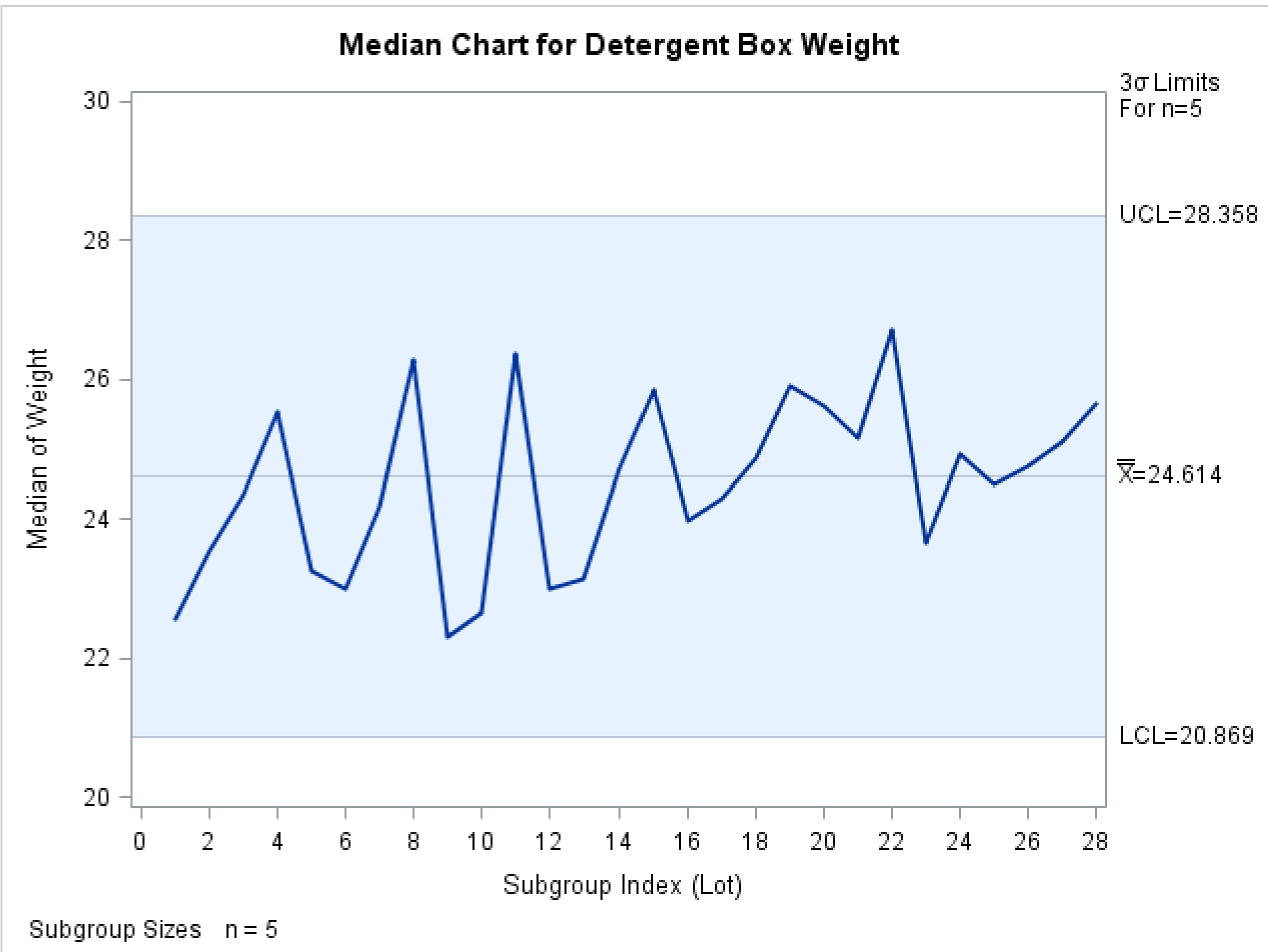


You can also request that the central line indicate the average of the subgroup means. The following statements create a median chart with this value for the central line:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart data=Detergent;
   mchart Weight*Lot / ndecimal    = 3
                       odstitle    = title
                       medcentral = avgmean;
run;
```

The MEDCENTRAL= option specifies the value used for the central line. In this case, MEDCENTRAL=AVGMEAN is specified to request a central line indicating the average of the subgroup means. The resulting chart is shown in Output 17.14.2.

Output 17.14.2 Central Line is Average of Subgroup Means

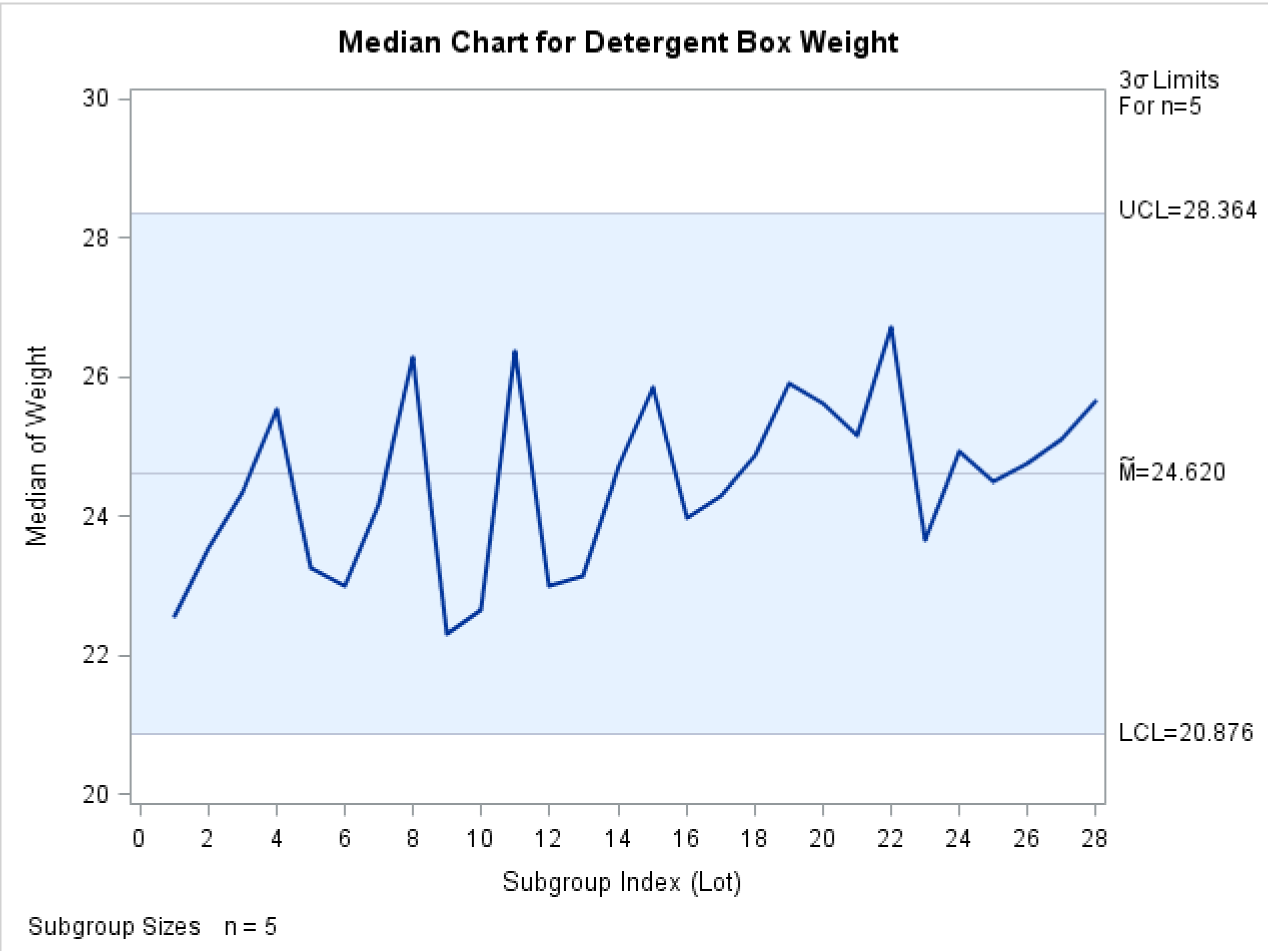


If you specify MEDCENTRAL=MEDMED, the median of the subgroup medians is used for the central line, as demonstrated by the following statements:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart data=Detergent;
   mchart Weight*Lot / ndecimal   = 3
                       odstitle   = title
                       medcentral = medmed;
run;
```

The resulting chart is shown in Output 17.14.3.

Output 17.14.3 Central Line is Median of Subgroup Medians

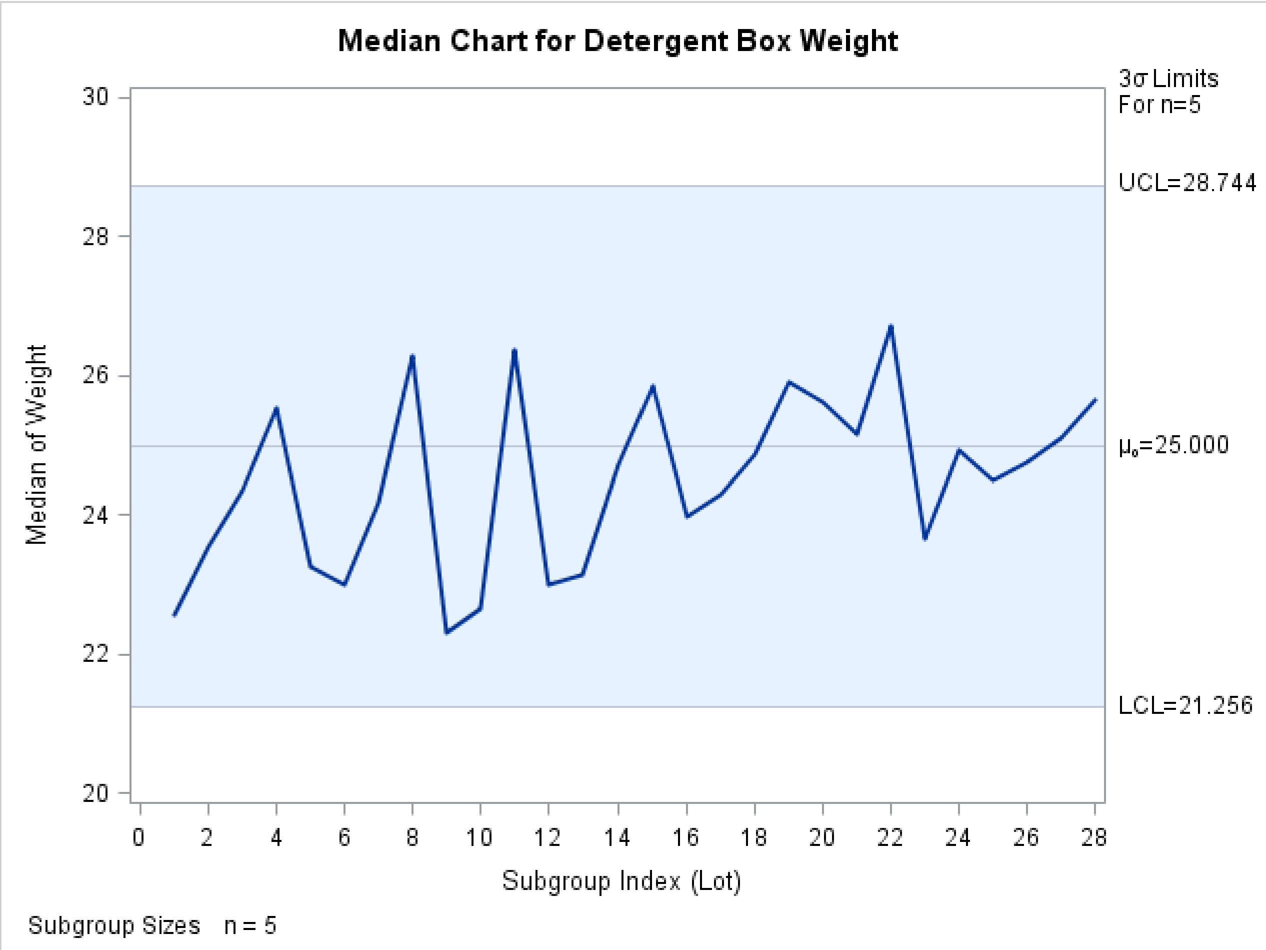


In some situations a standard value for the process mean (μ_0) is available. For instance, extensive startup testing provides an estimate of the process mean. If specified, this value is used for the central line. The following statements create a median chart for the detergent box weights with $\mu_0 = 25$:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart data=Detergent;
   mchart Weight*Lot / ndecimal = 3
                       mu0      = 25
                       odstitle = title
                       xsymbol  = mu0;
run;
```

The MU0= option specifies the standard value for the process mean, and the XSYMBOL= option specifies the label for the central line. In this case, XSYMBOL=MU0 is specified to indicate that the central line represents a standard value. The resulting chart is shown in Output 17.14.4.

Output 17.14.4 Median Chart for Detergent Box Weight Data



Note that you can also provide μ_0 with the _MEAN_ variable in a LIMITS= data set. For example, the following DATA step creates a data set (Dlims) which contains the same standard value specified in the preceding MCHART statement:

```
data Dlims;
   _var_    = "Weight  ";
   _subgrp_ = "Lot     ";
   _mean_   = 25;
run;
```

The _VAR_ and _SUBGRP_ variables are required if this data set is to be read as a LIMITS= data set in the PROC SHEWHART statement. These values must match the names of the *process* and *subgroup-variable* specified in the MCHART statement. The following statements specify the data set Dlims as a LIMITS= data set and create a median chart (not shown here) identical to the one in Output 17.14.4:

```
title 'Median Chart for Detergent Box Weight';
proc shewhart data=Detergent limits=Dlims;
   mchart Weight*Lot / xsymbol  = mu0
                       odstitle = title
                       ndecimal = 3;
run;
```

For more information, see "Constructing Median Charts" on page 1538.

Example 17.15: Estimating the Process Standard Deviation

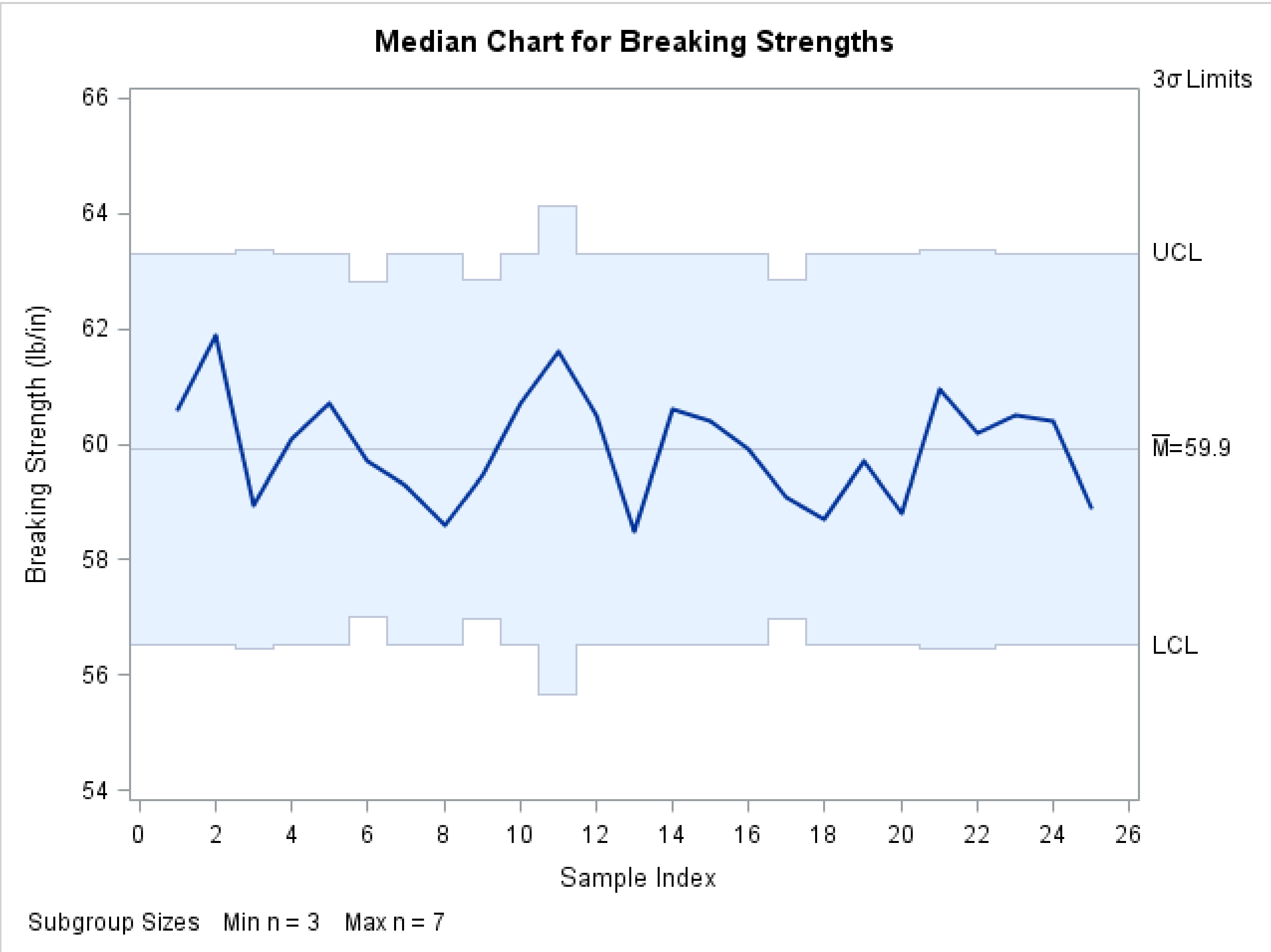
The following data set (Wire) contains breaking strength measurements recorded in pounds per inch for 25 samples from a metal wire manufacturing process. The subgroup sample sizes vary between 3 and 7.

```
data Wire;
   input Sample Size @;
   do i=1 to Size;
      input Breakstrength @@;
      output;
   end;
   drop i Size;
   label Breakstrength  ='Breaking Strength (lb/in)'
         Sample ='Sample Index';
   datalines;
 1  5 60.6 62.3 62.0 60.4 59.9
 2  5 61.9 62.1 60.6 58.9 65.3
 3  4 57.8 60.5 60.1 57.7
 4  5 56.8 62.5 60.1 62.9 58.9
 5  5 63.0 60.7 57.2 61.0 53.5
 6  7 58.7 60.1 59.7 60.1 59.1 57.3 60.9
 7  5 59.3 61.7 59.1 58.1 60.3
 8  5 61.3 58.5 57.8 61.0 58.6
 9  6 59.5 58.3 57.5 59.4 61.5 59.6
10  5 61.7 60.7 57.2 56.5 61.5
11  3 63.9 61.6 60.9
12  5 58.7 61.4 62.4 57.3 60.5
13  5 56.8 58.5 55.7 63.0 62.7
14  5 62.1 60.6 62.1 58.7 58.3
15  5 59.1 60.4 60.4 59.0 64.1
16  5 59.9 58.8 59.2 63.0 64.9
17  6 58.8 62.4 59.4 57.1 61.2 58.6
18  5 60.3 58.7 60.5 58.6 56.2
19  5 59.2 59.8 59.7 59.3 60.0
20  5 62.3 56.0 57.0 61.8 58.8
21  4 60.5 62.0 61.4 57.7
22  4 59.3 62.4 60.4 60.0
23  5 62.4 61.3 60.5 57.7 60.2
24  5 61.2 55.5 60.2 60.4 62.4
25  5 59.0 66.1 57.7 58.5 58.9
;
```

The following statements request a median chart, shown in Output 17.15.1, for the wire breaking strength measurements:

```
title 'Median Chart for Breaking Strengths';
ods graphics on;
proc shewhart data=Wire;
   mchart Breakstrength*Sample / odstitle=title;
run;
```

Output 17.15.1 Median Chart with Varying Sample Sizes



Note that the control limits vary with the subgroup sample size. The sample size legend in the lower left corner displays the minimum and maximum subgroup sample sizes.

By default, the control limits shown in Output 17.15.1 are 3σ limits estimated from the data. You can use the STDDEVIATIONS option and the SMETHOD= option in the MCHART statement to control how the estimate of the process standard deviation σ is calculated. The STDDEVIATIONS option specifies that the estimate of σ is to be calculated from subgroup standard deviations rather than subgroup ranges, the default. The SMETHOD= option specifies the method for estimating σ. You can specify the following methods:

- NOWEIGHT
- MVLUE
- RMSDF

The NOWEIGHT method, which is the default, requests an unweighted average of subgroup estimates, the MVLUE method requests a minimum variance linear unbiased estimate, and the RMSDF method requests a weighted root-mean-square estimate. Note that the RMSDF method is only available if, in addition, you specify the STDDEVIATIONS option. For details, see "Methods for Estimating the Standard Deviation" on page 1548.

The following statements contain five MCHART statements, which calculate five different estimates for σ by specifying different combinations of options:

```
title 'Estimates of the Process Standard Deviation';
proc shewhart data=Wire;
   mchart Breakstrength*Sample / outlimits=Wirelim1
                                 nochart outindex ='NOWEIGHT-Ranges';
   mchart Breakstrength*Sample / outlimits=Wirelim2
                                 stddeviations
                                 nochart outindex ='NOWEIGHT-Stds';
   mchart Breakstrength*Sample / outlimits=Wirelim3
                                 smethod  =mvlue
                                 nochart outindex ='MVLUE   -Ranges';
   mchart Breakstrength*Sample / outlimits=Wirelim4
                                 stddeviations
                                 smethod  =mvlue
                                 nochart outindex ='MVLUE   -Stds';
   mchart Breakstrength*Sample / outlimits=Wirelim5
                                 stddeviations
                                 smethod  =rmsdf
                                 nochart outindex ='RMSDF   -Stds';
run;
```

The OUTLIMITS= option names the data set containing the control limit information. The _STDDEV_ variable in the OUTLIMITS= data set contains the estimate of the process standard deviation. The OUTINDEX= option specifies the value of the _INDEX_ variable in the OUTLIMITS= data set and is used in this example to identify the estimation method. The following statements create a data set named Wlimits, which contains the five different estimates. This data set is listed in Output 17.15.2.

```
data Wlimits;
   set Wirelim1 Wirelim2 Wirelim3 Wirelim4 Wirelim5;
   keep _index_ _stddev_;
run;
```

Output 17.15.2 The Data Set Wlimits

The Wlimits Data Set

INDEX	_STDDEV_
NOWEIGHT-Ranges	2.11146
NOWEIGHT-Stds	2.15453
MVLUE -Ranges	2.11240
MVLUE -Stds	2.14790
RMSDF -Stds	2.17479

The median chart shown in Output 17.14.1 uses the estimate listed first in Output 17.15.2 ($\sigma = 2.11146$), since the MCHART statement used to create this chart omitted the STDDEVIATIONS option and the SMETHOD= option.

MRCHART Statement: SHEWHART Procedure

Overview: MRCHART Statement

The MRCHART statement creates charts for subgroup medians and ranges, which are used to analyze the central tendency and variability of a process.

You can use options in the MRCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted medians and ranges or as probability limits
- tabulate subgroup sample sizes, subgroup medians, subgroup ranges, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes, subgroup medians, and subgroup ranges in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify the method for estimating the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the charts more readable

- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing charts of medians and ranges with the MRCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 3, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: MRCHART Statement

This section introduces the MRCHART statement with simple examples that illustrate commonly used options. Complete syntax for the MRCHART statement is presented in the section "Syntax: MRCHART Statement" on page 1569, and advanced examples are given in the section "Examples: MRCHART Statement" on page 1592.

Creating Charts for Medians and Ranges from Raw Data

NOTE: See *Median and Range Charts Examples* in the SAS/QC Sample Library.

A consumer products company weighs detergent boxes (in pounds) to determine whether the fill process is in control. The following statements create a SAS data set named Detergent, which contains the weights for five boxes in each of 28 lots. A lot is considered a rational subgroup.

```
data Detergent;
   input Lot @;
   do i=1 to 5;
      input Weight @;
      output;
   end;
   drop i;
   datalines;
 1 17.39 26.93 19.34 22.56 24.49
 2 23.63 23.57 23.54 20.56 22.17
 3 24.35 24.58 23.79 26.20 21.55
 4 25.52 28.02 28.44 25.07 23.39
 5 23.25 21.76 29.80 23.09 23.70
 6 23.01 22.67 24.70 20.02 26.35
 7 23.86 24.19 24.61 26.05 24.18
 8 26.00 26.82 28.03 26.27 25.85
 9 21.58 22.31 25.03 20.86 26.94
```

```
10 22.64 21.05 22.66 29.26 25.02
11 26.38 27.50 23.91 26.80 22.53
12 23.01 23.71 25.26 20.21 22.38
13 23.15 23.53 22.98 21.62 26.99
14 26.83 23.14 24.73 24.57 28.09
15 26.15 26.13 20.57 25.86 24.70
16 25.81 23.22 23.99 23.91 27.57
17 25.53 22.87 25.22 24.30 20.29
18 24.88 24.15 25.29 29.02 24.46
19 22.32 25.96 29.54 25.92 23.44
20 25.63 26.83 20.95 24.80 27.25
21 21.68 21.11 26.07 25.17 27.63
22 26.72 27.05 24.90 30.08 25.22
23 31.58 22.41 23.67 23.47 24.90
24 28.06 23.44 24.92 24.64 27.42
25 21.10 22.34 24.96 26.50 24.51
26 23.80 24.03 24.75 24.82 27.21
27 25.10 26.09 27.21 24.28 22.45
28 25.53 22.79 26.26 25.85 25.64
;
```

A partial listing of Detergent is shown in Figure 17.43.

Figure 17.43 Partial Listing of the Data Set Detergent

The Data Set DETERGENT

Lot	Weight
1	17.39
1	26.93
1	19.34
1	22.56
1	24.49
2	23.63
2	23.57
2	23.54
2	20.56
2	22.17
3	24.35
3	24.58
3	23.79
3	26.20
3	21.55
4	25.52

The data set Detergent is said to be in "strung-out" form, since each observation contains the lot number and weight of a single box. The first five observations contain the weights for the first lot, the second five observations contain the weights for the second lot, and so on. Because the variable Lot classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable Weight contains the weights and is referred to as the *process variable* (or *process* for short).

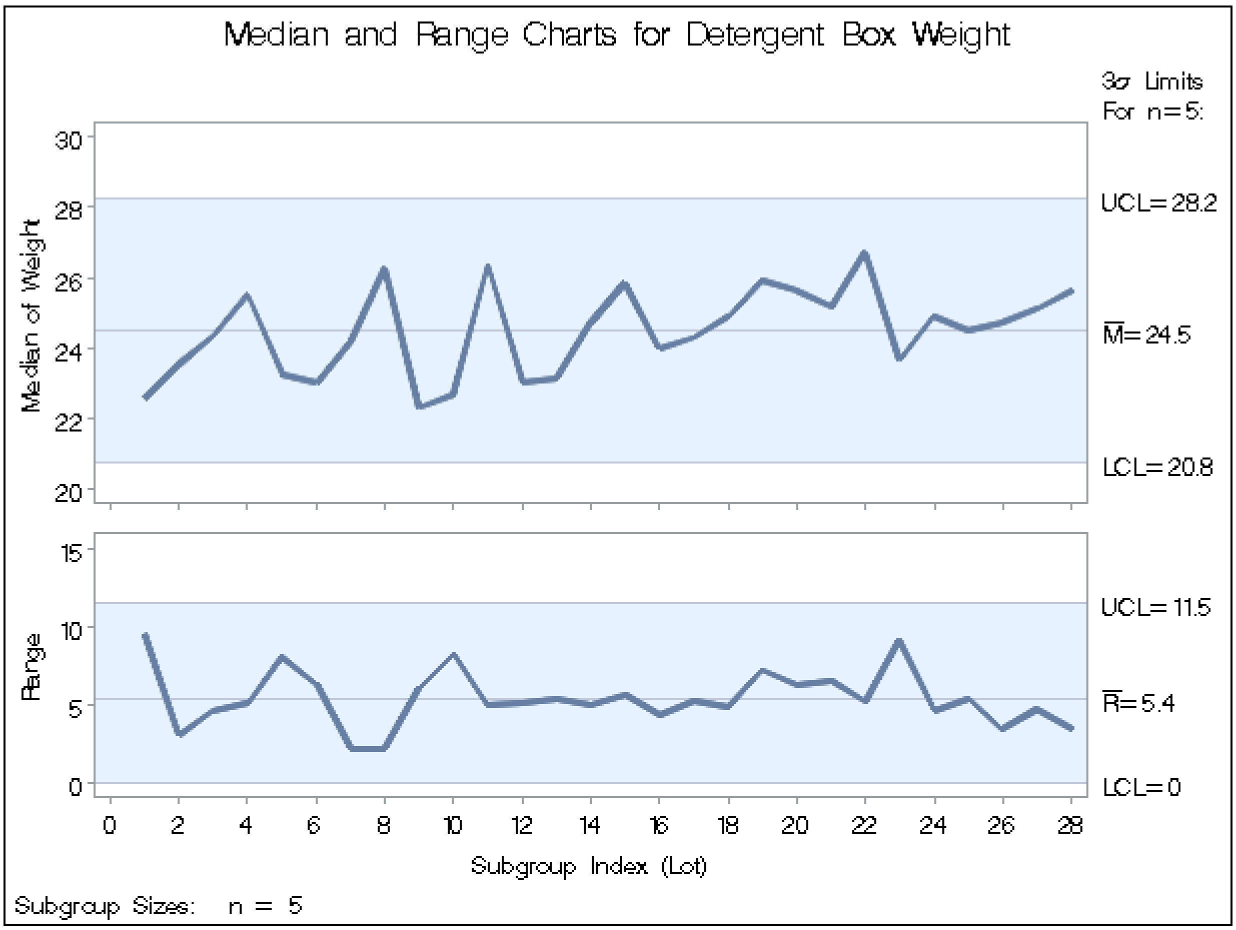
You can use median and range charts to determine whether the fill process is in control. The following statements create the charts shown in Figure 17.44:

```
ods graphics off;
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart data=Detergent;
   mrchart Weight*Lot ;
run;
```

This example illustrates the basic form of the MRCHART statement. After the keyword MRCHART, you specify the *process* to analyze (in this case, Weight) followed by an asterisk and the *subgroup-variable* (Lot).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Figure 17.44 Median and Range Charts (Traditional Graphics)



Each point on the median chart represents the median of the measurements for a particular lot. For instance, the weights for the first lot are 17.39, 19.34, 22.56, 24.49, and 26.93, and consequently, the median plotted for this lot is 22.56. Each point on the range chart represents the range of the measurements for a particular batch. For instance, the range plotted for the first lot is 26.93 – 17.39 = 9.54. Since all of the points lie within the control limits, you can conclude that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 17.32. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1567.

For computational details, see "Constructing Charts for Medians and Ranges" on page 1581. For more details on reading raw data, see "DATA= Data Set" on page 1588.

Creating Charts for Medians and Ranges from Summary Data

NOTE: See *Median and Range Charts Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create median and range charts using raw data (process measurements). However, in many applications, the data are provided as subgroup summary statistics. This example illustrates how you can use the MRCHART statement with data of this type.

The following data set (Detsum) provides the data from the preceding example in summarized form. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Lot). The variable WeightM contains the subgroup medians, the variable WeightR contains the subgroup ranges, and the variable WeightN contains the subgroup sample sizes (these are all five).

```
data Detsum;
   input Lot WeightM WeightR;
   WeightN = 5;
   datalines;
 1  22.56  9.54
 2  23.54  3.07
 3  24.35  4.65
 4  25.52  5.05
 5  23.25  8.04
 6  23.01  6.33
 7  24.19  2.19
 8  26.27  2.18
 9  22.31  6.08
10  22.66  8.21
11  26.38  4.97
12  23.01  5.05
13  23.15  5.37
14  24.73  4.95
15  25.86  5.58
16  23.99  4.35
17  24.30  5.24
18  24.88  4.87
19  25.92  7.22
20  25.63  6.30
21  25.17  6.52
22  26.72  5.18
23  23.67  9.17
24  24.92  4.62
25  24.51  5.40
26  24.75  3.41
27  25.10  4.76
28  25.64  3.47
;
```

A partial listing of Detsum is shown in Figure 17.45.

Figure 17.45 The Summary Data Set Detsum

Summary Data for Detergent Box Weights

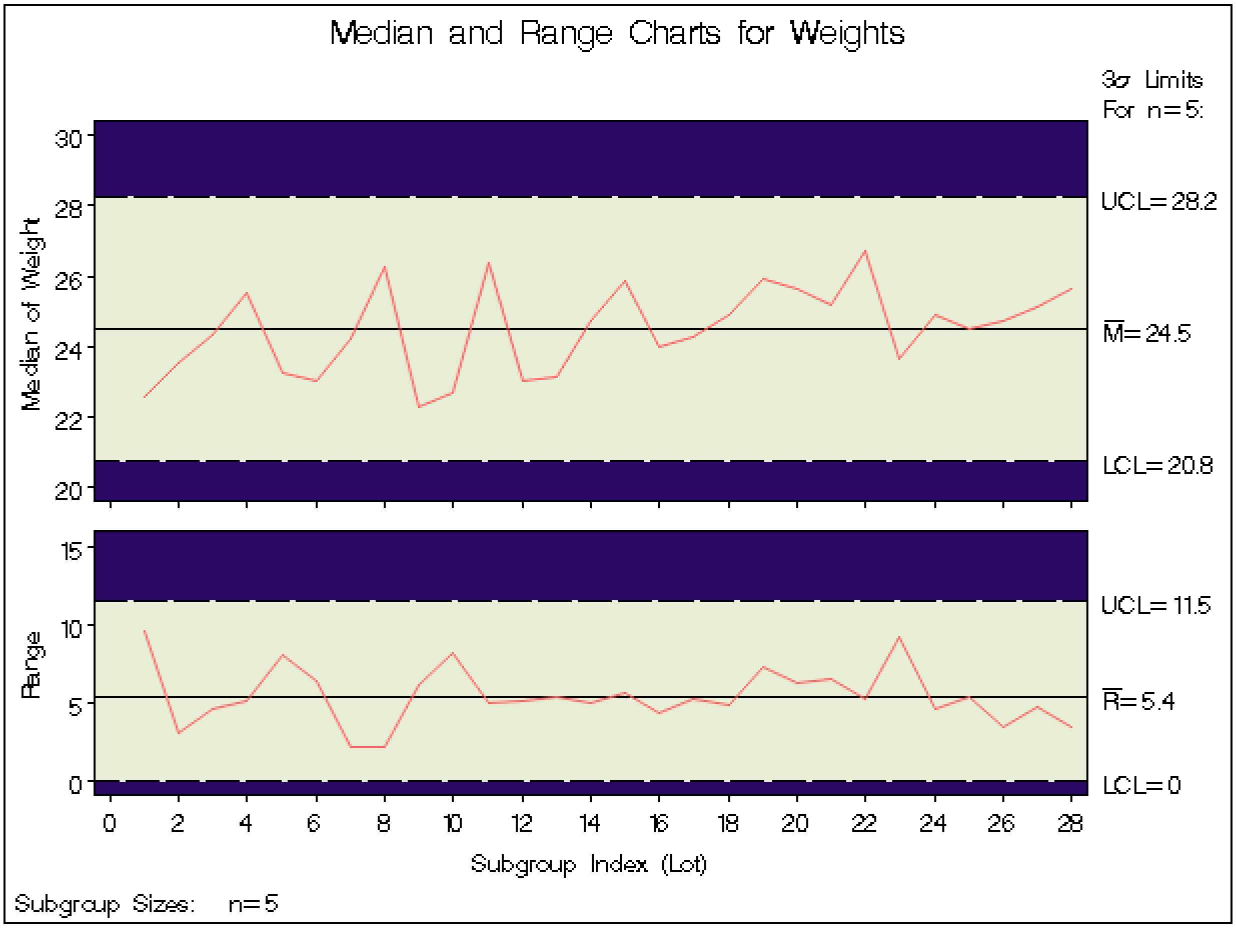
Lot	WeightM	WeightR	WeightN
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5

You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext=swiss;
symbol color = rose h = .8;
title 'Median and Range Charts for Weights';
proc shewhart history=Detsum;
   mrchart Weight*Lot / cframe   = vipb
                        cinfill  = ywh
                        cconnect = rose
                        coutfill = salmon;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement and MRCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The charts are shown in Figure 17.46.

Figure 17.46 Median and Range Charts from Summary Data Set Detsum (Traditional Graphics with NOGSTYLE)



Note that Weight is *not* the name of a SAS variable in the data set Detsum but is, instead, the common prefix for the names of the three SAS variables WeightM, WeightR, and WeightN. The suffix characters *M*, *R*, and *N* indicate *median*, *range*, and *sample size*, respectively. This naming convention enables you to specify three subgroup summary variables in the HISTORY= data set with a single name (Weight), referred to as the *process*. The name Lot specified after the asterisk is the name of the *subgroup-variable*.

In general, a HISTORY= input data set used with the MRCHART statement must contain the following variables:

- subgroup variable
- subgroup median variable
- subgroup range variable
- subgroup sample size variable

Furthermore, the names of the subgroup median, range, and sample size variables must begin with the prefix *process* specified in the MRCHART statement and end with the special suffix characters *M*, *R*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. Suppose that, instead of the variables WeightM, WeightR, and WeightN, the data set Detsum contained summary variables named medians, ranges, and sizes. The following statements would temporarily rename medians, ranges, and sizes to WeightM, WeightR, and WeightN, respectively:

```
proc shewhart
   history=Detsum (rename=(medians = WeightM
                           ranges  = WeightR
                           sizes   = WeightN ));
   mrchart Weight*Lot;
run;
```

In summary, the interpretation of *process* depends on the input data set:

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1589.

Saving Summary Statistics

NOTE: See *Median and Range Charts Examples* in the SAS/QC Sample Library.

In this example, the MRCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Detergent and create a summary data set named Dethist:

```
proc shewhart data=Detergent;
   mrchart Weight*Lot / outhistory = Dethist
                        nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the charts, which would be identical to the charts in Figure 17.44. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the MRCHART statement. A complete list of options is presented in the section "Syntax: MRCHART Statement" on page 1569.

Figure 17.47 contains a partial listing of Dethist.

Figure 17.47 The Summary Data Set Dethist

Summary Data Set DETHIST for Detergent Box Weights

Lot	WeightM	WeightR	WeightN
1	22.56	9.54	5
2	23.54	3.07	5
3	24.35	4.65	5
4	25.52	5.05	5
5	23.25	8.04	5

There are four variables in the data set Dethist.

- Lot contains the subgroup index.
- WeightM contains the subgroup medians.
- WeightR contains the subgroup ranges.
- WeightN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *M*, *R*, and *N* to the *process* Weight specified in the MRCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1585.

Saving Control Limits

NOTE: See *Median and Range Charts Examples* in the SAS/QC Sample Library.

You can save the control limits for median and range charts in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1567) or modify the limits with a DATA step program.

The following statements read measurements from the data set Detergent (see "Creating Charts for Medians and Ranges from Raw Data" on page 1557) and save the control limits displayed in Figure 17.44 in a data set named Detlim:

```
proc shewhart data=Detergent;
   mrchart Weight*Lot / outlimits=Detlim
                        nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set Detlim is listed in Figure 17.48.

Figure 17.48 The Data Set Detlim Containing Control Limit Information

Control Limits for Detergent Box Weights

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLM_	_MEAN_	_UCLM_
Weight	Lot	ESTIMATE	5	.002909021	3	20.7554	24.4996	28.2439

LCLR	_R_	_UCLR_	_STDDEV_
0	5.42036	11.4613	2.33041

The data set Detlim contains one observation with the limits for *process* Weight. The variables _LCLM_ and _UCLM_ contain the control limits for the medians, and the variable _MEAN_ contains the central line. The variables _LCLR_ and _UCLR_ contain the control limits for the ranges, and the variable _R_ contains the central line. The values of _MEAN_ and _STDDEV_ are estimates of the process mean and process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. For more information, see "OUTLIMITS= Data Set" on page 1584.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Detergent;
   mrchart Weight*Lot / outtable=Dtable
                        nochart;
run;
```

This data set contains one observation for each subgroup sample. The variables _SUBMED_, _SUBR_, and _SUBN_ contain the subgroup medians, subgroup ranges, and subgroup sample sizes. The variables _LCLM_ and _UCLM_ contain the control limits for the median chart, and the variables _LCLR_ and _UCLR_ contain the control limits for the range chart. The variable _MEAN_ contains the central line for the median chart, and the variable _R_ contains the central line for the range chart. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1586.

The data set Dtable is listed in Figure 17.49.

Figure 17.49 The Data Set Dtable

Summary Statistics and Control Limit Information

VAR	Lot	_SIGMAS_	_LIMITN_	_SUBN_	_LCLM_	_SUBMED_	_MEAN_	_UCLM_	_STDDEV_
Weight	1	3	5	5	20.7554	22.56	24.4996	28.2439	2.33041
Weight	2	3	5	5	20.7554	23.54	24.4996	28.2439	2.33041
Weight	3	3	5	5	20.7554	24.35	24.4996	28.2439	2.33041
Weight	4	3	5	5	20.7554	25.52	24.4996	28.2439	2.33041
Weight	5	3	5	5	20.7554	23.25	24.4996	28.2439	2.33041
Weight	6	3	5	5	20.7554	23.01	24.4996	28.2439	2.33041
Weight	7	3	5	5	20.7554	24.19	24.4996	28.2439	2.33041
Weight	8	3	5	5	20.7554	26.27	24.4996	28.2439	2.33041
Weight	9	3	5	5	20.7554	22.31	24.4996	28.2439	2.33041
Weight	10	3	5	5	20.7554	22.66	24.4996	28.2439	2.33041
Weight	11	3	5	5	20.7554	26.38	24.4996	28.2439	2.33041
Weight	12	3	5	5	20.7554	23.01	24.4996	28.2439	2.33041
Weight	13	3	5	5	20.7554	23.15	24.4996	28.2439	2.33041
Weight	14	3	5	5	20.7554	24.73	24.4996	28.2439	2.33041
Weight	15	3	5	5	20.7554	25.86	24.4996	28.2439	2.33041
Weight	16	3	5	5	20.7554	23.99	24.4996	28.2439	2.33041
Weight	17	3	5	5	20.7554	24.30	24.4996	28.2439	2.33041
Weight	18	3	5	5	20.7554	24.88	24.4996	28.2439	2.33041
Weight	19	3	5	5	20.7554	25.92	24.4996	28.2439	2.33041
Weight	20	3	5	5	20.7554	25.63	24.4996	28.2439	2.33041
Weight	21	3	5	5	20.7554	25.17	24.4996	28.2439	2.33041
Weight	22	3	5	5	20.7554	26.72	24.4996	28.2439	2.33041

EXLIM	_LCLR_	_SUBR_	_R_	_UCLR_	_EXLIMR_
	0	9.54	5.42036	11.4613	
	0	3.07	5.42036	11.4613	
	0	4.65	5.42036	11.4613	
	0	5.05	5.42036	11.4613	
	0	8.04	5.42036	11.4613	
	0	6.33	5.42036	11.4613	
	0	2.19	5.42036	11.4613	
	0	2.18	5.42036	11.4613	
	0	6.08	5.42036	11.4613	
	0	8.21	5.42036	11.4613	
	0	4.97	5.42036	11.4613	
	0	5.05	5.42036	11.4613	
	0	5.37	5.42036	11.4613	
	0	4.95	5.42036	11.4613	
	0	5.58	5.42036	11.4613	
	0	4.35	5.42036	11.4613	
	0	5.24	5.42036	11.4613	
	0	4.87	5.42036	11.4613	
	0	7.22	5.42036	11.4613	
	0	6.30	5.42036	11.4613	
	0	6.52	5.42036	11.4613	
	0	5.18	5.42036	11.4613	

Figure 17.49 *continued*

Summary Statistics and Control Limit Information

VAR	Lot	_SIGMAS_	_LIMITN_	_SUBN_	_LCLM_	_SUBMED_	_MEAN_	_UCLM_	_STDDEV_
Weight	23	3	5	5	20.7554	23.67	24.4996	28.2439	2.33041
Weight	24	3	5	5	20.7554	24.92	24.4996	28.2439	2.33041
Weight	25	3	5	5	20.7554	24.51	24.4996	28.2439	2.33041
Weight	26	3	5	5	20.7554	24.75	24.4996	28.2439	2.33041
Weight	27	3	5	5	20.7554	25.10	24.4996	28.2439	2.33041
Weight	28	3	5	5	20.7554	25.64	24.4996	28.2439	2.33041

EXLIM	_LCLR_	_SUBR_	_R_	_UCLR_	_EXLIMR_
	0	9.17	5.42036	11.4613	
	0	4.62	5.42036	11.4613	
	0	5.40	5.42036	11.4613	
	0	3.41	5.42036	11.4613	
	0	4.76	5.42036	11.4613	
	0	3.47	5.42036	11.4613	

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Dtable and display charts (not shown here) identical to those in Figure 17.44:

```
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart table=Dtable;
   mrchart Weight*Lot;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see “Specialized Control Charts: SHEWHART Procedure” on page 2096). For more information, see “TABLE= Data Set” on page 1590.

Reading Preestablished Control Limits

NOTE: See *Median and Range Charts Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Detlim saved control limits computed from the measurements in Detergent. This example shows how these limits can be applied to new data provided in the following data set:

```
data Detergent2;
   input Lot @;
   do i=1 to 5;
      input Weight @;
      output;
   end;
   drop i;
   datalines;
29 16.66 27.49 18.87 22.53 24.72
30 23.74 23.67 23.64 20.26 22.09
31 24.56 24.82 23.92 26.67 21.38
32 25.89 28.73 29.21 25.38 23.47
33 23.32 21.61 30.75 23.13 23.82
```

```
34 23.04 22.65 24.96 19.64 26.84
35 24.01 24.38 24.86 26.50 24.37
36 26.43 27.36 28.74 26.74 26.27
37 21.41 22.24 25.34 20.59 27.51
38 22.62 20.81 22.64 30.15 25.32
39 26.86 28.14 24.06 27.35 22.49
40 23.03 23.83 25.59 19.85 22.33
41 23.19 23.63 23.00 21.46 27.57
42 27.38 23.18 24.99 24.81 28.82
43 26.60 26.58 20.26 26.27 24.96
44 26.22 23.28 24.15 24.06 28.23
45 25.90 22.88 25.55 24.50 19.95
46 16.66 27.49 18.87 22.53 24.72
47 23.74 23.67 23.64 20.26 22.09
48 24.56 24.82 23.92 26.67 21.38
49 25.89 28.73 29.21 25.38 23.47
50 23.32 21.61 30.75 23.13 23.82
;
```

The following statements create median and range charts for the data in Detergent2 using the control limits in Detlim:

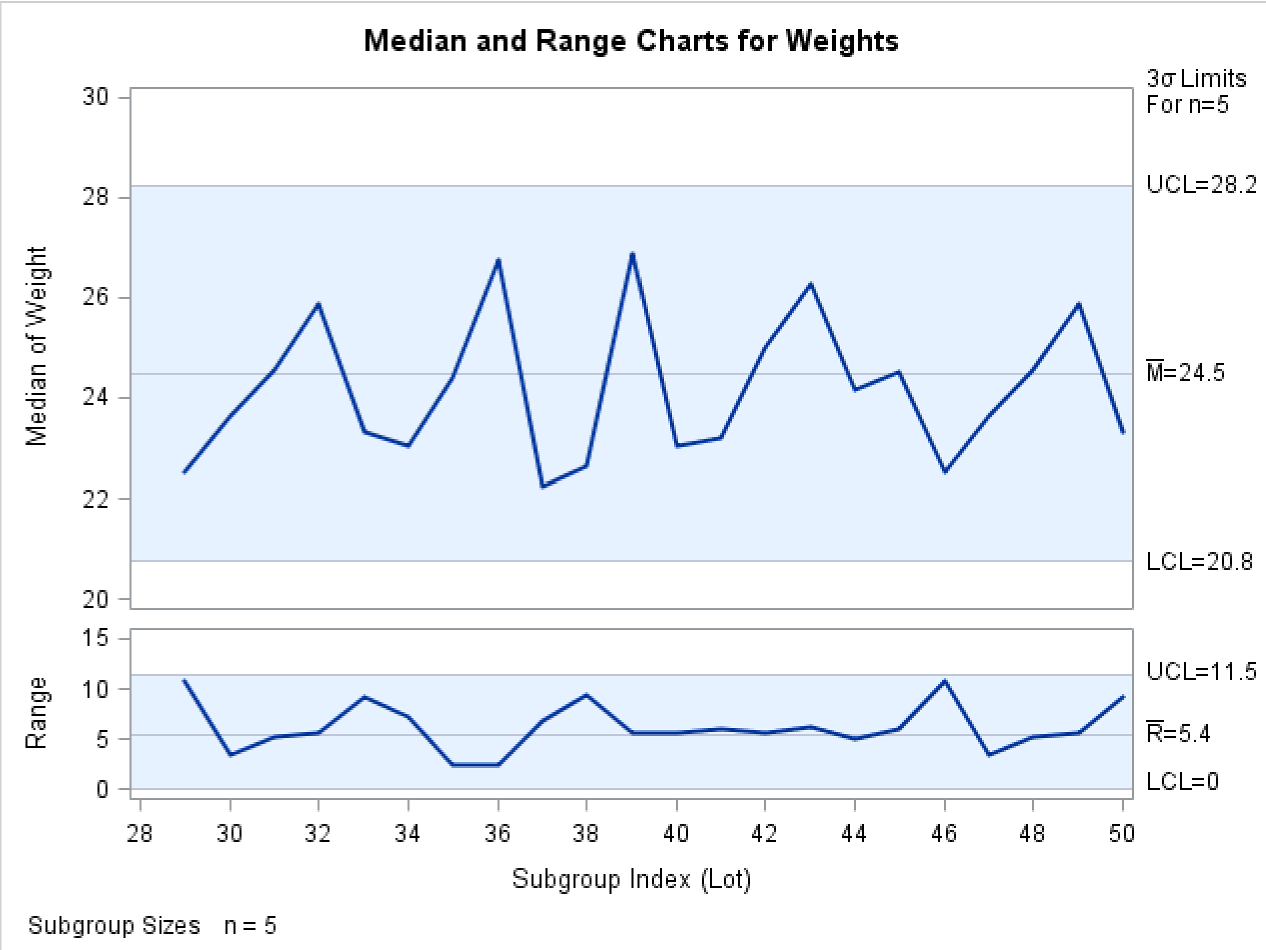
```
ods graphics on;
title 'Median and Range Charts for Weights';
proc shewhart data=Detergent2 limits=Detlim;
   mrchart Weight*Lot / odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the median and range charts are created using ODS Graphics instead of traditional graphics. The charts are shown in Figure 17.50.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Weight
- the value of _SUBGRP_ matches the *subgroup-variable* name Lot

Figure 17.50 Median and Range Charts for Second Set of Detergent Box Weights (ODS Graphics)



The charts indicate that the process is in control, since all the medians and ranges lie within the control limits.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1588 for details concerning the variables that you must provide.

Syntax: MRCHART Statement

The basic syntax for the MRCHART statement is as follows:

MRCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

MRCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=**'*character*'> / < *options* > **;**

You can use any number of MRCHART statements in the SHEWHART procedure. The components of the MRCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Medians and Ranges from Raw Data" on page 1557.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Medians and Ranges from Summary Data" on page 1560.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1564.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct median and range charts for Weight, LENGTH, and Width:

```
proc shewhart data=Measures;
   mrchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding MRCHART statement, DAY is the subgroup variable. For details, see "Subgroup Variables" on page 1871.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2026 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the medians and ranges.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2025 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create median and range charts using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   mrchart Weight*Day='*';
run;
```

options

enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The section "Summary of Options", which follows, lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1946 describes each option in detail.

Summary of Options

The following tables list the MRCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1946.

Table 17.31 MRCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	requests probability limits for chart
LIMITN=	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	specifies width of control limits in terms of multiple k of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	specifies color for area inside control limits
CLIMITS=	specifies color of control limits, central line, and related labels
LCLLABEL=	specifies label for lower control limit on median chart
LCLLABEL2=	specifies label for lower control limit on R chart
LIMLABSUBCHAR=	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	specifies line type for control limits

Table 17.31 *continued*

Option	Description
NDECIMAL=	specifies number of digits to right of decimal place in default labels for control limits and central line on median chart
NDECIMAL2=	specifies number of digits to right of decimal place in default labels for control limits and central line on *R* chart
NOCTL	suppresses display of central line on median chart
NOCTL2	suppresses display of central line on *R* chart
NOLCL	suppresses display of lower control limit on median chart
NOLCL2	suppresses display of lower control limit on *R* chart
NOLIMIT0	suppresses display of zero lower control limit on *R* chart
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit on median chart
NOUCL2	suppresses display of upper control limit on *R* chart
RSYMBOL=	specifies label for central line on *R* chart
UCLLABEL=	specifies label for upper control limit on median chart
UCLLABEL2=	specifies label for upper control limit on *R* chart
WLIMITS=	specifies width for control limits and central line
XSYMBOL=	specifies label for central line on median chart
Process Mean and Standard Deviation Options	
MEDCENTRAL=	specifies method for estimating process mean μ
MU0=	specifies known value of μ_0 for process mean μ
SIGMA0=	specifies known value σ_0 for process standard deviation σ
SMETHOD=	specifies method for estimating process standard deviation σ
TYPE=	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	labels every point on median chart
ALLLABEL2=	labels every point on *R* chart
CLABEL=	specifies color for labels
CCONNECT=	specifies color for line segments that connect points on chart
CFRAMELAB=	specifies fill color for frame around labeled points
CNEEDLES=	specifies color for needles that connect points to central line
COUT=	specifies color for portions of line segments that connect points outside control limits

Table 17.31 *continued*

Option	Description
COUTFILL=	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	specifies angle at which labels are drawn
LABELFONT=	specifies software font for labels (alias for the TEST-FONT= option)
LABELHEIGHT=	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=	labels points outside control limits on median chart
OUTLABEL2=	labels points outside control limits on *R* chart
SYMBOLLEGEND=	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=	specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	computes zone widths independently above and below center line
NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	suppresses tests across *phase* boundaries
TESTS=	specifies tests for special causes for the median chart
TESTS2=	specifies tests for special causes for the *R* chart
TEST2RESET=	enables tests for special causes to be reset for the *R* chart
TEST2RUN=	specifies length of pattern for Test 2
TEST3RUN=	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL=	provides labels for points where test is positive
TESTLABEL*n*=	specifies label for *n*th test for special causes
TESTNMETHOD=	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=	enables tests for special causes to be reset
WESTGARD=	requests that Westgard rules be applied to the median chart
ZONELABELS	adds labels A, B, and C to zone lines for median chart
ZONE2LABELS	adds labels A, B, and C to zone lines for *R* chart
ZONES	adds lines to median chart delineating zones A, B, and C
ZONES2	adds lines to *R* chart delineating zones A, B, and C
ZONEVALPOS=	specifies position of ZONEVALUES labels
ZONEVALUES	labels median chart zone lines with their values
ZONE2VALUES	labels *R* zone lines with their values

Table 17.31 *continued*

Option	Description
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	specifies color for labels indicating points where test is positive
CTESTSYMBOL=	specifies color for symbol used to plot points where test is positive
CZONES=	specifies color for lines and labels delineating zones A, B, and C
LTESTS=	specifies type of line connecting points where test is positive
LZONES=	specifies line type for lines delineating zones A, B, and C
TESTFONT=	specifies software font for labels at points where test is positive
TESTHEIGHT=	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	specifies plot symbol for points where test is positive
TESTSYMBOLHT=	specifies symbol height for points where test is positive
WTESTS=	specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill colors for frame for plot area
CTEXT=	specifies color for tick mark values and axis labels
DISCRETE	produces horizontal axis for discrete numeric group values
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default to *R* chart
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	specifies splitting character for axis labels

Table 17.31 *continued*

Option	Description
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis of median chart
VAXIS2=	specifies major tick mark values for vertical axis of *R* chart
VFORMAT=	specifies format for primary vertical axis tick mark labels
VFORMAT2=	specifies format for secondary vertical axis tick mark labels
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=	specifies width of axis lines
Plot Layout Options	
ALLN	plots means for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of charts
NOCHART2	suppresses creation of *R* chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
SEPARATE	displays median and *R* charts on separate screens or pages
TOTPANELS=	specifies number of pages or screens to be used to display chart
YPCT1=	specifies length of vertical axis on median chart as a percentage of sum of lengths of vertical axes for median and *R* charts
ZEROSTD	displays median chart regardless of whether $\hat{\sigma} = 0$

Table 17.31 *continued*

Option	Description
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= and HREF2= options
CVREF=	specifies color for lines requested by VREF= and VREF2= options
HREF=	specifies position of reference lines perpendicular to horizontal axis on median chart
HREF2=	specifies position of reference lines perpendicular to horizontal axis on *R* chart
HREFDATA=	specifies position of reference lines perpendicular to horizontal axis on median chart
HREF2DATA=	specifies position of reference lines perpendicular to horizontal axis on *R* chart
HREFLABELS=	specifies labels for HREF= lines
HREF2LABELS=	specifies labels for HREF2= lines
HREFLABPOS=	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	specifies line type for HREF= and HREF2= lines
LVREF=	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	specifies position of reference lines perpendicular to vertical axis on median chart
VREF2=	specifies position of reference lines perpendicular to vertical axis on *R* chart
VREFLABELS=	specifies labels for VREF= lines
VREF2LABELS=	specifies labels for VREF2= lines
VREFLABPOS=	position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=	specifies line type for grid requested with the ENDGRID option
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme points are clipped
CLIPLEGEND=	specifies text for clipping legend
CLIPLEGPOS=	specifies position of clipping legend

Table 17.31 *continued*

Option	Description
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features to median chart
ANNOTATE2=	specifies annotate data set that adds features to *R* chart
DESCRIPTION=	specifies description of median chart's GRSEG catalog entry
DESCRIPTION2=	specifies description of *R* chart's GRSEG catalog entry
FONT=	specifies software font for labels and legends on charts
NAME=	specifies name of median chart's GRSEG catalog entry
NAME2=	specifies name of *R* chart's GRSEG catalog entry
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	shades areas between control limits and connected points lying outside the limits
STARFILL=	specifies a variable identfying groups of stars filled with different colors
STARS=	specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	specifies the control limit infill transparency
MARKERS	plots subgroup points with markers
NOBLOCKREF	suppresses block and phase reference lines
NOBLOCKREFFILL	suppresses block and phase wall fills
NOFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOPHASEREF	suppresses block and phase reference lines
NOPHASEREFFILL	suppresses block and phase wall fills
NOREF	suppresses block and phase reference lines
NOREFFILL	suppresses block and phase wall fills
NOSTARFILLLEGEND	suppresses legend for levels of a STARFILL= variable

Table 17.31 *continued*

Option	Description
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	specifies a graph footnote
ODSFOOTNOTE2=	specifies a secondary graph footnote
ODSLEGENDEXPAND	specifies that legend entries contain all levels observed in the data
ODSTITLE=	specifies a graph title
ODSTITLE2=	specifies a secondary graph title
OUTFILLTRANSPARENCY=	specifies control limit outfill transparency
OVERLAYURL=	specifies URLs to associate with overlay points
OVERLAY2URL=	specifies URLs to associate with overlay points on secondary chart
PHASEPOS=	specifies vertical position of phase legend
PHASEREFLEVEL=	associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	draws central line labels using a simulated software font
STARTRANSPARENCY=	specifies star fill transparency
URL=	specifies a variable whose values are URLs to be associated with subgroups
URL2=	specifies a variable whose values are URLs to be associated with subgroups on secondary chart
Input Data Set Options	
MISSBREAK	specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	creates output data set containing subgroup summary statistics
OUTINDEX=	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	creates output data set containing control limits
OUTTABLE=	creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes

Table 17.31 *continued*

Option	Description
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	specifies α value and type for computing capability index confidence limits
LSL=	specifies list of lower specification limits
TARGET=	specifies list of target values
USL=	specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for *block-variable* legend
BLOCKLABTYPE=	specifies text size of *block-variable* legend
BLOCKPOS=	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE=	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	specifies color for STARCIRCLES= circles
CSTARFILL=	specifies color for filling stars
CSTAROUT=	specifies outline color for stars exceeding inner or outer circles
CSTARS=	specifies color for outlines of stars
LSTARCIRCLES=	specifies line types for STARCIRCLES= circles
LSTARS=	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	specifies reference circles for stars

Table 17.31 *continued*

Option	Description
STARINRADIUS=	specifies inner radius of stars
STARLABEL=	specifies vertices to be labeled
STARLEGEND=	specifies style of legend for star vertices
STARLEGENDLAB=	specifies label for STARLEGEND= legend
STAROUTRADIUS=	specifies outer radius of stars
STARSPECS=	specifies method used to standardize vertex variables
STARSTART=	specifies angle for first vertex
STARTYPE=	specifies graphical style of star
STARVERTICES=	superimposes star at each point on median chart
WSTARCIRCLES=	specifies width of STARCIRCLES= circles
WSTARS=	specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	specifies colors for primary chart overlay line segments
CCOVERLAY2=	specifies colors for secondary chart overlay line segments
COVERLAY=	specifies colors for primary chart overlay plots
COVERLAY2=	specifies colors for secondary chart overlay plots
COVERLAYCLIP=	specifies color for clipped points on overlays
LOVERLAY=	specifies line types for primary chart overlay line segments
LOVERLAY2=	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=	specifies variables to overlay on primary chart
OVERLAY2=	specifies variables to overlay on secondary chart
OVERLAY2HTML=	specifies links to associate with secondary chart overlay points
OVERLAY2ID=	specifies labels for secondary chart overlay points
OVERLAY2SYM=	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
OVERLAYHTML=	specifies links to associate with primary chart overlay points
OVERLAYID=	specifies labels for primary chart overlay points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols for primary chart overlays
OVERLAYSYMHT=	specifies symbol heights for primary chart overlays
WOVERLAY=	specifies widths of primary chart overlay line segments
WOVERLAY2=	specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	specifies a variable whose values create links to be associated with subgroups
HTML2=	specifies variable whose values create links to be associated with subgroups on secondary chart

Table 17.31 *continued*

Option	Description
HTML_LEGEND=	specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	specifies plot character for clipped points
CONNECTCHAR=	specifies character used to form line segments that connect points on chart
HREFCHAR=	specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	specifies characters indicating *symbol-variable*
TESTCHAR=	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	specifies line character for VREF= and VREF2= lines
ZONECHAR=	specifies character for lines that delineate zones for tests for special causes

Details: MRCHART Statement

Constructing Charts for Medians and Ranges

The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	mean of measurements in ith subgroup
R_i	range of measurements in ith subgroup
n_i	sample size of ith subgroup
N	the number of subgroups
x_{ij}	jth measurement in the ith subgroup, $j = 1, 2, 3, \ldots, n_i$
$x_{i(j)}$	jth largest measurement in the ith subgroup. Then

$$x_{i(1)} \leq x_{i(2)} \leq \ldots \leq x_{i(n_i)}$$

$\bar{\bar{X}}$	weighted average of subgroup means
M_i	median of the measurements in the ith subgroup:

$$M_i = \begin{cases} x_{i((n_i+1)/2)} & \text{if } n_i \text{ is odd} \\ (x_{i(n_i/2)} + x_{i((n_i/2)+1)})/2 & \text{if } n_i \text{ is even} \end{cases}$$

$\bar{M}$ — average of the subgroup medians:

$$\bar{M} = (n_1 M_1 + \ldots + n_N M_N)/(n_1 + \ldots + n_N)$$

$\tilde{M}$ — median of the subgroup medians. Denote the jth largest median by $M_{(j)}$ so that $M_{(1)} \leq M_{(2)} \leq \ldots \leq M_{(N)}$.

$$\tilde{M} = \begin{cases} M_{((N+1)/2)} & \text{if } N \text{ is odd} \\ (M_{(N/2)} + M_{(N/2)+1})/2 & \text{if } N \text{ is even} \end{cases}$$

$e_M(n)$ — standard error of the median of n independent, normally distributed variables with unit standard deviation (the value of $e_M(n)$ can be calculated with the STDMED function in a DATA step)

$Q_p(n)$ — 100pth percentile ($0 < p < 1$) of the distribution of the median of n independent observations from a normal population with unit standard deviation

$d_2(n)$ — expected value of the range of n independent normally distributed variables with unit standard deviation

$d_3(n)$ — standard error of the range of n independent observations from a normal population with unit standard deviation

z_p — 100pth percentile of the standard normal distribution

$D_p(n)$ — 100pth percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on a median chart indicates the value of a subgroup median (M_i). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the value plotted for this subgroup is $M_{10} = 15$. Each point on a range chart indicates the value of a subgroup range (R_i). For example, the value plotted for the tenth subgroup is $R_{10} = 19 - 12 = 7$.

Central Lines

On a median chart, the value of the central line indicates an estimate for μ, which is computed as

- $\bar{M}$ by default
- $\bar{\bar{X}}$ when you specify MEDCENTRAL=AVGMEAN
- $\tilde{M}$ when you specify MEDCENTRAL=MEDMED
- μ_0 when you specify μ_0 with the MU0= option

On the range chart, by default, the central line for the ith subgroup indicates an estimate for the expected value of R_i, which is computed as $d_2(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $d_2(n_i)\sigma_0$. The central line on the range chart varies with n_i.

Control Limits

You can compute the limits

- as a specified multiple (k) of the standard errors of M_i and R_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that M_i or R_i exceeds its limits

The following table provides the formulas for the limits:

Table 17.32 Limits for Median and Range Charts

Control Limits	
Median Chart	LCL = lower limit = $\bar{M} - k\hat{\sigma}e_M(n_i)$
	UCL = upper limit = $\bar{M} + k\hat{\sigma}e_M(n_i)$
Range Chart	LCL = lower control limit = $\max(d_2(n_i)\hat{\sigma} - kd_3(n_i)\hat{\sigma}, 0)$
	UCL = upper control limit = $d_2(n_i)\hat{\sigma} + kd_3(n_i)\hat{\sigma}$
Probability Limits	
Median Chart	LCL = lower limit = $\bar{M} - Q_{\alpha/2}(n_i)\hat{\sigma}$
	UCL = upper limit = $\bar{M} + Q_{1-\alpha/2}(n_i)\hat{\sigma}$
Range Chart	LCL = lower limit = $D_{\alpha/2}\hat{\sigma}$
	UCL = upper limit = $D_{1-\alpha/2}\hat{\sigma}$

In Table 17.32, replace $\bar{M}$ with $\overline{\overline{X}}$ if you specify MEDCENTRAL=AVGMEAN, and replace $\bar{M}$ with $\tilde{M}$ if you specify MEDCENTRAL=MEDMED. Replace $\bar{M}$ with μ_0 if you specify μ_0 with the MU0= option, and replace $\hat{\sigma}$ with σ_0 if you specify σ_0 with the SIGMA0= option.

The formulas assume that the data are normally distributed. Note that the limits for both charts vary with n_i and that the probability limits for R_i are asymmetric around the central line.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in the LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in the LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 17.33 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLM	lower control limit for subgroup median
LCLR	lower control limit for subgroup range
LIMITN	sample size associated with the control limits
LSL	lower specification limit
MEAN	estimate of process mean ($\bar{M}$, $\tilde{M}$, $\overline{\overline{X}}$, or μ_0)
_R_	value of central line on range chart
SIGMAS	multiple (k) of standard error of M_i or R_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the MRCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLM	upper control limit for subgroup median
UCLR	upper control limit for subgroup range
USL	upper specification limit
VAR	*process* specified in the MRCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variables _LIMITN_, _LCLM_, _UCLM_, _LCLR_, _R_, and _UCLR_.

2. If the limits are defined in terms of a multiple k of the standard errors of M_i and R_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - F_{med}(k,n))$, where $F_{med}(\cdot,n)$ is the cumulative distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value V, this value is assigned to _ALPHA_.

3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = F_{med}^{-1}(1-\alpha/2,n)$, where $F_{med}^{-1}(\cdot,n)$ is the inverse distribution function of the median of a random sample of n standard normally distributed observations, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value V, this value is assigned to _SIGMAS_.

4. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1874 for computational details.

5. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the MRCHART statement. For an example of an OUTLIMITS= data set, see "Saving Control Limits" on page 1564.

OUTHISTORY= Data Set

The OUTHISTORY= option saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup median variable named by *process* suffixed with *M*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Variables containing subgroup medians, ranges, and sample sizes are created for each *process* specified in the MRCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   mrchart (Width Diameter)*lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthM, WidthR, WidthN, DiameterM, DiameterR, and DiameterN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1563.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on median chart
EXLIMR	control limit exceeded on range chart
LCLM	lower control limit for median
LCLR	lower control limit for range
LIMITN	nominal sample size associated with the control limits
MEAN	estimate of process mean ($\bar{M}$, $\tilde{M}$, $\overline{\overline{X}}$, or μ_0)
_R_	average range
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBM	subgroup median
SUBN	subgroup sample size
SUBR	subgroup range
TESTS	tests for special causes signaled on median chart
TESTS2	tests for special causes signaled on range chart
UCLM	upper control limit for mean
UCLR	upper control limit for range
VAR	*process* specified in the MRCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).

2. The variable _TESTS_ is saved if you specify the TESTS= option. The kth character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.

3. The variable _TESTS2_ is saved if you specify the TESTS2= option. The kth character of a value of _TESTS2_ is k if Test k is positive at that subgroup.

4. The variables _EXLIM_, _EXLIMR_, _TESTS_, and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example of an OUTTABLE= data set, see "Saving Control Limits" on page 1564.

ODS Tables

The following table summarizes the ODS tables that you can request with the MRCHART statement.

Table 17.34 ODS Tables Produced with the MRCHART Statement

Table Name	Description	Options
MRCHART	median and *R* chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*). **NOTE:** In SAS/QC 13.1 the SHEWHART procedure does not support the creation of graphs that are editable with the ODS Graphics Editor.

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. MRCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1946.

When ODS Graphics is in effect, the MRCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 17.35.

Table 17.35 ODS Graphics Produced by the MRCHART Statement

ODS Graph Name	Plot Description
MRChart	median and *R* chart

See Chapter 3, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the MRCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the MRCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the *i*th subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the *i*th subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see “Displaying Stratification in Phases” on page 2031).

For an example of a DATA= data set, see “Creating Charts for Medians and Ranges from Raw Data” on page 1557.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   mrchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLM_, _MEAN_, _UCLM_, _LCLR_, _R_, and _UCLR_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_ , which are used to calculate the control limits according to the equations in Table 17.32

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1567.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedures or to read output data sets created with SAS summarization procedures, such as PROC UNIVARIATE.

A HISTORY= data set used with the MRCHART statement must contain the following variables:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup median variable for each *process*
- a subgroup range variable for each *process*
- a subgroup sample size variable for each *process*

The names of the subgroup mean, subgroup median, subgroup range, and subgroup sample size variables must be the *process* name concatenated with the special suffix characters *X*, *M*, *R*, and *N*, respectively. You must provide the subgroup mean variable only if you specify the MEDCENTRAL=AVGMEAN option.

For example, consider the following statements:

```
proc shewhart history=Summary;
   mrchart (Weight Yieldstrength)*Batch / medcentral=avgmean;
run;
```

The data set Summary must include the variables Batch, WeightX, WeightM, WeightR, WeightN, YieldstrengthX, YieldstrengthM, YieldstrengthR, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2031 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Medians and Ranges from Summary Data" on page 1560.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure or to read data sets created by other SAS procedures. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2096.

The following table lists the variables required in a TABLE= data set used with the MRCHART statement:

Table 17.36 Variables Required in a TABLE= Data Set

Variable	Description
LCLM	lower control limit for median
LCLR	lower control limit for range
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
_R_	average range
subgroup-variable	values of the *subgroup-variable*
SUBM	subgroup median
SUBN	subgroup sample size
SUBR	subgroup range
UCLM	upper control limit for median
UCLR	upper control limit for range

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes for subgroup medians and must be a character variable of length 8.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes for subgroup ranges and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1564.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, two methods are available for estimating the process standard deviation σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the ith subgroup.

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is equal to $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which n_i $geq2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

See Example 17.16 for illustrations of the default and MVLUE methods.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical (median chart)	DATA=	*process*
Vertical (median chart)	HISTORY=	subgroup median variable
Vertical (median chart)	TABLE=	_SUBMED_

You can specify distinct labels for the vertical axes of the median and *R* charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the median chart, and the second part labels the vertical axis of the *R* chart.

For an example, see Example 17.17.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: MRCHART Statement

This section provides advanced examples of the MRCHART statement.

Example 17.16: Working with Unequal Subgroup Sample Sizes

NOTE: See *Median and Range Charts-Unequal Subgroup Sizes* in the SAS/QC Sample Library.

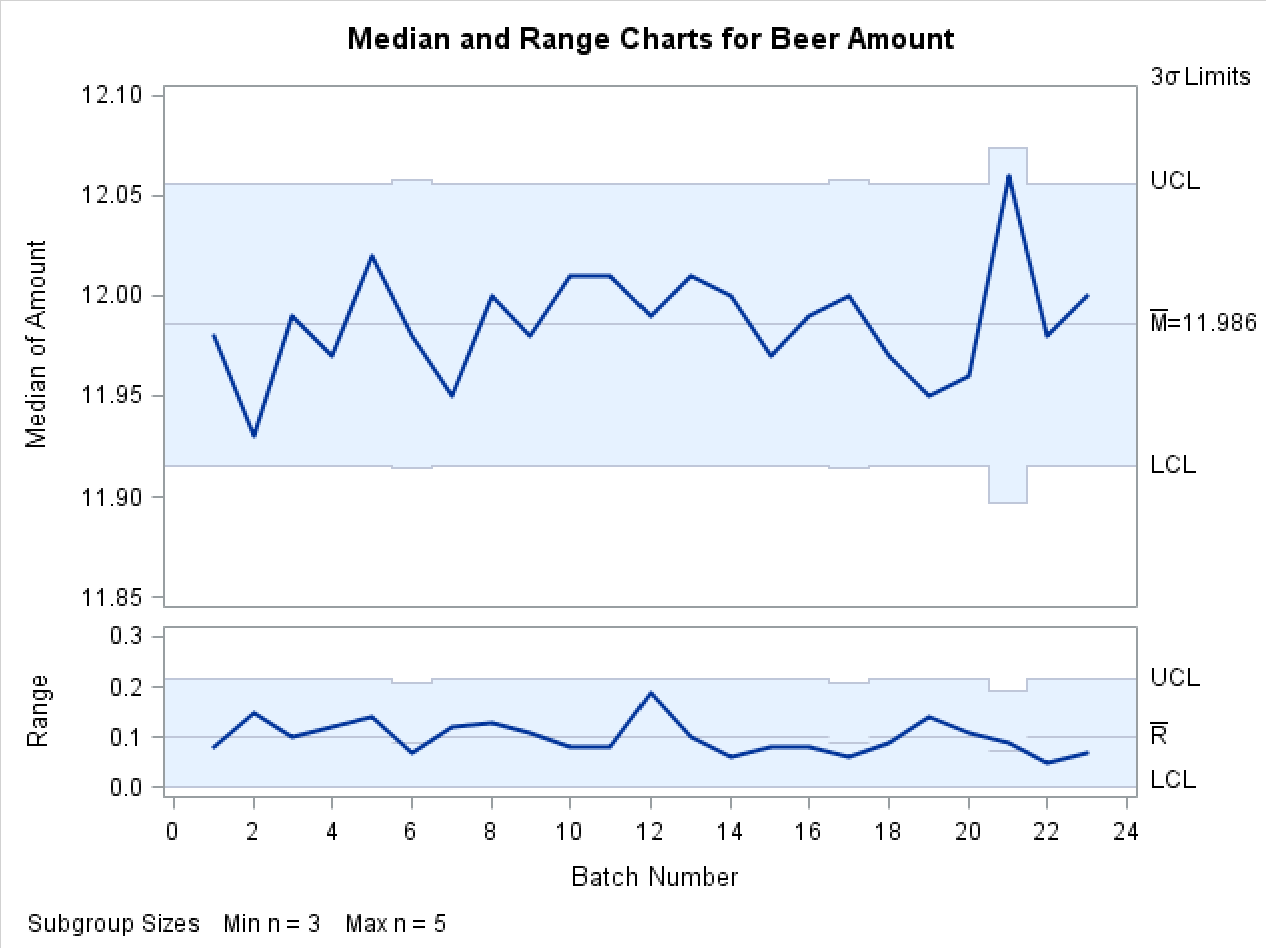
A brewery monitors its bottling process to ensure that each bottle is filled with the proper amount of beer. The following data set contains the amount of beer recorded in fluid ounces for 23 batches:

```
data Beer;
   input Batch size @;
   do i=1 to size;
      input Amount @@;
      output;
   end;
   drop i size;
   label Batch  ='Batch Number';
   datalines;
 1  5  12.01 11.97 11.93 11.98 12.00
 2  5  11.88 11.98 11.93 12.03 11.92
 3  5  11.93 11.99 12.00 12.03 11.95
 4  5  11.98 11.94 12.02 11.90 11.97
 5  5  12.02 12.02 11.98 12.04 11.90
 6  4  11.98 11.98 12.00 11.93
 7  5  11.93 11.95 12.02 11.91 12.03
 8  5  12.00 11.98 12.02 11.89 12.01
 9  5  11.98 11.93 11.99 12.02 11.91
10  5  11.97 12.02 12.05 12.01 11.97
11  5  12.02 12.01 11.97 12.02 11.94
12  5  11.93 11.83 11.99 12.02 12.01
13  5  12.01 11.98 11.94 12.04 12.01
14  5  11.98 11.96 12.02 12.00 12.00
15  5  11.97 11.99 12.03 11.95 11.96
16  5  11.99 11.95 11.96 12.03 12.01
17  4  11.99 11.97 12.03 12.01
18  5  11.94 11.96 11.98 12.03 11.97
19  5  11.97 11.87 11.90 12.01 11.95
20  5  11.96 11.94 11.96 11.98 12.05
21  3  12.06 12.07 11.98
22  5  12.01 11.98 11.96 11.97 12.00
23  5  12.00 12.02 12.03 11.99 11.96
;
```

A batch is regarded as a rational subgroup. Five bottles of beer are supposed to be tested in each batch. However, in batch 6 and batch 17 only four bottles are tested, and in batch 21 only three bottles are tested. The following statements request median and range charts, shown in Output 17.16.1, for the beer amounts:

```
ods graphics on;
title 'Median and Range Charts for Beer Amount';
proc shewhart data=Beer;
   mrchart Amount*Batch / odstitle=title;
run;
```

Output 17.16.1 Median and Range Charts with Varying Sample Sizes



Since none of the subgroup medians or subgroup ranges fall outside their respective control limits, you can conclude that the process is in control.

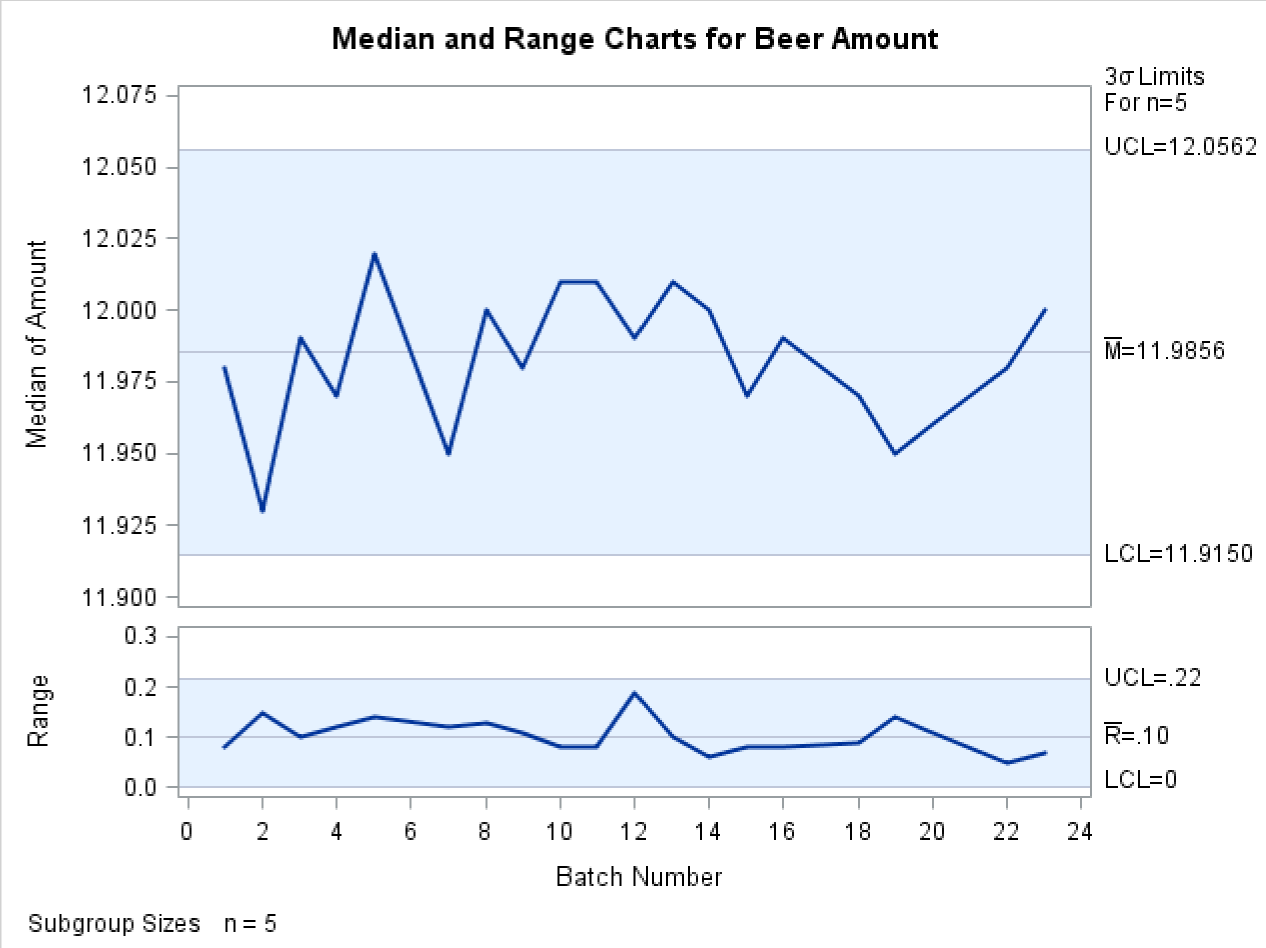
Note that the central line on the range chart and the control limits on both charts vary with the subgroup sample size. The subgroup sample size legend displays the minimum and maximum subgroup sample sizes.

The SHEWHART procedure provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for the control limits, as illustrated by the following statements:

```
title 'Median and Range Charts for Beer Amount';
proc shewhart data=Beer;
   mrchart Amount*Batch / limitn=5 odstitle=title;
run;
```

The resulting charts are shown in Output 17.16.2.

Output 17.16.2 Control Limits Based on Fixed Sample Size

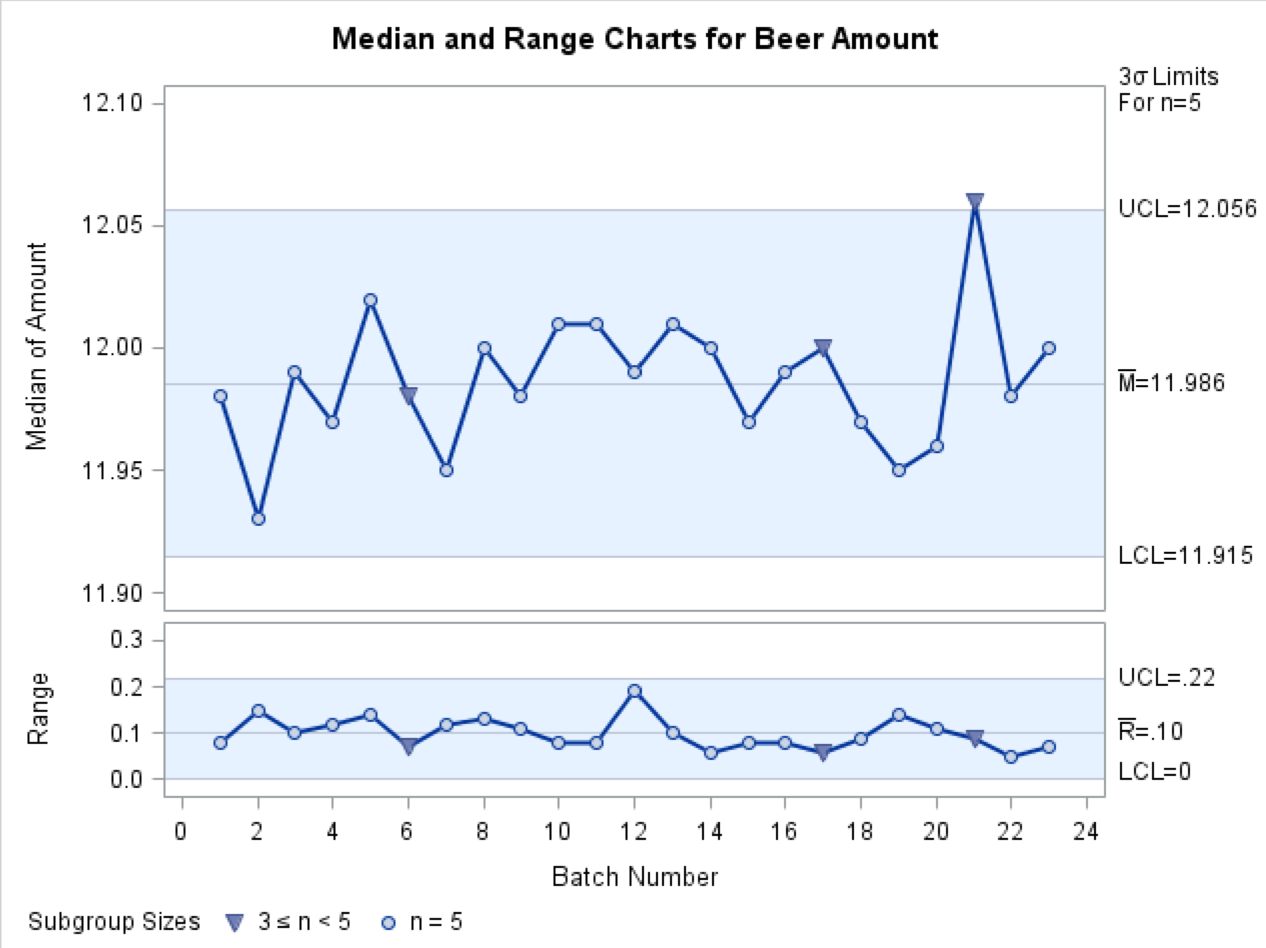


Note that the points displayed on the chart are those corresponding to subgroups whose sample size matches the nominal sample size (five) specified with the LIMITN= option. Points are not plotted for batches 6, 17, and 21. To display points for all subgroups (regardless of subgroup sample size), specify the ALLN option. The following statements produce the charts shown in Output 17.16.3:

```
title 'Median and Range Charts for Beer Amount';
proc shewhart data=Beer;
   mrchart Amount*Batch / limitn   = 5
                          odstitle = title
                          alln
                          nmarkers;
run;
```

The NMARKERS option requests special symbols that identify points for which the subgroup sample size differs from the nominal sample size. In Output 17.16.3, the median amount for batch 21 exceeds the upper control limits, indicating that the process is not in control. This illustrates the approximate nature of fixed control limits used with subgroup samples of varying sizes.

Output 17.16.3 Displaying All Subgroups Regardless of Sample Size



You can use the SMETHOD= option to determine how the process standard deviation σ is to be estimated when the subgroup sample sizes vary. The default method computes σ as an unweighted average of subgroup estimates of σ. The MVLUE method assigns greater weight to estimates of σ from subgroups with larger sample sizes. If the subgroup sample sizes are constant, the MVLUE method reduces to the NOWEIGHT method.

For details, see "Methods for Estimating the Standard Deviation" on page 1591. The following statements estimate σ using both methods:

```
proc shewhart data=Beer;
   mrchart Amount*Batch / outindex  = 'Default'
                          outlimits = Blim1
                          nochart;
   mrchart Amount*Batch / smethod   = mvlue
                          outindex  = 'MVLUE'
                          outlimits = Blim2
                          nochart;
run;

data Blimits;
   set Blim1 Blim2;
run;
```

The estimates are saved as values of the variable _STDDEV_ in the data set Blimits, which is listed in Output 17.16.4. The bookkeeping variable _INDEX_ identifies the estimate.

Output 17.16.4 The Data Set Blimits

The Data Set Blimits

VAR	_SUBGRP_	_INDEX_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLM_	_MEAN_
Amount	Batch	Default	ESTIMATE	V	V	3	V	11.9856
Amount	Batch	MVLUE	ESTIMATE	V	V	3	V	11.9856

UCLM	_LCLR_	_R_	_UCLR_	_STDDEV_
V	V	V	V	0.043938
V	V	V	V	0.044004

In the data set Blimits, the variables _LIMITN_, _ALPHA_, _LCLM_, _UCLM_, _LCLR_, _R_, and _UCLR_ have been assigned the special missing value *V*. This indicates that the quantities represented by these variables vary with the subgroup sample size.

Example 17.17: Specifying Axis Labels

NOTE: See *Median and Range Charts-Specifying Axis Labels* in the SAS/QC Sample Library.

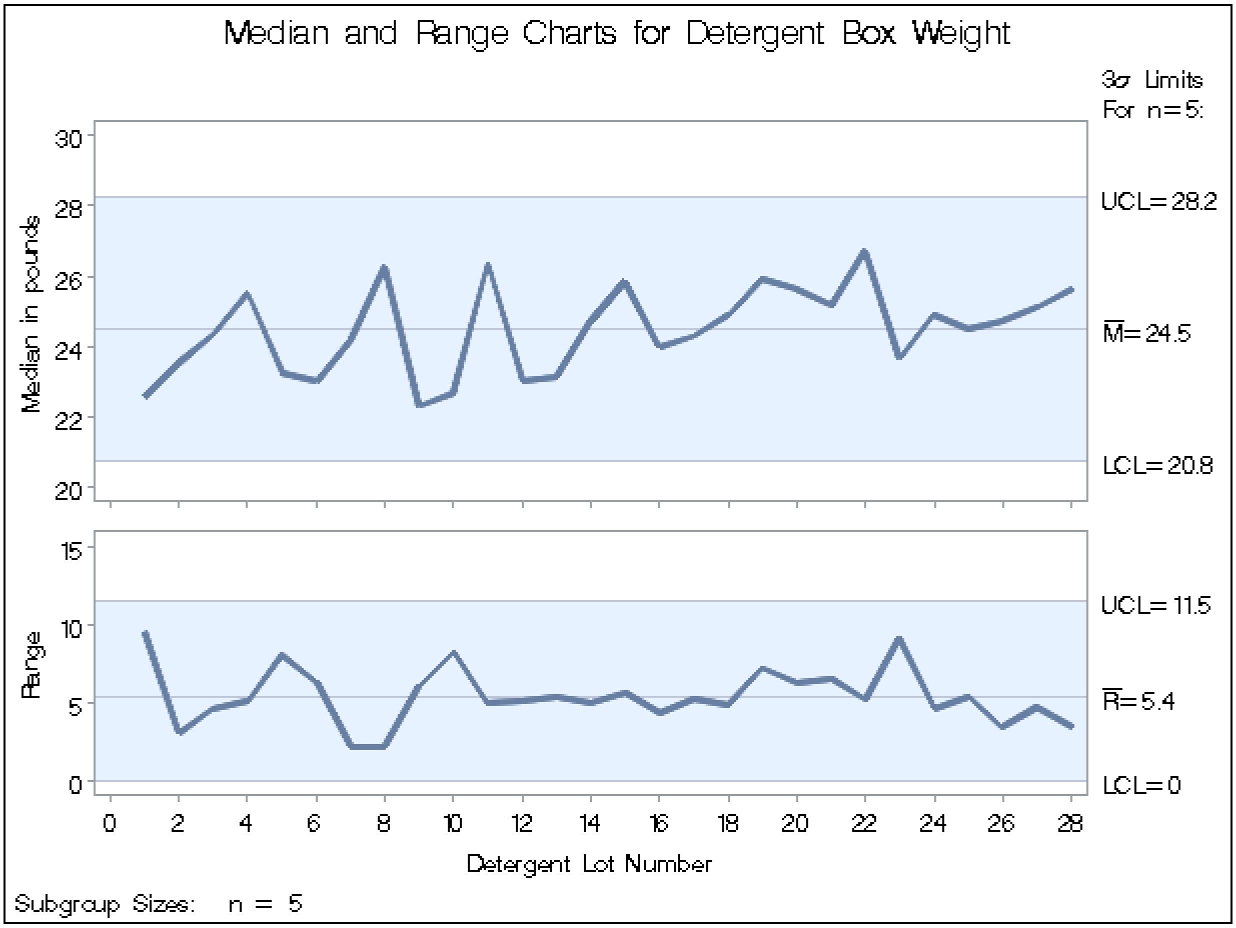
This example illustrates various methods for specifying axis labels and other axis features for median and range charts. For further details, see "Labeling Axes" on page 2063.

The charts in Figure 17.44, which are based on the data set Detergent introduced in the section "Getting Started: MRCHART Statement" on page 1557, display default labels for the horizontal and vertical axes. You can specify axis labels by associating labels with the *process* and *subgroup* variables as illustrated by the following statements:

```
ods graphics off;
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart data=Detergent;
   mrchart Weight*Lot / split   = '/';
   label Lot    = 'Detergent Lot Number'
         Weight = 'Median in pounds/Range';
run;
```

The charts are shown in Output 17.17.1. The horizontal axis label is the label associated with the *subgroup-variable* Lot. The vertical axis label for the median chart, referred to as the primary vertical axis label, is the first portion of the label associated with the *process* variable Weight, up to but not including the split character, which is specified with the SPLIT= option. The vertical axis label for the range chart, referred to as the secondary vertical axis label, is the second portion of the label associated with Weight.

Output 17.17.1 Customized Axis Labels Using Variable Labels



When the input data set is a HISTORY= data set, the vertical axis labels are determined by the label associated with the subgroup median variable. This is illustrated by the following statements, which use the data set Detsum introduced in "Creating Charts for Medians and Ranges from Summary Data" on page 1560:

```
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart history=Detsum;
   mrchart Weight*Lot / split = '/';
   label Lot      = 'Detergent Lot Number'
         WeightM = 'Median (pounds)/Range';
run;
```

The charts are identical to those in Output 17.17.1.

When the input data set is a TABLE= data set, the vertical axis labels are determined by the label associated with the subgroup median variable _SUBMED_. This is illustrated by the following statements, which use the data set Dtable introduced in Figure 17.49:

```
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart table=Dtable;
   mrchart Weight*Lot / split = '/';
   label Lot      = 'Detergent Lot Number'
         _submed_ = 'Median (pounds)/Range';
run;
```

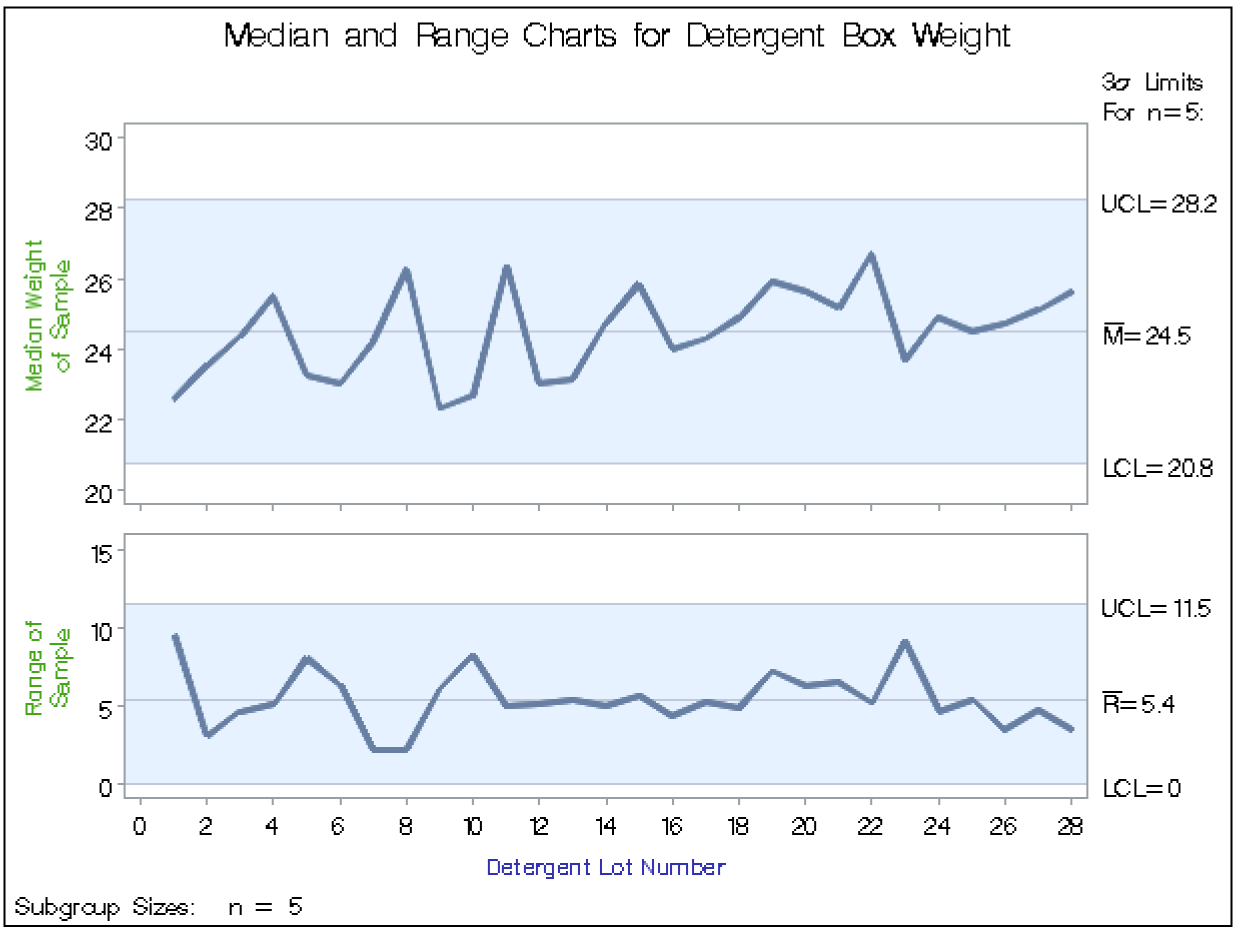
The charts are identical to those in Output 17.17.1.

When you are creating traditional graphics, you can use AXIS statements to enhance the appearance of the axes. This method is illustrated by the following statements:

```
title 'Median and Range Charts for Detergent Box Weight';
proc shewhart data=Detergent;
   mrchart Weight*Lot / haxis   = axis1
                        vaxis   = axis2
                        vaxis2  = axis3;
axis1 label=(c=bib  f=simplex 'Detergent Lot Number'          );
axis2 label=(c=vilg f=simplex 'Median Weight' j=c 'of Sample' );
axis3 label=(c=vilg f=simplex 'Range of'      j=c 'Sample'    );
run;
```

The charts are shown in Output 17.17.2.

Output 17.17.2 Customized Axis Labels Using AXIS Statements



You can use AXIS statements to customize a variety of axis features. For details, see *SAS/GRAPH: Reference*.

NPCHART Statement: SHEWHART Procedure

Overview: NPCHART Statement

The NPCHART statement creates np charts for the numbers of nonconforming (defective) items in subgroup samples.

You can use options in the NPCHART statement to

- compute control limits from the data based on a multiple of the standard error of the numbers of nonconforming items or as probability limits
- tabulate subgroup sample sizes, numbers of nonconforming items, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes and proportions of nonconforming items in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) proportion of nonconforming items for computing control limits
- specify the data as counts, proportions, or percentages of nonconforming items
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing np charts with the NPCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 3, “SAS/QC Graphics,” for more information about producing these different kinds of graphs.

Getting Started: NPCHART Statement

This section introduces the NPCHART statement with simple examples that illustrate commonly used options. Complete syntax for the NPCHART statement is presented in the section "Syntax: NPCHART Statement" on page 1610, and advanced examples are given in the section "Examples: NPCHART Statement" on page 1630.

Creating np Charts from Count Data

NOTE: See *np Chart Examples* in the SAS/QC Sample Library.

An electronics company manufactures circuits in batches of 500 and uses an np chart to monitor the number of failing circuits. Thirty batches are examined, and the failures in each batch are counted. The following statements create a SAS data set named Circuits,[5], which contains the failure counts:

```
data Circuits;
   input Batch Fail @@;
   datalines;
1    5   2   6   3  11   4   6    5   4
6    9   7  17   8  10   9  12   10   9
11   8  12   7  13   7  14  15   15   8
16  18  17  12  18  16  19   4   20   7
21  17  22  12  23   8  24   7   25  15
26   6  27   8  28  12  29   7   30   9
;
```

A partial listing of Circuits is shown in Figure 17.51.

Figure 17.51 The Data Set Circuits

Number of Failing Circuits

Batch	Fail
1	5
2	6
3	11
4	6
5	4

There is a single observation for each batch. The variable Batch identifies the subgroup sample and is referred to as the *subgroup-variable*. The variable Fail contains the number of nonconforming items in each subgroup sample and is referred to as the *process variable* (or *process* for short).

The following statements create the np chart shown in Figure 17.52:

```
ods graphics off;
title 'np Chart for the Number of Failing Circuits';
proc shewhart data=Circuits;
   npchart Fail*Batch / subgroupn = 500;
run;
```

[5]This data set is also used in the "Getting Started" section of "PCHART Statement: SHEWHART Procedure" on page 1639

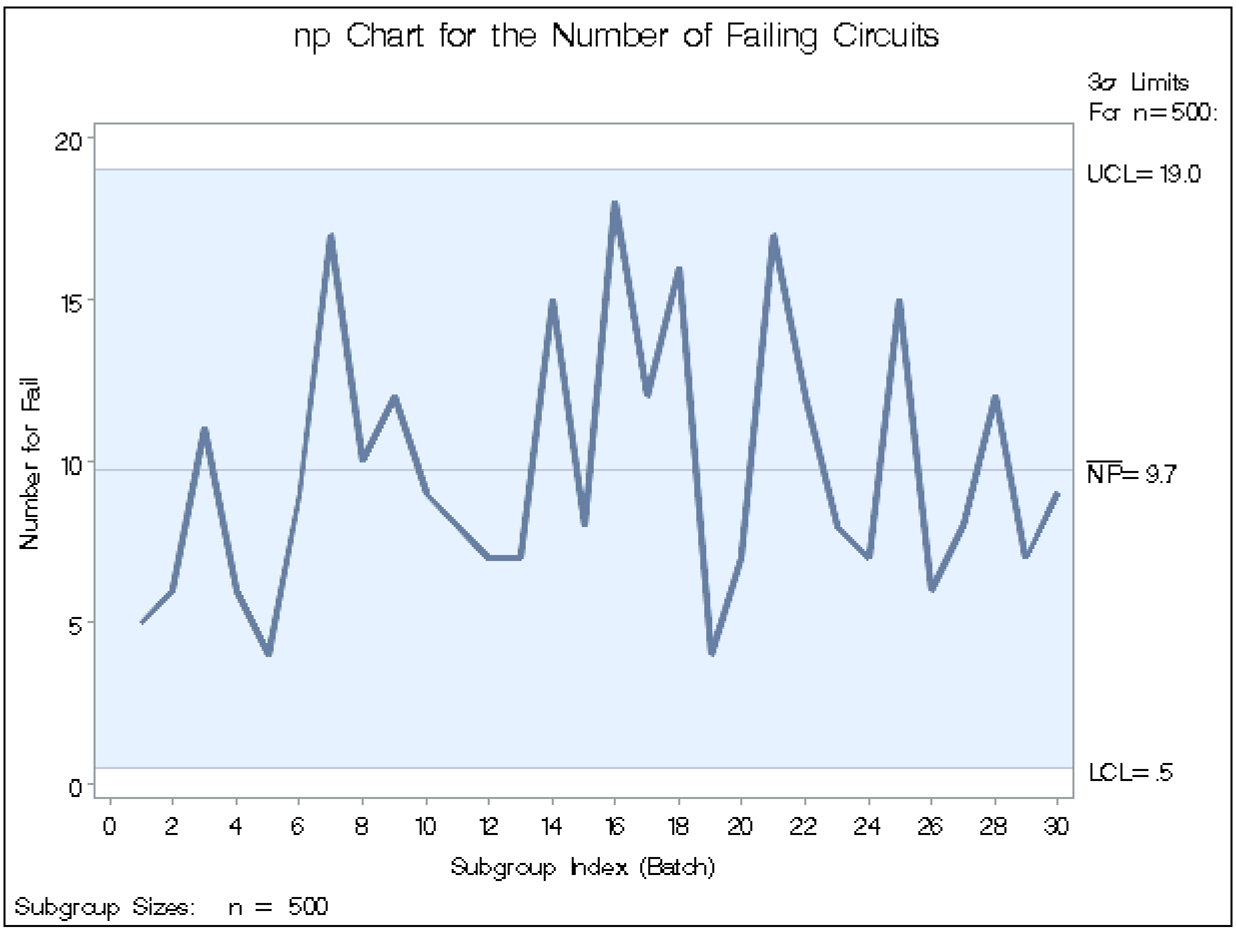
This example illustrates the basic form of the NPCHART statement. After the keyword NPCHART, you specify the *process* to analyze (in this case, Fail), followed by an asterisk and the *subgroup-variable* (Batch).

The input data set is specified with the DATA= option in the PROC SHEWHART statement. The SUBGROUPN= option specifies the number of items in each subgroup sample and is required with a DATA= input data set. The SUBGROUPN= option specifies one of the following:

- a constant subgroup sample size (in this case)
- a variable in the input data set whose values provide the subgroup sample sizes (see the next example)

Options such as SUBGROUPN= are specified after the slash (/) in the NPCHART statement. A complete list of options is presented in the section “Syntax: NPCHART Statement” on page 1610.

Figure 17.52 *np* Chart for Circuit Failures (Traditional Graphics)



Each point on the *np* chart represents the number of nonconforming items for a particular subgroup. For instance, the value plotted for the first batch is 5.

Since all the points fall within the control limits, it can be concluded that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in “Control Limits” on page 1622. You can also read control limits from an input data set; see “Reading Preestablished Control Limits” on page 1609. For computational details, see “Constructing Charts for Number Nonconforming (np Charts)” on page 1621. For more details on reading raw data, see “DATA= Data Set” on page 1626.

Creating np Charts from Summary Data

NOTE: See *np Chart Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create *np* charts using raw data (counts of nonconforming items). However, in many applications, the data are provided in summarized form as proportions or percentages of nonconforming items. This example illustrates how you can use the NPCHART statement with data of this type.

The following data set provides the data from the preceding example in summarized form:

```
data Cirprop;
   input Batch pFailed @@;
   sizes=500;
   datalines;
 1  0.010   2  0.012   3  0.022   4  0.012   5  0.008
 6  0.018   7  0.034   8  0.020   9  0.024  10  0.018
11  0.016  12  0.014  13  0.014  14  0.030  15  0.016
16  0.036  17  0.024  18  0.032  19  0.008  20  0.014
21  0.034  22  0.024  23  0.016  24  0.014  25  0.030
26  0.012  27  0.016  28  0.024  29  0.014  30  0.018
;
```

A partial listing of Cirprop is shown in Figure 17.53. The subgroups are still indexed by Batch. The variable pFailed contains the proportions of nonconforming items, and the variable Sampsize contains the subgroup sample sizes.

Figure 17.53 The Data Set Cirprop

Subgroup Proportions of Nonconforming Items

Batch	pFailed	sizes
1	0.010	500
2	0.012	500
3	0.022	500
4	0.012	500
5	0.008	500

The following statements create an *np* chart identical to the one in Figure 17.52:

```
title 'np Chart for the Number of Failing Circuits';
proc shewhart data=Cirprop;
   npchart pFailed*Batch / subgroupn=Sampsize
                           dataunit =proportion;
label pFailed = 'Number of FAIL';
run;
```

The DATAUNIT= option specifies that the values of the *process* (pFailed) are proportions of nonconforming items. By default, the values of the *process* are assumed to be counts of nonconforming items (see the previous example).

Alternatively, you can read the data set Cirprop by specifying it as a HISTORY= data set in the PROC SHEWHART statement. A HISTORY= data set used with the NPCHART statement must contain the following variables:

- subgroup variable
- subgroup proportion of nonconforming items variable
- subgroup sample size variable

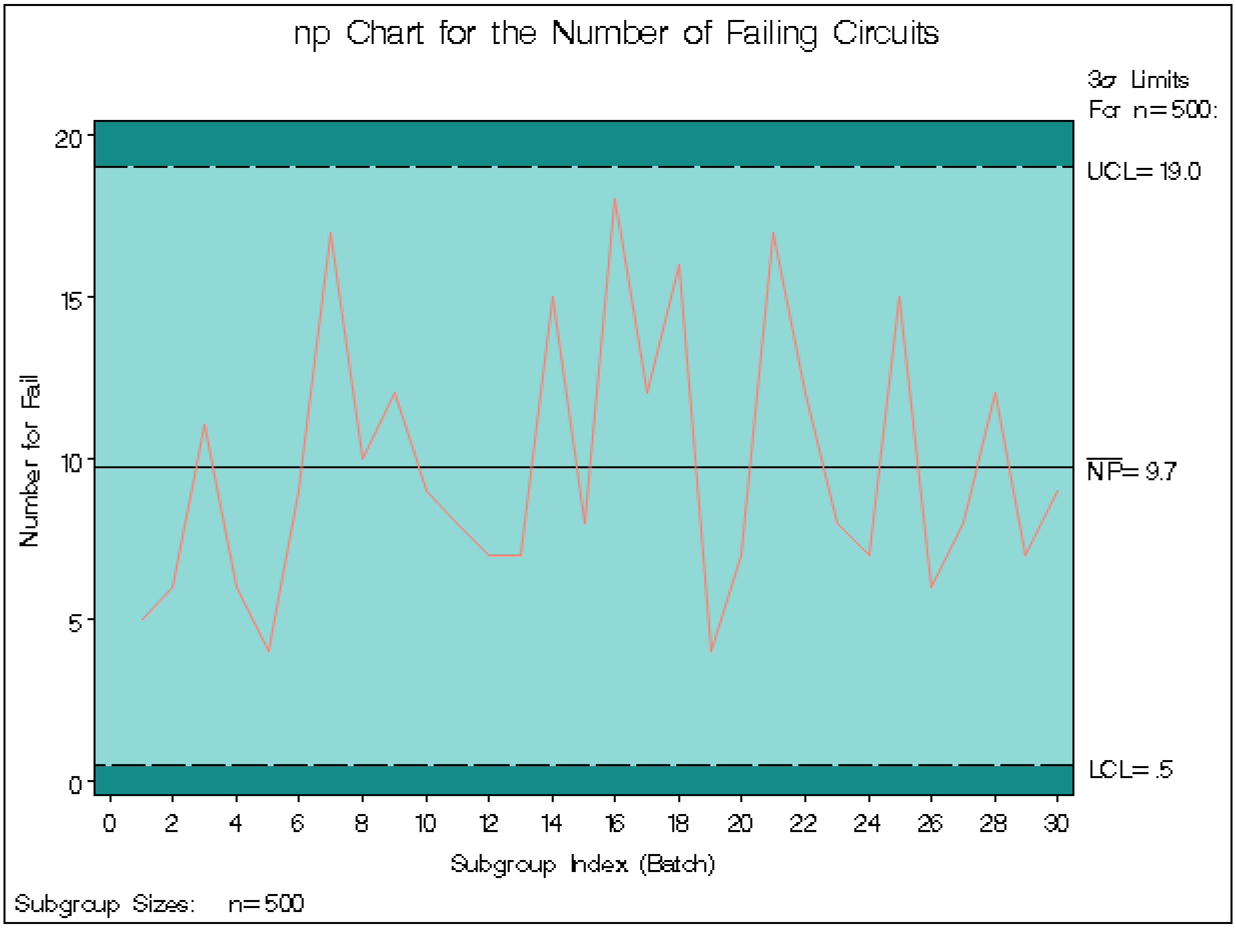
Furthermore, the names of the subgroup proportion and sample size variables must begin with the *process* name specified in the NPCHART statement and end with the special suffix characters *P* and *N*, respectively.

To specify Cirprop as a HISTORY= data set and Fail as the *process*, you must rename the variables pFailed and Sampsize to FailP and FailN, respectively. The following statements temporarily rename pFailed and Sampsize for the duration of the procedure step:

```
options nogstyle;
goptions ftext=swiss;
title 'np Chart for the Number of Failing Circuits';
proc shewhart history=Cirprop(rename=(pFailed =FailP
                                      sizes=FailN ));
   npchart Fail*Batch / cframe    = vibg
                        cinfill   = vlibg
                        coutfill  = salmon
                        cconnect  = salmon;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the NPCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting *np* chart is shown in Figure 17.54.

Figure 17.54 np Chart for Circuit Failures (Traditional Graphics with NOGSTYLE)



In this example, it is more convenient to use Cirprop as a DATA= data set than as a HISTORY= data set. As illustrated in the next example, it is generally more convenient to use the HISTORY= option for input data sets that have been created previously by the SHEWHART procedure as OUTHISTORY= data sets.

For more information, see "HISTORY= Data Set" on page 1628.

Saving Proportions of Nonconforming Items

NOTE: See *np Chart Examples* in the SAS/QC Sample Library.

In this example, the NPCHART statement is used to create a data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read the number of nonconforming items from the data set Circuits (see "Creating np Charts from Count Data" on page 1601) and create a summary data set named Cirhist:

```
proc shewhart data=Circuits;
   npchart Fail*Batch / subgroupn  = 500
                        outhistory = Cirhist
                        nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 17.52. Figure 17.55 contains a partial listing of Cirhist.

Figure 17.55 The Data Set Cirhist

Subgroup Proportions of Failing Circuits

Batch	FailP	FailN
1	0.010	500
2	0.012	500
3	0.022	500
4	0.012	500
5	0.008	500

There are three variables in the data set Cirhist.

- Batch contains the subgroup index.
- FailP contains the subgroup proportion of nonconforming items.
- FailN contains the subgroup sample size.

Note that the variables containing the subgroup proportions of nonconforming items and subgroup sample sizes are named by adding the suffix characters *P* and *N* to the *process* Fail specified in the NPCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1624.

Saving Control Limits

NOTE: See *np Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for an *np* chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1609) or modify the limits with a DATA step program.

The following statements read the number of nonconforming items per subgroup from the data set Circuits (see "Creating np Charts from Count Data" on page 1601) and save the control limits displayed in Figure 17.52 in a data set named Cirlim:

```
proc shewhart data=Circuits;
   npchart Fail*Batch / subgroupn=500
                        outlimits=Cirlim
                        nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Cirlim is listed in Figure 17.56.

Figure 17.56 The Data Set Cirlim Containing Control Limit Information

Control Limits for the Number of Failing Circuits

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_P_	_LCLNP_	_NP_	_UCLNP_
Fail	Batch	ESTIMATE	500	.002320877	3	0.019467	0.46539	9.73333	19.0013

The data set Cirlim contains one observation with the limits for *process* Fail. The variables _LCLNP_ and _UCLNP_ contain the lower and upper control limits, and the variable _NP_ contains the central line. The variable _P_ contains the average proportion of nonconforming items. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _P_ is an estimate or a standard value.

For more information, see "OUTLIMITS= Data Set" on page 1623.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Circuits;
   npchart Fail*Batch / subgroupn=500
                        outtable=Cirtable
                        nochart;
run;
```

The Cirtable data set contains one observation for each subgroup sample. The variables _SUBNP_ and _SUBN_ contain the subgroup numbers of nonconforming items and subgroup sample sizes, respectively. The variables _LCLNP_ and _UCLNP_ contain the lower and upper control limits, and the variable _NP_ contains the central line. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1624.

The data set Cirtable is listed in Figure 17.57.

Figure 17.57 The Data Set Cirtable

Number Nonconforming and Control Limit Information

VAR	Batch	_SIGMAS_	_LIMITN_	_SUBN_	_LCLNP_	_SUBNP_	_NP_	_UCLNP_	_EXLIM_
Fail	1	3	500	500	0.46539	5	9.73333	19.0013	
Fail	2	3	500	500	0.46539	6	9.73333	19.0013	
Fail	3	3	500	500	0.46539	11	9.73333	19.0013	
Fail	4	3	500	500	0.46539	6	9.73333	19.0013	
Fail	5	3	500	500	0.46539	4	9.73333	19.0013	
Fail	6	3	500	500	0.46539	9	9.73333	19.0013	
Fail	7	3	500	500	0.46539	17	9.73333	19.0013	
Fail	8	3	500	500	0.46539	10	9.73333	19.0013	
Fail	9	3	500	500	0.46539	12	9.73333	19.0013	
Fail	10	3	500	500	0.46539	9	9.73333	19.0013	
Fail	11	3	500	500	0.46539	8	9.73333	19.0013	
Fail	12	3	500	500	0.46539	7	9.73333	19.0013	
Fail	13	3	500	500	0.46539	7	9.73333	19.0013	
Fail	14	3	500	500	0.46539	15	9.73333	19.0013	
Fail	15	3	500	500	0.46539	8	9.73333	19.0013	
Fail	16	3	500	500	0.46539	18	9.73333	19.0013	
Fail	17	3	500	500	0.46539	12	9.73333	19.0013	
Fail	18	3	500	500	0.46539	16	9.73333	19.0013	
Fail	19	3	500	500	0.46539	4	9.73333	19.0013	
Fail	20	3	500	500	0.46539	7	9.73333	19.0013	
Fail	21	3	500	500	0.46539	17	9.73333	19.0013	
Fail	22	3	500	500	0.46539	12	9.73333	19.0013	
Fail	23	3	500	500	0.46539	8	9.73333	19.0013	
Fail	24	3	500	500	0.46539	7	9.73333	19.0013	
Fail	25	3	500	500	0.46539	15	9.73333	19.0013	
Fail	26	3	500	500	0.46539	6	9.73333	19.0013	
Fail	27	3	500	500	0.46539	8	9.73333	19.0013	
Fail	28	3	500	500	0.46539	12	9.73333	19.0013	
Fail	29	3	500	500	0.46539	7	9.73333	19.0013	
Fail	30	3	500	500	0.46539	9	9.73333	19.0013	

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Cirtable and display an np chart (not shown here) identical to the chart in Figure 17.52:

```
title 'np Chart for the Number of Failing Circuits';
proc shewhart table=Cirtable;
   npchart Fail*Batch;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2096). For more information, see "TABLE= Data Set" on page 1629.

Reading Preestablished Control Limits

NOTE: See *np Chart Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Cirlim saved control limits computed from the data in Circuits. This example shows how these limits can be applied to new data provided in the following data set:

```
data Circuit2;
   input Batch Fail;
   datalines;
31  12   32   9   33  16   34   9
35   3   36   8   37  20   38   4
39   8   40   6   41  12   42  16
43   9   44   2   45  10   46   8
47  14   48  10   49  11   50   9
;
```

The following statements create an np chart for the data in Circuit2 using the control limits in Cirlim:

```
ods graphics on;
title 'np Chart for the Proportion of Failing Circuits';
proc shewhart data=Circuit2 limits=Cirlim;
   npchart Fail*Batch / subgroupn = 500
                        odstitle  = title;
run;
```

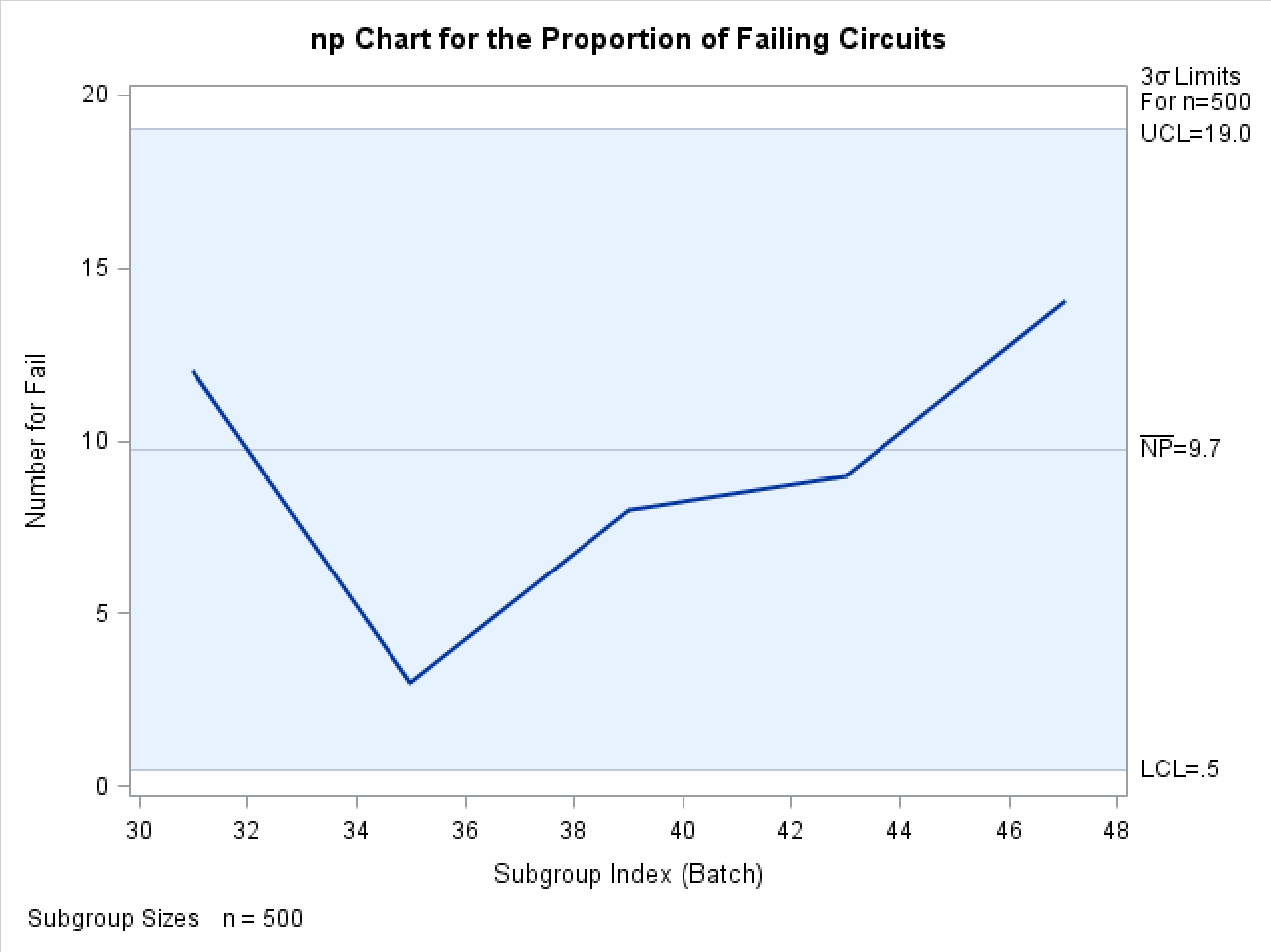
The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the np chart is created using ODS Graphics instead of traditional graphics.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Fail
- the value of _SUBGRP_ matches the *subgroup-variable* name Batch

The resulting np chart is shown in Figure 17.58.

Figure 17.58 *np* Chart for Second Set of Circuit Failures (ODS Graphics)



The number of nonconforming items in the 37th batch exceeds the upper control limit, signaling that the process is out of control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step; see Example 17.21 for an example. See "LIMITS= Data Set" on page 1627 for details concerning the variables that you must provide.

Syntax: NPCHART Statement

The basic syntax for the NPCHART statement is as follows:

NPCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

NPCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'*> / < *options* > **;**

You can use any number of NPCHART statements in the SHEWHART procedure. The components of the NPCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If numbers of nonconforming items are read from a DATA= data set, *process* must be the name of the variable containing the numbers. For an example, see "Creating np Charts from Count Data" on page 1601.
- If proportions of nonconforming items are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating np Charts from Summary Data" on page 1603.
- If numbers of nonconforming items and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1606.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct *np* charts for Rejects and Reworks:

```
proc shewhart data=Measures;
   npchart (Rejects Reworks)*Sample / subgroupn=100;
run;
```

Note that when data are read from a DATA= data set, the SUBGROUPN= option, which specifies subgroup sample sizes, is required.

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding NPCHART statement, Sample is the subgroup variable. For details, see "Subgroup Variables" on page 1871.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2026 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot numbers of nonconforming items.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2025 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create an *np* chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   npchart Rejects*Day='*' / subgroupn=100;
run;
```

options

enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The "Summary of Options" on page 1612 section, which follows, lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1946 describes each option in detail.

Summary of Options

The following tables list the NPCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1946.

Table 17.37 NPCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	requests probability limits for chart
LIMITN=	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
PROBLIMITS=	requests probability limits at discrete values
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
ACTUALALPHA	displays the actual probability of a point being outside the control limits in the control limits legend
CINFILL=	specifies color for area inside control limits
CLIMITS=	specifies color of control limits, central line, and related labels
LCLLABEL=	specifies label for lower control limit
LIMLABSUBCHAR=	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit

Table 17.37 *continued*

Option	Description
LLIMITS=	specifies line type for control limits
NDECIMAL=	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMIT0	suppresses display of lower control limit if it is 0
NOLIMIT1	suppresses display of upper control limit if it is 1 (100%)
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit
NPSYMBOL=	specifies label for central line
UCLLABEL=	specifies label for upper control limit
WLIMITS=	specifies width for control limits and central line
Standard Value Options	
P0=	specifies known (standard) value p_0 for proportion of nonconforming items
TYPE=	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	labels every point on np chart
CLABEL=	specifies color for labels
CCONNECT=	specifies color for line segments that connect points on chart
CFRAMELAB=	specifies fill color for frame around labeled points
CNEEDLES=	specifies color for needles that connect points to central line
COUT=	specifies color for portions of line segments that connect points outside control limits
COUTFILL=	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	specifies angle at which labels are drawn
LABELFONT=	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=	labels points outside control limits
SYMBOLLEGEND=	specifies LEGEND statement for levels of *symbol-variable*

Table 17.37 *continued*

Option	Description
SYMBOLORDER=	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=	specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	computes zone widths independently above and below center line
NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	suppresses tests across *phase* boundaries
TESTS=	specifies tests for special causes
TEST2RUN=	specifies length of pattern for Test 2
TEST3RUN=	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL=	provides labels for points where test is positive
TESTLABEL*n*=	specifies label for *n*th test for special causes
TESTNMETHOD=	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=	enables tests for special causes to be reset
WESTGARD=	requests that Westgard rules be applied
ZONELABELS	adds labels A, B, and C to zone lines
ZONES	adds lines delineating zones A, B, and C
ZONEVALPOS=	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	specifies color for labels indicating points where test is positive
CTESTSYMBOL=	specifies color for symbol used to plot points where test is positive
CZONES=	specifies color for lines and labels delineating zones A, B, and C
LTESTS=	specifies type of line connecting points where test is positive
LZONES=	specifies line type for lines delineating zones A, B, and C
TESTFONT=	specifies software font for labels at points where test is positive
TESTHEIGHT=	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	specifies plot symbol for points where test is positive
TESTSYMBOLHT=	specifies symbol height for points where test is positive

Table 17.37 *continued*

Option	Description
WTESTS=	specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill colors for frame for plot area
CTEXT=	specifies color for tick mark values and axis labels
DISCRETE	produces horizontal axis for discrete numeric group values
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis
VFORMAT=	specifies format for vertical axis tick mark labels
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=	specifies width of axis lines
Plot Layout Options	
ALLN	plots means for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart

Table 17.37 *continued*

Option	Description
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	specifies number of pages or screens to be used to display chart
ZEROSTD	displays np chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= options
CVREF=	specifies color for lines requested by VREF= options
HREF=	specifies position of reference lines perpendicular to horizontal axis
HREFDATA=	specifies position of reference lines perpendicular to horizontal axis
HREFLABELS=	specifies labels for HREF= lines
HREFLABPOS=	specifies position of HREFLABELS= labels
LHREF=	specifies line type for HREF= lines
LVREF=	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	specifies position of reference lines perpendicular to vertical axis
VREFLABELS=	specifies labels for VREF= lines
VREFLABPOS=	position of VREFLABELS= labels
Grid Options	
CGRID=	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=	specifies line type for grid requested with the ENDGRID option
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme points are clipped
CLIPLEGEND=	specifies text for clipping legend

Table 17.37 *continued*

Option	Description
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features chart
DESCRIPTION=	specifies description of *np* chart's GRSEG catalog entry
FONT=	specifies software font for labels and legends on charts
NAME=	specifies name of *np* chart's GRSEG catalog entry
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	shades areas between control limits and connected points lying outside the limits
STARFILL=	specifies a variable identfying groups of stars filled with different colors
STARS=	specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	specifies the control limit infill transparency
MARKERS	plots subgroup points with markers
NOBLOCKREF	suppresses block and phase reference lines
NOBLOCKREFFILL	suppresses block and phase wall fills
NOFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOPHASEREF	suppresses block and phase reference lines
NOPHASEREFFILL	suppresses block and phase wall fills
NOREF	suppresses block and phase reference lines
NOREFFILL	suppresses block and phase wall fills
NOSTARFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	specifies a graph footnote
ODSFOOTNOTE2=	specifies a secondary graph footnote

Table 17.37 *continued*

Option	Description
ODSLEGENDEXPAND	specifies that legend entries contain all levels observed in the data
ODSTITLE=	specifies a graph title
ODSTITLE2=	specifies a secondary graph title
OUTFILLTRANSPARENCY=	specifies control limit outfill transparency
OVERLAYURL=	specifies URLs to associate with overlay points
PHASEPOS=	specifies vertical position of phase legend
PHASEREFLEVEL=	associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	draws central line labels using a simulated software font
STARTRANSPARENCY=	specifies star fill transparency
URL=	specifies a variable whose values are URLs to be associated with subgroups
Input Data Set Options	
DATAUNIT	specifies that input values are proportions or percentages (rather than counts) of nonconforming items
MISSBREAK	specifies that observations with missing values are not to be processed
SUBGROUPN	specifies subgroup sample sizes as constant number *n* or as values of variable in a DATA= data set
Output Data Set Options	
OUTHISTORY=	creates output data set containing subgroup summary statistics
OUTINDEX=	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	creates output data set containing control limits
OUTTABLE=	creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Table 17.37 *continued*

Option	Description
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for *block-variable* legend
BLOCKLABTYPE=	specifies text size of *block-variable* legend
BLOCKPOS=	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE=	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	specifies color for STARCIRCLES= circles
CSTARFILL=	specifies color for filling stars
CSTAROUT=	specifies outline color for stars exceeding inner or outer circles
CSTARS=	specifies color for outlines of stars
LSTARCIRCLES=	specifies line types for STARCIRCLES= circles
LSTARS=	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	specifies reference circles for stars
STARINRADIUS=	specifies inner radius of stars
STARLABEL=	specifies vertices to be labeled
STARLEGEND=	specifies style of legend for star vertices
STARLEGENDLAB=	specifies label for STARLEGEND= legend
STAROUTRADIUS=	specifies outer radius of stars
STARSPECS=	specifies method used to standardize vertex variables
STARSTART=	specifies angle for first vertex
STARTYPE=	specifies graphical style of star
STARVERTICES=	superimposes star at each point on chart
WSTARCIRCLES=	specifies width of STARCIRCLES= circles

Table 17.37 *continued*

Option	Description
WSTARS=	specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	specifies colors for overlay line segments
COVERLAY=	specifies colors for overlay plots
COVERLAYCLIP=	specifies color for clipped points on overlays
LOVERLAY=	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=	specifies variables to overlay on chart
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
OVERLAYHTML=	specifies links to associate with overlay points
OVERLAYID=	specifies labels for overlay points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols for overlays
OVERLAYSYMHT=	specifies symbol heights for overlays
WOVERLAY=	specifies widths of overlay line segments
Options for Interactive Control Charts	
HTML=	specifies a variable whose values create links to be associated with subgroups
HTML_LEGEND=	specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	specifies plot character for clipped points
CONNECTCHAR=	specifies character used to form line segments that connect points on chart
HREFCHAR=	specifies line character for HREF= lines
SYMBOLCHARS=	specifies characters indicating *symbol-variable*
TESTCHAR=	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	specifies line character for VREF= lines
ZONECHAR=	specifies character for lines that delineate zones for tests for special causes

Details: NPCHART Statement

Constructing Charts for Number Nonconforming (np Charts)

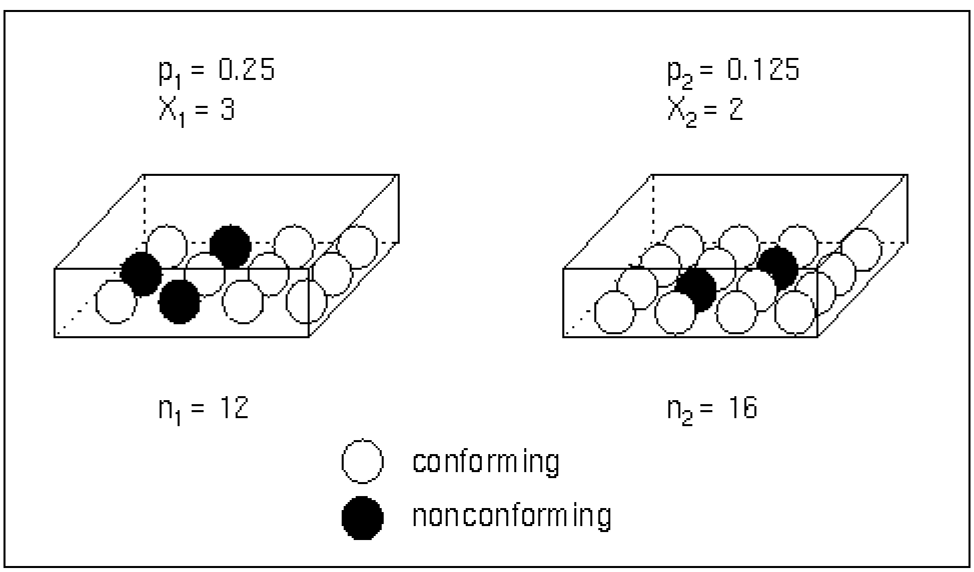
The following notation is used in this section:

p	expected proportion of nonconforming items produced by the process
p_i	proportion of nonconforming items in the ith subgroup
X_i	number of nonconforming items in the ith subgroup
n_i	number of items in the ith subgroup
$\bar{p}$	average proportion of nonconforming items taken across subgroups: $$\bar{p} = \frac{n_1 p_1 + \cdots + n_N p_N}{n_1 + \cdots + n_N} = \frac{X_1 + \cdots + X_N}{n_1 + \cdots + n_N}$$
N	number of subgroups
$I_T(\alpha, \beta)$	incomplete beta function: $$I_T(\alpha,\beta) = (\Gamma(\alpha+\beta)/\Gamma(\alpha)\Gamma(\beta)) \int_0^T t^{\alpha-1}(1-t)^{\beta-1}dt$$ for $0 < T < 1$, $\alpha > 0$, and $\beta > 0$, where $\Gamma(\cdot)$ is the gamma function

Plotted Points

Each point on an np chart represents the observed number (X_i) of nonconforming items in a subgroup. For example, suppose the first subgroup (see Figure 17.59) contains 12 items, of which three are nonconforming. The point plotted for the first subgroup is $X_1 = 3$.

Figure 17.59 Proportions Versus Counts



Note that a p chart displays the proportion of nonconforming items p_i. You can use the PCHART statement to create p charts; see "PCHART Statement: SHEWHART Procedure" on page 1639.

Central Line

By default, the central line on an np chart indicates an estimate for $n_i p$, which is computed as $n_i \bar{p}$. If you specify a known value (p_0) for p, the central line indicates the value of $n_i p_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of X_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that X_i exceeds the limits

The lower and upper control limits, LCL and UCL respectively, are computed as

$$\begin{aligned}\text{LCL} &= \max\left(n_i \bar{p} - k\sqrt{n_i \bar{p}(1-\bar{p})}\ , 0\right)\\ \text{UCL} &= \min\left(n_i \bar{p} + k\sqrt{n_i \bar{p}(1-\bar{p})}\ , n_i\right)\end{aligned}$$

A lower probability limit for X_i can be determined using the fact that

$$\begin{aligned}P\{X_i < \text{LCL}\} &= 1 - P\{X_i \geq \text{LCL}\}\\ &= 1 - I_{\bar{p}}(\text{LCL}, n_i + 1 - \text{LCL})\\ &= I_{1-\bar{p}}(n_i + 1 - \text{LCL}, \text{LCL})\end{aligned}$$

Refer to Johnson, Kotz, and Kemp (1992). This assumes that the process is in statistical control and that X_i is binomially distributed. The lower probability limit LCL is then calculated by setting

$$I_{1-\bar{p}}(n_i + 1 - \text{LCL}, \text{LCL}) = \alpha/2$$

and solving for LCL. Similarly, the upper probability limit for X_i can be determined using the fact that

$$\begin{aligned}P\{X_i > \text{UCL}\} &= P\{X_i > \text{UCL}\}\\ &= I_{\bar{p}}(\text{UCL} + 1, n_i - \text{UCL})\end{aligned}$$

The upper probability limit UCL is then calculated by setting

$$I_{\bar{p}}(\text{UCL} + 1, n_i - \text{UCL}) = \alpha/2$$

and solving for UCL. The probability limits are asymmetric about the central line. Note that both the control limits and probability limits vary with n_i.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify p_0 with the P0= option or with the variable _P_ in the LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 17.38 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLNP	lower control limit for number of nonconforming items
LIMITN	sample size associated with the control limits
_NP_	average number of nonconforming items ($n_i \bar{p}$ or $n_i p_0$)
_P_	average proportion of nonconforming items ($\bar{p}$ or p_0)
SIGMAS	multiple (k) of standard error of X_i
SUBGRP	*subgroup-variable* specified in the NPCHART statement
TYPE	type (standard or estimate) of _NP_
UCLNP	upper control limit for number of nonconforming items
VAR	*process* specified in the NPCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variables _LIMITN_, _LCLNP_, _UCLNP_, _NP_, and _SIGMAS_.
2. If the limits are defined in terms of a multiple k of the standard error of X_i, the value of _ALPHA_ is computed as $\alpha = P\{X_i < \text{\_LCLNP\_}\} + P\{X_i > \text{\_UCLNP\_}\}$, using the incomplete beta function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = (\text{\_UCLNP\_} - \text{\_NP\_})/\sqrt{\text{\_NP\_}(1 - \text{\_NP\_})/\text{\_LIMITN\_}}$. If _LIMITN_ has the special missing value V, this value is assigned to _SIGMAS_.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the NPCHART statement. For an example, see "Saving Control Limits" on page 1606.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- the subgroup proportion of nonconforming items variable named by the *process* suffixed with *P*
- a subgroup sample size variable named by the *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the NPCHART statement. For example, consider the following statements:

```
proc shewhart data=Input;
   npchart (Rework Rejected)*Batch / outhistory=Summary
                                     subgroupn =30;
run;
```

The data set Summary contains variables named Batch, ReworkP, ReworkN, RejectedP, and RejectedN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Proportions of Nonconforming Items" on page 1606.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on np chart
LCLNP	lower control limit for number of nonconforming items
LIMITN	nominal sample size associated with the control limits
_NP_	average number of nonconforming items
SIGMAS	multiple (k) of the standard error of X_i associated with the control limits
subgroup	values of the subgroup variable
SUBNP	subgroup number of nonconforming items
SUBN	subgroup sample size
TESTS	tests for special causes signaled on np chart
UCLNP	upper control limit for number of nonconforming items
VAR	*process* specified in the NPCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The kth character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request the first four tests (the tests appropriate for np charts) and Tests 2 and 4 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, a 4 for the fourth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see “Saving Control Limits” on page 1606.

ODS Tables

The following table summarizes the ODS tables that you can request with the NPCHART statement.

Table 17.39 ODS Tables Produced with the NPCHART Statement

Table Name	Description	Options
NPCHART	np chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*). **NOTE:** In SAS/QC 13.1 the SHEWHART procedure does not support the creation of graphs that are editable with the ODS Graphics Editor.

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. NPCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1946.

When ODS Graphics is in effect, the NPCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 17.40.

Table 17.40 ODS Graphics Produced by the NPCHART Statement

ODS Graph Name	Plot Description
NPChart	np chart

See Chapter 3, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (counts of nonconforming items) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the NPCHART statement must be a SAS variable in the DATA= data set. This variable provides counts for subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the NPCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a count for each *process* and a value for the *subgroup-variable*. The data set must contain one observation for each subgroup. Note that you can specify the DATAUNIT= option in the NPCHART statement to read proportions or percentages of nonconforming items instead of counts. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

When you use a DATA= data set with the NPCHART statement, the SUBGROUPN= option (which specifies the subgroup sample size) is required. By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2031.

For an example of a DATA= data set, see "Creating np Charts from Count Data" on page 1601.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   npchart Rejects*Batch  / subgroupn=100;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLNP_, _NP_, and _UCLNP_, which specify the control limits directly
- the variable _P_, which is used to calculate the control limits according to the equations in the section "Control Limits" on page 1622

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.

- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE' and 'STANDARD'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1609.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to create your own HISTORY= data set.

A HISTORY= data set used with the NPCHART statement must contain

- the *subgroup-variable*
- a subgroup proportion of nonconforming items variable for each *process*
- a subgroup sample size variable for each *process*

The names of the proportion sample size variables must be the *process* name concatenated with the special suffix characters *P* and *N*, respectively.

For example, consider the following statements:

```
proc shewhart history=Summary;
   npchart ( Rework Rejected)*Batch / subgroupn=50;
run;
```

The data set Summary must include the variables Batch, ReworkP, ReworkN, RejectedP, and RejectedN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2031 for an example).

For an example of a HISTORY= data set, see "Creating np Charts from Summary Data" on page 1603.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2096.

The following table lists the variables required in a TABLE= data set used with the NPCHART statement:

Table 17.41 Variables Required in a TABLE= Data Set

Variable	Description
LCLNP	lower control limit for number of nonconforming items
LIMITN	nominal sample size associated with the control limits
_NP_	average number of nonconforming items
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBNP	subgroup number of nonconforming items
UCLNP	upper control limit for number of nonconforming items

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1606.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup number nonconforming variable
Vertical	TABLE=	_SUBNP_

For an example, see "Labeling Axes" on page 2063.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: NPCHART Statement

This section provides advanced examples of the NPCHART statement.

Example 17.18: Applying Tests for Special Causes

NOTE: See *np Charts-Tests for Special Causes* in the SAS/QC Sample Library.

This example shows how you can apply tests for special causes to make np charts more sensitive to special causes of variation. The following statements create a SAS data set named Circuit3, which contains the number of failing circuits for 20 batches from the circuit manufacturing process introduced in the section "Creating np Charts from Count Data" on page 1601:

```
data Circuit3;
   input Batch Fail @@;
   datalines;
 1 12     2 21     3 16     4  9
 5  3     6  4     7  6     8  9
 9 11    10 13    11 12    12  7
13  2    14 14    15  9    16  8
17 14    18 10    19 11    20  9
;
```

The following statements create the np chart, apply several tests to the chart, and tabulate the results:

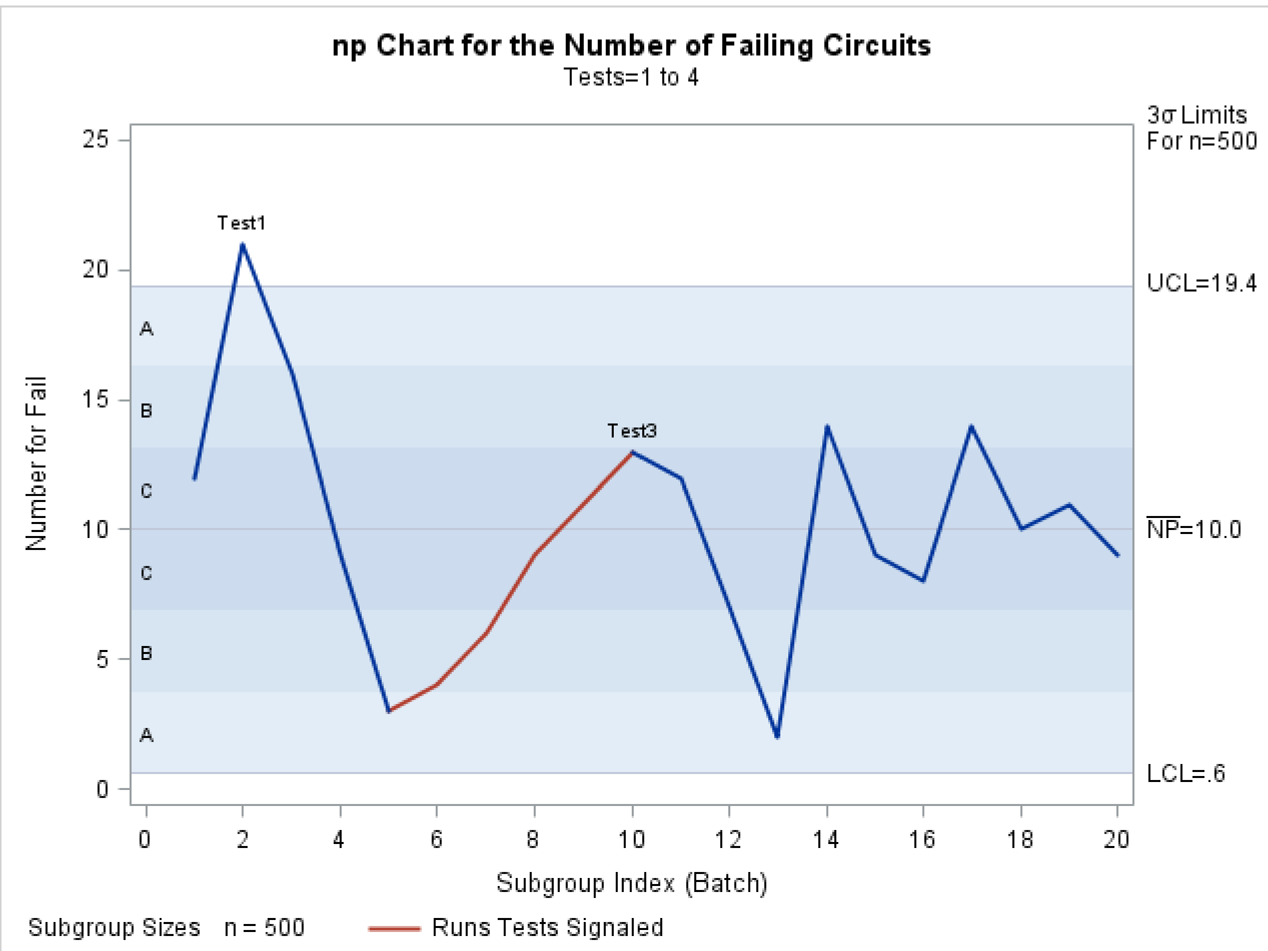
```
ods graphics on;
title1'np Chart for the Number of Failing Circuits';
title2 'Tests=1 to 4';
proc shewhart data=Circuit3;
   npchart Fail*Batch / subgroupn = 500
                        tests=1 to 4
                        table
```

```
                         tabletest
                         tablelegend
                         zones
                         zonelabels
                         odstitle  = title
                         odstitle2 = title2;
run;
```

The chart is shown in Output 17.18.1, and the printed output is shown in Output 17.18.2. The TESTS= option requests Tests 1, 2, 3, and 4, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2073. The TABLETESTS option requests a table of counts of nonconforming items and control limits, with a column indicating which subgroups tested positive for special causes. The TABLELEGEND option adds a legend describing the tests. The ZONELABELS option displays zone lines and zone labels on the chart. The zones are used to define the tests.

Output 17.18.1 and Output 17.18.2 indicate that Test 1 is positive at batch 2 and Test 3 is positive at batch 10.

Output 17.18.1 Tests for Special Causes Displayed on np Chart



Output 17.18.2 Tabular Form of np Chart

np Chart for the Number of Failing Circuits
Tests=1 to 4

The SHEWHART Procedure

np Chart Summary for Fail

Batch	Subgroup Sample Size	3 Sigma Limits with n=500 for Number: Lower Limit	Subgroup Number	Upper Limit	Special Tests Signaled
1	500	0.60851449	12.000000	19.391486	
2	500	0.60851449	21.000000	19.391486	1
3	500	0.60851449	16.000000	19.391486	
4	500	0.60851449	9.000000	19.391486	
5	500	0.60851449	3.000000	19.391486	
6	500	0.60851449	4.000000	19.391486	
7	500	0.60851449	6.000000	19.391486	
8	500	0.60851449	9.000000	19.391486	
9	500	0.60851449	11.000000	19.391486	
10	500	0.60851449	13.000000	19.391486	3
11	500	0.60851449	12.000000	19.391486	
12	500	0.60851449	7.000000	19.391486	
13	500	0.60851449	2.000000	19.391486	
14	500	0.60851449	14.000000	19.391486	
15	500	0.60851449	9.000000	19.391486	
16	500	0.60851449	8.000000	19.391486	
17	500	0.60851449	14.000000	19.391486	
18	500	0.60851449	10.000000	19.391486	
19	500	0.60851449	11.000000	19.391486	
20	500	0.60851449	9.000000	19.391486	

Test Descriptions	
Test 1	One point beyond Zone A (outside control limits)
Test 3	Six points in a row steadily increasing or decreasing

Example 17.19: Specifying Standard Average Proportion

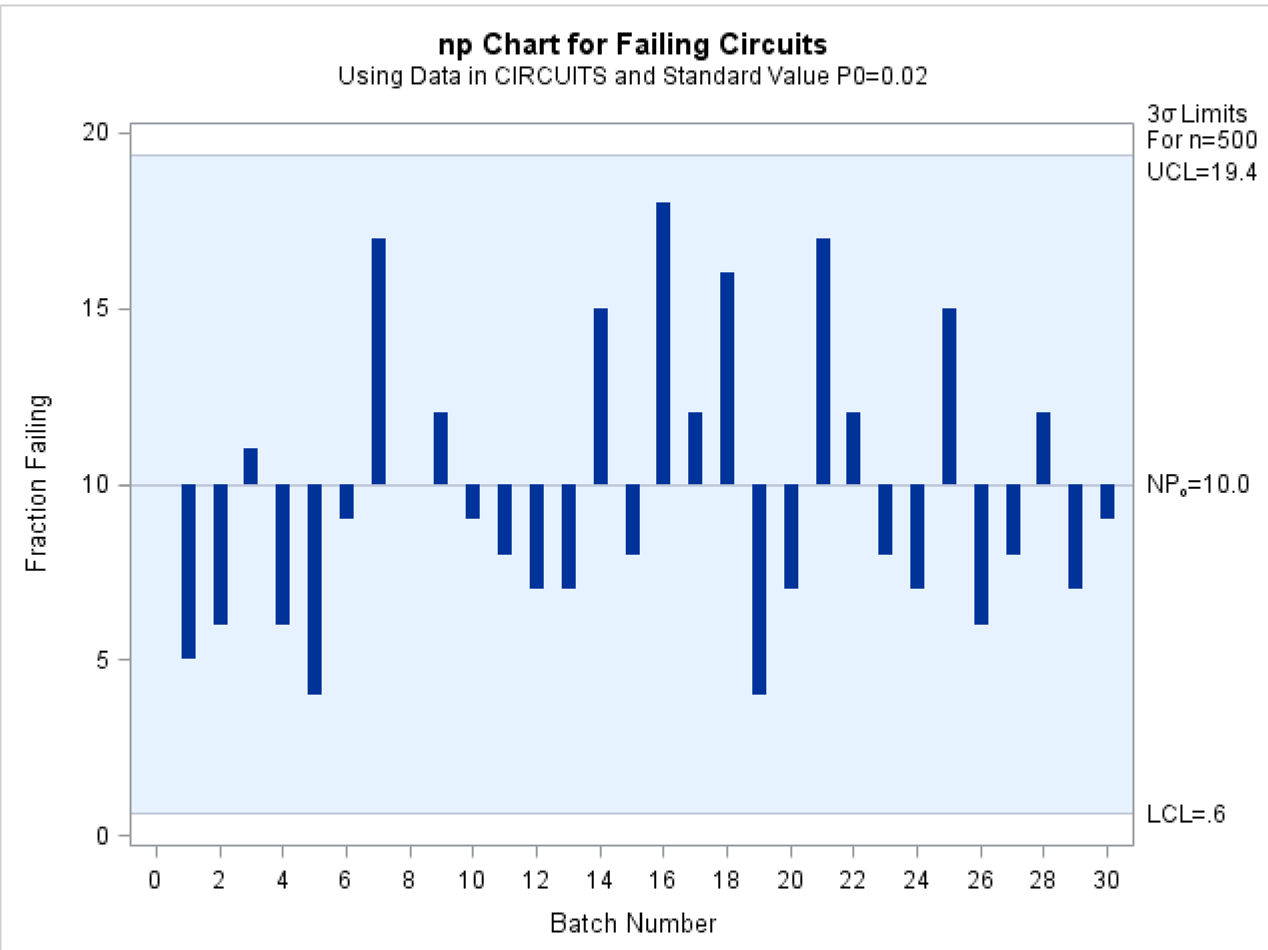
NOTE: See *Specifying a Known Proportion for np Charts* in the SAS/QC Sample Library.

In some situations, a standard (known) value (p_0) is available for the expected proportion of nonconforming items, based on extensive testing or previous sampling. This example illustrates how you can specify p_0 to create an np chart.

An np chart is used to monitor the number of failing circuits in the data set Circuits, which is introduced in "Creating np Charts from Count Data" on page 1601. The expected proportion of failing circuits is known to be $p_0 = 0.02$. The following statements create an np chart, shown in Output 17.19.1, using p_0 to compute the control limits:

```
ods graphics on;
title1 'np Chart for Failing Circuits';
title2 'Using Data in CIRCUITS and Standard Value P0=0.02';
proc shewhart data=Circuits;
   npchart Fail*Batch / subgroupn = 500
                        p0        = 0.02
                        npsymbol  = np0
                        nolegend
                        needles
                        odstitle  = title
                        odstitle2 = title2;
   label Batch ='Batch Number'
         Fail  ='Fraction Failing';
run;
```

Output 17.19.1 An np Chart with Standard Value of p_0



The chart indicates that the process is in control. The P0= option specifies p_0. The NPSYMBOL= option specifies a label for the central line indicating that the line represents a standard value. The NEEDLES option connects points to the central line with vertical needles. The NOLEGEND option suppresses the default legend for subgroup sample sizes. Labels for the vertical and horizontal axes are provided with the LABEL statement. For details concerning axis labeling, see "Axis Labels" on page 1629.

Alternatively, you can specify p_0 using the variable _P_ in a LIMITS= data set, as follows:

```
data Climits;
   length _var_ _subgrp_ _type_ $8;
   _p_       = 0.02;
   _subgrp_ = 'Batch';
   _var_     = 'Fail';
   _type_    = 'STANDARD';
   _limitn_ = 500;

proc shewhart data=Circuits limits=Climits;
   npchart Fail*Batch / subgroupn = 500
                         npsymbol  = np0
                         nolegend
                         needles;
   label Batch ='Batch Number'
         Fail  ='Fraction Failing';
run;
```

The bookkeeping variable _TYPE_ indicates that _P_ has a standard value. The chart produced by these statements is identical to the chart in Output 17.19.1.

Example 17.20: Working with Unequal Subgroup Sample Sizes

NOTE: See *np Charts with Unequal Subgroup Sample Sizes* in the SAS/QC Sample Library.

The following statements create a SAS data set named Battery, which contains the number of alkaline batteries per lot failing an acceptance test. The number of batteries tested in each lot varies but is approximately 150.

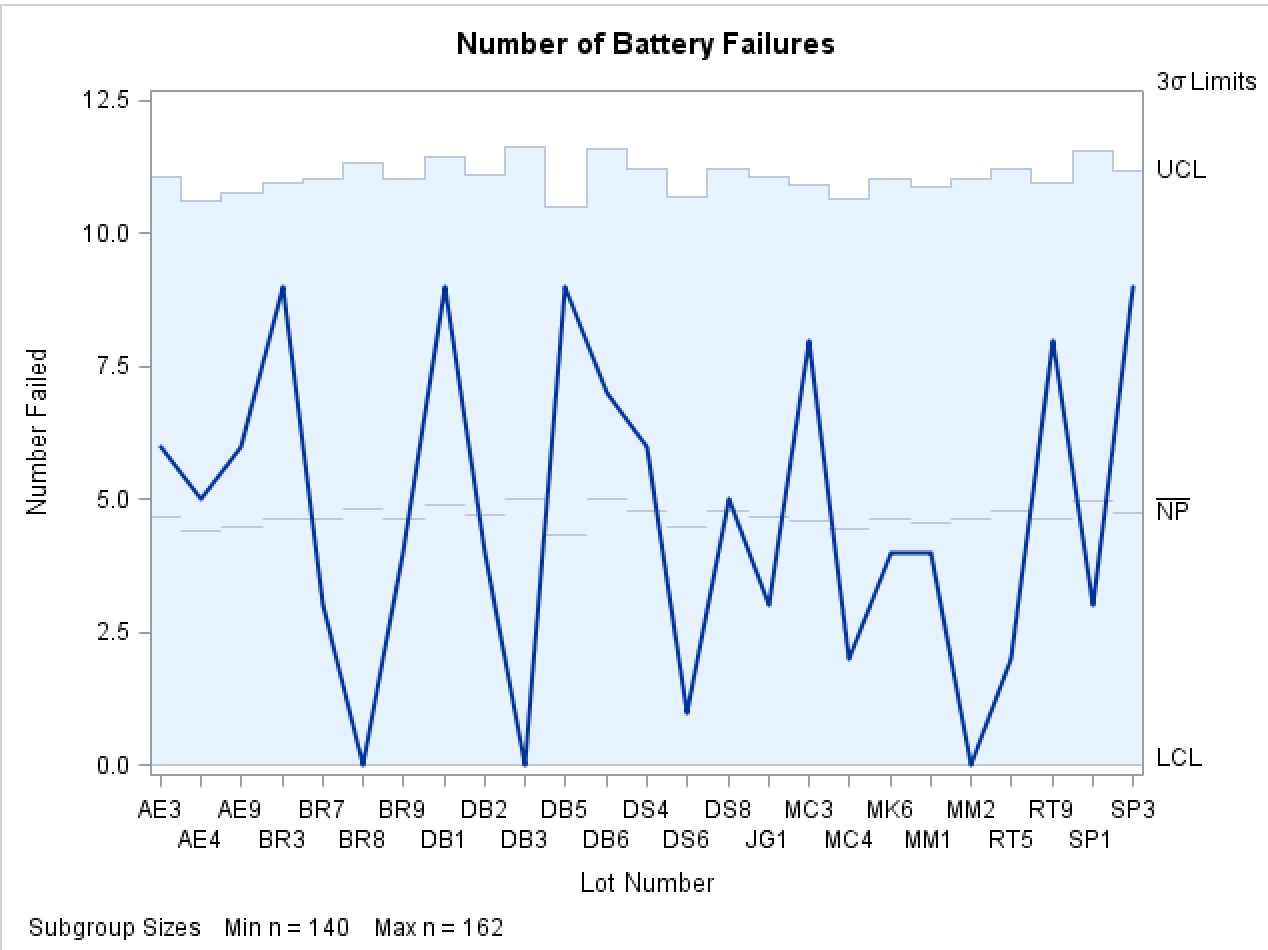
```
data Battery;
   length Lot $3;
   input Lot nFailed Sampsize @@;
   label nFailed ='Number Failed'
         Lot      = 'Lot Number'
         Sampsize='Number Sampled';
   datalines;
AE3   6   151      AE4   5   142      AE9   6   145
BR3   9   149      BR7   3   150      BR8   0   156
BR9   4   150      DB1   9   158      DB2   4   152
DB3   0   162      DB5   9   140      DB6   7   161
DS4   6   154      DS6   1   144      DS8   5   154
JG1   3   151      MC3   8   148      MC4   2   143
MK6   4   150      MM1   4   147      MM2   0   150
RT5   2   154      RT9   8   149      SP1   3   160
SP3   9   153
;
```

The variable nFailed contains the number of battery failures, the variable Lot contains the lot number, and the variable Sampsize contains the lot sample size. The following statements request an np chart for this data:

```
ods graphics on;
title 'Number of Battery Failures';
proc shewhart data=Battery;
   npchart nFailed*Lot / subgroupn = Sampsize
                         outlimits = Batlim
                         odstitle  = title;
   label nFailed='Number Failed';
run;
```

The chart is shown in Output 17.20.1, and the OUTLIMITS= data set Batlim is listed in Output 17.20.2.

Output 17.20.1 An np Chart with Varying Subgroup Sample Sizes



Note that the upper control limit and central line on the np chart vary with the subgroup sample size. The lower control limit is truncated at zero. The sample size legend indicates the minimum and maximum subgroup sample sizes.

Output 17.20.2 The Control Limits Data Set Batlim

Control Limits for Battery Failures

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_P_	_LCLNP_	_NP_	_UCLNP_
nFailed	Lot	ESTIMATE	V	V	3	0.031010	V	V	V

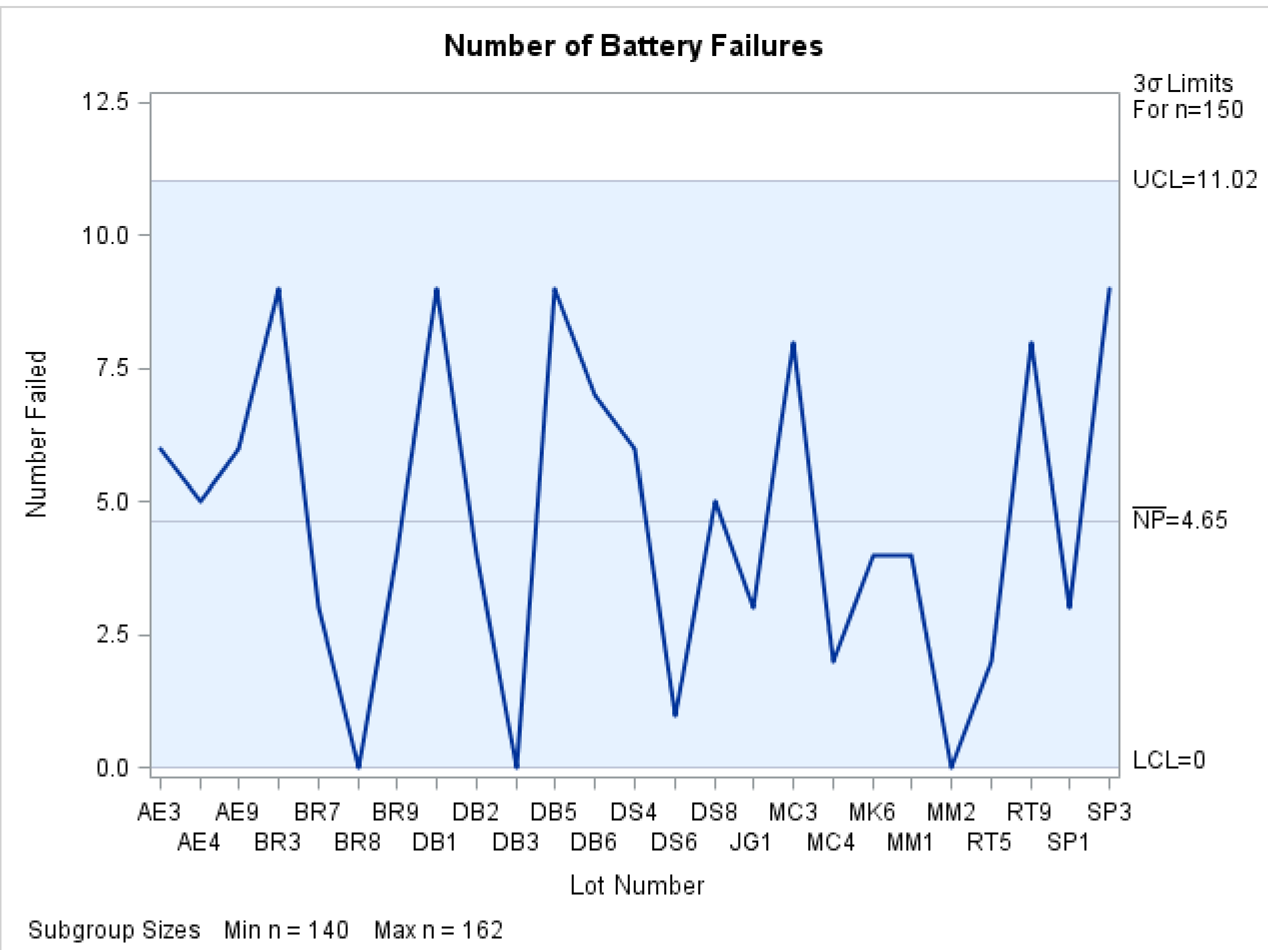
The variables in Batlim whose values vary with subgroup sample size are assigned the special missing value *V*.

The SHEWHART procedure provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for computing the control limits, as illustrated by the following statements:

```
title 'Number of Battery Failures';
proc shewhart data=Battery;
   npchart nFailed*Lot / subgroupn = Sampsize
                         limitn    = 150
                         odstitle  = title
                         alln;
   label nFailed='Number Failed';
run;
```

The ALLN option specifies that all points (regardless of subgroup sample size) are to be displayed. By default, only points for subgroups whose sample size matches the LIMITN= value are displayed. The chart is shown in Output 17.20.3.

Output 17.20.3 Control Limits Based on Fixed Subgroup Sample Size



All the points are inside the control limits, indicating that the process is in statistical control. Since there is relatively little variation in the sample sizes, the control limits in Output 17.20.3 provide a close approximation to the exact control limits in Output 17.20.1, and the same conclusions can be drawn from both charts. In general, you should be careful when interpreting charts that use a nominal sample size to compute control limits, since these limits are only approximate when the sample sizes vary.

Example 17.21: Specifying Control Limit Information

NOTE: See *np Charts-Specifying Control Limit Info* in the SAS/QC Sample Library.

This example shows how to use the DATA step to create LIMITS= data sets for use with the NPCHART statement. The variables _VAR_ and _SUBGRP_ are required. These variables must be character variables whose lengths are no greater than 32, and their values must match the *process* and *subgroup-variable* specified in the NPCHART statement. In addition, you must provide one of the following:

- the variables _LCLNP_, _NP_, and _UCLNP_
- the variable _P_

The following DATA step creates a data set named Climits1, which provides a complete set of control limits for an *np* chart:

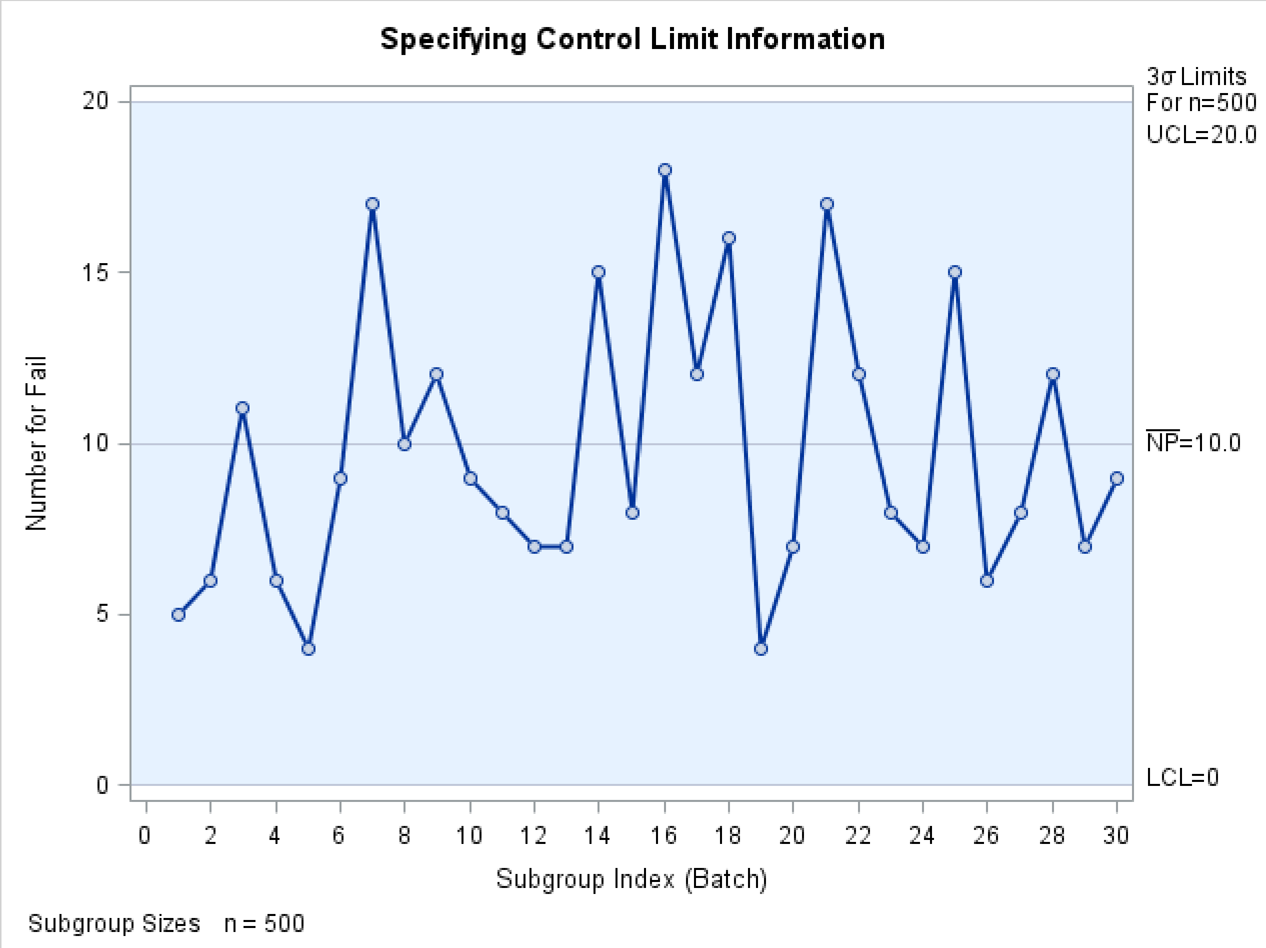
```
data Climits1;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'Fail';
   _subgrp_ = 'Batch';
   _limitn_ = 500;
   _type_   = 'STANDARD';
   _lclnp_  =  0;
   _np_     = 10;
   _uclnp_  = 20;
run;
```

The following statements read the control limits from the data set Climits1 and apply them to the count data in the data set Circuits, which is introduced in “Creating np Charts from Count Data” on page 1601:

```
ods graphics on;
title 'Specifying Control Limit Information';
proc shewhart data=Circuits limits=Climits1;
   npchart Fail*Batch / subgroupn = 500
                        odstitle  = title
                        markers;
run;
```

The chart is shown in Output 17.21.1.

Output 17.21.1 Control Limit Information Read from Climits1

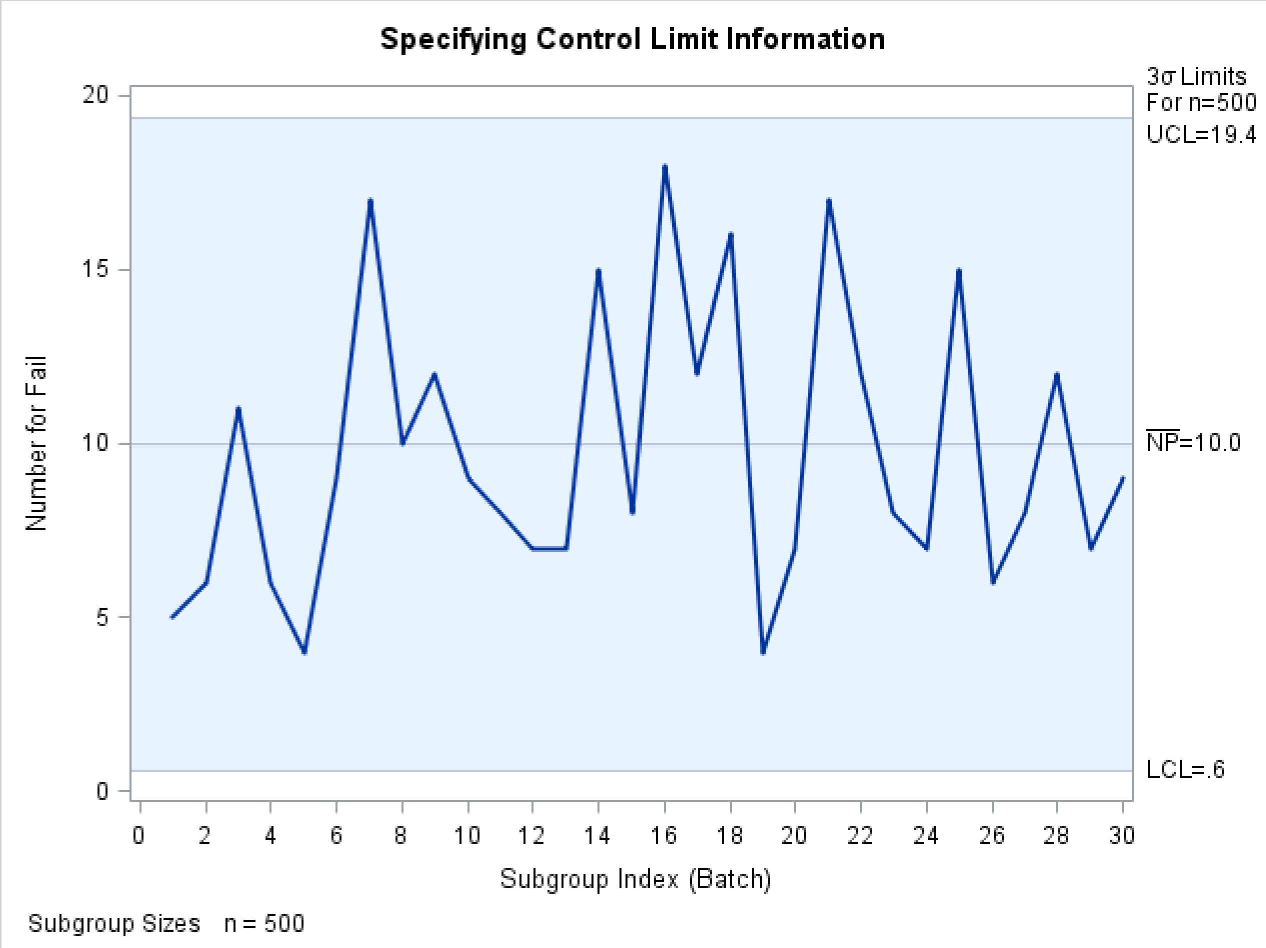


The following DATA step creates a data set named Climits2, which provides a value for the expected proportion of nonconforming items (_P_). This parameter is then used to compute the control limits for the data in Circuits according to the equations in "Control Limits" on page 1622.

```
data Climits2;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'Fail';
   _subgrp_ = 'Batch';
   _limitn_ = 500;
   _type_   = 'STANDARD';
   _p_      = .02;
run;

title 'Specifying Control Limit Information';
proc shewhart data=Circuits limits=Climits2;
   npchart Fail*Batch / subgroupn = 500
                        odstitle  = title;
run;
```

The chart is shown in Output 17.21.2. Note that the control limits are not the same as those shown in Output 17.21.1.

Output 17.21.2 Control Limit Information Read from Climits2



PCHART Statement: SHEWHART Procedure

Overview: PCHART Statement

The PCHART statement creates *p* charts for the proportions of nonconforming (defective) items in subgroup samples.

You can use options in the PCHART statement to

- compute control limits from the data based on a multiple of the standard error of the proportions or as probability limits
- tabulate subgroup sample sizes, proportions of nonconforming items, control limits, and other information
- save control limits in an output data set

- save subgroup sample sizes and proportions of nonconforming items in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) proportion of nonconforming items for computing control limits
- specify the data as counts, proportions, or percentages of nonconforming items
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing *p* charts with the PCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 3, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: PCHART Statement

This section introduces the PCHART statement with simple examples that illustrate commonly used options. Complete syntax for the PCHART statement is presented in the section "Syntax: PCHART Statement" on page 1650, and advanced examples are given in the section "Examples: PCHART Statement" on page 1670.

Creating p Charts from Count Data

NOTE: See *p Chart Examples* in the SAS/QC Sample Library.

An electronics company manufactures circuits in batches of 500 and uses a *p* chart to monitor the proportion of failing circuits. Thirty batches are examined, and the failures in each batch are counted. The following statements create a SAS data set named Circuits,[6] which contains the failure counts:

[6] This data set is also used in the "Getting Started" section of "NPCHART Statement: SHEWHART Procedure" on page 1600.

```
data Circuits;
   input Batch Fail @@;
   datalines;
1    5   2    6   3  11   4   6    5   4
6    9   7   17   8  10   9  12   10   9
11   8  12    7  13   7  14  15   15   8
16  18  17   12  18  16  19   4   20   7
21  17  22   12  23   8  24   7   25  15
26   6  27    8  28  12  29   7   30   9
;
```

A partial listing of Circuits is shown in Figure 17.60.

Figure 17.60 The Data Set Circuits

Number of Failing Circuits

Batch	Fail
1	5
2	6
3	11
4	6
5	4

There is a single observation for each batch. The variable Batch identifies the subgroup sample and is referred to as the *subgroup-variable*. The variable Fail contains the number of nonconforming items in each subgroup sample and is referred to as the *process variable* (or *process* for short).

The following statements create the *p* chart shown in Figure 17.61:

```
ods graphics off;
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart data=Circuits;
   pchart Fail*Batch / subgroupn = 500;
run;
```

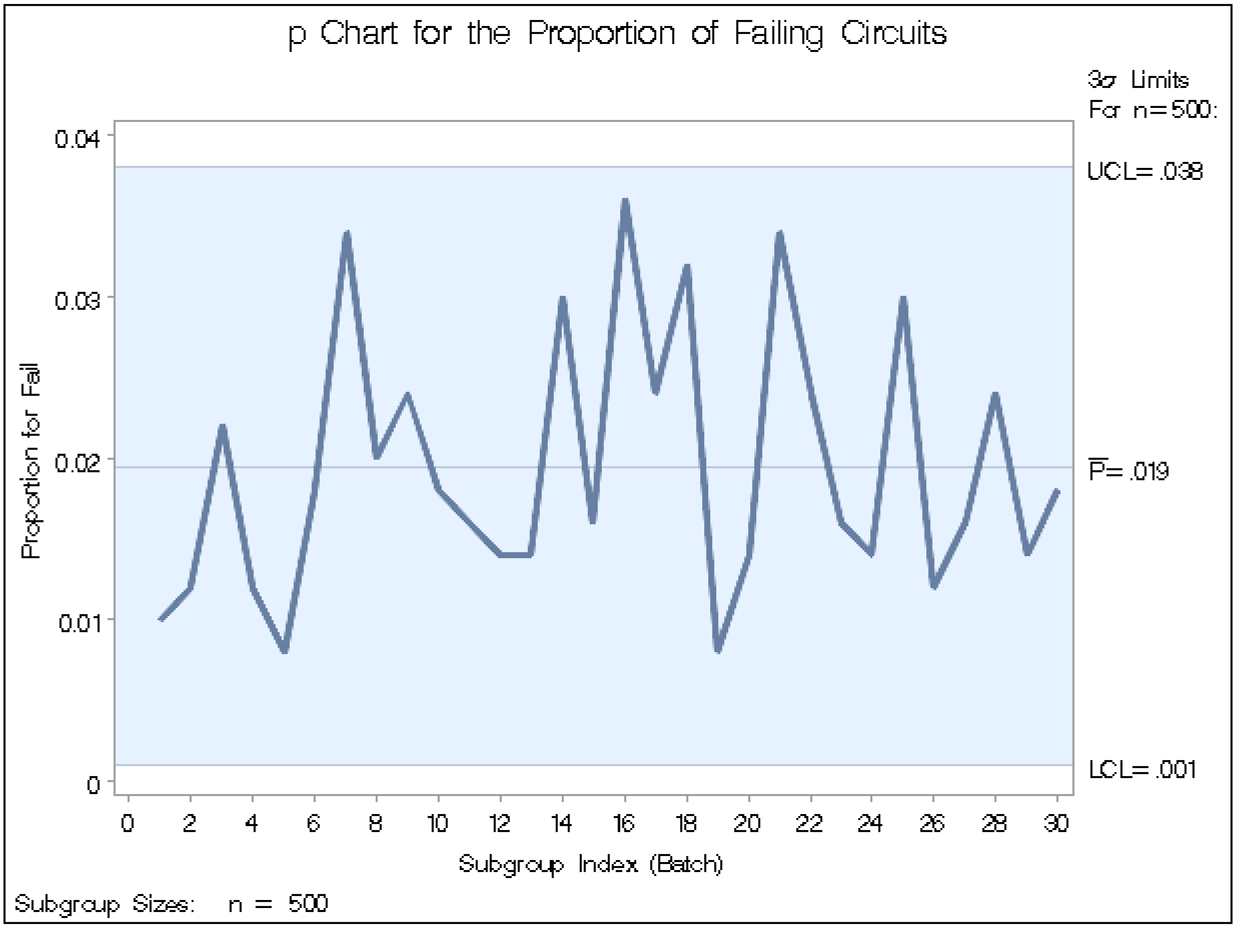
This example illustrates the basic form of the PCHART statement. After the keyword PCHART, you specify the *process* to analyze (in this case, Fail), followed by an asterisk and the *subgroup-variable* (Batch).

The input data set is specified with the DATA= option in the PROC SHEWHART statement. The SUBGROUPN= option specifies the number of items in each subgroup sample and is required with a DATA= input data set. The SUBGROUPN= option specifies one of the following:

- a constant subgroup sample size (as in this case)
- a variable in the input data set whose values provide the subgroup sample sizes (see the next example)

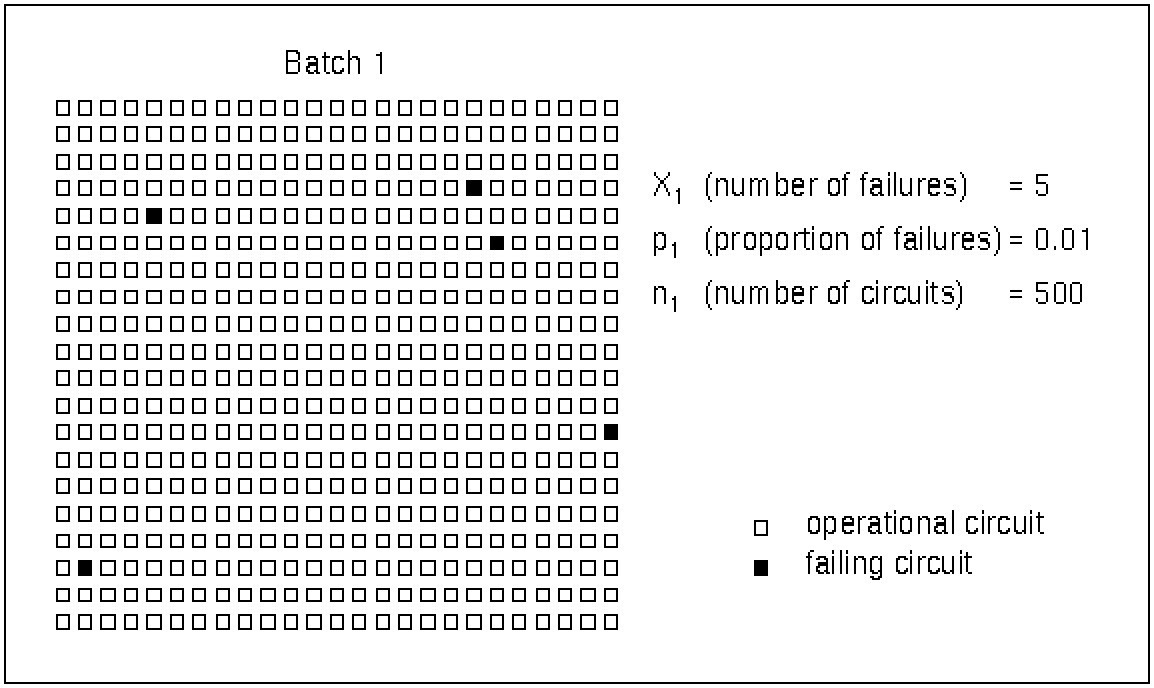
Options such as SUBGROUPN= are specified after the slash (/) in the PCHART statement. A complete list of options is presented in the section “Syntax: PCHART Statement” on page 1650.

Figure 17.61 *p* Chart for Circuit Failures (Traditional Graphics)



Each point on the *p* chart represents the proportion of nonconforming items for a particular subgroup. For instance, the value plotted for the first batch is $5/500 = 0.01$, as illustrated in Figure 17.62.

Figure 17.62 Proportions Versus Counts



Since all the points fall within the control limits, it can be concluded that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in "Control Limits" on page 1662. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1649. For computational details, see "Constructing Charts for Proportion Nonconforming (p Charts)" on page 1661. For more details on reading counts of nonconforming items, see "DATA= Data Set" on page 1666.

Creating p Charts from Summary Data

NOTE: See *p Chart Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create *p* charts using raw data (counts of nonconforming items). However, in many applications, the data are provided in summarized form as proportions or percentages of nonconforming items. This example illustrates how you can use the PCHART statement with data of this type.

The following data set provides the data from the preceding example in summarized form:

```
data Cirprop;
   input Batch pFailed @@;
   Sampsize=500;
   datalines;
 1  0.010   2  0.012   3  0.022   4  0.012   5  0.008
 6  0.018   7  0.034   8  0.020   9  0.024  10  0.018
11  0.016  12  0.014  13  0.014  14  0.030  15  0.016
16  0.036  17  0.024  18  0.032  19  0.008  20  0.014
21  0.034  22  0.024  23  0.016  24  0.014  25  0.030
26  0.012  27  0.016  28  0.024  29  0.014  30  0.018
;
```

A partial listing of Cirprop is shown in Figure 17.63. The subgroups are still indexed by Batch. The variable pFailed contains the proportions of nonconforming items, and the variable Sampsize contains the subgroup sample sizes.

Figure 17.63 The Data Set Cirprop

Number of Failing Circuits

Batch	Fail
1	5
2	6
3	11
4	6
5	4

The following statements create a *p* chart identical to the one in Figure 17.61:

```
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart data=Cirprop;
   pchart pFailed*Batch / subgroupn=Sampsize
                          dataunit =proportion;
label pfailed = 'Proportion for Fail';
run;
```

The DATAUNIT= option specifies that the values of the *process* (pFailed) are proportions of nonconforming items. By default, the values of the *process* are assumed to be counts of nonconforming items (see the previous example).

Alternatively, you can read the data set Cirprop by specifying it as a HISTORY= data set in the PROC SHEWHART statement. A HISTORY= data set used with the PCHART statement must contain the following variables:

- subgroup variable
- subgroup proportion of nonconforming items variable
- subgroup sample size variable

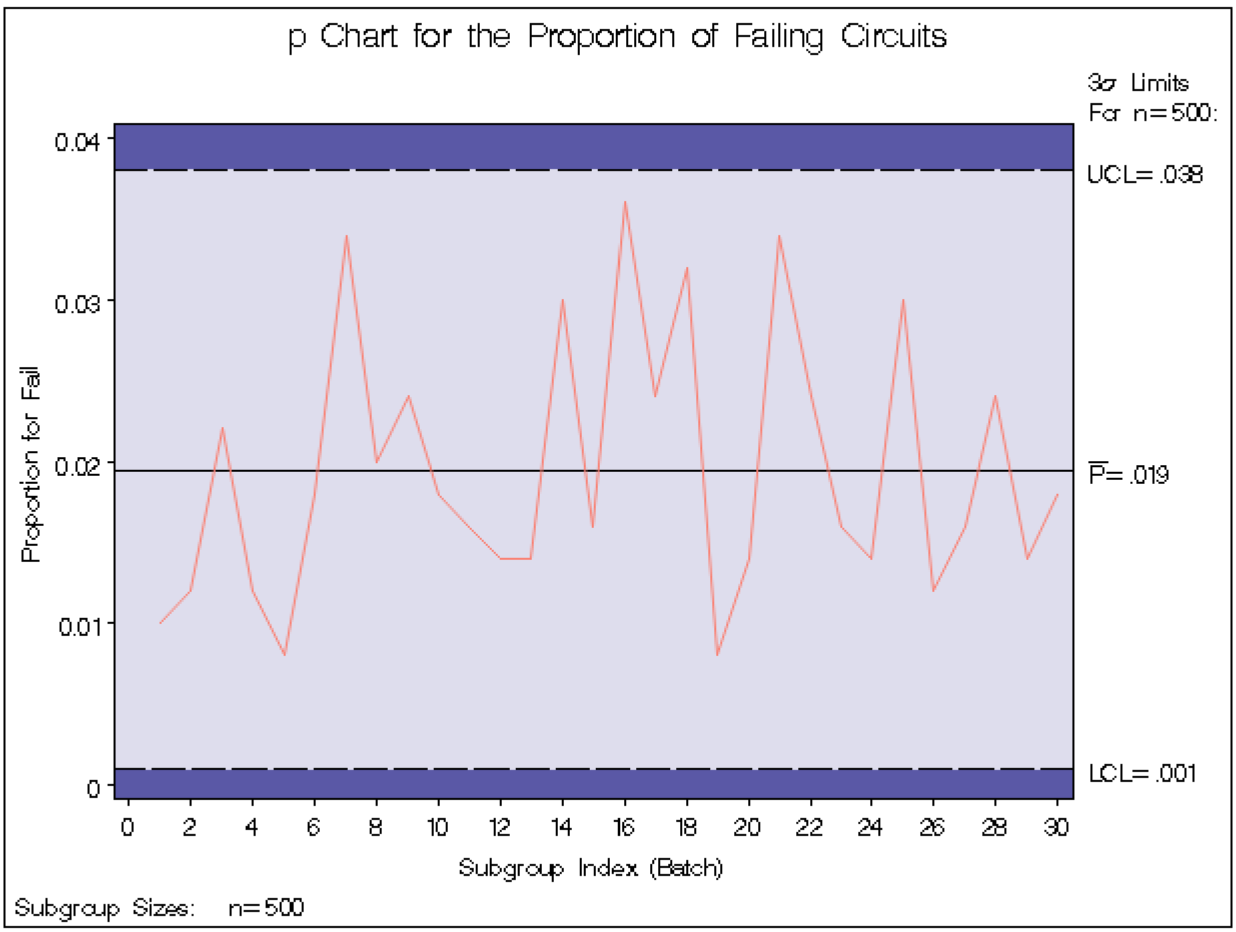
Furthermore, the names of the subgroup proportion and sample size variables must begin with the *process* name specified in the PCHART statement and end with the special suffix characters *P* and *N*, respectively.

To specify Cirprop as a HISTORY= data set and Fail as the *process*, you must rename the variables pFailed and Sampsize to FailP and FailN, respectively. The following statements temporarily rename pFailed and Sampsize for the duration of the procedure step:

```
options nogstyle;
goptions ftext=swiss;
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart history=Cirprop(rename=(pFailed=FailP
                              Sampsize=FailN ));
   pchart Fail*Batch / cframe    = lib
                       cinfill   = bwh
                       coutfill  = yellow
                       cconnect  = salmon;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the PCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting *p* chart is shown in Figure 17.64.

Figure 17.64 *p* Chart from Subgroup Proportions (Traditional Graphics with NOGSTYLE)



In this example, it is more convenient to use Cirprop as a DATA= data set than as a HISTORY= data set. In general, it is more convenient to use the HISTORY= option for input data sets that have been previously created by the SHEWHART procedure as OUTHISTORY= data sets, as illustrated in the next example. For more information, see "HISTORY= Data Set" on page 1668.

Saving Proportions of Nonconforming Items

NOTE: See *p Chart Examples* in the SAS/QC Sample Library.

In this example, the PCHART statement is used to create a data set that can later be read by the SHEWHART procedure (as in the preceding example). The following statements read the number of nonconforming items from the data set Circuits (see "Creating p Charts from Count Data" on page 1640) and create a summary data set named Cirhist:

```
proc shewhart data=Circuits;
   pchart Fail*Batch / subgroupn  = 500
                       outhistory = Cirhist
                       nochart ;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 17.61. Figure 17.65 contains a partial listing of Cirhist.

Figure 17.65 The Data Set Cirhist

Subgroup Proportions and Control Limit Information

Batch	FailP	FailN
1	0.010	500
2	0.012	500
3	0.022	500
4	0.012	500
5	0.008	500

There are three variables in the data set Cirhist.

- Batch contains the subgroup index.
- FailP contains the subgroup proportion of nonconforming items.
- FailN contains the subgroup sample size.

Note that the variables containing the subgroup proportions of nonconforming items and subgroup sample sizes are named by adding the suffix characters *P* and *N* to the *process* Fail specified in the PCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets. For more information, see "OUTHISTORY= Data Set" on page 1664.

Saving Control Limits

NOTE: See *p Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for a *p* chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1649) or modify the limits with a DATA step program.

The following statements read the number of nonconforming items per subgroup from the data set Circuits (see "Creating p Charts from Count Data" on page 1640) and save the control limits displayed in Figure 17.61 in a data set named Cirlim:

```
proc shewhart data=Circuits;
   pchart Fail*Batch / subgroupn = 500
                       outlimits = Cirlim
                       nochart ;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Cirlim is listed in Figure 17.66.

Figure 17.66 The Data Set Cirlim Containing Control Limit Information

Control Limits for the Proportion of Failing Circuits

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLP_	_P_	_UCLP_
Fail	Batch	ESTIMATE	500	.002320877	3	.000930786	0.019467	0.038003

The data set Cirlim contains one observation with the limits for *process* Fail. The variables _LCLP_ and _UCLP_ contain the lower and upper control limits, and the variable _P_ contains the central line. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _P_ is an estimate or standard value.

For more information, see "OUTLIMITS= Data Set" on page 1663.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Circuits;
   pchart Fail*Batch / subgroupn = 500
                       outtable  = Cirtable
                       nochart ;
run;
```

The data set Cirtable is listed in Figure 17.67.

Figure 17.67 The Data Set Cirtable

Subgroup Proportions and Control Limit Information

VAR	Batch	_SIGMAS_	_LIMITN_	_SUBN_	_LCLP_	_SUBP_	_P_	_UCLP_	_EXLIM_
Fail	1	3	500	500	.000930786	0.010	0.019467	0.038003	
Fail	2	3	500	500	.000930786	0.012	0.019467	0.038003	
Fail	3	3	500	500	.000930786	0.022	0.019467	0.038003	
Fail	4	3	500	500	.000930786	0.012	0.019467	0.038003	
Fail	5	3	500	500	.000930786	0.008	0.019467	0.038003	
Fail	6	3	500	500	.000930786	0.018	0.019467	0.038003	
Fail	7	3	500	500	.000930786	0.034	0.019467	0.038003	
Fail	8	3	500	500	.000930786	0.020	0.019467	0.038003	
Fail	9	3	500	500	.000930786	0.024	0.019467	0.038003	
Fail	10	3	500	500	.000930786	0.018	0.019467	0.038003	
Fail	11	3	500	500	.000930786	0.016	0.019467	0.038003	
Fail	12	3	500	500	.000930786	0.014	0.019467	0.038003	
Fail	13	3	500	500	.000930786	0.014	0.019467	0.038003	
Fail	14	3	500	500	.000930786	0.030	0.019467	0.038003	
Fail	15	3	500	500	.000930786	0.016	0.019467	0.038003	
Fail	16	3	500	500	.000930786	0.036	0.019467	0.038003	
Fail	17	3	500	500	.000930786	0.024	0.019467	0.038003	
Fail	18	3	500	500	.000930786	0.032	0.019467	0.038003	
Fail	19	3	500	500	.000930786	0.008	0.019467	0.038003	
Fail	20	3	500	500	.000930786	0.014	0.019467	0.038003	
Fail	21	3	500	500	.000930786	0.034	0.019467	0.038003	
Fail	22	3	500	500	.000930786	0.024	0.019467	0.038003	
Fail	23	3	500	500	.000930786	0.016	0.019467	0.038003	
Fail	24	3	500	500	.000930786	0.014	0.019467	0.038003	
Fail	25	3	500	500	.000930786	0.030	0.019467	0.038003	
Fail	26	3	500	500	.000930786	0.012	0.019467	0.038003	
Fail	27	3	500	500	.000930786	0.016	0.019467	0.038003	
Fail	28	3	500	500	.000930786	0.024	0.019467	0.038003	
Fail	29	3	500	500	.000930786	0.014	0.019467	0.038003	
Fail	30	3	500	500	.000930786	0.018	0.019467	0.038003	

This data set contains one observation for each subgroup sample. The variables _SUBP_ and _SUBN_ contain the subgroup proportions of nonconforming items and subgroup sample sizes. The variables _LCLP_ and _UCLP_ contain the lower and upper control limits, and the variable _P_ contains the central line. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1664.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read the information in Cirtable and display a *p* chart (not shown here) identical to the chart in Figure 17.61:

```
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart table=Cirtable;
   pchart Fail*Batch;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2096). For more information, see "TABLE= Data Set" on page 1669.

Reading Preestablished Control Limits

NOTE: See *p Chart Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Cirlim saved control limits computed from the data in Circuits. This example shows how these limits can be applied to new data provided in the following data set:

```
data Circuit2;
   input Batch Fail @@;
   datalines;
31  12   32   9   33  16   34   9
35   3   36   8   37  20   38   4
39   8   40   6   41  12   42  16
43   9   44   2   45  10   46   8
47  14   48  10   49  11   50   9
;
```

The following statements create a *p* chart for the data in Circuit2 using the control limits in Cirlim:

```
ods graphics on;
title 'p Chart for the Proportion of Failing Circuits';
proc shewhart data=Circuit2 limits=Cirlim;
   pchart Fail*Batch / subgroupn = 500
                       odstitle  = title;
run;
```

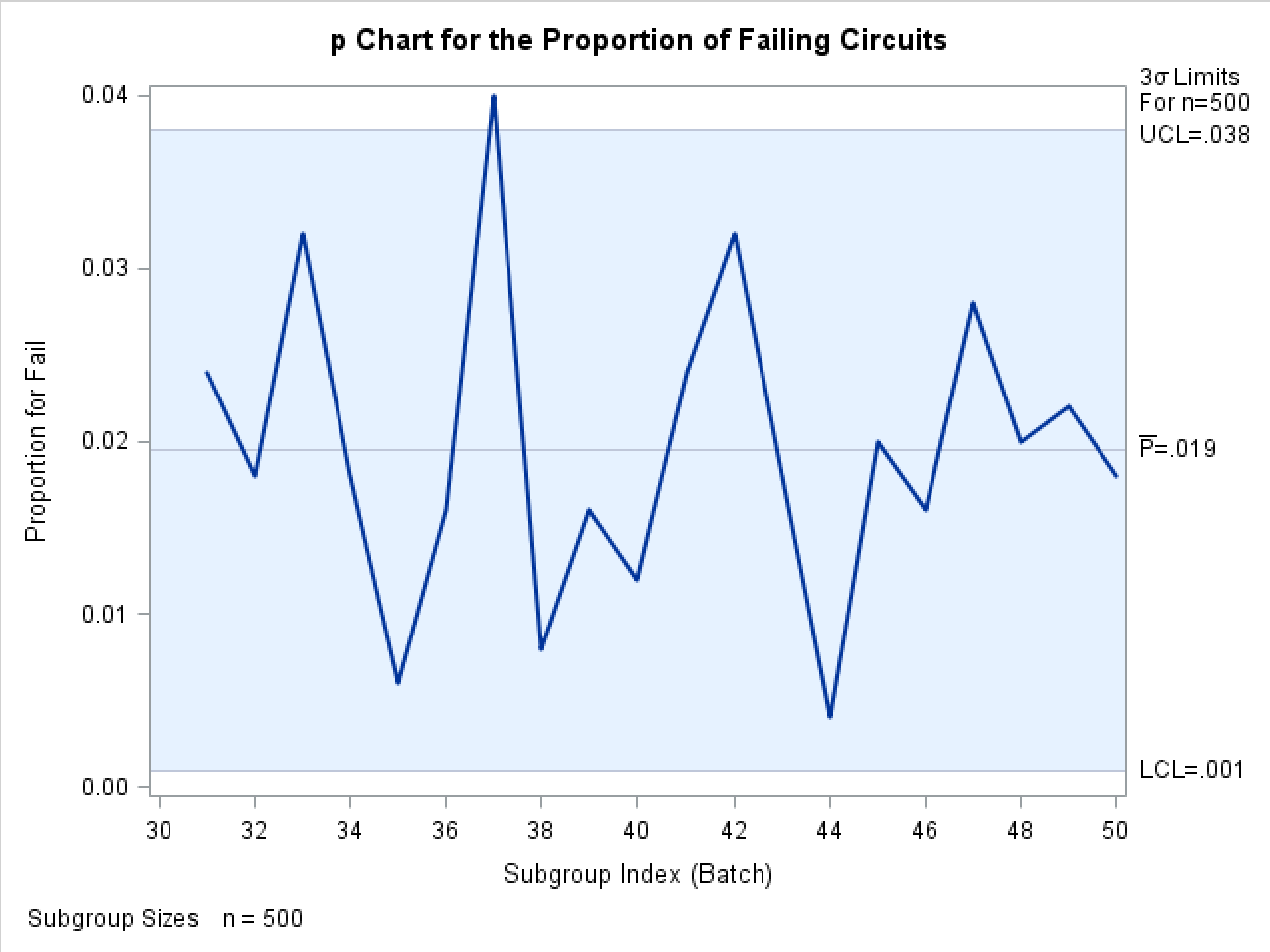
The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the *p* chart is created by using ODS Graphics instead of traditional graphics.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Fail
- the value of _SUBGRP_ matches the *subgroup-variable* name Batch

The resulting *p* chart is shown in Figure 17.68.

Figure 17.68 p Chart for Second Set of Circuit Failures (ODS Graphics)



The proportion of nonconforming items in the 37th batch exceeds the upper control limit, signaling that the process is out of control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1667 for details concerning the variables that you must provide.

Syntax: PCHART Statement

The basic syntax for the PCHART statement is as follows:

PCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

PCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=**'*character*' > / < *options* > **;**

You can use any number of PCHART statements in the SHEWHART procedure. The components of the PCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If numbers of nonconforming items are read from a DATA= data set, *process* must be the name of the variable containing the numbers. For an example, see "Creating p Charts from Summary Data" on page 1643.
- If proportions of nonconforming items are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating p Charts from Summary Data" on page 1643.
- If proportions of nonconforming items and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1647.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct *p* charts for Rejects and Reworks:

```
proc shewhart data=Measures;
   pchart (Rejects Reworks)*Sample / subgroupn=100;
run;
```

Note that when data are read from a DATA= data set, the SUBGROUPN= option, which specifies subgroup sample sizes, is required.

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding PCHART statement, Sample is the subgroup variable. For details, see "Subgroup Variables" on page 1871.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2026 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot proportions of nonconforming items.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2025 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create a *p* chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   pchart Rejects*Day='*' / subgroupn=100;
run;
```

options

enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options", which follows, lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1946 describes each option in detail.

Summary of Options

The following tables list the PCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1946.

Table 17.42 PCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	requests probability limits for chart
LIMITN=	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
PROBLIMITS=	requests probability limits at discrete values
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
ACTUALALPHA	displays the actual probability of a point being outside the control limits in the control limits legend
CINFILL=	specifies color for area inside control limits
CLIMITS=	specifies color of control limits, central line, and related labels
LCLLABEL=	specifies label for lower control limit
LIMLABSUBCHAR=	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit

Table 17.42 *continued*

Option	Description
LLIMITS=	specifies line type for control limits
NDECIMAL=	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMIT0	suppresses display of lower control limit if it is 0
NOLIMIT1	suppresses display of upper control limit if it is 1 (100%)
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit
PSYMBOL=	specifies label for central line
UCLLABEL=	specifies label for upper control limit
WLIMITS=	specifies width for control limits and central line
Standard Value Options	
P0=	specifies known (standard) value p_0 for proportion of nonconforming items p
TYPE=	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	labels every point on p chart
CLABEL=	specifies color for labels
CCONNECT=	specifies color for line segments that connect points on chart
CFRAMELAB=	specifies fill color for frame around labeled points
CNEEDLES=	specifies color for needles that connect points to central line
COUT=	specifies color for portions of line segments that connect points outside control limits
COUTFILL=	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	specifies angle at which labels are drawn
LABELFONT=	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=	labels points outside control limits
SYMBOLLEGEND=	specifies LEGEND statement for levels of *symbol-variable*

Table 17.42 *continued*

Option	Description
SYMBOLORDER=	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=	specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	computes zone widths independently above and below center line
NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	suppresses tests across *phase* boundaries
TESTS=	specifies tests for special causes
TEST2RUN=	specifies length of pattern for Test 2
TEST3RUN=	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL=	provides labels for points where test is positive
TESTLABEL*n*=	specifies label for *n*th test for special causes
TESTNMETHOD=	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=	enables tests for special causes to be reset
WESTGARD=	requests that Westgard rules be applied
ZONELABELS	adds labels A, B, and C to zone lines
ZONES	adds lines delineating zones A, B, and C
ZONEVALPOS=	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	specifies color for labels indicating points where test is positive
CTESTSYMBOL=	specifies color for symbol used to plot points where test is positive
CZONES=	specifies color for lines and labels delineating zones A, B, and C
LTESTS=	specifies type of line connecting points where test is positive
LZONES=	specifies line type for lines delineating zones A, B, and C
TESTFONT=	specifies software font for labels at points where test is positive
TESTHEIGHT=	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	specifies plot symbol for points where test is positive
TESTSYMBOLHT=	specifies symbol height for points where test is positive

Table 17.42 *continued*

Option	Description
WTESTS=	specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill colors for frame for plot area
CTEXT=	specifies color for tick mark values and axis labels
DISCRETE	produces horizontal axis for discrete numeric group values
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis
VFORMAT=	specifies format for vertical axis tick mark labels
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=	specifies width of axis lines
YSCALE=	scales vertical axis in percent units (rather than proportions)
Plot Layout Options	
ALLN	plots means for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart

Table 17.42 *continued*

Option	Description
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	specifies number of pages or screens to be used to display chart
ZEROSTD	displays p chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= options
CVREF=	specifies color for lines requested by VREF= options
HREF=	specifies position of reference lines perpendicular to horizontal axis
HREFDATA=	specifies position of reference lines perpendicular to horizontal axis
HREFLABELS=	specifies labels for HREF= lines
HREFLABPOS=	specifies position of HREFLABELS= labels
LHREF=	specifies line type for HREF= lines
LVREF=	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	specifies position of reference lines perpendicular to vertical axis
VREFLABELS=	specifies labels for VREF= lines
VREFLABPOS=	position of VREFLABELS= labels
Grid Options	
CGRID=	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=	specifies line type for grid requested with the ENDGRID option
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme points are clipped
CLIPLEGEND=	specifies text for clipping legend

Table 17.42 *continued*

Option	Description
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features chart
DESCRIPTION=	specifies description of *p* chart's GRSEG catalog entry
FONT=	specifies software font for labels and legends on charts
NAME=	specifies name of *p* chart's GRSEG catalog entry
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	shades areas between control limits and connected points lying outside the limits
STARFILL=	specifies a variable identfying groups of stars filled with different colors
STARS=	specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	specifies the control limit infill transparency
MARKERS	plots subgroup points with markers
NOBLOCKREF	suppresses block and phase reference lines
NOBLOCKREFFILL	suppresses block and phase wall fills
NOFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOPHASEREF	suppresses block and phase reference lines
NOPHASEREFFILL	suppresses block and phase wall fills
NOREF	suppresses block and phase reference lines
NOREFFILL	suppresses block and phase wall fills
NOSTARFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	specifies a graph footnote
ODSFOOTNOTE2=	specifies a secondary graph footnote

Table 17.42 *continued*

Option	Description
ODSLEGENDEXPAND	specifies that legend entries contain all levels observed in the data
ODSTITLE=	specifies a graph title
ODSTITLE2=	specifies a secondary graph title
OUTFILLTRANSPARENCY=	specifies control limit outfill transparency
OVERLAYURL=	specifies URLs to associate with overlay points
PHASEPOS=	specifies vertical position of phase legend
PHASEREFLEVEL=	associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	draws central line labels using a simulated software font
STARTRANSPARENCY=	specifies star fill transparency
URL=	specifies a variable whose values are URLs to be associated with subgroups
Input Data Set Options	
DATAUNIT	specifies that input values are proportions or percentages (rather than counts) of nonconforming items
MISSBREAK	specifies that observations with missing values are not to be processed
SUBGROUPN	specifies subgroup sample sizes as constant number *n* or as values of variable in a DATA= data set
Output Data Set Options	
OUTHISTORY=	creates output data set containing subgroup summary statistics
OUTINDEX=	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	creates output data set containing control limits
OUTTABLE=	creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Table 17.42 *continued*

Option	Description
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for *block-variable* legend
BLOCKLABTYPE=	specifies text size of *block-variable* legend
BLOCKPOS=	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE=	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	specifies color for STARCIRCLES= circles
CSTARFILL=	specifies color for filling stars
CSTAROUT=	specifies outline color for stars exceeding inner or outer circles
CSTARS=	specifies color for outlines of stars
LSTARCIRCLES=	specifies line types for STARCIRCLES= circles
LSTARS=	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	specifies reference circles for stars
STARINRADIUS=	specifies inner radius of stars
STARLABEL=	specifies vertices to be labeled
STARLEGEND=	specifies style of legend for star vertices
STARLEGENDLAB=	specifies label for STARLEGEND= legend
STAROUTRADIUS=	specifies outer radius of stars
STARSPECS=	specifies method used to standardize vertex variables
STARSTART=	specifies angle for first vertex
STARTYPE=	specifies graphical style of star
STARVERTICES=	superimposes star at each point on chart
WSTARCIRCLES=	specifies width of STARCIRCLES= circles

Table 17.42 *continued*

Option	Description
WSTARS=	specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	specifies colors for overlay line segments
COVERLAY=	specifies colors for overlay plots
COVERLAYCLIP=	specifies color for clipped points on overlays
LOVERLAY=	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=	specifies variables to overlay on chart
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
OVERLAYHTML=	specifies links to associate with overlay points
OVERLAYID=	specifies labels for overlay points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols for overlays
OVERLAYSYMHT=	specifies symbol heights for overlays
WOVERLAY=	specifies widths of overlay line segments
Options for Interactive Control Charts	
HTML=	specifies a variable whose values create links to be associated with subgroups
HTML_LEGEND=	specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	specifies plot character for clipped points
CONNECTCHAR=	specifies character used to form line segments that connect points on chart
HREFCHAR=	specifies line character for HREF= lines
SYMBOLCHARS=	specifies characters indicating *symbol-variable*
TESTCHAR=	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	specifies line character for VREF= lines
ZONECHAR=	specifies character for lines that delineate zones for tests for special causes

Details: PCHART Statement

Constructing Charts for Proportion Nonconforming (p Charts)

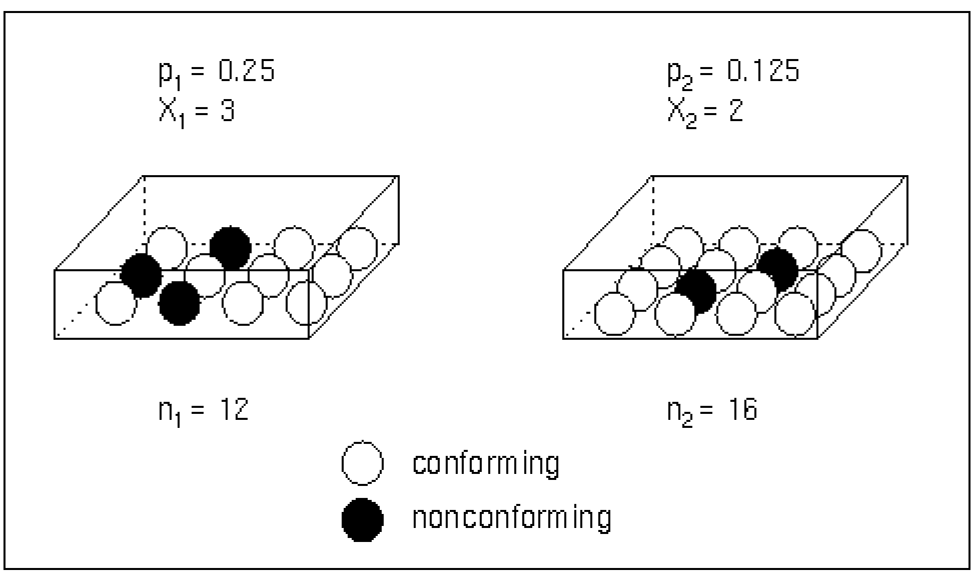
The following notation is used in this section:

p	expected proportion of nonconforming items produced by the process
p_i	proportion of nonconforming items in the *i*th subgroup
X_i	number of nonconforming items in the *i*th subgroup
n_i	number of items in the *i*th subgroup
$\bar{p}$	average proportion of nonconforming items taken across subgroups: $$\bar{p} = \frac{n_1 p_1 + \cdots + n_N p_N}{n_1 + \cdots + n_N} = \frac{X_1 + \cdots + X_N}{n_1 + \cdots + n_N}$$
N	number of subgroups
$I_T(\alpha, \beta)$	incomplete beta function: $$I_T(\alpha, \beta) = (\Gamma(\alpha + \beta)/\Gamma(\alpha)\Gamma(\beta)) \int_0^T t^{\alpha-1}(1-t)^{\beta-1} dt$$ for $0 < T < 1$, $\alpha > 0$, and $\beta > 0$, where $\Gamma(\cdot)$ is the gamma function

Plotted Points

Each point on a p chart represents the observed proportion ($p_i = X_i/n_i$) of nonconforming items in a subgroup. For example, suppose the second subgroup (see Figure 17.69) contains 16 items, of which two are nonconforming. The point plotted for the second subgroup is $p_2 = 2/16 = 0.125$.

Figure 17.69 Proportions Versus Counts



Note that an np chart displays the number (count) of nonconforming items X_i. You can use the NPCHART statement to create np charts; see "NPCHART Statement: SHEWHART Procedure" on page 1600.

Central Line

By default, the central line on a p chart indicates an estimate of p that is computed as $\bar{p}$. If you specify a known value (p_0) for p, the central line indicates the value of p_0.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of p_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that p_i exceeds the limits

The lower and upper control limits, LCL and UCL, respectively, are computed as

$$\begin{aligned} \text{LCL} &= \max\left(\bar{p} - k\sqrt{\bar{p}(1-\bar{p})/n_i}\ , 0\right) \\ \text{UCL} &= \min\left(\bar{p} + k\sqrt{\bar{p}(1-\bar{p})/n_i}\ , 1\right) \end{aligned}$$

A lower probability limit for p_i can be determined using the fact that

$$\begin{aligned} P\{p_i < \text{LCL}\} &= 1 - P\{p_i \geq \text{LCL}\} \\ &= 1 - P\{X_i \geq n_i\text{LCL}\} \\ &= 1 - I_{\bar{p}}(n_i\text{LCL}, n_i + 1 - n_i\text{LCL}) \\ &= I_{1-\bar{p}}(n_i + 1 - n_i\text{LCL}, n_i\text{LCL}) \end{aligned}$$

Refer to Johnson, Kotz, and Kemp (1992). This assumes that the process is in statistical control and that X_i is binomially distributed. The lower probability limit LCL is then calculated by setting

$$I_{1-\bar{p}}(n_i + 1 - n_i\text{LCL}, n_i\text{LCL}) = \alpha/2$$

and solving for LCL. Similarly, the upper probability limit for p_i can be determined using the fact that

$$\begin{aligned} P\{p_i > \text{UCL}\} &= P\{p_i > \text{UCL}\} \\ &= P\{X_i > n_i\text{UCL}\} \\ &= I_{\bar{p}}(n_i\text{UCL} + 1, n_i - n_i\text{UCL}) \end{aligned}$$

The upper probability limit UCL is then calculated by setting

$$I_{\bar{p}}(n_i\text{UCL} + 1, n_i - n_i\text{UCL}) = \alpha/2$$

and solving for UCL. The probability limits are asymmetric around the central line. Note that both the control limits and probability limits vary with n_i.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.

- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify p_0 with the P0= option or with the variable _P_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 17.43 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLP	lower control limit for proportion of nonconforming items
LIMITN	nominal sample size associated with the control limits
_P_	average proportion of nonconforming items ($\bar{p}$ or p_0)
SIGMAS	multiple (k) of standard error of p_i
SUBGRP	*subgroup-variable* specified in the PCHART statement
TYPE	type (standard or estimate) of _P_
UCLP	upper control limit for proportion of nonconforming items
VAR	*process* specified in the PCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLP_, _UCLP_, and _SIGMAS_.
2. If the limits are defined in terms of a multiple k of the standard error of p_i, the value of _ALPHA_ is computed as $\alpha = P\{p_i < \text{\_LCLP\_}\} + P\{p_i > \text{\_UCLP\_}\}$, using the incomplete beta function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = (\text{\_UCLP\_} - \text{\_P\_})/\sqrt{\text{\_P\_}(1 - \text{\_P\_})/\text{\_LIMITN\_}}$. If _LIMITN_ has the special missing value *V*, this value is assigned to _SIGMAS_.
4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the PCHART statement. For an example, see "Saving Control Limits" on page 1647.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup proportion of nonconforming items variable named by *process* suffixed with *P*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the PCHART statement. For example, consider the following statements:

```
proc shewhart data=Input;
   pchart (Rework Rejected)*Batch / outhistory=Summary
                                    subgroupn =30;
run;
```

The data set Summary contains variables named Batch, ReworkP, ReworkN, RejectedP, and RejectedN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Proportions of Nonconforming Items" on page 1646.

Note that an OUTHISTORY= data set created with the PCHART statement can be reused as a HISTORY= data set by either the PCHART statement or the NPCHART statement.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The variables shown in the following table are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on *p* chart
LCLP	lower control limit for proportion of nonconforming items
LIMITN	nominal sample size associated with the control limits
_P_	average proportion of nonconforming items
SIGMAS	multiple (k) of the standard error of p_i associated with the control limits
subgroup	values of the subgroup variable
SUBP	subgroup proportion of nonconforming items
SUBN	subgroup sample size
TESTS	tests for special causes signaled on *p* chart
UCLP	upper control limit for proportion of nonconforming items
VAR	*process* specified in the PCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The *k*th character of a value of _TESTS_ is *k* if Test *k* is positive at that subgroup. For example, if you request the first four tests (the tests appropriate for *p* charts) and Tests 2 and 4 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, a 4 for the fourth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1647.

ODS Tables

The following table summarizes the ODS tables that you can request with the PCHART statement.

Table 17.44 ODS Tables Produced with the PCHART Statement

Table Name	Description	Options
PCHART	*p* chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLE-OUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*). **NOTE:** In SAS/QC 13.1 the SHEWHART procedure does not support the creation of graphs that are editable with the ODS Graphics Editor.

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. PCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1946.

When ODS Graphics is in effect, the PCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 17.45.

Table 17.45 ODS Graphics Produced by the PCHART Statement

ODS Graph Name	Plot Description
PChart	*p* chart

See Chapter 3, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (counts of nonconforming items) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the PCHART statement must be a SAS variable in the DATA= data set. This variable provides counts for subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the PCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a count for each *process* and a value for the *subgroup-variable*. The data set must contain one observation for each subgroup. Note that you can specify the DATAUNIT= option in the PCHART statement to read proportions or percentages of nonconforming items instead of counts. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

When you use a DATA= data set with the PCHART statement, the SUBGROUPN= option (which specifies the subgroup sample size) is required. By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2031).

For an example of a DATA= data set, see "Creating p Charts from Count Data" on page 1640.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   pchart Rejects*Batch / subgroupn= 100;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLP_, _P_, and _UCLP_, which specify the control limits directly
- the variable _P_, without providing _LCLP_ and _UCLP_. The value of _P_ is used to calculate the control limits according to the equations in "Control Limits" on page 1662.

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.

- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE' and 'STANDARD'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1649.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to create your own HISTORY= data set.

A HISTORY= data set used with the PCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup proportion of nonconforming items variable for each *process*
- a subgroup sample size variable for each *process*

The names of the proportion sample size variables must be the *process* name concatenated with the special suffix characters *P* and *N*, respectively.

For example, consider the following statements:

```
proc shewhart history=Summary;
   pchart (Rework Rejected)*Batch / subgroupn=50;
run;
```

The data set Summary must include the variables Batch, ReworkP, ReworkN, RejectedP, and RejectedN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2031 for an example).

For an example of a HISTORY= data set, see "Creating p Charts from Summary Data" on page 1643.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2096.

The following table lists the variables required in a TABLE= data set used with the PCHART statement:

Table 17.46 Variables Required in a TABLE= Data Set

Variable	Description
LCLP	lower control limit for proportion of nonconforming items
LIMITN	nominal sample size associated with the control limits
_P_	average proportion of nonconforming items
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBP	subgroup proportion of nonconforming items
UCLP	upper control limit for proportion of nonconforming items

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1647.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup proportion nonconforming variable
Vertical	TABLE=	_SUBP_

For example, the following sets of statements specify the label *Proportion Nonconforming* for the vertical axis of the *p* chart:

```
proc shewhart data=Circuits;
   pchart Fail*Batch / subgroupn=500;
   label Fail = 'Proportion Nonconforming';
run;

proc shewhart history=Cirhist;
   pchart Fail*Batch ;
   label FailP = 'Proportion Nonconforming';
run;

proc shewhart table=Cirtable;
   pchart Fail*Batch ;
   label _SUBP_ = 'Proportion Nonconforming';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: PCHART Statement

This section provides advanced examples of the PCHART statement.

Example 17.22: Applying Tests for Special Causes

NOTE: See *p Charts-Tests for Special Causes* in the SAS/QC Sample Library.

This example shows how you can apply tests for special causes to make *p* charts more sensitive to special causes of variation. The following statements create a SAS data set named Circuit3, which contains the number of failing circuits for 20 batches from the circuit manufacturing process introduced in "Creating p Charts from Count Data" on page 1640:

```
data Circuit3;
   input Batch Fail @@;
   datalines;
 1 12     2 21     3 16     4  9
 5  3     6  4     7  6     8  9
 9 11    10 13    11 12    12  7
13  2    14 14    15  9    16  8
17 14    18 10    19 11    20  9
;
```

The following statements create the *p* chart, apply several tests to the chart, and tabulate the results:

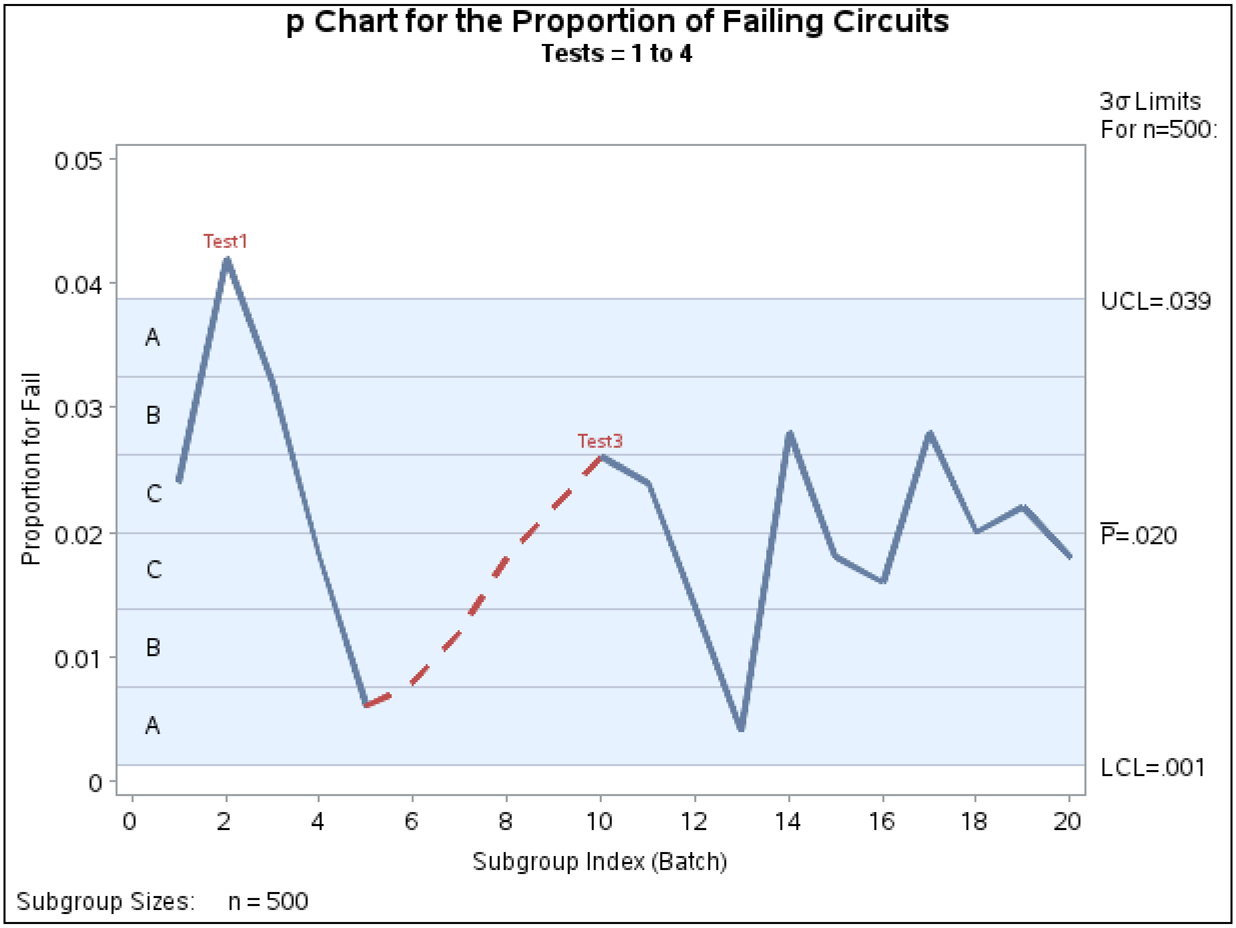
```
ods graphics off;
title1'p Chart for the Proportion of Failing Circuits';
title2 'Tests = 1 to 4';
proc shewhart data=Circuit3;
   pchart Fail*Batch / subgroupn = 500
                       tests     = 1 to 4
                       zones
                       zonelabels
                       ltests    = 20
                       table
                       tabletest
                       tablelegend;
run;
```

The chart is shown in Output 17.22.1, and the printed output is shown in Output 17.22.2. The TESTS= option requests Tests 1, 2, 3, and 4, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2073. The TABLETESTS option requests a table of proportions of nonconforming items and control limits, with a column indicating which subgroups tested positive for special causes. The TABLELEGEND option adds a legend describing the tests that are positive.

The ZONELABELS option displays zone lines and zone labels on the chart. The zones are used to define the tests. The LTESTS= option specifies the line type used to connect the points in a pattern for a test that is signaled.

Output 17.22.1 and Output 17.22.2 indicate that Test 1 is positive at batch 2 and Test 3 is positive at batch 10.

Output 17.22.1 Tests for Special Causes Displayed on p Chart



Output 17.22.2 Tabular Form of p Chart

p Chart for the Proportion of Failing Circuits
Tests = 1 to 4

The SHEWHART Procedure

p Chart Summary for Fail

3 Sigma Limits with n=500 for Proportion

Batch	Subgroup Sample Size	Lower Limit	Subgroup Proportion	Upper Limit	Special Tests Signaled
1	500	0.00121703	0.02400000	0.03878297	
2	500	0.00121703	0.04200000	0.03878297	1
3	500	0.00121703	0.03200000	0.03878297	
4	500	0.00121703	0.01800000	0.03878297	
5	500	0.00121703	0.00600000	0.03878297	
6	500	0.00121703	0.00800000	0.03878297	
7	500	0.00121703	0.01200000	0.03878297	
8	500	0.00121703	0.01800000	0.03878297	
9	500	0.00121703	0.02200000	0.03878297	
10	500	0.00121703	0.02600000	0.03878297	3
11	500	0.00121703	0.02400000	0.03878297	
12	500	0.00121703	0.01400000	0.03878297	
13	500	0.00121703	0.00400000	0.03878297	
14	500	0.00121703	0.02800000	0.03878297	
15	500	0.00121703	0.01800000	0.03878297	
16	500	0.00121703	0.01600000	0.03878297	
17	500	0.00121703	0.02800000	0.03878297	
18	500	0.00121703	0.02000000	0.03878297	
19	500	0.00121703	0.02200000	0.03878297	
20	500	0.00121703	0.01800000	0.03878297	

Test Descriptions	
Test 1	One point beyond Zone A (outside control limits)
Test 3	Six points in a row steadily increasing or decreasing

Example 17.23: Specifying Standard Average Proportion

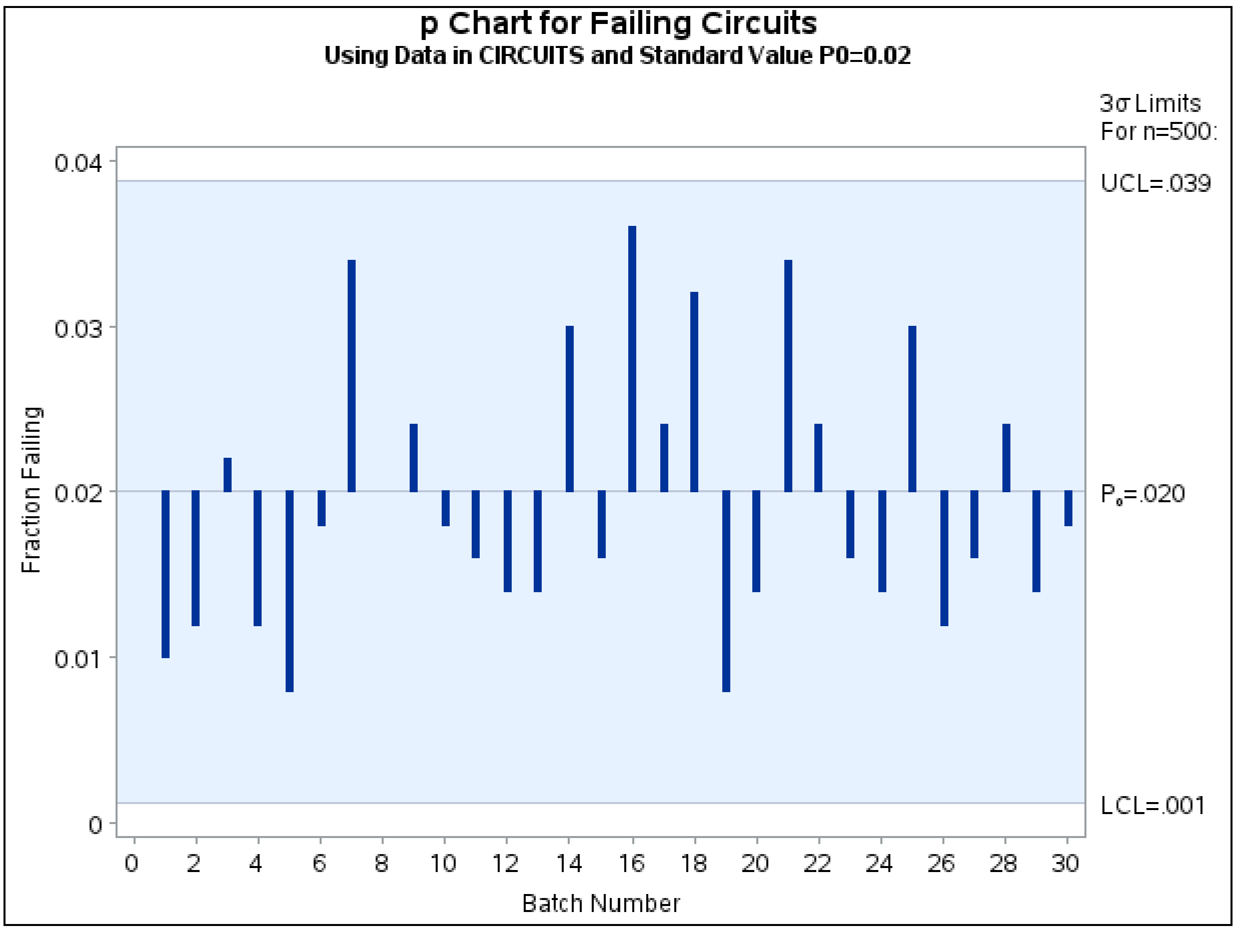
NOTE: See *p Charts-Specifying Std Average Proportion* in the SAS/QC Sample Library.

In some situations, a standard (known) value (p_0) is available for the expected proportion of nonconforming items, based on extensive testing or previous sampling. This example illustrates how you can specify p_0 to create a p chart.

A p chart is used to monitor the proportion of failing circuits in the data set Circuits, which is introduced in "Creating p Charts from Count Data" on page 1640. The expected proportion is known to be $p_0 = 0.02$. The following statements create a p chart, shown in Output 17.23.1, using p_0 to compute the control limits:

```
ods graphics off;
title1 'p Chart for Failing Circuits';
title2 'Using Data in CIRCUITS and Standard Value P0=0.02';
proc shewhart data=Circuits;
   pchart Fail*Batch / subgroupn = 500
                       p0        = 0.02
                       psymbol   = p0
                       wneedles  = 3
                       nolegend;
   label Batch = 'Batch Number'
         Fail  = 'Fraction Failing';
run;
```

Output 17.23.1 A p Chart with Standard Value p_0



The chart indicates that the process is in control. The P0= option specifies p_0. The PSYMBOL= option specifies a label for the central line indicating that the line represents a standard value. The NEEDLES option connects points to the central line with vertical needles. The NOLEGEND option suppresses the default legend for subgroup sample sizes. Labels for the vertical and horizontal axes are provided with the LABEL statement. For details concerning axis labeling, see "Axis Labels" on page 1669.

Alternatively, you can specify p_0 using the variable _P_ in a LIMITS= data set, as follows:

```
data Climits;
   length _var_ _subgrp_ _type_ $8;
   _p_      = 0.02;
   _subgrp_ = 'Batch';
   _var_    = 'Fail';
   _type_   = 'STANDARD';
   _limitn_ = 500;
run;

proc shewhart data=Circuits limits=Climits;
   pchart Fail*Batch / subgroupn = 500
                       psymbol   = p0
                       nolegend
                       needles;
   label batch ='Batch Number'
         fail  ='Fraction Failing';
run;
```

The bookkeeping variable _TYPE_ indicates that _P_ has a standard value. The chart produced by these statements is identical to the chart in Output 17.23.1.

Example 17.24: Working with Unequal Subgroup Sample Sizes

NOTE: See *p Charts with Unequal Subgroup Sample Sizes* in the SAS/QC Sample Library.

The following statements create a SAS data set named Battery, which contains the number of alkaline batteries per lot failing an acceptance test. The number of batteries tested in each lot varies but is approximately 150.

```
data Battery;
   length lot $3;
   input lot nFailed Sampsize @@;
   label nFailed  = 'Number Failed'
         lot      = 'Lot Number'
         Sampsize = 'Number Sampled';
   datalines;
AE3  6  151     AE4  5  142     AE9  6  145
BR3  9  149     BR7  3  150     BR8  0  156
BR9  4  150     DB1  9  158     DB2  4  152
DB3  0  162     DB5  9  140     DB6  7  161
DS4  6  154     DS6  1  144     DS8  5  154
JG1  3  151     MC3  8  148     MC4  2  143
MK6  4  150     MM1  4  147     MM2  0  150
RT5  2  154     RT9  8  149     SP1  3  160
SP3  9  153
;
```

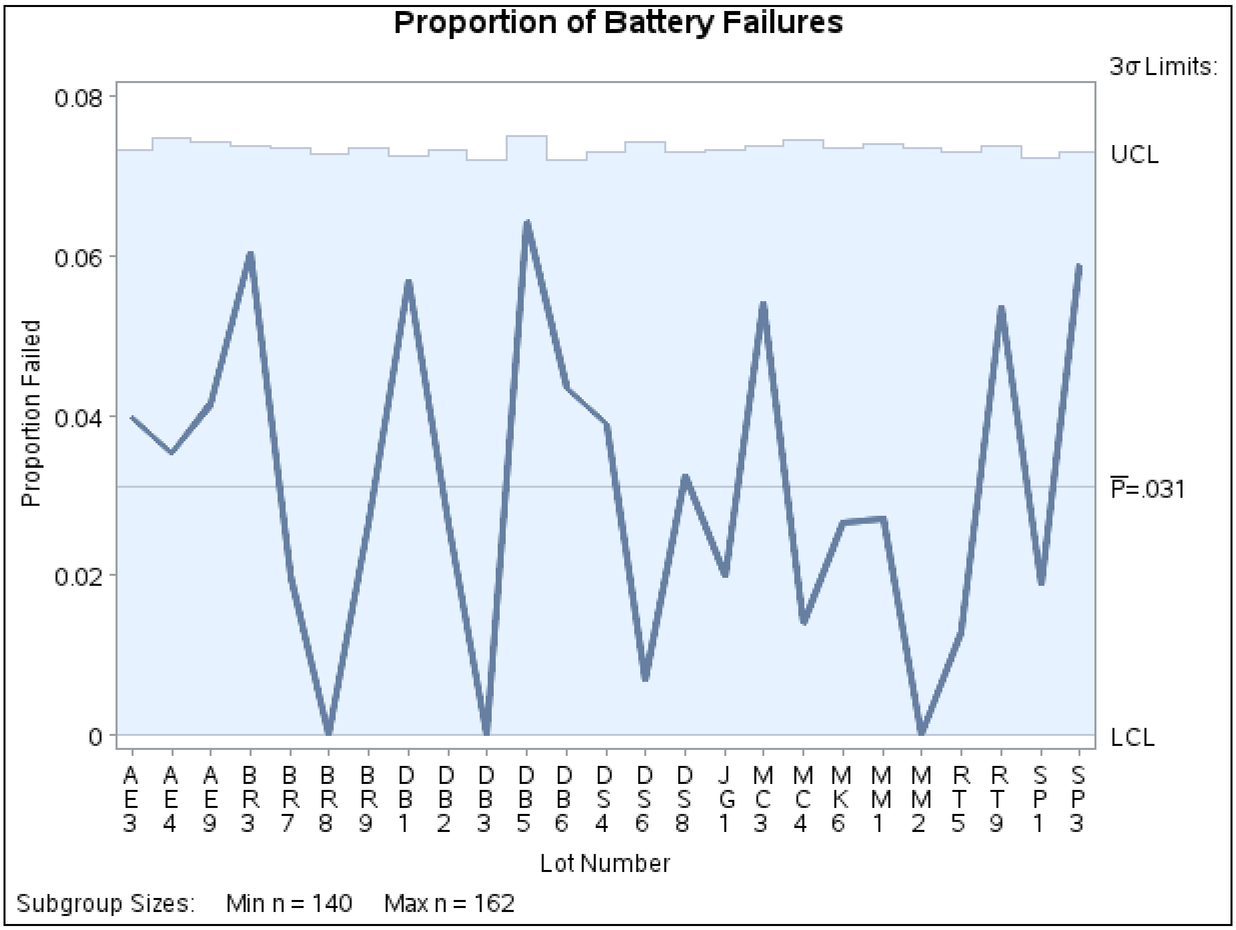
The variable nFailed contains the number of battery failures, the variable Lot contains the lot number, and the variable Sampsize contains the lot sample size. The following statements request a *p* chart for this data:

```
ods graphics off;
title 'Proportion of Battery Failures';
proc shewhart data=Battery;
   pchart nFailed*Lot / subgroupn = Sampsize
                        turnhlabels
                        font      = 'Lucida Console'
                        outlimits = Batlim;
   label nFailed = 'Proportion Failed';
run;
```

Here the FONT= option is used to specify the name of a hardware font to be used for the *p* chart. In this case the requested font is Lucida Console, a Windows TrueType font. See *SAS/GRAPH: Reference* and *SAS Companion for Microsoft Windows* for more information about hardware and TrueType fonts.

The chart is shown in Output 17.24.1 and the OUTLIMITS= data set Batlim is listed in Output 17.24.2.

Output 17.24.1 A *p* Chart with Varying Subgroup Sample Sizes



Note that the upper control limit varies with the subgroup sample size. The lower control limit is truncated at zero. The sample size legend indicates the minimum and maximum subgroup sample sizes.

Output 17.24.2 Listing of the Control Limits Data Set Batlim

Control Limits for Battery Failures

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLP_	_P_	_UCLP_
nFailed	lot	ESTIMATE	V	V	3	V	0.031010	V

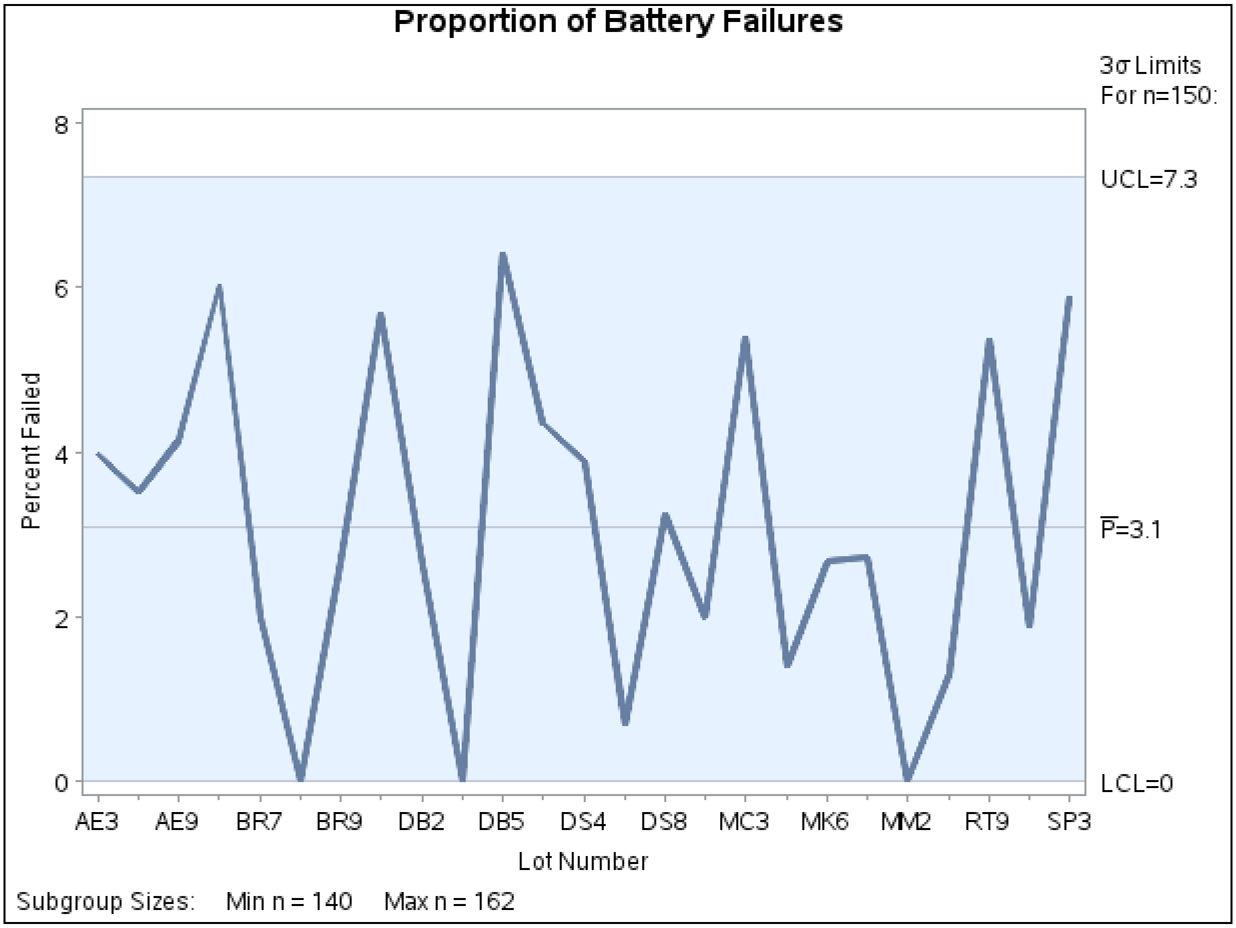
The variables in Batlim whose values vary with subgroup sample size are assigned the special missing value *V*.

The SHEWHART procedure provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for computing the control limits, as illustrated by the following statements:

```
title 'Proportion of Battery Failures';
proc shewhart data=Battery;
   pchart nFailed*Lot / subgroupn = Sampsize
                        limitn    = 150
                        alln
                        outlimits = Clim2
                        yscale    = percent;
   label nFailed = 'Percent Failed';
run;
```

The ALLN option specifies that all points (regardless of subgroup sample size) are to be displayed. By default, only points for subgroups whose sample size matches the LIMITN= value are displayed. The YSCALE= option specifies that the vertical axis is to be scaled in percentages rather than proportions. The chart is shown in Output 17.24.3.

Output 17.24.3 Control Limits Based on Fixed Subgroup Sample Size



All the points are inside the control limits, indicating that the process is in statistical control. Since there is relatively little variation in the sample sizes, the control limits in Output 17.24.3 provide a close approximation to the exact control limits in Output 17.24.1, and the same conclusions can be drawn from both charts. In general, care should be taken when interpreting charts that use a nominal sample size to compute control limits, since these limits are only approximate when the sample sizes vary.

Example 17.25: Creating a Chart with Revised Control Limits

NOTE: See *p Charts with Revised Control Limits* in the SAS/QC Sample Library.

The following statements create a SAS data set named CircOne, which contains the number of failing circuits for 30 batches produced by the circuit manufacturing process introduced in the section "Getting Started: PCHART Statement" on page 1640:

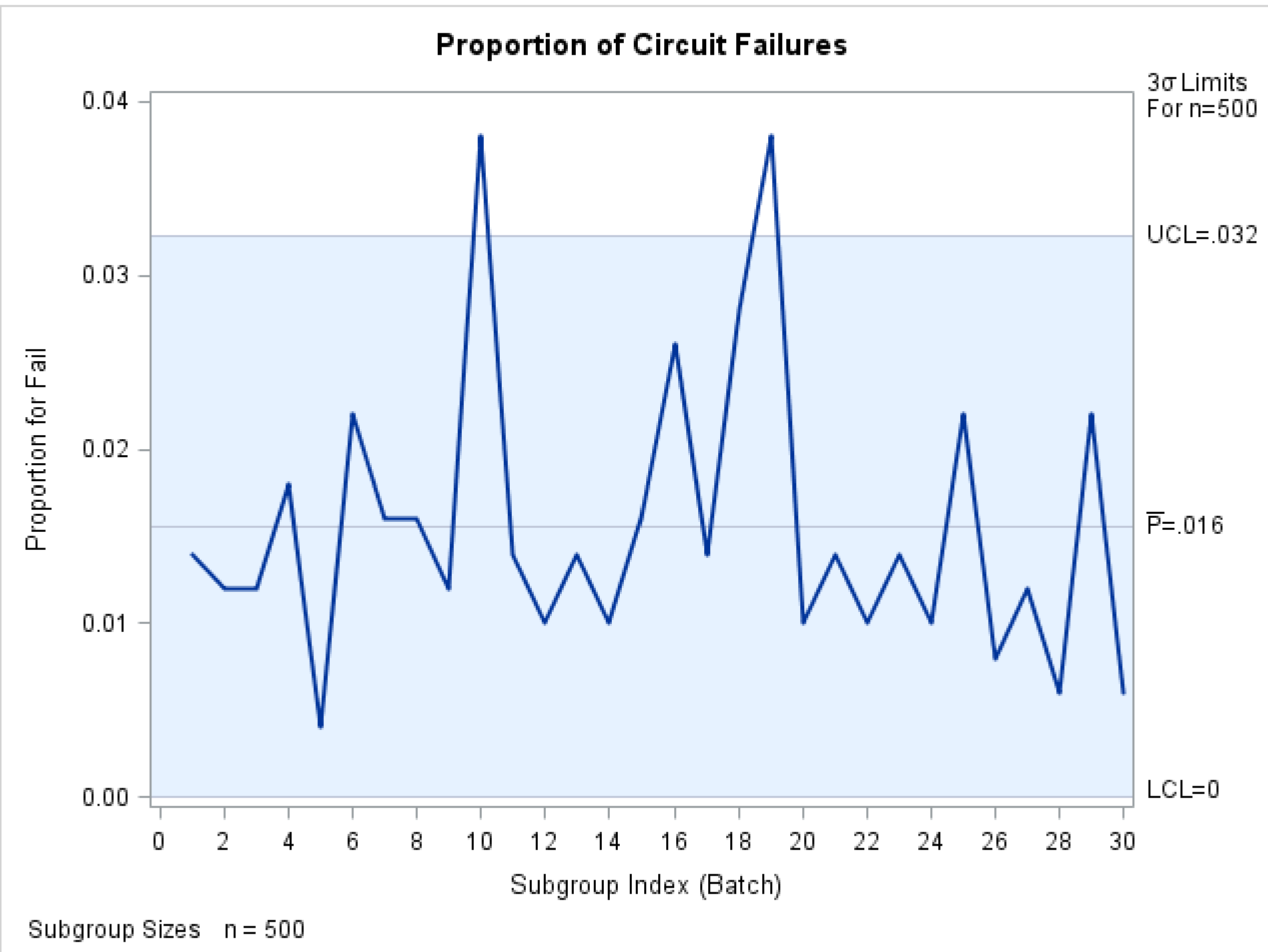
```
data CircOne;
   input Batch Fail @@;
   datalines;
 1   7   2   6   3   6   4   9    5   2
 6  11   7   8   8   8   9   6   10  19
```

```
11    7  12    5  13    7  14    5    15    8
16   13  17    7  18   14  19   19    20    5
21    7  22    5  23    7  24    5    25   11
26    4  27    6  28    3  29   11    30    3
;
```

A *p* chart is used to monitor the proportion of failing circuits. The following statements create the chart shown in Output 17.25.1:

```
ods graphics on;
title 'Proportion of Circuit Failures';
proc shewhart data=CircOne;
   pchart Fail*Batch / subgroupn = 500
                       outindex  = 'Trial Limits'
                       outlimits = Faillim1
                       odstitle  = title;
run;
```

Output 17.25.1 A *p* Chart for Circuit Failures



Batches 10 and 19 have unusually high proportions of failing circuits. Subsequent investigation identifies special causes for both batches, and it is decided to eliminate these batches from the data set and recompute the control limits. The following statements create a data set named Faillim2 that contains the revised control limits:

```
proc shewhart data=CircOne;
   where Batch ^= 10 and Batch ^= 19;
   pchart Fail*Batch / subgroupn = 500
                       nochart
                       outindex  ='Revised Limits'
                       outlimits = Faillim2;
run;


data Faillims;
   set Faillim1 Faillim2;
run;
```

The data set Faillims, which contains the true and revised control limits, is listed in Output 17.25.2.

Output 17.25.2 Listing of the Data Set Faillims

Proportion of Circuit Failures

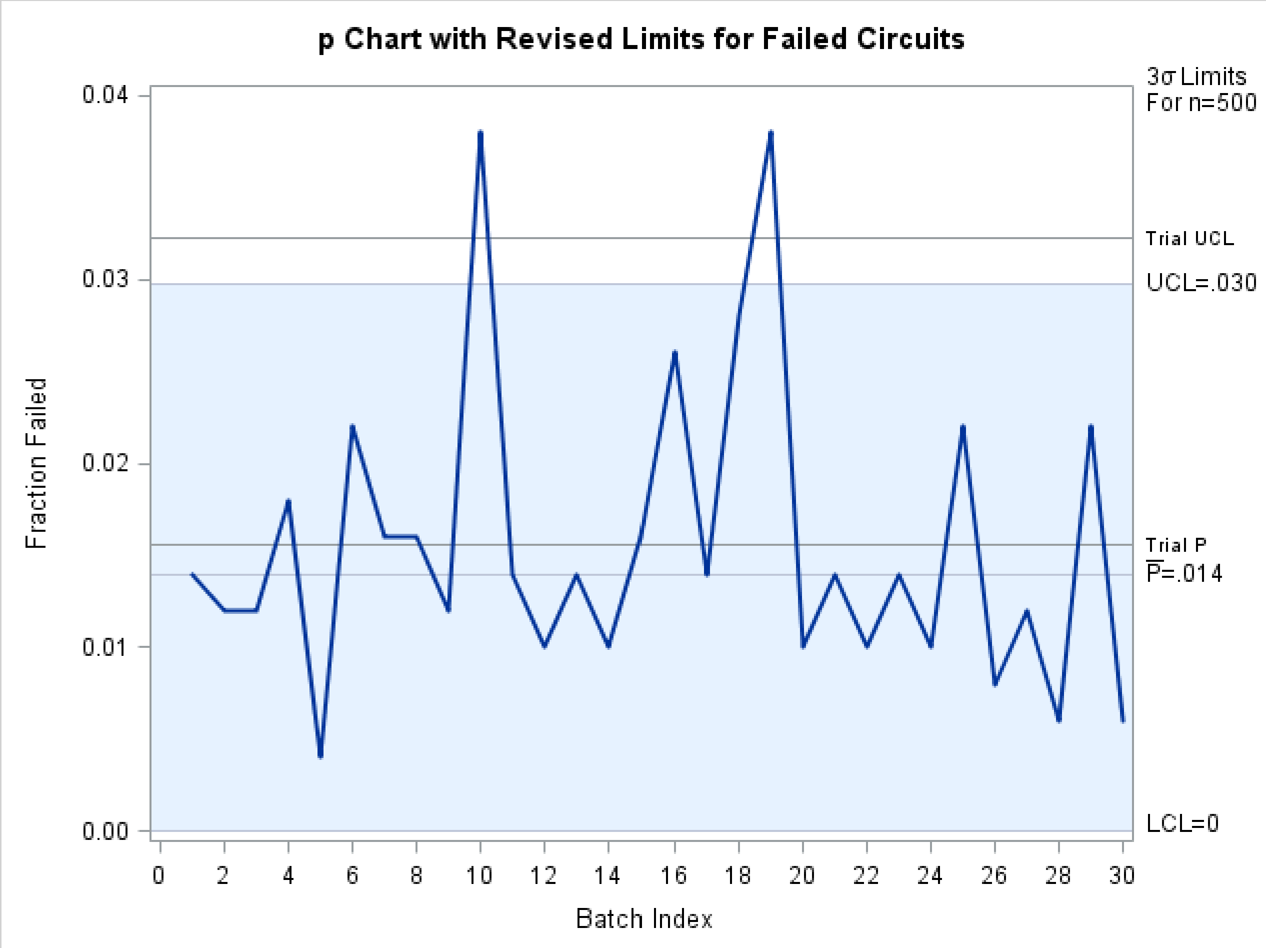
VAR	_SUBGRP_	_INDEX_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLP_	_P_	_UCLP_
Fail	Batch	Trial Limits	ESTIMATE	500	.002421958	3	0	0.0156	0.032226
Fail	Batch	Revised Limits	ESTIMATE	500	.002477491	3	0	0.0140	0.029763

The following statements create a *p* chart displaying both sets of control limits:

```
title 'p Chart with Revised Limits for Failed Circuits';
proc shewhart data=CircOne limits=Faillims;
   pchart Fail*Batch / subgroupn  = 500
                       readindex  = 'Revised Limits'
                       vref       = 0.0156 0.032226
                       vreflabels = ('Trial P' 'Trial UCL')
                       vreflabpos = 3
                       odstitle   = title
                       nolegend;
   label Fail  = 'Fraction Failed'
         Batch = 'Batch Index';
run;
ods graphics off;
```

The READINDEX= option is used to select the revised limits displayed on the *p* chart in Output 17.25.3. See “Displaying Multiple Sets of Control Limits” on page 2033. The VREF=, VREFLABELS=, and VREFLABPOS= options are used to display and label the trial limits. You can also pass in the values of the trial limits with macro variables. For an illustration of this technique, see Example 17.6.

Output 17.25.3 p Chart with Revised Limits



Example 17.26: OC Curve for Chart

NOTE: See *OC Curve for a p Chart* in the SAS/QC Sample Library.

This example uses the GPLOT procedure and the OUTLIMITS= data set Faillim2 from the previous example to plot an OC curve for the p chart shown in Output 17.25.3.

The OC curve displays β (the probability that p_i lies within the control limits) as a function of p (the true proportion nonconforming). The computations are exact, assuming that the process is in control and that the number of nonconforming items (X_i) has a binomial distribution.

The value of β is computed as follows:

$$\begin{aligned}
\beta &= P(p_i \leq \text{UCL}) - P(p_i < \text{LCL}) \\
&= P(X_i \leq n\text{UCL}) - P(X_i < n\text{LCL}) \\
&= P(X_i < n\text{UCL}) + P(X_i = n\text{UCL}) - P(X_i < n\text{LCL}) \\
&= I_{1-p}(n + 1 - n\text{UCL}, n\text{UCL}) + P(X_i = n\text{UCL}) - I_{1-p}(n + 1 - n\text{LCL}, n\text{LCL}) \\
&= I_p(n\text{LCL}, n + 1 - n\text{LCL}) + P(X_i = n\text{UCL}) - I_p(n\text{UCL}, n + 1 - n\text{UCL})
\end{aligned}$$

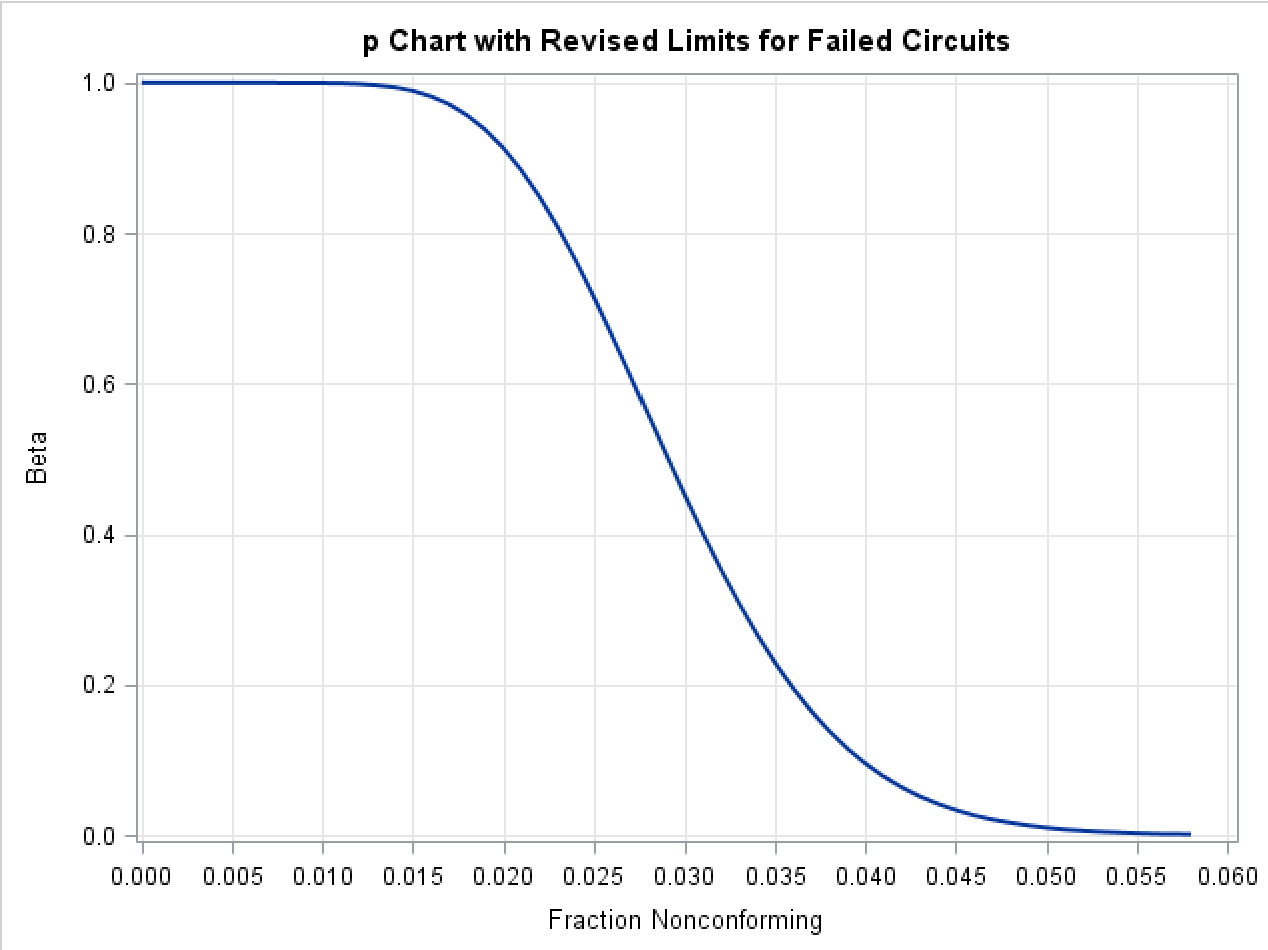
Here, $I_p(\cdot,\cdot)$ denotes the incomplete beta function. The following DATA step computes β (the variable BETA) as a function of p (the variable P):

```
data ocpchart;
   set Faillim2;
   keep beta fraction _lclp_ _p_ _uclp_;
   nucl=_limitn_*_uclp_;
   nlcl=_limitn_*_lclp_;
   do p=0 to 500;
      fraction=p/1000;
      if nucl=floor(nucl) then
         adjust=probbnml(fraction,_limitn_,nucl) -
                probbnml(fraction,_limitn_,nucl-1);
      else adjust=0;
      if nlcl=0 then
         beta=1 - probbeta(fraction,nucl,_limitn_-nucl+1) + adjust;
      else beta=probbeta(fraction,nlcl,_limitn_-nlcl+1) -
                probbeta(fraction,nucl,_limitn_-nucl+1) +
                adjust;
      if beta >= 0.001 then output;
   end;
   call symput('lcl', put(_lclp_,5.3));
   call symput('mean',put(_p_,   5.3));
   call symput('ucl', put(_uclp_,5.3));
run;
```

The following statements display the OC curve shown in Output 17.26.1:

```
proc sgplot data=ocpchart;
   series x=fraction y=beta / lineattrs=(thickness=2);
   xaxis values=(0 to 0.06 by 0.005) grid;
   yaxis grid;
   label fraction = 'Fraction Nonconforming'
         beta     = 'Beta';
run;
```

Output 17.26.1 OC Curve for p Chart



RCHART Statement: SHEWHART Procedure

Overview: RCHART Statement

The RCHART statement creates an R chart for subgroup ranges, which is used to analyze the variability of a process.[7]

You can use options in the RCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted ranges or as probability limits
- tabulate subgroup sample sizes, subgroup ranges, control limits, and other information

[7]You can also use s charts for this purpose; see "SCHART Statement: SHEWHART Procedure" on page 1721. In general, s charts are recommended with large subgroup sample sizes ($n_i \geq 10$).

- save control limits in an output data set
- save subgroup sample sizes, subgroup means, and subgroup ranges in an output data set
- read preestablished control limits from a data set
- specify the method for estimating the process standard deviation
- specify a known (standard) process standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing *R* charts with the RCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 3, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: RCHART Statement

This section introduces the RCHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the RCHART statement is presented in the section "Syntax: RCHART Statement" on page 1696, and advanced examples are given in the section "Examples: RCHART Statement" on page 1715.

Creating Range Charts from Raw Data

NOTE: See *Range Chart (R Chart) Examples* in the SAS/QC Sample Library.

A disk drive manufacturer performs a battery of tests to evaluate its drives. The following statements create a data set named Disks, which contains the time (in milliseconds) required to complete one of these tests for six drives in each of 25 lots:

```
data Disks;
   input Lot @;
   do i=1 to 6;
      input Time @;
      output;
   end;
   drop i;
   datalines;
  1 8.05 7.90 8.04 8.06 8.01 7.99
  2 8.03 8.06 8.02 8.02 7.97 8.03
  3 8.00 7.94 7.97 7.95 8.00 8.01
  4 8.00 8.06 8.06 7.99 7.97 7.96
  5 7.93 8.01 8.00 8.09 8.06 8.02
  6 7.98 7.99 8.01 8.09 8.00 7.97
  7 8.00 7.94 7.93 8.03 7.93 8.08
  8 8.01 7.98 7.98 8.07 8.05 8.09
  9 7.97 7.96 8.01 8.11 8.06 8.07
 10 7.93 8.03 8.03 8.00 7.93 8.03
 11 8.00 8.00 8.02 7.92 7.98 8.01
 12 7.98 7.93 8.01 7.97 8.02 8.00
 13 8.06 7.93 7.98 7.98 8.02 7.96
 14 8.05 7.98 8.05 7.99 7.95 7.99
 15 7.94 8.01 7.97 8.04 7.91 8.03
 16 8.03 8.03 8.02 8.06 8.00 7.97
 17 8.03 7.94 8.05 8.05 8.04 7.94
 18 7.99 7.99 7.86 7.99 8.06 8.03
 19 7.95 7.96 7.99 7.96 7.94 8.12
 20 8.03 8.07 7.98 7.97 8.00 8.04
 21 8.04 7.90 8.03 8.02 7.98 7.97
 22 7.95 8.05 7.98 8.01 7.97 8.15
 23 8.06 8.00 8.03 8.02 7.99 7.95
 24 7.97 8.02 8.00 7.96 7.96 8.00
 25 8.12 7.97 7.99 8.09 8.05 8.00
;
```

A partial listing of Disks is shown in Figure 17.70.

Figure 17.70 Partial Listing of the Data Set Disks

The Data Set DISKS

Lot	Time
1	8.05
1	7.90
1	8.04
1	8.06
1	8.01
1	7.99
2	8.03
2	8.06
2	8.02
2	8.02
2	7.97
2	8.03
3	8.00
3	7.94
3	7.97
3	7.95
3	8.00
3	8.01

The data set Disks is said to be in "strung-out" form since each observation contains the lot number and test time for a single disk drive. The first five observations contain the times for the first lot, the second five observations contain the times for the second lot, and so on. Because the variable Lot classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable Time contains the time measurements and is referred to as the *process variable* (or *process* for short).

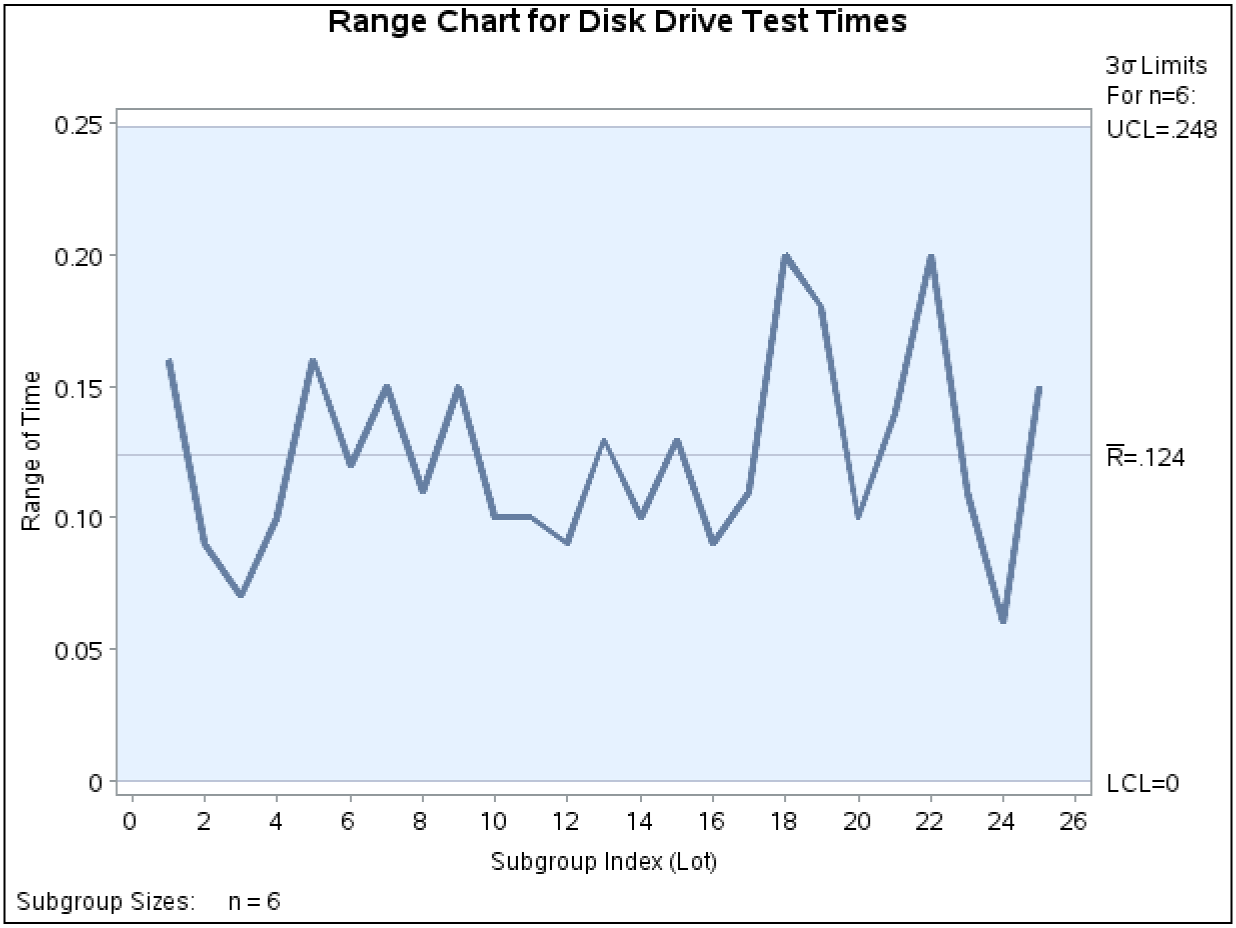
You can use an *R* chart to determine whether the variability in the performance of the disk drives is in control. The following statements create the *R* chart shown in Figure 17.71:

```
ods graphics off;
title 'Range Chart for Disk Drive Test Times';
proc shewhart data=Disks;
   rchart Time*Lot;
run;
```

This example illustrates the basic form of the RCHART statement. After the keyword RCHART, you specify the *process* to analyze (in this case, Time), followed by an asterisk and the *subgroup-variable* (Lot).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Figure 17.71 *R* Chart for the Data Set Disks (Traditional Graphics)



Each point on the *R* chart represents the range of the measurements for a particular lot. For instance, the range plotted for the first lot is $8.06 - 7.90 = 0.16$. Since all of the subgroup ranges lie within the control limits, you can conclude that the variability in the performance of the disk drives is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 17.48. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1694.

For computational details, see "Constructing Range Charts" on page 1706. For more details on reading raw data, see "DATA= Data Set" on page 1711.

Creating Range Charts from Summary Data

NOTE: See *Range Chart (R Chart) Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create *R* charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the RCHART statement with data of this type.

The following data set (Disksum) provides the data from the preceding example in summarized form:

```
data Disksum;
   input Lot TimeX TimeR;
   TimeN=6;
   datalines;
  1  8.00833  0.16
  2  8.02167  0.09
  3  7.97833  0.07
  4  8.00667  0.10
  5  8.01833  0.16
  6  8.00667  0.12
  7  7.98500  0.15
  8  8.03000  0.11
  9  8.03000  0.15
 10  7.99167  0.10
 11  7.98833  0.10
 12  7.98500  0.09
 13  7.98833  0.13
 14  8.00167  0.10
 15  7.98333  0.13
 16  8.01833  0.09
 17  8.00833  0.11
 18  7.98667  0.20
 19  7.98667  0.18
 20  8.01500  0.10
 21  7.99000  0.14
 22  8.01833  0.20
 23  8.00833  0.11
 24  7.98500  0.06
 25  8.03667  0.15
;
```

A partial listing of Disksum is shown in Figure 17.72. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Lot). The variable TimeX contains the subgroup means, the variable TimeR contains the subgroup ranges, and the variable TimeN contains the subgroup sample sizes (these are all six).

Figure 17.72 The Summary Data Set Disksum

The Summary Data Set of Disk Drive Test Times

Lot	TimeX	TimeR	TimeN
1	8.00833	0.16	6
2	8.02167	0.09	6
3	7.97833	0.07	6
4	8.00667	0.10	6
5	8.01833	0.16	6

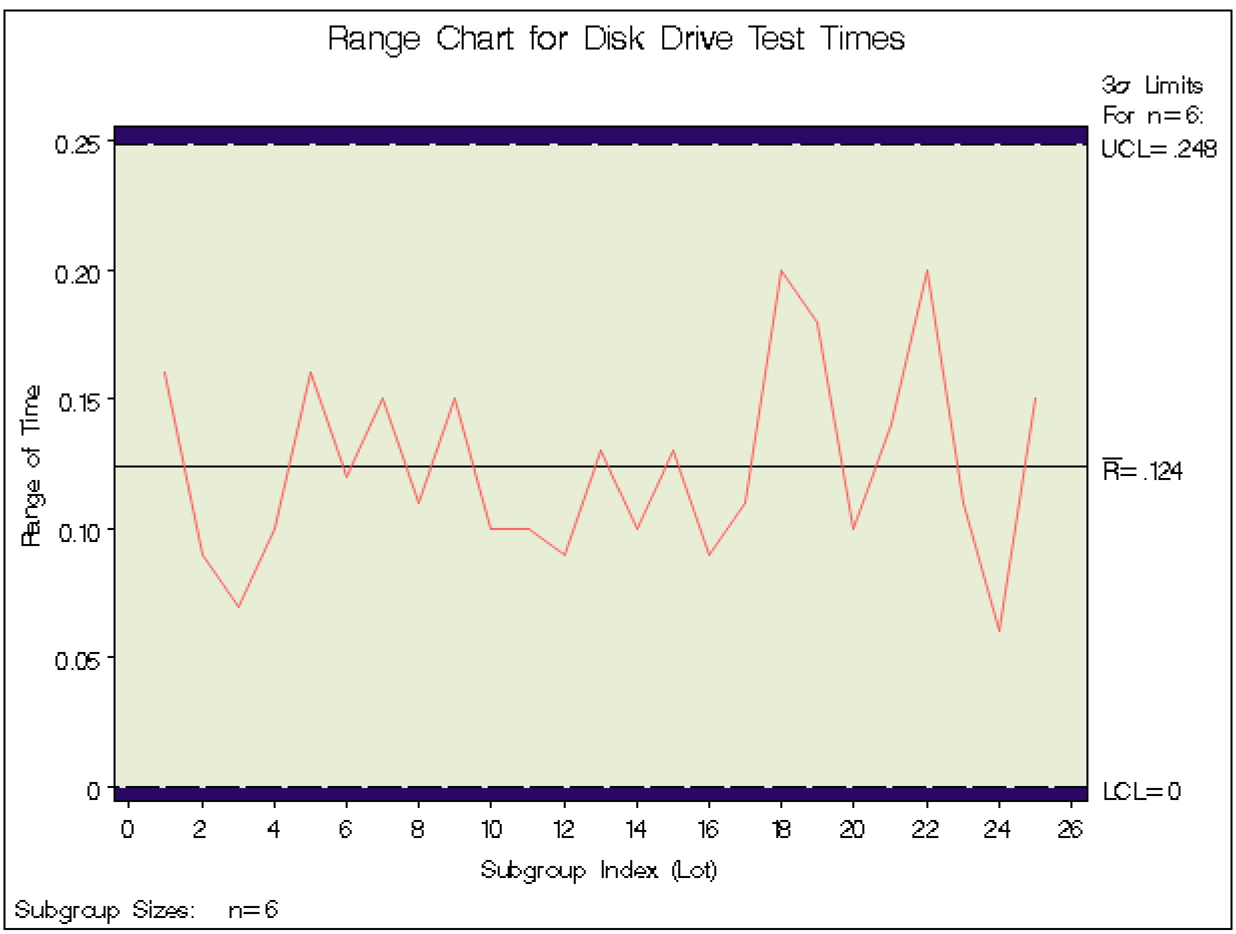
You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext=swiss;
symbol color = rose h = .8;
title 'Range Chart for Disk Drive Test Times';
proc shewhart history=Disksum;
   rchart Time*Lot / cframe   = vipb
                     cinfill  = ywh
                     cconnect = rose;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement and RCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting *R* chart is shown in Figure 17.73.

Note that Time is *not* the name of a SAS variable in the data set Disksum but is, instead, the common prefix for the names of the SAS variables TimeR and TimeN. The suffix characters *R* and *N* indicate *range* and *sample size*, respectively. Thus, you can specify two subgroup summary variables in the HISTORY= data set with a single name (Time), which is referred to as the *process*. The name Lot specified after the asterisk is the name of the *subgroup-variable*.

Figure 17.73 *R* Chart from the Summary Data Set Disksum (Traditional Graphics with NOGSTYLE)



In general, a HISTORY= input data set used with the RCHART statement must contain the following variables:

- subgroup variable
- subgroup range variable
- subgroup sample size variable

Furthermore, the names of the subgroup range and sample size variables must begin with the *process* name specified in the RCHART statement and end with the special suffix characters *R* and *N*, respectively. If the names do not follow this convention, you can use the RENAME optionin the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see page 1843).

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see “HISTORY= Data Set” on page 1712.

Saving Summary Statistics

NOTE: See *Range Chart (R Chart) Examples* in the SAS/QC Sample Library.

In this example, the RCHART statement procedure is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Disks and create a summary data set named Diskhist:

```
proc shewhart data=Disks;
   rchart Time*Lot / outhistory = Diskhist
   nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 17.71. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the RCHART statement. A complete list of options is presented in the section “Syntax: RCHART Statement” on page 1696.

Figure 17.74 contains a partial listing of Diskhist.

Figure 17.74 The Summary Data Set Diskhist

Summary Data Set for Disk Times

Lot	TimeX	TimeR	TimeN
1	8.00833	0.16	6
2	8.02167	0.09	6
3	7.97833	0.07	6
4	8.00667	0.10	6
5	8.01833	0.16	6

There are four variables in the data set Diskhist.

- Lot contains the subgroup index.
- TimeX contains the subgroup means.
- TimeR contains the subgroup ranges.
- TimeN contains the subgroup sample sizes.

The subgroup mean variable is included in the OUTHISTORY= data set even though it is not required by the RCHART statement. This enables the data set to be used as a HISTORY= data set with the BOXCHART, XCHART, and XRCHART statements, as well as with the RCHART statement. Note that the summary statistic variables are named by adding the suffix characters *X*, *R*, and *N* to the *process* Time specified in the RCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1708.

Saving Control Limits

NOTE: See *Range Chart (R Chart) Examples* in the SAS/QC Sample Library.

You can save the control limits for an *R* chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1694) or modify the limits with a DATA step program.

The following statements read measurements from the data set Disks (see "Creating Range Charts from Raw Data" on page 1685) and save the control limits displayed in Figure 17.71 in a data set named Disklim:

```
title 'Control Limits for Disk Times';
proc shewhart data=Disks;
   rchart Time*Lot / outlimits = Disklim
                     nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Disklim is listed in Figure 17.75.

Figure 17.75 The Data Set Disklim Containing Control Limit Information

Control Limits for Disk Times

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
Time	Lot	ESTIMATE	6	.004447667	3	7.94314	8.00307	8.06299

LCLR	_R_	_UCLR_	_STDDEV_
0	0.124	0.24847	0.048927

The data set Disklim contains one observation with the limits for *process* Time. The variables _LCLR_ and _UCLR_ contain the lower and upper control limits, and the variable _R_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. The variables _LCLX_ and _UCLX_, which contain the lower and upper control limits for subgroup means, are included so that the data set Disklim can be used to create an $\bar{X}$ chart (see "XRCHART Statement: SHEWHART Procedure" on page 1837). For more information, see "OUTLIMITS= Data Set" on page 1707.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
title 'Summary Statistics and Control Limit Information';
proc shewhart data=Disks;
  rchart Time*Lot / outtable=Disktab
                    nochart;
run;
```

The data set Disktab is listed in Figure 17.76.

Figure 17.76 The Data Set Disktab

Summary Statistics and Control Limit Information

VAR	Lot	_SIGMAS_	_LIMITN_	_SUBN_	_LCLR_	_SUBR_	_R_	_UCLR_	_STDDEV_	_EXLIM_
Time	1	3	6	6	0	0.16	0.124	0.24847	0.048927	
Time	2	3	6	6	0	0.09	0.124	0.24847	0.048927	
Time	3	3	6	6	0	0.07	0.124	0.24847	0.048927	
Time	4	3	6	6	0	0.10	0.124	0.24847	0.048927	
Time	5	3	6	6	0	0.16	0.124	0.24847	0.048927	
Time	6	3	6	6	0	0.12	0.124	0.24847	0.048927	
Time	7	3	6	6	0	0.15	0.124	0.24847	0.048927	
Time	8	3	6	6	0	0.11	0.124	0.24847	0.048927	
Time	9	3	6	6	0	0.15	0.124	0.24847	0.048927	
Time	10	3	6	6	0	0.10	0.124	0.24847	0.048927	
Time	11	3	6	6	0	0.10	0.124	0.24847	0.048927	
Time	12	3	6	6	0	0.09	0.124	0.24847	0.048927	
Time	13	3	6	6	0	0.13	0.124	0.24847	0.048927	
Time	14	3	6	6	0	0.10	0.124	0.24847	0.048927	
Time	15	3	6	6	0	0.13	0.124	0.24847	0.048927	
Time	16	3	6	6	0	0.09	0.124	0.24847	0.048927	
Time	17	3	6	6	0	0.11	0.124	0.24847	0.048927	
Time	18	3	6	6	0	0.20	0.124	0.24847	0.048927	
Time	19	3	6	6	0	0.18	0.124	0.24847	0.048927	
Time	20	3	6	6	0	0.10	0.124	0.24847	0.048927	
Time	21	3	6	6	0	0.14	0.124	0.24847	0.048927	
Time	22	3	6	6	0	0.20	0.124	0.24847	0.048927	
Time	23	3	6	6	0	0.11	0.124	0.24847	0.048927	
Time	24	3	6	6	0	0.06	0.124	0.24847	0.048927	
Time	25	3	6	6	0	0.15	0.124	0.24847	0.048927	

This data set contains one observation for each subgroup sample. The variables _SUBR_ and _SUBN_ contain the subgroup ranges and subgroup sample sizes. The variables _LCLR_ and _UCLR_ contain the lower and upper control limits, and the variable _R_ contains the central line. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1709. An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Disktab and display an *R* chart (not shown here) identical to the chart in Figure 17.71:

```
title 'Range Chart for Disk Drive Test Times';
proc shewhart table=Disktab;
   rchart Time*Lot;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2096). For more information, see "TABLE= Data Set" on page 1713.

Reading Preestablished Control Limits

NOTE: See *Range Chart (R Chart) Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Disklim saved control limits computed from the measurements in Disks. This example shows how these limits can be applied to new data provided in the following data set:

```
data Disks2;
   input Lot @;
   do i=1 to 6;
      input Time @;
      output;
   end;
   drop i;
   datalines;
26 7.93 7.97 7.89 7.81 7.88 7.92
27 7.86 7.91 7.87 7.89 7.83 7.87
28 7.93 7.95 7.90 7.89 7.88 7.90
29 7.97 8.00 7.86 7.89 7.84 7.78
30 7.91 7.93 7.98 7.93 7.83 7.88
31 7.85 7.94 7.88 7.98 7.96 7.84
32 7.86 8.01 7.88 7.95 7.90 7.89
33 7.87 7.93 7.96 7.89 7.81 8.00
34 7.87 7.97 7.95 7.89 7.92 7.84
35 7.92 7.97 7.90 7.88 7.89 7.86
36 7.96 7.90 7.90 7.84 7.90 8.00
37 7.92 7.90 7.98 7.92 7.94 7.94
38 7.88 7.99 8.02 7.98 7.88 7.92
39 7.89 7.91 7.92 7.90 7.94 7.94
40 7.84 7.88 7.91 7.98 7.87 7.93
41 7.91 7.87 7.96 7.91 7.89 7.92
42 7.96 7.93 7.86 7.93 7.86 7.94
43 7.84 7.82 7.87 7.91 7.91 8.01
44 7.93 7.91 7.92 7.88 7.91 7.86
45 7.95 7.92 7.93 7.90 7.86 8.00
;
```

The following statements create an *R* chart using the control limits in Disklim:

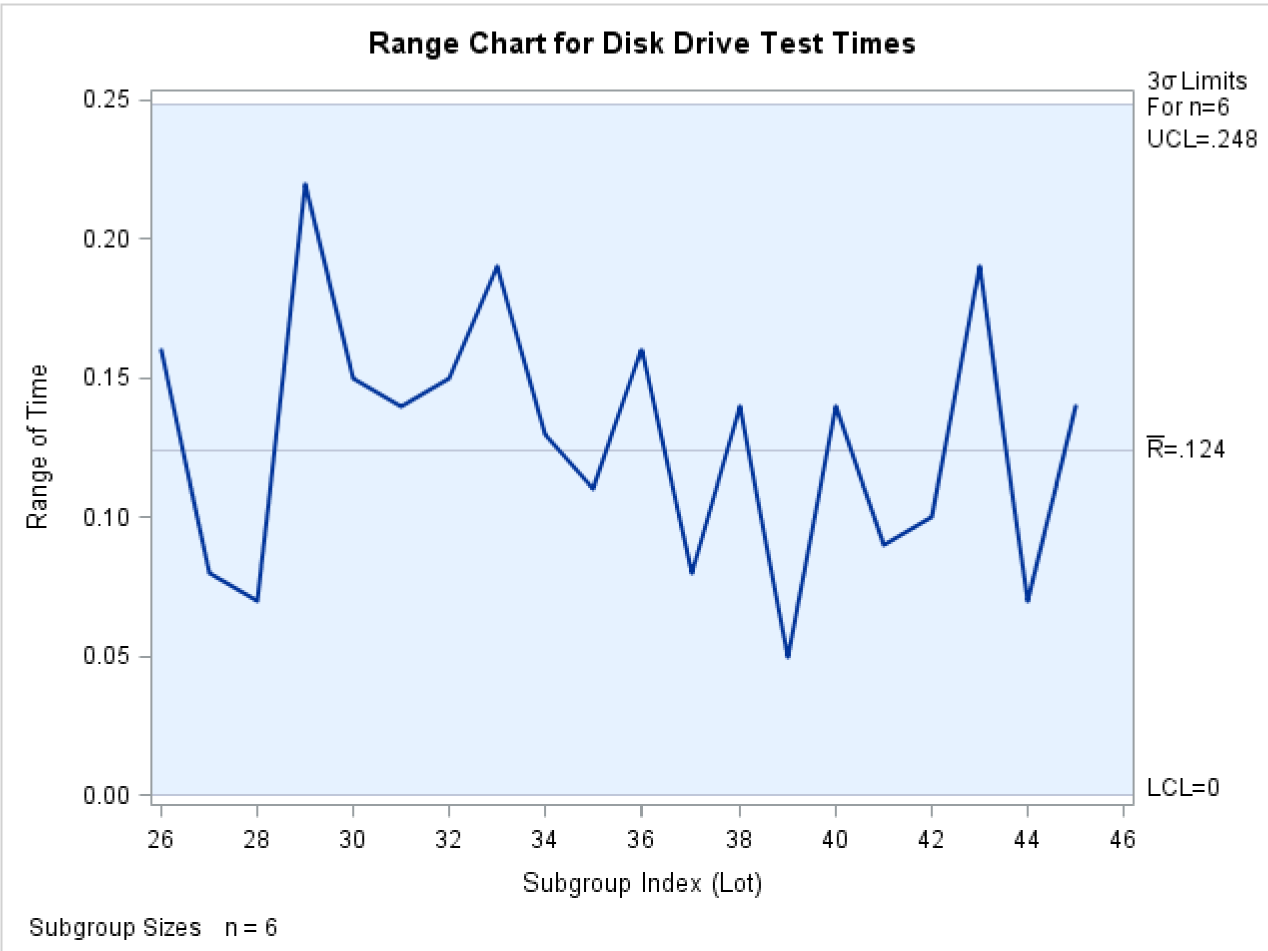
```
ods graphics on;
title 'Range Chart for Disk Drive Test Times';
proc shewhart data=Disks2 limits=Disklim;
   rchart Time*Lot / odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the *R* chart is created using ODS Graphics instead of traditional graphics. The chart is shown in Figure 17.77.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Time
- the value of _SUBGRP_ matches the *subgroup-variable* name Lot

Figure 17.77 *R* Chart for Second Set of Disk Drive Test Times (ODS Graphics)



All the ranges lie within the control limits, indicating that the variability in disk drive performance is still in statistical control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See Example 17.28 and "LIMITS= Data Set" on page 1711 for details concerning the variables that you must provide.

Syntax: RCHART Statement

The basic syntax for the RCHART statement is as follows:

RCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

RCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=**'*character*' > / < *options* > **;**

You can use any number of RCHART statements in the SHEWHART procedure. The components of the RCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Range Charts from Raw Data" on page 1685.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Range Charts from Summary Data" on page 1687.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1691.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct *R* charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   rchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding RCHART statement, Day is the subgroup variable. For details, see "Subgroup Variables" on page 1871.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2026 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the ranges.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2025 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create an *R* chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   rchart Weight*Day='*';
run;
```

options

enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options", which follows, lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1946 describes each option in detail.

Summary of Options

The following tables list the RCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1946.

Table 17.47 RCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	requests probability limits for chart
LIMITN=	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	specifies color for area inside control limits

Table 17.47 *continued*

Option	Description
CLIMITS=	specifies color of control limits, central line, and related labels
LCLLABEL=	specifies label for lower control limit
LIMLABSUBCHAR=	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	specifies line type for control limits
NDECIMAL=	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMIT0	suppresses display of zero lower control limit on *R* chart
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit
RSYMBOL=	specifies label for central line on *R* chart
UCLLABEL=	specifies label for upper control limit
WLIMITS=	specifies width for control limits and central line
Process Mean and Standard Deviation Options	
SIGMA0=	specifies known value σ_0 for process standard deviation σ
SMETHOD=	specifies method for estimating process standard deviation σ
TYPE=	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL2=	labels every point on *R* chart
CLABEL=	specifies color for labels
CCONNECT=	specifies color for line segments that connect points on chart
CFRAMELAB=	specifies fill color for frame around labeled points
CNEEDLES=	specifies color for needles that connect points to central line
COUT=	specifies color for portions of line segments that connect points outside control limits
COUTFILL=	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	specifies angle at which labels are drawn
LABELFONT=	specifies software font for labels (alias for the TESTFONT= option)

Table 17.47 *continued*

Option	Description
LABELHEIGHT=	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL2=	labels points outside control limits on *R* chart
SYMBOLLEGEND=	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=	specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	computes zone widths independently above and below center line
NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	suppresses tests across *phase* boundaries
TESTS2=	specifies tests for special causes for the *R* chart
TEST2RESET=	enables tests for special causes to be reset for the *R* chart
TEST2RUN=	specifies length of pattern for Test 2
TEST3RUN=	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL=	provides labels for points where test is positive
TESTLABEL*n*=	specifies label for *n*th test for special causes
TESTNMETHOD=	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=	enables tests for special causes to be reset
WESTGARD=	requests that Westgard rules be applied to the *R* chart
ZONE2LABELS	adds labels A, B, and C to zone lines for *R* chart
ZONES2	adds lines to *R* chart delineating zones A, B, and C
ZONEVALPOS=	specifies position of ZONEVALUES labels
ZONE2VALUES	labels *R* zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	specifies color for labels indicating points where test is positive
CTESTSYMBOL=	specifies color for symbol used to plot points where test is positive
CZONES=	specifies color for lines and labels delineating zones A, B, and C
LTESTS=	specifies type of line connecting points where test is positive
LZONES=	specifies line type for lines delineating zones A, B, and C

Table 17.47 *continued*

Option	Description
TESTFONT=	specifies software font for labels at points where test is positive
TESTHEIGHT=	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	specifies plot symbol for points where test is positive
TESTSYMBOLHT=	specifies symbol height for points where test is positive
WTESTS=	specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill colors for frame for plot area
CTEXT=	specifies color for tick mark values and axis labels
DISCRETE	produces horizontal axis for discrete numeric group values
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default to *R* chart
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis
VFORMAT=	specifies format for vertical axis tick mark labels
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=	specifies width of axis lines
Plot Layout Options	
ALLN	plots means for all subgroups
BILEVEL	creates control charts using half-screens and half-pages

Table 17.47 *continued*

Option	Description
EXCHART	creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	specifies number of pages or screens to be used to display chart
ZEROSTD	displays *R* chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= options
CVREF=	specifies color for lines requested by VREF= = options
HREF=	specifies position of reference lines perpendicular to horizontal axis
HREFDATA=	specifies position of reference lines perpendicular to horizontal axis
HREFLABELS=	specifies labels for HREF= lines
HREFLABPOS=	specifies position of HREFLABELS= labels
LHREF=	specifies line type for HREF= lines
LVREF=	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	specifies position of reference lines perpendicular to vertical axis
VREFLABELS=	specifies labels for VREF= lines
VREFLABPOS=	position of VREFLABELS= labels
Grid Options	
CGRID=	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=	specifies line type for grid requested with the ENDGRID option

Table 17.47 *continued*

Option	Description
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme points are clipped
CLIPLEGEND=	specifies text for clipping legend
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features to chart
DESCRIPTION=	specifies description of *R* chart's GRSEG catalog entry
FONT=	specifies software font for labels and legends on charts
NAME=	specifies name of *R* chart's GRSEG catalog entry
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	shades areas between control limits and connected points lying outside the limits
STARFILL=	specifies a variable identfying groups of stars filled with different colors
STARS=	specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	specifies the control limit infill transparency
MARKERS	plots subgroup points with markers
NOBLOCKREF	suppresses block and phase reference lines
NOBLOCKREFFILL	suppresses block and phase wall fills
NOFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOPHASEREF	suppresses block and phase reference lines
NOPHASEREFFILL	suppresses block and phase wall fills

Table 17.47 *continued*

Option	Description
NOREF	suppresses block and phase reference lines
NOREFFILL	suppresses block and phase wall fills
NOSTARFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	specifies a graph footnote
ODSFOOTNOTE2=	specifies a secondary graph footnote
ODSLEGENDEXPAND	specifies that legend entries contain all levels observed in the data
ODSTITLE=	specifies a graph title
ODSTITLE2=	specifies a secondary graph title
OUTFILLTRANSPARENCY=	specifies control limit outfill transparency
OVERLAYURL=	specifies URLs to associate with overlay points
PHASEPOS=	specifies vertical position of phase legend
PHASEREFLEVEL=	associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	draws central line labels using a simulated software font
STARTRANSPARENCY=	specifies star fill transparency
URL=	specifies a variable whose values are URLs to be associated with subgroups
Input Data Set Options	
MISSBREAK	specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	creates output data set containing subgroup summary statistics
OUTINDEX=	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	creates output data set containing control limits
OUTTABLE=	creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded

Table 17.47 *continued*

Option	Description
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	specifies α value and type for computing capability index confidence limits
LSL=	specifies list of lower specification limits
TARGET=	specifies list of target values
USL=	specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for *block-variable* legend
BLOCKLABTYPE=	specifies text size of *block-variable* legend
BLOCKPOS=	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE=	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	specifies color for STARCIRCLES= circles
CSTARFILL=	specifies color for filling stars
CSTAROUT=	specifies outline color for stars exceeding inner or outer circles
CSTARS=	specifies color for outlines of stars
LSTARCIRCLES=	specifies line types for STARCIRCLES= circles
LSTARS=	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	specifies reference circles for stars
STARINRADIUS=	specifies inner radius of stars
STARLABEL=	specifies vertices to be labeled

Table 17.47 *continued*

Option	Description
STARLEGEND=	specifies style of legend for star vertices
STARLEGENDLAB=	specifies label for STARLEGEND= legend
STAROUTRADIUS=	specifies outer radius of stars
STARSPECS=	specifies method used to standardize vertex variables
STARSTART=	specifies angle for first vertex
STARTYPE=	specifies graphical style of star
STARVERTICES=	superimposes star at each point on chart
WSTARCIRCLES=	specifies width of STARCIRCLES= circles
WSTARS=	specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	specifies colors for overlay line segments
COVERLAY=	specifies colors for overlay plots
COVERLAYCLIP=	specifies color for clipped points on overlays
LOVERLAY=	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=	specifies variables to overlay on chart
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
OVERLAYHTML=	specifies links to associate with overlay points
OVERLAYID=	specifies labels for overlay points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols for overlays
OVERLAYSYMHT=	specifies symbol heights for overlays
WOVERLAY=	specifies widths of overlay line segments
Options for Interactive Control Charts	
HTML=	specifies a variable whose values create links to be associated with subgroups
HTML_LEGEND=	specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	specifies plot character for clipped points
CONNECTCHAR=	specifies character used to form line segments that connect points on chart
HREFCHAR=	specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	specifies characters indicating *symbol-variable*
TESTCHAR=	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	specifies line character for VREF= and VREF2= lines
ZONECHAR=	specifies character for lines that delineate zones for tests for special causes

Details: RCHART Statement

Constructing Range Charts

The following notation is used in this section:

σ	process standard deviation (standard deviation of the population of measurements)
R_i	range of measurements in ith subgroup
n_i	sample size of ith subgroup
$d_2(n)$	expected value of the range of n independent normally distributed variables with unit standard deviation
$d_3(n)$	standard error of the range of n independent observations from a normal population with unit standard deviation
$D_p(n)$	100pth percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on an R chart indicates the value of a subgroup range (R_i). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the value plotted for this subgroup is $R_{10} = 19 - 12 = 7$.

Central Line

By default, the central line for the ith subgroup indicates an estimate of the expected value of R_i, which is computed as $d_2(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $d_2(n_i)\sigma_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of R_i above and below the central line. The default limits are computed with k = 3 (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that R_i exceeds the limits

The following table provides the formulas for the limits:

Table 17.48 Limits for R Charts

Control Limits
LCL = lower limit = $\max(d_2(n_i)\hat{\sigma} - kd_3(n_i)\hat{\sigma}, 0)$ UCL = upper limit = $d_2(n_i)\hat{\sigma} + kd_3(n_i)\hat{\sigma}$
Probability Limits
LCL = lower limit = $D_{\alpha/2}\hat{\sigma}$ UCL = upper limit = $D_{1-\alpha/2}\hat{\sigma}$

The formulas assume that the data are normally distributed. Note that the control limits vary with n_i and that the probability limits for R_i are asymmetric around the central line. If a standard value σ_0 is available for σ, replace $\hat{\sigma}$ with σ_0 in Table 17.48.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables are saved:

Table 17.49 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits with the OUTINDEX= option
LCLR	lower control limit for subgroup range
LCLX	lower control limit for subgroup mean
LIMITN	sample size associated with the control limits
LSL	lower specification limit
MEAN	process mean ($\overline{\overline{X}}$)
_R_	value of central line on R chart
SIGMAS	multiple (k) of standard error of R_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the RCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLR	upper control limit for subgroup range
UCLX	upper control limit for subgroup mean
USL	upper specification limit
VAR	*process* specified in the RCHART statement

Notes:

1. The variables _LCLX_, _MEAN_, and _UCLX_ are saved to enable the OUTLIMITS= data set to be used as a LIMITS= data set with the BOXCHART, XCHART, and XRCHART statements.

2. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLR_, _R_, and _UCLR_.

3. If the limits are defined in terms of a multiple k of the standard error of R_i, the value of _ALPHA_ is computed as

   $$F_R(\_LCLR\_/\_STDDEV\_) + 1 - F_R(\_UCLR\_/\_STDDEV\_)$$

   where $F_R(\cdot)$ is the cumulative distribution function of the range of a sample of n observations from a normal population with unit standard deviation, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _ALPHA_.

4. If the limits are probability limits, the value of _SIGMAS_ is computed as $(\_UCLR\_ - \_R\_)/e$, where e is the standard error of the range of n observations from a normal population with unit standard deviation. If _LIMITN_ has the special missing value *V*, this value is assigned to _SIGMAS_.

5. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1874 for computational details.

6. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the RCHART statement. For an example, see "Saving Control Limits" on page 1691.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup sample size variable named by *process* suffixed with *N*

The subgroup mean variable is saved so that the data set can be reused as a HISTORY= data set with the BOXCHART, XCHART, and XRCHART statements, as well as the RCHART statement.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the RCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   rchart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthX, WidthR, WidthN, DiameterX, DiameterR, and DiameterN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1690.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on *R* chart
LCLR	lower control limit for range
LIMITN	nominal sample size associated with the control limits
_R_	average range
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBN	subgroup sample sizes
SUBR	subgroup range
TESTS2	tests for special causes signaled on *R* chart
UCLR	upper control limit for range
VAR	*process* specified in the RCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= option, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS2_ is saved if you specify the TESTS2= option.
3. The variables _EXLIM_ and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1691.

ODS Tables

The following table summarizes the ODS tables that you can request with the RCHART statement.

Table 17.50 ODS Tables Produced with the RCHART Statement

Table Name	Description	Options
RCHART	*R* chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*). **NOTE:** In SAS/QC 13.1 the SHEWHART procedure does not support the creation of graphs that are editable with the ODS Graphics Editor.

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. RCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1946.

When ODS Graphics is in effect, the RCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 17.51.

Table 17.51 ODS Graphics Produced by the RCHART Statement

ODS Graph Name	Plot Description
RChart	*R* chart

See Chapter 3, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the RCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped in subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the RCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a raw measurement for each *process* and a value for the *subgroup-variable*. If the *i*th subgroup contains n_i items, there should be n_i consecutive observations for which the value of the subgroup variable is the index of the *i*th subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2031.

For an example of a DATA= data set, see "Creating Range Charts from Raw Data" on page 1685.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   rchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 17.49. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLR_, _R_, and _UCLR_, which specify the control limits directly
- the variable _STDDEV_, which is used to calculate the control limits according to the equations in Table 17.48

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1694.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to read output data sets created with SAS summarization procedures, such as the MEANS procedure.

A HISTORY= data set used with the RCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup range variable for each *process*
- a subgroup sample size variable for each *process*

The names of the subgroup range and subgroup sample size variables must be the prefix *process* concatenated with the special suffix characters *R* and *N*, respectively.

For example, consider the following statements:

```
proc shewhart history=Summary;
   rchart (Weight Yieldstrength)*Batch;
run;
```

The data set Summary must include the variables Batch, WeightR, WeightN, YieldstrengthR, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as phases) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2031 for an example).

For an example of a HISTORY= data set, see "Creating Range Charts from Summary Data" on page 1687.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2096.

The following table lists the variables required in a TABLE= data set used with the RCHART statement:

Table 17.52 Variables Required in a TABLE= Data Set

Variable	Description
LCLR	lower control limit for range
LIMITN	nominal sample size associated with the control limits
_R_	average range
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBR	subgroup range
UCLR	upper control limit for range

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1691.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, two methods (referred to as default and MVLUE) are available for estimating σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the ith subgroup.

$$R_i = \max_{1 \leq j \leq n_i} (x_{ij}) - \min_{1 \leq j \leq n_i} (x_{ij})$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The unbiasing factor $d_3(n_i)$ is defined so that, if the observations are normally distributed, the expected value of σ_{R_i} is $d_3(n_i)\sigma$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup range variable
Vertical	TABLE=	_SUBR_

For an example, see "Labeling Axes" on page 2063.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: RCHART Statement

This section provides advanced examples of the RCHART statement.

Example 17.27: Computing Probability Limits

NOTE: See *An R Chart with Probability Limits* in the SAS/QC Sample Library.

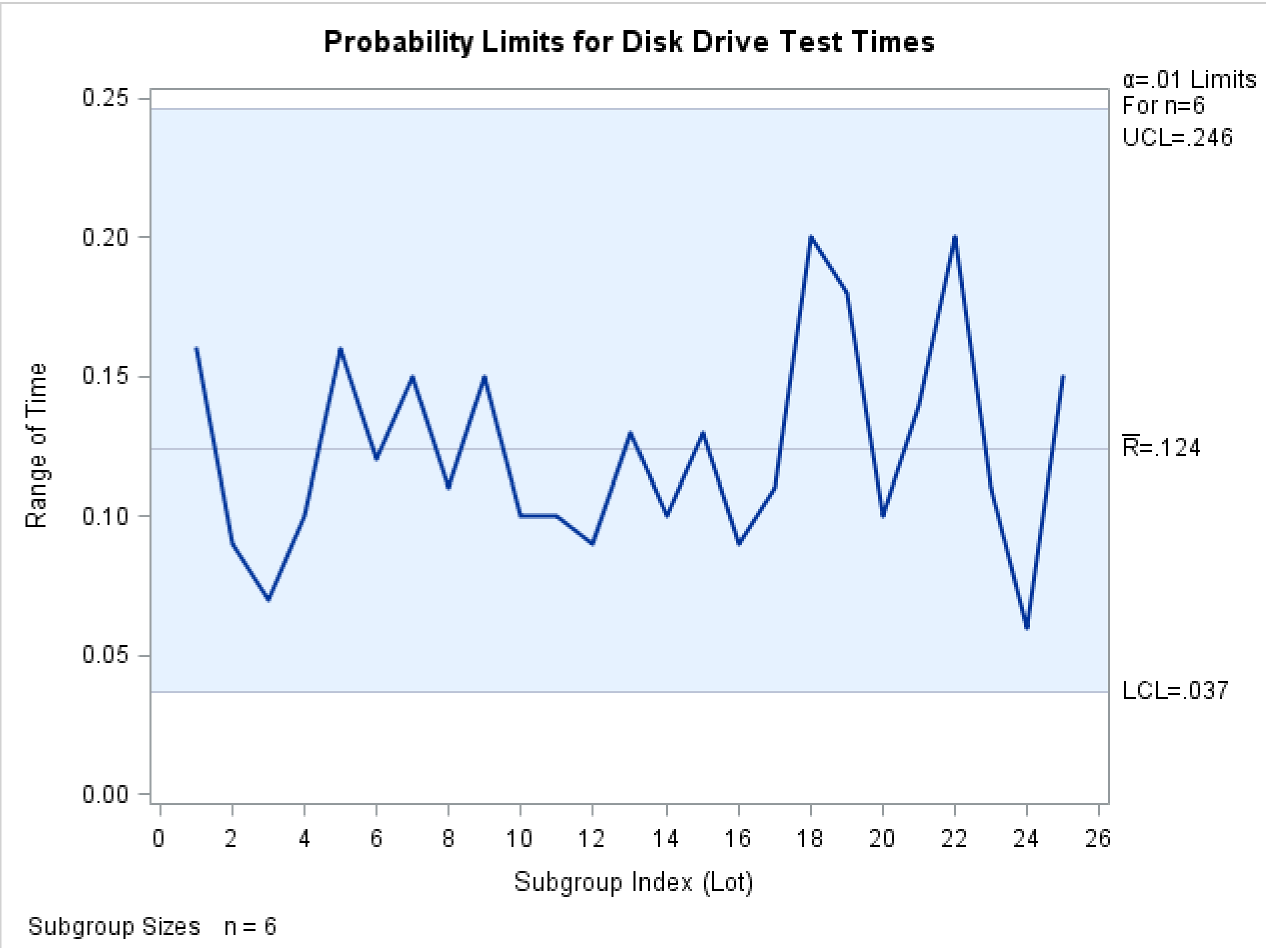
This example demonstrates how to create R charts with probability limits. The following statements read the disk drive test times from the data set Disks (see "Creating Range Charts from Raw Data" on page 1685) and create the R chart shown in Output 17.27.1:

```
ods graphics on;
title 'Probability Limits for Disk Drive Test Times';
proc shewhart data=Disks;
   rchart Time*Lot / alpha     = .01
                     outlimits = Dlimits
                     odstitle  = title;
run;
```

The ALPHA= option specifies the probability (α) that a subgroup range exceeds its limits. Here, the limits are computed so that the probability that a range is less than the lower limit is $\alpha/2 = 0.005$, and the probability that a range is greater than the upper limit is $\alpha/2 = 0.005$. This assumes that the measurements are normally distributed. The OUTLIMITS= option names an output data set that saves the probability limits. A listing of Dlimits is shown in Output 17.27.2.

The variable _ALPHA_ saves the value of α. Note that, in this case, the upper probability limit is equivalent to an upper 2.95σ limit.

Output 17.27.1 *R* Chart with Probability Limits



Output 17.27.2 Probability Limits Data Set

Probablity Limits for Disk Drive Test Times

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
Time	Lot	ESTIMATE	6	0.01	2.94688	7.95162	8.00307	8.05452

LCLR	_R_	_UCLR_	_STDDEV_
0.036645	0.124	0.24627	0.048927

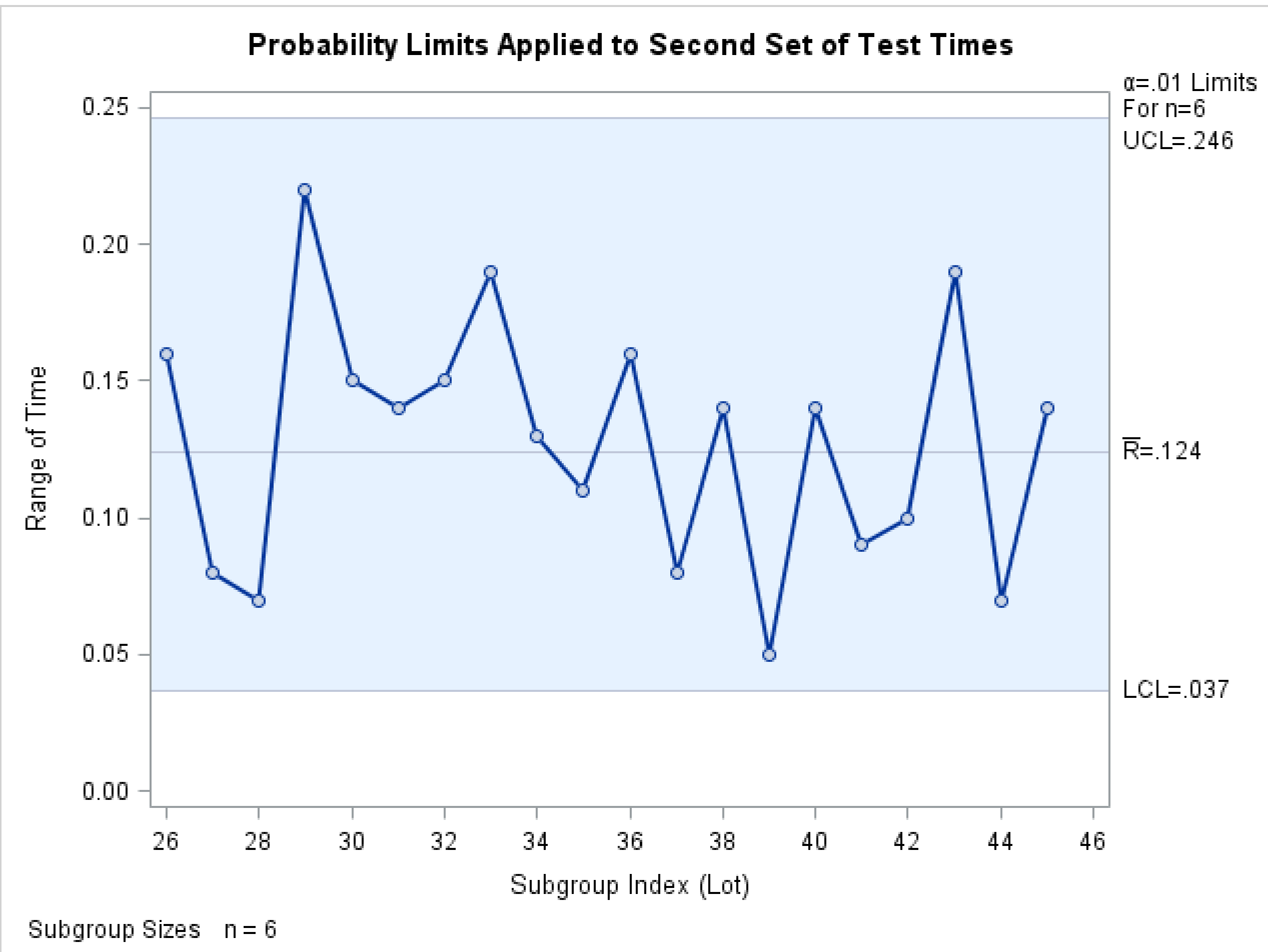
Since all the points fall within the probability limits, it can be concluded that the variability in the disk drive performance is in statistical control.

The following statements apply the limits in Dlimits to the times in the data set Disks2 (see "Reading Preestablished Control Limits" on page 1694):

```
title 'Probability Limits Applied to Second Set of Test Times';
proc shewhart data=Disks2 limits=Dlimits;
   rchart Time*Lot / readalpha
                     markers
                     odstitle = title;
run;
```

The READALPHA option specifies that the variable _ALPHA_, rather than the variable _SIGMAS_, is to be read from the LIMITS= data set. Thus the limits displayed in the chart, shown in Output 17.27.3, are probability limits.

Output 17.27.3 Reading Probability Limits from a LIMITS= Data Set



Example 17.28: Specifying Control Limit Information

NOTE: See *Specifying Control Limit Info for R Chart* in the SAS/QC Sample Library.

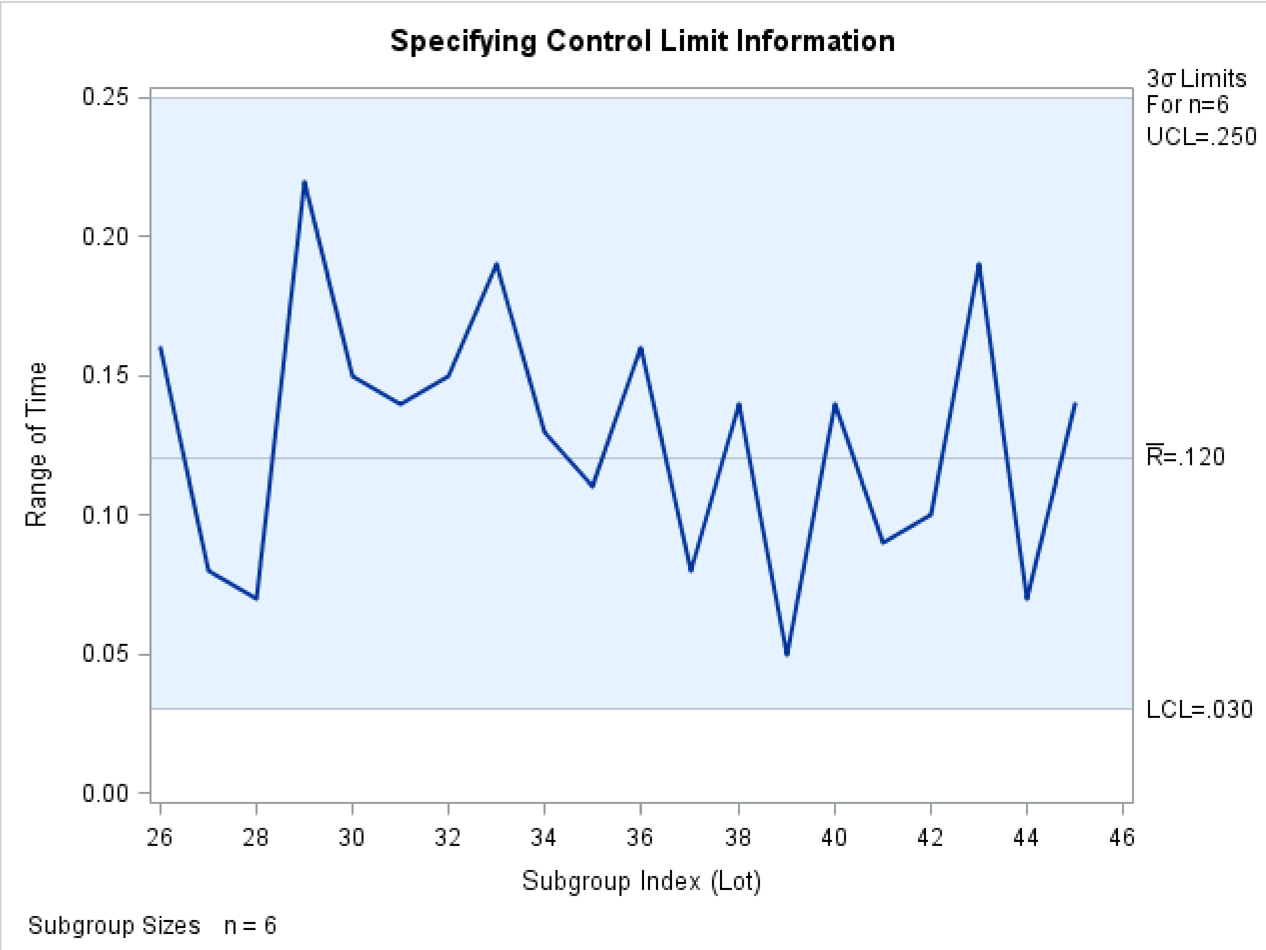
This example illustrates how you can use a DATA step program to create a LIMITS= data set. You can provide previously established values for the limits and central line with the variables _LCLR_, _R_, and _UCLR_, as in the following statements:

```
data Dlimits2;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'Time';
   _subgrp_ = 'Lot';
   _type_   = 'STANDARD';
   _limitn_ = 6;
   _lclr_   = .03;
   _r_      = .12;
   _uclr_   = .25;
run;
```

The following statements apply the control limits in Dlimits2 to the measurements in DISKS2 (see "Reading Preestablished Control Limits" on page 1694) and create the *R* chart shown in Output 17.28.1:

```
ods graphics on;
title 'Specifying Control Limit Information';
proc shewhart data=Disks2 limits=Dlimits2;
   rchart Time*Lot / odstitle=title;
run;
```

Output 17.28.1 Reading Control Limits from Dlimits2



In some cases, a standard value (σ_0) may be available for the process standard deviation. The following DATA step creates a data set named Dlimits3 that provides this value:

```
data Dlimits3;
   length _var_ _subgrp_ _type_ $8;
   _var_     = 'Time';
   _subgrp_  = 'Lot';
   _stddev_  = .045;
   _limitn_  = 6;
   _type_    = 'STDSIGMA';
run;
```

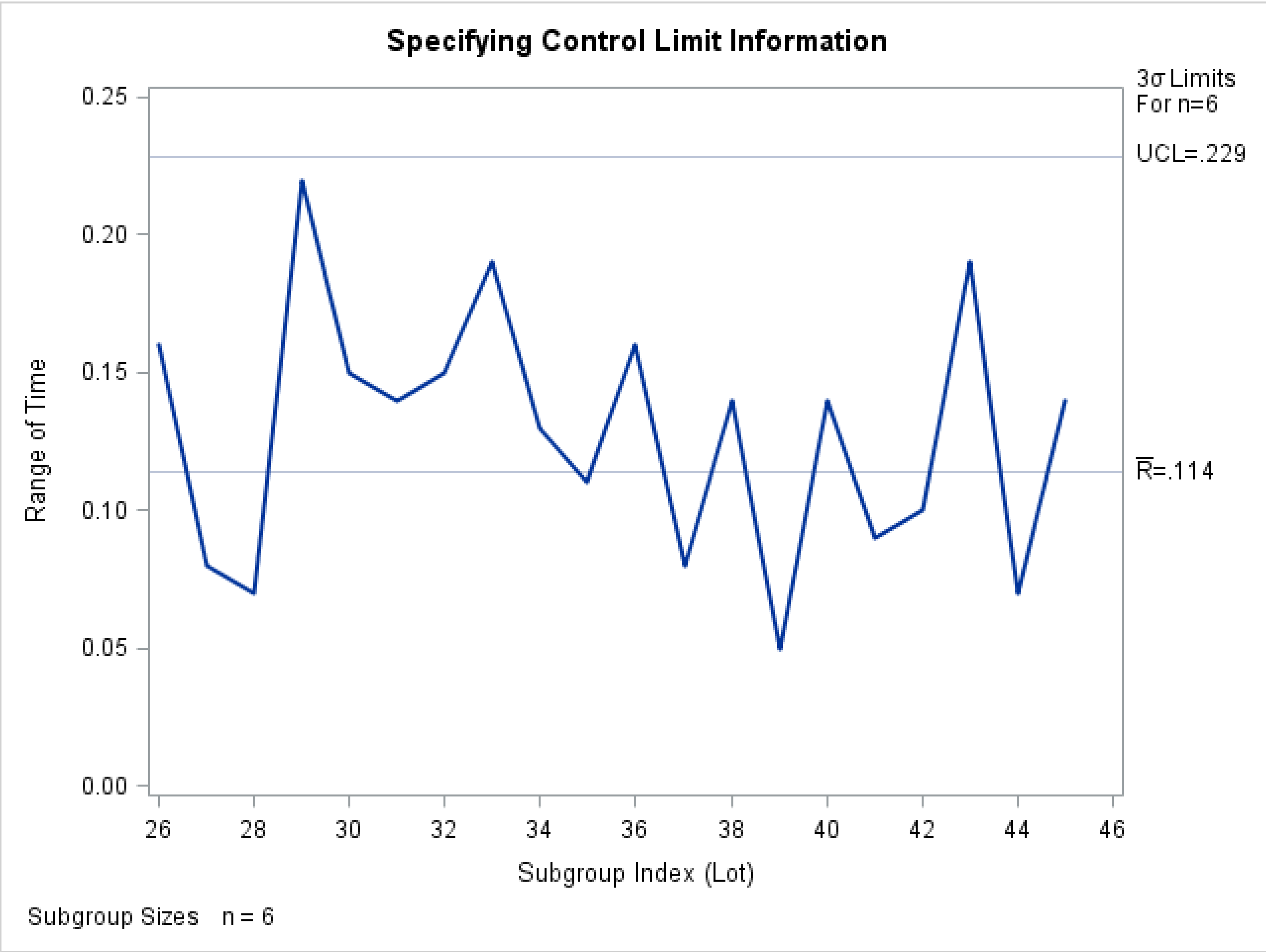
The variable _TYPE_ is a bookkeeping variable whose value indicates that the value of _STDDEV_ is a standard value rather than an estimate.

The following statements read the value of σ_0 from Dlimits3 and create the *R* chart shown in Output 17.28.2:

```
title 'Specifying Control Limit Information';
proc shewhart data=Disks2 limits=Dlimits3;
   rchart Time*Lot / nolimit0 odstitle=title;
run;
```

The NOLIMIT0 option suppresses the display of a fixed lower control limit if the value of the limit is zero (which is the case in this example).

Output 17.28.2 Reading in Standard Value for Process Standard Deviation



Instead of specifying σ_0 with the variable _STDDEV_ in a LIMITS= data set, you can use the SIGMA0= option in the RCHART statement. The following statements create an *R* chart identical to the chart shown in Output 17.28.2:

```
proc shewhart data=Disks;
   rchart Time*Lot / sigma0=.045 nolimit0;
run;
```

For more information, see “LIMITS= Data Set” on page 1711.

SCHART Statement: SHEWHART Procedure

Overview: SCHART Statement

The SCHART statement creates an *s* chart for subgroup standard deviations, which is used to analyze the variability of a process.[8]

You can use options in the SCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted standard deviations or as probability limits
- tabulate subgroup sample sizes, subgroup standard deviations, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes, subgroup means, and subgroup standard deviations in an output data set
- read preestablished control limits from a data set
- specify a method for estimating the process standard deviation
- specify a known (standard) process standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing *s* charts with the SCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 3, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

[8] You can also use *R* charts for this purpose; see "RCHART Statement: SHEWHART Procedure" on page 1683. In general, *s* charts are recommended with large subgroup sample sizes ($n_i \geq 10$).

Getting Started: SCHART Statement

This section introduces the SCHART statement with simple examples that illustrate commonly used options. Complete syntax for the SCHART statement is presented in the section "Syntax: SCHART Statement" on page 1732, and advanced examples are given in the section "Examples: SCHART Statement" on page 1752.

Creating Standard Deviation Charts from Raw Data

NOTE: See *Standard Deviation Chart (s Chart) Example* in the SAS/QC Sample Library.

A petroleum company uses a turbine to heat water into steam, which is then pumped into the ground to make oil less viscous and easier to extract. This heating process occurs 20 times daily, and the amount of power (in kilowatts) used to heat the water to the desired temperature is recorded. The following statements create a SAS data set named Turbine, which contains the power output measurements for 20 days:

```
data Turbine;
   informat Day date7.;
   format Day date5.;
   input Day @;
   do i=1 to 10;
      input KWatts @;
      output;
   end;
   drop i;
   datalines;
04JUL94 3196 3507 4050 3215 3583 3617 3789 3180 3505 3454
04JUL94 3417 3199 3613 3384 3475 3316 3556 3607 3364 3721
05JUL94 3390 3562 3413 3193 3635 3179 3348 3199 3413 3562
05JUL94 3428 3320 3745 3426 3849 3256 3841 3575 3752 3347
06JUL94 3478 3465 3445 3383 3684 3304 3398 3578 3348 3369
06JUL94 3670 3614 3307 3595 3448 3304 3385 3499 3781 3711
07JUL94 3448 3045 3446 3620 3466 3533 3590 3070 3499 3457

   ... more lines ...

23JUL94 3756 3145 3571 3331 3725 3605 3547 3421 3257 3574
;
```

A partial listing of Turbine is shown in Figure 17.78.

Figure 17.78 Partial Listing of the Data Set Turbine

Kilowatt Power Output Data

Obs	Day	KWatts
1	04JUL	3196
2	04JUL	3507
3	04JUL	4050
4	04JUL	3215
5	04JUL	3583
6	04JUL	3617
7	04JUL	3789
8	04JUL	3180
9	04JUL	3505
10	04JUL	3454
11	04JUL	3417
12	04JUL	3199
13	04JUL	3613
14	04JUL	3384
15	04JUL	3475
16	04JUL	3316
17	04JUL	3556
18	04JUL	3607
19	04JUL	3364
20	04JUL	3721
21	05JUL	3390
22	05JUL	3562
23	05JUL	3413
24	05JUL	3193
25	05JUL	3635

The data set Turbine is said to be in "strung-out" form, since each observation contains the day and power output for a single heating. The first 20 observations contain the power outputs for the first day, the second 20 observations contain the power outputs for the second day, and so on. Because the variable Day classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable KWatts contains the power output measurements and is referred to as the *process variable* (or *process* for short).

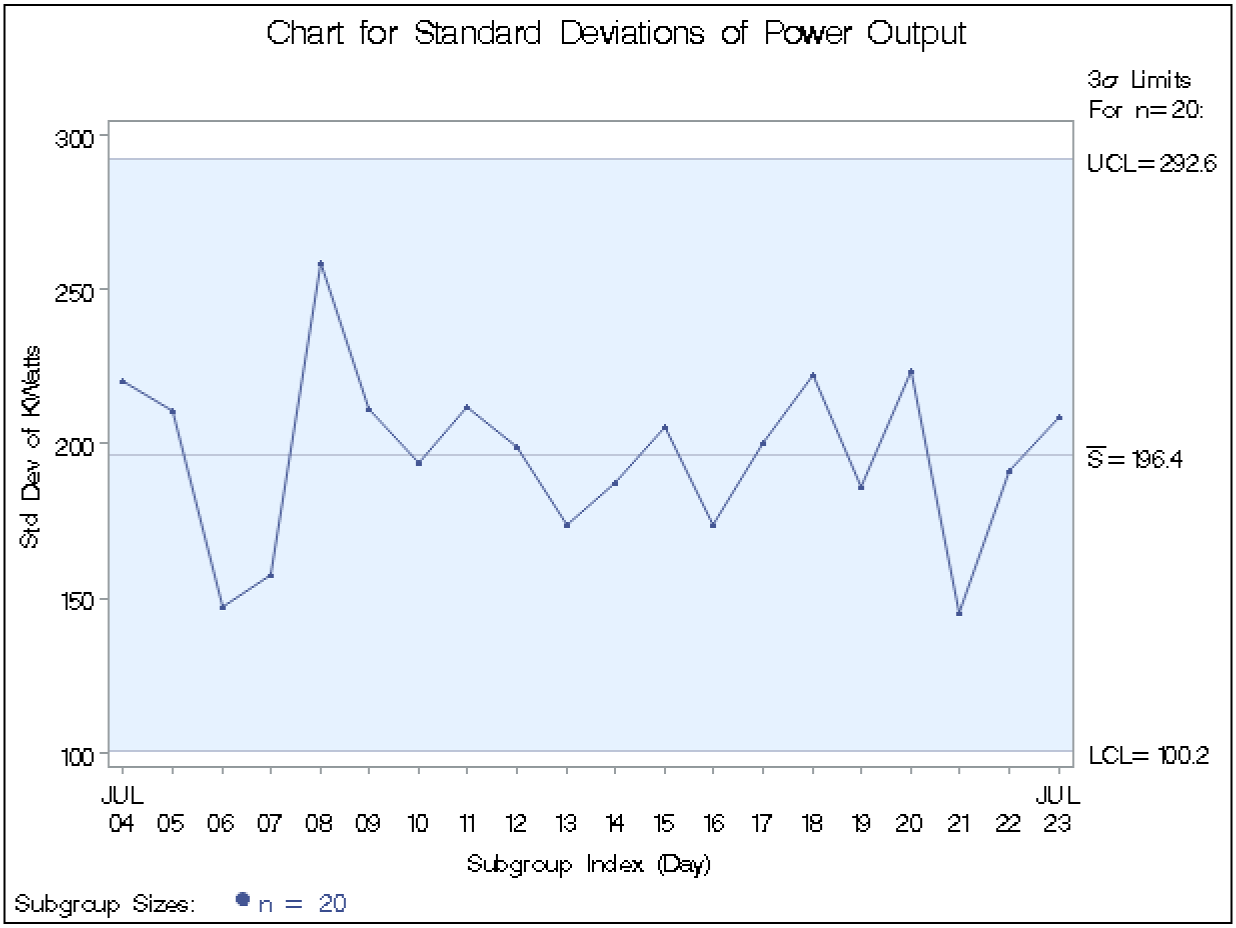
You can use an *s* chart to determine whether the variability in the heating process is in control. The following statements create the *s* chart shown in Figure 17.79:

```
ods graphics off;
symbol v=dot h=.8;
title 'Chart for Standard Deviations of Power Output';
proc shewhart data=Turbine;
   schart KWatts*Day;
run;
```

This example illustrates the basic form of the SCHART statement. After the keyword SCHART, you specify the *process* to analyze (in this case, KWatts), followed by an asterisk and the *subgroup-variable* (Day).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Figure 17.79 s Chart for Power Output Data (Traditional Graphics)



Each point on the chart represents the standard deviation of the measurements for a particular day. For instance, the standard deviation plotted for the first day is

$$\sqrt{\frac{(3196-3487.4)^2+(3507-3487.4)^2+\cdots+(3721-3487.4)^2}{19}} = 220.26$$

Since all of the subgroup standard deviations lie within the control limits, you can conclude that the variability of the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 17.54. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1730.

For computational details, see "Constructing Charts for Standard Deviations" on page 1742. For more details on reading raw data, see "DATA= Data Set" on page 1748.

Creating Standard Deviation Charts from Subgroup Summary Data

NOTE: See *Standard Deviation Chart (s Chart) Example* in the SAS/QC Sample Library.

The previous example illustrates how you can create *s* charts using raw data (process measurements). However, in many applications, the data are provided as subgroup summary statistics. This example illustrates how you can use the SCHART statement with data of this type.

The following data set (Oilsum) provides the data from the preceding example in summarized form:

```
data Oilsum;
   input Day KWattsX KWattsS KWattsN;
   informat Day date7. ;
   format Day date5. ;
   label Day   ='Date of Measurement';
   datalines;
 04JUL94 3487.40 220.260 20
 05JUL94 3471.65 210.427 20
 06JUL94 3488.30 147.025 20
 07JUL94 3434.20 157.637 20
 08JUL94 3475.80 258.949 20
 09JUL94 3518.10 211.566 20
 10JUL94 3492.65 193.779 20
 11JUL94 3496.40 212.024 20
 12JUL94 3398.50 199.201 20
 13JUL94 3456.05 173.455 20
 14JUL94 3493.60 187.465 20
 15JUL94 3563.30 205.472 20
 16JUL94 3519.05 173.676 20
 17JUL94 3474.20 200.576 20
 18JUL94 3443.60 222.084 20
 19JUL94 3586.35 185.724 20
 20JUL94 3486.45 223.474 20
 21JUL94 3492.90 145.267 20
 22JUL94 3432.80 190.994 20
 23JUL94 3496.90 208.858 20
;
```

A partial listing of Oilsum is shown in Figure 17.80. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Day). The variable KWattsX contains the subgroup means, the variable KWattsS contains the subgroup standard deviations, and the variable KWattsN contains the subgroup sample sizes (these are all 20).

Figure 17.80 The Summary Data Set Oilsum

Summary Data Set for Power Outputs

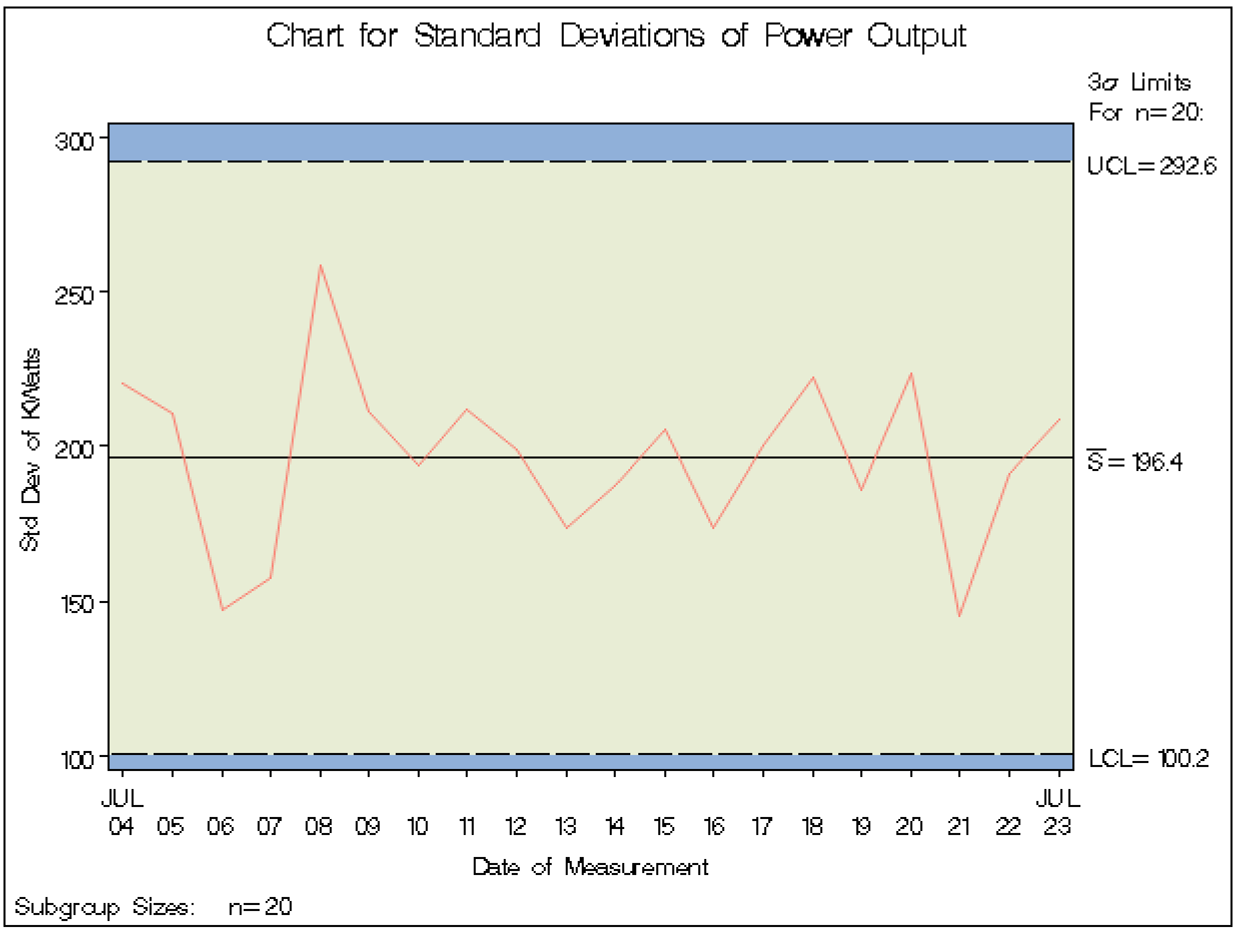
Day	KWattsX	KWattsS	KWattsN
04JUL	3487.40	220.260	20
05JUL	3471.65	210.427	20
06JUL	3488.30	147.025	20
07JUL	3434.20	157.637	20
08JUL	3475.80	258.949	20

You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext=swiss;
title 'Chart for Standard Deviations of Power Output';
proc shewhart history=Oilsum;
   schart KWatts*Day / cframe   = vligb
                       cinfill = ywh
                       cconnect = salmon;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting *s* chart is shown in Figure 17.81.

Figure 17.81 *s* Chart for Power Output Data (Traditional Graphics with NOGSTYLE)



Note that KWatts is *not* the name of a SAS variable in the data set Oilsum but is, instead, the common prefix for the names of the SAS variables KWattsS and KWattsN. The suffix characters *S* and *N* indicate *standard deviation* and *sample size*, respectively. Thus, you can specify two subgroup summary variables in the HISTORY= data set with a single name (KWatts), which is referred to as the *process*. The name Day, specified after the asterisk, is the name of the *subgroup-variable*.

In general, a HISTORY= input data set used with the SCHART statement must contain the following variables:

- subgroup variable
- subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of the subgroup standard deviation and sample size variables must begin with the *process* name specified in the SCHART statement and end with the special suffix characters *S* and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see "Creating Charts for Means and Ranges from Summary Data" on page 1841).

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1749.

Saving Summary Statistics

NOTE: See *Standard Deviation Chart (s Chart) Example* in the SAS/QC Sample Library.

In this example, the SCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Turbine and create a summary data set named Turbhist:

```
proc shewhart data=Turbine;
   schart KWatts*Day / outhistory = Turbhist
                       nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 17.79. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the SCHART statement. A complete list of options is presented in the section "Syntax: SCHART Statement" on page 1732.

Figure 17.82 contains a partial listing of Turbhist.

Figure 17.82 The Summary Data Set Turbhist

Summary Data Set for Power Output

Day	KWattsX	KWattsS	KWattsN
04JUL	3487.40	220.260	20
05JUL	3471.65	210.427	20
06JUL	3488.30	147.025	20
07JUL	3434.20	157.637	20
08JUL	3475.80	258.949	20

There are four variables in the data set Turbhist.

- Day contains the subgroup index.
- KWattsX contains the subgroup means.
- KWattsS contains the subgroup standard deviations.
- KWattsN contains the subgroup sample sizes.

The subgroup mean variable is included even though it is not required by the SCHART statement. This enables the data set to be used as a HISTORY= data set with the BOXCHART, XCHART, and XSCHART statements, as well as with the SCHART statement. Note that the summary statistic variables are named by adding the suffix characters *X*, *S*, and *N* to the *process* KWatts specified in the SCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1745.

Saving Control Limits

NOTE: See *Standard Deviation Chart (s Chart) Example* in the SAS/QC Sample Library.

You can save the control limits for an *s* chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1730) or modify the limits with a DATA step program.

The following statements read measurements from the data set Turbine (see "Creating Standard Deviation Charts from Raw Data" on page 1722) and save the control limits displayed in Figure 17.79 in a data set named Turblim:

```
proc shewhart data=Turbine;
   schart KWatts*Day / outlimits=Turblim
                       nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Turblim is listed in Figure 17.83.

Figure 17.83 The Data Set Turblim Containing Control Limit Information

Control Limits for Power Output Data

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
KWatts	Day	ESTIMATE	20	.002792725	3	3351.92	3485.41	3618.90

LCLS	_S_	_UCLS_	_STDDEV_
100.207	196.396	292.584	198.996

The data set Turblim contains one observation with the limits for *process* KWatts. The variables _LCLS_ and _UCLS_ contain the lower and upper control limits, and the variable _S_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. The variables _LCLX_ and _UCLX_, which contain the lower and upper control limits for subgroup means, are included so that the data set Turblim can be used to create an $\bar{X}$ chart (see "XSCHART Statement: SHEWHART Procedure" on page 1886). For more information, see "OUTLIMITS= Data Set" on page 1743.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Turbine;
   schart KWatts*Day / outtable=Turbtab
                       nochart;
run;
```

The data set Turbtab is listed in Figure 17.84.

Figure 17.84 The OUTTABLE= Data Set Turbtab

Summary Statistics and Control Limit Information

VAR	Day	_SIGMAS_	_LIMITN_	_SUBN_	_LCLS_	_SUBS_	_S_	_UCLS_	_STDDEV_	_EXLIM_
KWatts	04JUL	3	20	20	100.207	220.260	196.396	292.584	198.996	
KWatts	05JUL	3	20	20	100.207	210.427	196.396	292.584	198.996	
KWatts	06JUL	3	20	20	100.207	147.025	196.396	292.584	198.996	
KWatts	07JUL	3	20	20	100.207	157.637	196.396	292.584	198.996	
KWatts	08JUL	3	20	20	100.207	258.949	196.396	292.584	198.996	
KWatts	09JUL	3	20	20	100.207	211.566	196.396	292.584	198.996	
KWatts	10JUL	3	20	20	100.207	193.779	196.396	292.584	198.996	
KWatts	11JUL	3	20	20	100.207	212.024	196.396	292.584	198.996	
KWatts	12JUL	3	20	20	100.207	199.201	196.396	292.584	198.996	
KWatts	13JUL	3	20	20	100.207	173.455	196.396	292.584	198.996	
KWatts	14JUL	3	20	20	100.207	187.465	196.396	292.584	198.996	
KWatts	15JUL	3	20	20	100.207	205.472	196.396	292.584	198.996	
KWatts	16JUL	3	20	20	100.207	173.676	196.396	292.584	198.996	
KWatts	17JUL	3	20	20	100.207	200.576	196.396	292.584	198.996	
KWatts	18JUL	3	20	20	100.207	222.084	196.396	292.584	198.996	
KWatts	19JUL	3	20	20	100.207	185.724	196.396	292.584	198.996	
KWatts	20JUL	3	20	20	100.207	223.474	196.396	292.584	198.996	
KWatts	21JUL	3	20	20	100.207	145.267	196.396	292.584	198.996	
KWatts	22JUL	3	20	20	100.207	190.994	196.396	292.584	198.996	
KWatts	23JUL	3	20	20	100.207	208.858	196.396	292.584	198.996	

This data set contains one observation for each subgroup sample. The variables _SUBS_ and _SUBN_ contain the subgroup standard deviations and subgroup sample sizes. The variables _LCLS_ and _UCLS_ contain the lower and upper control limits, and the variable _S_ contains the central line. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1746.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Turbtab and display an *s* chart (not shown here) identical to the chart in Figure 17.79:

```
title 'Chart for Standard Deviations of Power Output';
symbol v=dot;
proc shewhart table=Turbtab;
   schart KWatts*Day;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2096). For more information, see "TABLE= Data Set" on page 1750.

Reading Preestablished Control Limits

NOTE: See *Standard Deviation Chart (s Chart) Example* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Turblim saved control limits computed from the measurements in Turbine. This example shows how these limits can be applied to new data.

The following statements create an *s* chart for new measurements in the data set Turbine2 (not listed here) using the control limits in Turblim:

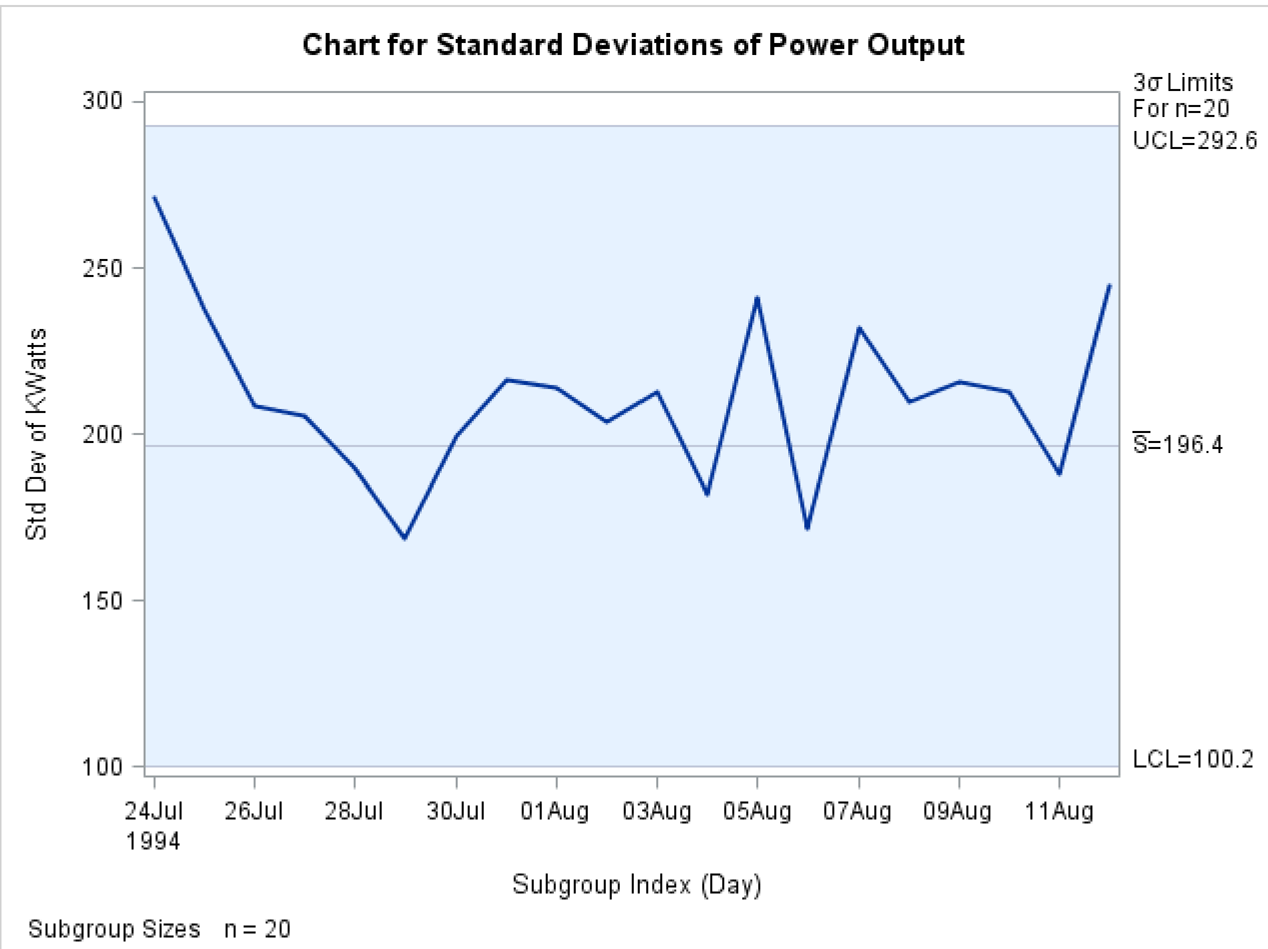
```
ods graphics on;
title 'Chart for Standard Deviations of Power Output';
proc shewhart data=Turbine2 limits=Turblim;
   schart KWatts*Day / odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the *s* chart is created by using ODS Graphics instead of traditional graphics. The chart is shown in Figure 17.85.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name KWatts
- the value of _SUBGRP_ matches the *subgroup-variable* name Day

Figure 17.85 *s* Chart for Second Set of Power Output Data (ODS Graphics)



All the standard deviations lie within the control limits, indicating that the variability of the heating process is still in statistical control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1748 for details concerning the variables that you must provide.

Syntax: SCHART Statement

The basic syntax for the SCHART statement is as follows:

SCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

SCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=**'*character*'> / < *options* > **;**

You can use any number of SCHART statements in the SHEWHART procedure. The components of the SCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Standard Deviation Charts from Raw Data" on page 1722.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Standard Deviation Charts from Subgroup Summary Data" on page 1725.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1728.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct *s* charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   schart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding SCHART statement, Day is the subgroup variable. For details, see "Subgroup Variables" on page 1871.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2026 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the subgroup standard deviations.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2025 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create an *s* chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   schart Weight*Day='*';
run;
```

options

enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options", which follows, lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1946 describes each option in detail.

Summary of Options

The following tables list the SCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1946.

Table 17.53 SCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	requests probability limits for chart
LIMITN=	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set

Table 17.53 *continued*

Option	Description
READINDEX=	reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	specifies width of control limits in terms of multiple k of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	specifies color for area inside control limits
CLIMITS=	specifies color of control limits, central line, and related labels
LCLLABEL=	specifies label for lower control limit
LIMLABSUBCHAR=	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	specifies line type for control limits
NDECIMAL=	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMIT0	suppresses display of zero lower control limit on s chart
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit
SSYMBOL=	specifies label for central line on s chart
UCLLABEL=	specifies label for upper control limit
WLIMITS=	specifies width for control limits and central line
Process Mean and Standard Deviation Options	
SIGMA0=	specifies known value σ_0 for process standard deviation σ
SMETHOD=	specifies method for estimating process standard deviation σ
TYPE=	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL2=	labels every point on s chart
CLABEL=	specifies color for labels
CCONNECT=	specifies color for line segments that connect points on chart
CFRAMELAB=	specifies fill color for frame around labeled points

Table 17.53 *continued*

Option	Description
CNEEDLES=	specifies color for needles that connect points to central line
COUT=	specifies color for portions of line segments that connect points outside control limits
COUTFILL=	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	specifies angle at which labels are drawn
LABELFONT=	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL2=	labels points outside control limits on s chart
SYMBOLLEGEND=	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=	specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	computes zone widths independently above and below center line
NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	suppresses tests across *phase* boundaries
TESTS2=	specifies tests for special causes for the s chart
TEST2RESET=	enables tests for special causes to be reset for the s chart
TEST2RUN=	specifies length of pattern for Test 2
TEST3RUN=	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL=	provides labels for points where test is positive
TESTLABEL*n*=	specifies label for *n*th test for special causes
TESTNMETHOD=	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=	enables tests for special causes to be reset
WESTGARD=	requests that Westgard rules be applied to the s chart
ZONE2LABELS	adds labels A, B, and C to zone lines for s chart
ZONES2	adds lines to s chart delineating zones A, B, and C
ZONEVALPOS=	specifies position of ZONEVALUES labels
ZONE2VALUES	labels s zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	specifies color for boxes enclosing labels indicating points where test is positive

Table 17.53 *continued*

Option	Description
CTESTS=	specifies color for labels indicating points where test is positive
CTESTSYMBOL=	specifies color for symbol used to plot points where test is positive
CZONES=	specifies color for lines and labels delineating zones A, B, and C
LTESTS=	specifies type of line connecting points where test is positive
LZONES=	specifies line type for lines delineating zones A, B, and C
TESTFONT=	specifies software font for labels at points where test is positive
TESTHEIGHT=	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	specifies plot symbol for points where test is positive
TESTSYMBOLHT=	specifies symbol height for points where test is positive
WTESTS=	specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill colors for frame for plot area
CTEXT=	specifies color for tick mark values and axis labels
DISCRETE	produces horizontal axis for discrete numeric group values
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default to s chart
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis

Table 17.53 *continued*

Option	Description
VFORMAT=	specifies format for vertical axis tick mark labels
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=	specifies width of axis lines
Plot Layout Options	
ALLN	plots means for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	specifies number of pages or screens to be used to display chart
ZEROSTD	displays s chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= options
CVREF=	specifies color for lines requested by VREF= = options
HREF=	specifies position of reference lines perpendicular to horizontal axis
HREFDATA=	specifies position of reference lines perpendicular to horizontal axis
HREFLABELS=	specifies labels for HREF= lines
HREFLABPOS=	specifies position of HREFLABELS= labels
LHREF=	specifies line type for HREF= lines
LVREF=	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	specifies position of reference lines perpendicular to vertical axis
VREFLABELS=	specifies labels for VREF= lines

Table 17.53 *continued*

Option	Description
VREFLABPOS=	position of VREFLABELS= labels
Grid Options	
CGRID=	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=	specifies line type for grid requested with the ENDGRID option
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme points are clipped
CLIPLEGEND=	specifies text for clipping legend
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features to chart
DESCRIPTION=	specifies description of s chart's GRSEG catalog entry
FONT=	specifies software font for labels and legends on charts
NAME=	specifies name of s chart's GRSEG catalog entry
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	shades areas between control limits and connected points lying outside the limits
STARFILL=	specifies a variable identfying groups of stars filled with different colors
STARS=	specifies a variable identfying groups of stars whose outlines are drawn with different colors

Table 17.53 *continued*

Option	Description
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	specifies the control limit infill transparency
MARKERS	plots subgroup points with markers
NOBLOCKREF	suppresses block and phase reference lines
NOBLOCKREFFILL	suppresses block and phase wall fills
NOFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOPHASEREF	suppresses block and phase reference lines
NOPHASEREFFILL	suppresses block and phase wall fills
NOREF	suppresses block and phase reference lines
NOREFFILL	suppresses block and phase wall fills
NOSTARFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	specifies a graph footnote
ODSFOOTNOTE2=	specifies a secondary graph footnote
ODSLEGENDEXPAND	specifies that legend entries contain all levels observed in the data
ODSTITLE=	specifies a graph title
ODSTITLE2=	specifies a secondary graph title
OUTFILLTRANSPARENCY=	specifies control limit outfill transparency
OVERLAYURL=	specifies URLs to associate with overlay points
PHASEPOS=	specifies vertical position of phase legend
PHASEREFLEVEL=	associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	draws central line labels using a simulated software font
STARTRANSPARENCY=	specifies star fill transparency
URL=	specifies a variable whose values are URLs to be associated with subgroups
Input Data Set Options	
MISSBREAK	specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	creates output data set containing subgroup summary statistics
OUTINDEX=	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	creates output data set containing control limits
OUTTABLE=	creates output data set containing subgroup summary statistics and control limits

Table 17.53 *continued*

Option	Description
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	specifies α value and type for computing capability index confidence limits
LSL=	specifies list of lower specification limits
TARGET=	specifies list of target values
USL=	specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for *block-variable* legend
BLOCKLABTYPE=	specifies text size of *block-variable* legend
BLOCKPOS=	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE=	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	specifies *phases* to be read from an input data set

Table 17.53 *continued*

Option	Description
Star Options	
CSTARCIRCLES=	specifies color for STARCIRCLES= circles
CSTARFILL=	specifies color for filling stars
CSTAROUT=	specifies outline color for stars exceeding inner or outer circles
CSTARS=	specifies color for outlines of stars
LSTARCIRCLES=	specifies line types for STARCIRCLES= circles
LSTARS=	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	specifies reference circles for stars
STARINRADIUS=	specifies inner radius of stars
STARLABEL=	specifies vertices to be labeled
STARLEGEND=	specifies style of legend for star vertices
STARLEGENDLAB=	specifies label for STARLEGEND= legend
STAROUTRADIUS=	specifies outer radius of stars
STARSPECS=	specifies method used to standardize vertex variables
STARSTART=	specifies angle for first vertex
STARTYPE=	specifies graphical style of star
STARVERTICES=	superimposes star at each point on chart
WSTARCIRCLES=	specifies width of STARCIRCLES= circles
WSTARS=	specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	specifies colors for overlay line segments
COVERLAY=	specifies colors for overlay plots
COVERLAYCLIP=	specifies color for clipped points on overlays
LOVERLAY=	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=	specifies variables to overlay on chart
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
OVERLAYHTML=	specifies links to associate with overlay points
OVERLAYID=	specifies labels for overlay points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols for overlays
OVERLAYSYMHT=	specifies symbol heights for overlays
WOVERLAY=	specifies widths of overlay line segments
Options for Interactive Control Charts	
HTML=	specifies a variable whose values create links to be associated with subgroups
HTML_LEGEND=	specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	creates an OUTTABLE= data set with additional graphics coordinate data

Table 17.53 *continued*

Option	Description
Options for Line Printer Charts	
CLIPCHAR=	specifies plot character for clipped points
CONNECTCHAR=	specifies character used to form line segments that connect points on chart
HREFCHAR=	specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	specifies characters indicating *symbol-variable*
TESTCHAR=	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	specifies line character for VREF= and VREF2= lines
ZONECHAR=	specifies character for lines that delineate zones for tests for special causes

Details: SCHART Statement

Constructing Charts for Standard Deviations

The following notation is used in this section:

σ process standard deviation (standard deviation of the population of measurements)

s_i standard deviation of measurements in ith subgroup

$$s_i = \sqrt{(1/(n_i - 1))((x_{i1} - \bar{X}_i)^2 + \cdots + (x_{in_i} - \bar{X}_i)^2)}$$

n_i sample size of ith subgroup

$c_4(n)$ expected value of the standard deviation of n independent normally distributed variables with unit standard deviation

$c_5(n)$ standard error of the standard deviation of n independent observations from a normal population with unit standard deviation

$\chi^2_p(n)$ 100pth percentile ($0 < p < 1$) of the χ^2 distribution with n degrees of freedom

Plotted Points

Each point on an s chart indicates the value of a subgroup standard deviation (s_i). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 13, the value plotted for this subgroup is

$$s_{10} = \sqrt{((12-15)^2 + (15-15)^2 + (19-15)^2 + (16-15)^2 + (13-15)^2)/4} = 2.739$$

Central Line

By default, the central line for the ith subgroup indicates an estimate for the expected value of s_i, which is computed as $c_4(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $c_4(n_i)\sigma_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of s_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that s_i exceeds the limits

The following table provides the formulas for the limits:

Table 17.54 Limits for s Charts

Control Limits
LCL = lower limit = $\max(c_4(n_i)\hat{\sigma} - kc_5(n_i)\hat{\sigma}, 0)$ UCL = upper limit = $c_4(n_i)\hat{\sigma} + kc_5(n_i)\hat{\sigma}$
Probability Limits
LCL = lower limit = $\hat{\sigma}\sqrt{\chi^2_{\alpha/2}(n_i - 1)/(n_i - 1)}$ UCL = upper limit = $\hat{\sigma}\sqrt{\chi^2_{1-\alpha/2}(n_i - 1)/(n_i - 1)}$

The formulas assume that the data are normally distributed. If a standard value σ_0 is available for σ, replace $\hat{\sigma}$ with σ_0 in Table 17.54. Note that the upper and lower limits vary with n_i and that the probability limits are asymmetric around the central line.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables are saved:

Table 17.55 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLS	lower control limit for subgroup standard deviation
LCLX	lower control limit for subgroup mean
LIMITN	sample size associated with the control limits
LSL	lower specification limit
MEAN	process mean ($\bar{\bar{X}}$ or μ_0)
_S_	value of central line on s chart
SIGMAS	multiple (k) of standard error of $\bar{X}_i$ or s_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the SCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLS	upper control limit for subgroup standard deviation
UCLX	upper control limit for subgroup mean
USL	upper specification limit
VAR	*process* specified in the SCHART statement

Notes:

1. The variables _LCLX_, _MEAN_, and _UCLX_ are saved to enable the OUTLIMITS= data set to be used as a LIMITS= data set with the BOXCHART, XCHART, and XSCHART statements.

2. If the control limits vary with subgroup sample size, the special missing value *V* is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLS_, _S_, and _UCLS_.

3. If the limits are defined in terms of a multiple k of the standard error of s_i, the value of _ALPHA_ is computed as

$$F_S(\text{\_LCLS\_}/\text{\_STDDEV\_}) + 1 - F_S(\text{\_UCLS\_}/\text{\_STDDEV\_})$$

   where $F_S(\cdot)$ is the cumulative distribution function of the standard deviation of a sample of n observations from a normal population with unit standard deviation, and n is the value of _LIMITN_. If _LIMITN_ has the special missing value *V*, this value is assigned to _ALPHA_.

4. If the limits are probability limits, the value of _SIGMAS_ is computed as $(\text{\_UCLS\_} - \text{\_S\_})/e$, where e is the standard error of the standard deviation of n observations from a normal population with unit standard deviation. If _LIMITN_ has the special missing value *V*, this value is assigned to _SIGMAS_.

5. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1874 for computational details.

6. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the SCHART statement. For an example, see "Saving Control Limits" on page 1728.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a subgroup sample size variable named by *process* suffixed with *N*

The subgroup mean variable is included so that the data set can be reused as a HISTORY= data set with the BOXCHART, XCHART, and XSCHART statements, as well as the SCHART statement.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the SCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   schart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthX, WidthS, WidthN, DiameterX, DiameterS, and DiameterN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1727.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on s chart
LCLS	lower control limit for standard deviation
LIMITN	nominal sample size associated with the control limits
_S_	average standard deviation
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBN	subgroup sample size
SUBS	subgroup standard deviation
TESTS2	tests for special causes signaled on s chart
UCLS	upper control limit for standard deviation
VAR	*process* specified in the SCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= option, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS2_ is saved if you specify the TESTS2= option.
3. The variables _EXLIM_ and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see “Saving Control Limits” on page 1728.

ODS Tables

The following table summarizes the ODS tables that you can request with the SCHART statement.

Table 17.56 ODS Tables Produced with the SCHART Statement

Table Name	Description	Options
SCHART	*s* chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*). **NOTE:** In SAS/QC 13.1 the SHEWHART procedure does not support the creation of graphs that are editable with the ODS Graphics Editor.

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. SCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1946.

When ODS Graphics is in effect, the SCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 17.57.

Table 17.57 ODS Graphics Produced by the SCHART Statement

ODS Graph Name	Plot Description
SChart	*s* chart

See Chapter 3, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the SCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements, which must be grouped into subgroup samples indexed by the values of the *subgroup-variable*. The *subgroup-variable*, which is specified in the SCHART statement, must also be a SAS variable in the DATA= data set.

Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the *i*th subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the *i*th subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2031.

For an example of a DATA= data set, see "Creating Standard Deviation Charts from Raw Data" on page 1722.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   schart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 17.55. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLS_, _S_, and _UCLS_, which specify the control limits directly
- the variable _STDDEV_ , which is used to calculate the control limits according to the equations in Table 17.54

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option. This must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'ESTIMATE', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1730.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedures or to read output data sets created with SAS summarization procedures, such as the MEANS procedure.

A HISTORY= data set used with the SCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup standard deviation variable for each *process*
- a subgroup sample size variable for each *process*

The names of the subgroup standard deviation and subgroup sample size variables must be the *process* name concatenated with the special suffix characters *S* and *N*, respectively. For example, consider the following statements:

```
proc shewhart history=Summary;
   schart (Weight Yieldstrength)*Batch;
run;
```

The data set Summary must include the variables Batch, WeightS, WeightN, YieldstrengthS, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*

- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2031 for an example).

For an example of a HISTORY= data set, see "Creating Standard Deviation Charts from Subgroup Summary Data" on page 1725.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2096.

The following table lists the variables required in a TABLE= data set used with the SCHART statement:

Table 17.58 Variables Required in a TABLE= Data Set

Variable	Description
LCLS	lower control limit for standard deviation
LIMITN	nominal sample size associated with the control limits
_S_	average standard deviation
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBS	subgroup standard deviation
UCLS	upper control limit for standard deviation

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.

- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1728.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, three methods (referred to as default, MVLUE, and RMSDF) are available for estimating σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \cdots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the ith subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}(x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the ith subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, then the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). This estimate is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

RMSDF Method

If you specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

where $n = n_1 + \cdots + n_N - (N - 1)$. The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications, it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup standard deviation variable
Vertical	TABLE=	_SUBS_

For an example, see “Labeling Axes” on page 2063.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: SCHART Statement

This section provides advanced examples of the SCHART statement.

Example 17.29: Specifying a Known Standard Deviation

NOTE: See *s Chart with Known Standard Deviation* in the SAS/QC Sample Library.

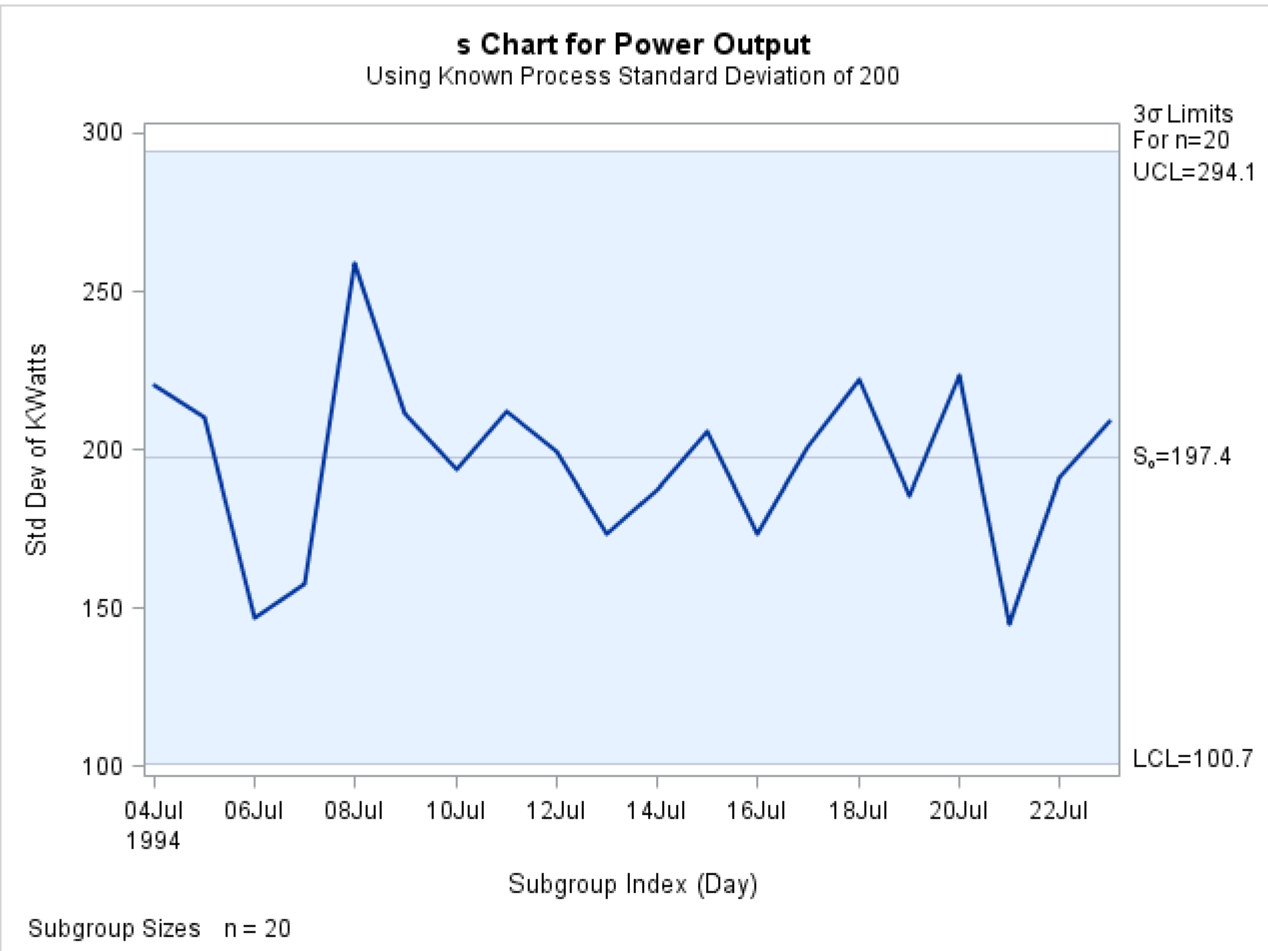
In some applications, a standard value σ_0 may be available for the process standard deviation σ. This example shows how you can specify σ_0 to compute the control limits.

Suppose that the amount of power needed to heat water in the heating process described in “Creating Standard Deviation Charts from Raw Data” on page 1722 has a known standard deviation of 200. The following statements specify this known value and create an *s* chart, shown in Output 17.29.1, for the power output measurements in the data set Turbine:

```
ods graphics on;
title 's Chart for Power Output';
title2 'Using Known Process Standard Deviation of 200';
proc shewhart data=Turbine;
   schart KWatts*Day / sigma0    = 200
                       ssymbol   = s0
                       odstitle  = title
                       odstitle2 = title2;
run;
```

The SIGMA0= option specifies σ_0, and the SSYMBOL= option specifies a label for the central line indicating that the central line is computed from σ_0. Since all the points lie within the limits, you can conclude that the variability of the process is stable.

Output 17.29.1 Reading in Standard Value for Process Standard Deviation



You can also specify σ_0 as the value of the variable _STDDEV_ in a LIMITS= data set, as illustrated by the following statements:

```
data Plimits;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'KWatts';
   _subgrp_ = 'Day';
   _type_   = 'STDSIGMA';
   _limitn_ = 20;
   _stddev_ = 200;
run;

title 'Chart Using Known Process Standard Deviation';
proc shewhart data=Turbine limits=Plimits;
   schart KWatts*Day / ssymbol=s0;
run;
```

The resulting *s* chart (not shown here) is identical to the one shown in Output 17.29.1. For more information, see “LIMITS= Data Set” on page 1748.

Example 17.30: Computing Average Run Lengths for s Charts

NOTE: See *Computing Average Run Lengths for s Charts* in the SAS/QC Sample Library.

This example illustrates how you can compute the average run length of an *s* chart. The data used here are the power measurements in the data set Turbine, which is introduced in “Creating Standard Deviation Charts from Raw Data” on page 1722.

The in-control average run length of a Shewhart chart is ARL $= \frac{1}{p}$, where p is the probability that a single point exceeds its control limits. Since this probability is saved as the value of the variable _ALPHA_ in an OUTLIMITS= data set, you can compute ARL for an *s* chart as follows:

```
title 'Average In-Control Run Length';
proc shewhart data=Turbine;
   schart KWatts*Day / outlimits=Turblim nochart;

data ARLcomp;
   keep _var_ _sigmas_ _alpha_ arl;
   set Turblim;
   arl = 1 / _alpha_;
run;
```

The data set ARLcomp is listed in Output 17.30.1, which shows that the ARL is equal to 358.

Output 17.30.1 The Data Set ARLcomp

Average In-Control Run Length

VAR	_ALPHA_	_SIGMAS_	arl
KWatts	.002792725	3	358.073

To compute out-of-control average run lengths, define f as the slippage factor for the process standard deviation σ, where $f > 1$. In other words, the “shifted” standard deviation to be detected by the chart is $f\sigma$. The following statements compute the ARL as a function of f:

```
data ARLshift;
   keep f f_std p arl_f;
   set Turblim;
   df = _limitn_ - 1;
   do f = 1 to 1.5 by 0.05;
      f_std = f * _stddev_;
      low   = df * ( _lcls_ / f_std )**2;
      upp   = df * ( _ucls_ / f_std )**2;
      p     = probchi( low, df ) + 1 - probchi( upp, df );
      arl_f = 1 / p;
      output;
   end;
run;
```

The data set ARLshift is listed in Output 17.30.2. For example, on average, 53 samples are required to detect a ten percent increase in σ (a shifted standard deviation of approximately 219). The computations use the fact that $(n_i - 1)s_i^2/\sigma^2$ has a χ^2 distribution with $n_i - 1$ degrees of freedom, assuming that the measurements are normally distributed.

Output 17.30.2 The Data Set ARLshift

Average Run Length Analysis

f	f_std	p	arl_f
1.00	198.996	0.00279	358.073
1.05	208.945	0.00758	131.922
1.10	218.895	0.01875	53.322
1.15	228.845	0.03984	25.102
1.20	238.795	0.07388	13.535
1.25	248.745	0.12239	8.171
1.30	258.694	0.18475	5.413
1.35	268.644	0.25834	3.871
1.40	278.594	0.33923	2.948
1.45	288.544	0.42298	2.364
1.50	298.494	0.50546	1.978

UCHART Statement: SHEWHART Procedure

Overview: UCHART Statement

The UCHART statement creates u charts for the numbers of nonconformities (defects) per inspection unit in subgroup samples containing arbitrary numbers of units.

You can use options in the UCHART statement to

- specify the number of inspection units per subgroup

- compute control limits from the data based on a multiple of the standard error of the plotted values or as probability limits
- tabulate subgroup summary statistics and control limits
- save control limits in an output data set
- save subgroup summary statistics in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a known (standard) value for the average number of nonconformities per inspection unit
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing *u* charts with the UCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 3, “SAS/QC Graphics,” for more information about producing these different kinds of graphs.

Getting Started: UCHART Statement

This section introduces the UCHART statement with simple examples that illustrate commonly used options. Complete syntax for the UCHART statement is presented in the section “Syntax: UCHART Statement” on page 1766, and advanced examples are given in the section “Examples: UCHART Statement” on page 1786.

Creating u Charts from Defect Count Data

NOTE: See *u Chart Examples* in the SAS/QC Sample Library.

A textile company uses a *u* chart to monitor the number of defects per square meter of fabric. The fabric is spooled onto rolls as it is inspected for defects. Each piece of fabric is one meter wide and 30 meters in length. The following statements create a SAS data set named Fabric, which contains the defect counts for 20 rolls:

```
data Fabric;
   input Roll Defects @@;
   datalines;
 1 12     2 11     3  9     4 15
 5  7     6  6     7  5     8 10
 9  8    10  8    11 14    12  5
13  9    14 13    15  7    16  5
17  8    18 11    19  7    20 12
;
```

A partial listing of Fabric is shown in Figure 17.86.

Figure 17.86 The Data Set Fabric

Number of Fabric Defects

Roll	Defects
1	12
2	11
3	9
4	15
5	7

There is a single observation per roll. The variable Roll identifies the subgroup sample and is referred to as the *subgroup-variable*. The variable Defects contains the number of nonconformities (defect count) for each subgroup sample and is referred to as the *process variable* (or *process* for short).

The following statements create the *u* chart shown in Figure 17.87:

```
ods graphics off;
title 'u Chart for Fabric Defects';
proc shewhart data=Fabric;
   uchart Defects*Roll / subgroupn = 30;
run;
```

This example illustrates the basic form of the UCHART statement. After the keyword UCHART, you specify the *process* to analyze (in this case, Defects), followed by an asterisk and the *subgroup-variable* (Roll).

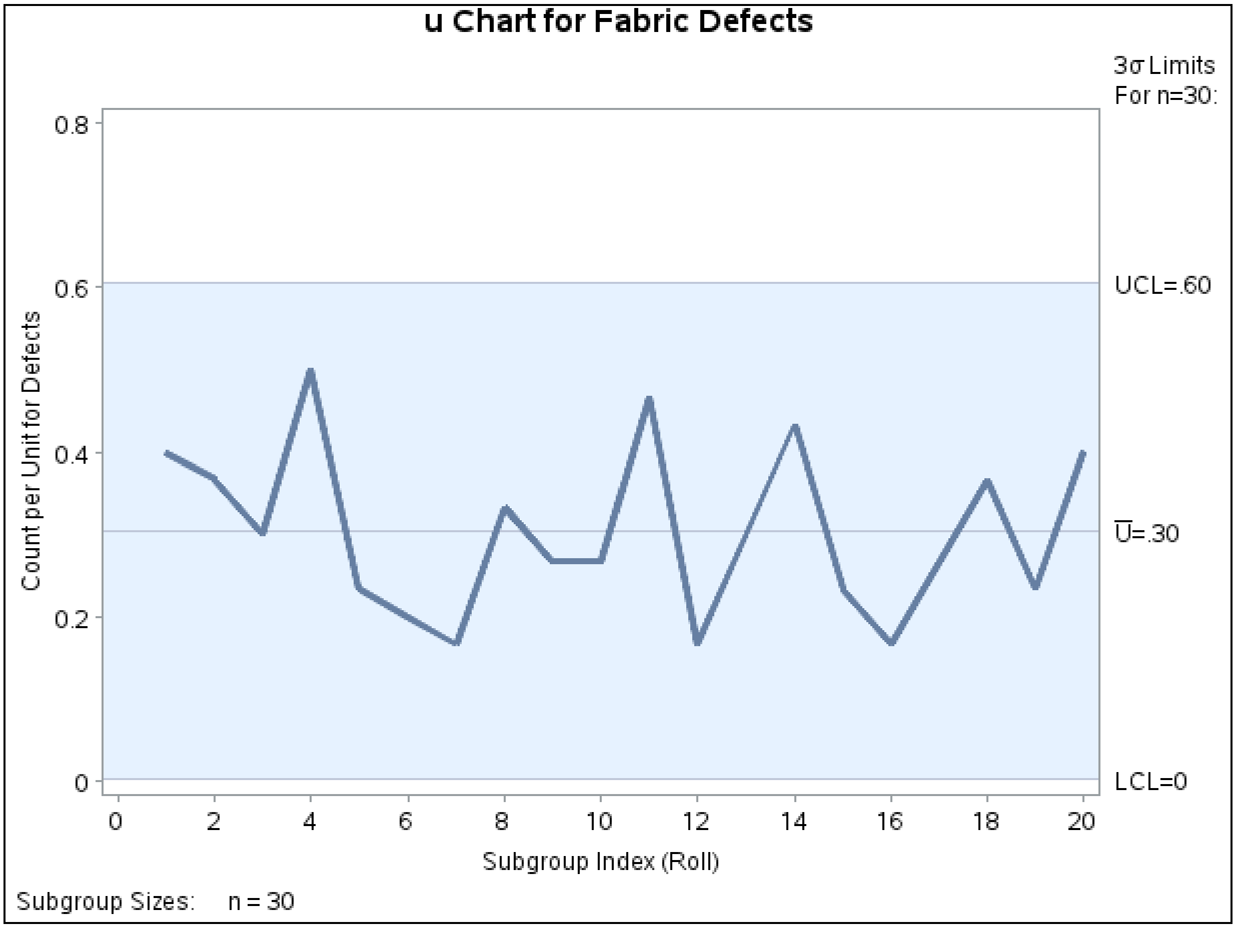
The SUBGROUPN= option specifies the number of inspection units in each subgroup sample and is required if the input data set is a DATA= data set. In this example, each square meter of fabric is an inspection unit, and each roll is a subgroup sample. The number of inspection units per subgroup can be thought of as the subgroup sample size.

You can use the SUBGROUPN= option to specify one of the following:

- a constant subgroup sample size (as in this example)
- an input variable name whose values contain the subgroup sample sizes (for an example, see "Saving Nonconformities per Unit" on page 1765)

Options such as SUBGROUPN= are specified after the slash (/) in the UCHART statement. A complete list of options is presented in the section "Syntax: UCHART Statement" on page 1766.

Figure 17.87 *u* Chart Example (Traditional Graphics)



The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Each point on the *u* chart represents the number of nonconformities per inspection unit for a particular subgroup. For instance, the value plotted for the first subgroup is $12/30 = 0.4$ (since there are 12 defects on the first roll and this roll contains 30 square meters of fabric). By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in "Control Limits" on page 1778. Since none of the points exceed the 3σ limits, the *u* chart indicates that the fabric manufacturing process is in statistical control.

See "Constructing Charts for Nonconformities per Unit (u Charts)" on page 1777 for details concerning *u* charts. For more details on reading defect count data, see "DATA= Data Set" on page 1782.

Saving Control Limits

NOTE: See *u Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for a *u* chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1761) or modify the limits with a DATA step program.

The following statements read defect counts from the data set Fabric (see "Creating u Charts from Defect Count Data" on page 1756) and save the control limits displayed in Figure 17.87 in a data set named Fablim:

```
proc shewhart data=Fabric;
   uchart Defects*Roll / subgroupn = 30
                         outlimits = Fablim
                         nochart;
run;
```

The SUBGROUPN= option specifies the number of inspection units in each subgroup sample. The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Fablim is listed in Figure 17.88.

Figure 17.88 The Data Set Fablim Containing Control Limit Information

Control Limits Data Set FABLIM

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLU_	_U_	_UCLU_
Defects	Roll	ESTIMATE	30	.002421390	3	.001671271	0.30333	0.60500

The data set Fablim contains one observation with the limits for *process* Defects. The variables _LCLU_ and _UCLU_ contain the lower and upper control limits, and the variable _U_ contains the central line. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the value of _U_ is an estimate or standard value. For more information, see "OUTLIMITS= Data Set" on page 1779.

Alternatively, you can use the OUTTABLE= option to create an output data set that saves both the control limits and the subgroup statistics, as illustrated by the following statements:

```
proc shewhart data=Fabric;
   uchart Defects*Roll / subgroupn = 30
                         outtable  = Fabtab
                         nochart;
run;
```

The data set Fabtab is listed in Figure 17.89.

Figure 17.89 The Data Set Fabtab

Number of Defects Per Square Meter and Control Limits

VAR	Roll	_SIGMAS_	_LIMITN_	_SUBN_	_LCLU_	_SUBU_	_U_	_UCLU_	_EXLIM_
Defects	1	3	30	30	.001671271	0.40000	0.30333	0.60500	
Defects	2	3	30	30	.001671271	0.36667	0.30333	0.60500	
Defects	3	3	30	30	.001671271	0.30000	0.30333	0.60500	
Defects	4	3	30	30	.001671271	0.50000	0.30333	0.60500	
Defects	5	3	30	30	.001671271	0.23333	0.30333	0.60500	
Defects	6	3	30	30	.001671271	0.20000	0.30333	0.60500	
Defects	7	3	30	30	.001671271	0.16667	0.30333	0.60500	
Defects	8	3	30	30	.001671271	0.33333	0.30333	0.60500	
Defects	9	3	30	30	.001671271	0.26667	0.30333	0.60500	
Defects	10	3	30	30	.001671271	0.26667	0.30333	0.60500	
Defects	11	3	30	30	.001671271	0.46667	0.30333	0.60500	
Defects	12	3	30	30	.001671271	0.16667	0.30333	0.60500	
Defects	13	3	30	30	.001671271	0.30000	0.30333	0.60500	
Defects	14	3	30	30	.001671271	0.43333	0.30333	0.60500	
Defects	15	3	30	30	.001671271	0.23333	0.30333	0.60500	
Defects	16	3	30	30	.001671271	0.16667	0.30333	0.60500	
Defects	17	3	30	30	.001671271	0.26667	0.30333	0.60500	
Defects	18	3	30	30	.001671271	0.36667	0.30333	0.60500	
Defects	19	3	30	30	.001671271	0.23333	0.30333	0.60500	
Defects	20	3	30	30	.001671271	0.40000	0.30333	0.60500	

This data set contains one observation for each subgroup sample. The variables _SUBU_ and _SUBN_ contain the number of nonconformities per unit in each subgroup and the number of inspection units per subgroup. The variables _LCLU_ and _UCLU_ contain the lower and upper control limits, and the variable _U_ contains the central line. The variables _VAR_ and Roll contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1780.

An OUTTABLE= data set can be read later as a TABLE= data set by the SHEWHART procedure. For example, the following statements read Fabtab and display a *u* chart (not shown here) identical to the chart in Figure 17.87:

```
title 'u Chart for Fabric Defects';
proc shewhart table=Fabtab;
   uchart Defects*Roll / subgroupn=30;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see “Specialized Control Charts: SHEWHART Procedure” on page 2096). For more information, see “TABLE= Data Set” on page 1785.

Reading Preestablished Control Limits

NOTE: See *u Chart Examples* in the SAS/QC Sample Library.

In the previous example, control limits were saved in a SAS data set named Fablim. This example shows how these limits can be applied to defect counts for an additional 20 rolls of fabric, which are provided in the following data set:

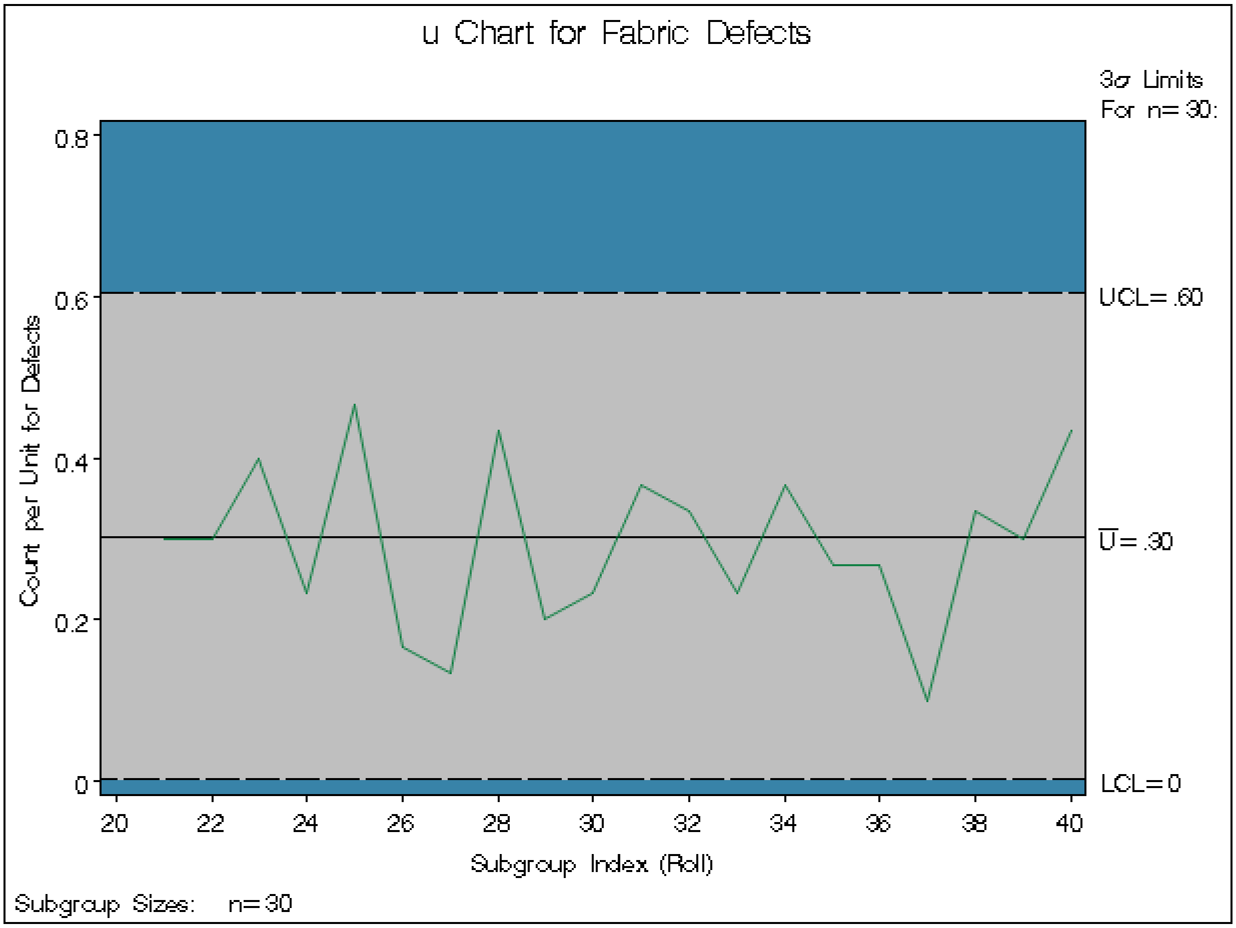
```
data Fabric2;
   input Roll Defects @@;
   datalines;
21  9    22  9    23 12    24  7    25 14
26  5    27  4    28 13    29  6    30  7
31 11    32 10    33  7    34 11    35  8
36  8    37  3    38 10    39  9    40 13
;
```

The following statements create a *u* chart for the second group of rolls using the control limits in Fablim:

```
options nogstyle;
goptions ftext=swiss;
symbol color = vig h = .8;
title 'u Chart for Fabric Defects';
proc shewhart data=Fabric2 limits=Fablim;
   uchart Defects*Roll / subgroupn = 30
                         cframe    = steel
                         cinfill   = ligr
                         cconnect  = vig
                         coutfill  = yellow;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement and UCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The chart is shown in Figure 17.90 and indicates that the process is in control.

Figure 17.90 A *u* Chart for Second Set of Fabric Rolls (Traditional Graphics with NOGSTYLE)



The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* Defects
- the value of _SUBGRP_ matches the *subgroup-variable* name Roll

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1783 for details concerning the variables that you must provide.

Creating u Charts from Nonconformities per Unit

NOTE: See *u Chart Examples* in the SAS/QC Sample Library.

In the previous example, the input data set provided the number of nonconformities for each subgroup sample. However, in some applications, as illustrated here, the data provide the number of nonconformities *per inspection unit* for each subgroup.

A clothing manufacturer ships shirts in boxes of ten. Prior to shipment, each shirt is inspected for flaws. Since the manufacturer is interested in the average number of flaws per shirt, the number of flaws found in each box is divided by ten and then recorded. The following statements create a SAS data set named Shirts, which contains the average number of flaws per shirt for 25 boxes:

```
data Shirts;
   input Box AvgdefU @@;
   AvgdefN=10;
   datalines;
 1  0.4    2  0.7    3  0.5    4  1.0    5  0.3
 6  0.2    7  0.0    8  0.4    9  0.4   10  0.6
11  0.2   12  0.7   13  0.3   14  0.1   15  0.3
16  0.6   17  0.6   18  0.3   19  0.7   20  0.3
21  0.0   22  0.1   23  0.5   24  0.6   25  0.4
;
```

Note that this is the same data set used in "Getting Started: CCHART Statement" on page 1437 of "CCHART Statement: SHEWHART Procedure" on page 1436. A partial listing of Shirts is shown in Figure 17.91.

Figure 17.91 The Data Set Shirts

Average Number of Shirt Flaws

Box	AvgdefU	AvgdefN
1	0.4	10
2	0.7	10
3	0.5	10
4	1.0	10
5	0.3	10

The data set Shirts contains three variables: the box number (Box), the average number of flaws per shirt (AvgdefU), and the number of shirts per box (AvgdefN). Here, a *subgroup* is a box of shirts, and an *inspection unit* is an individual shirt. Note that each subgroup contains ten inspection units.

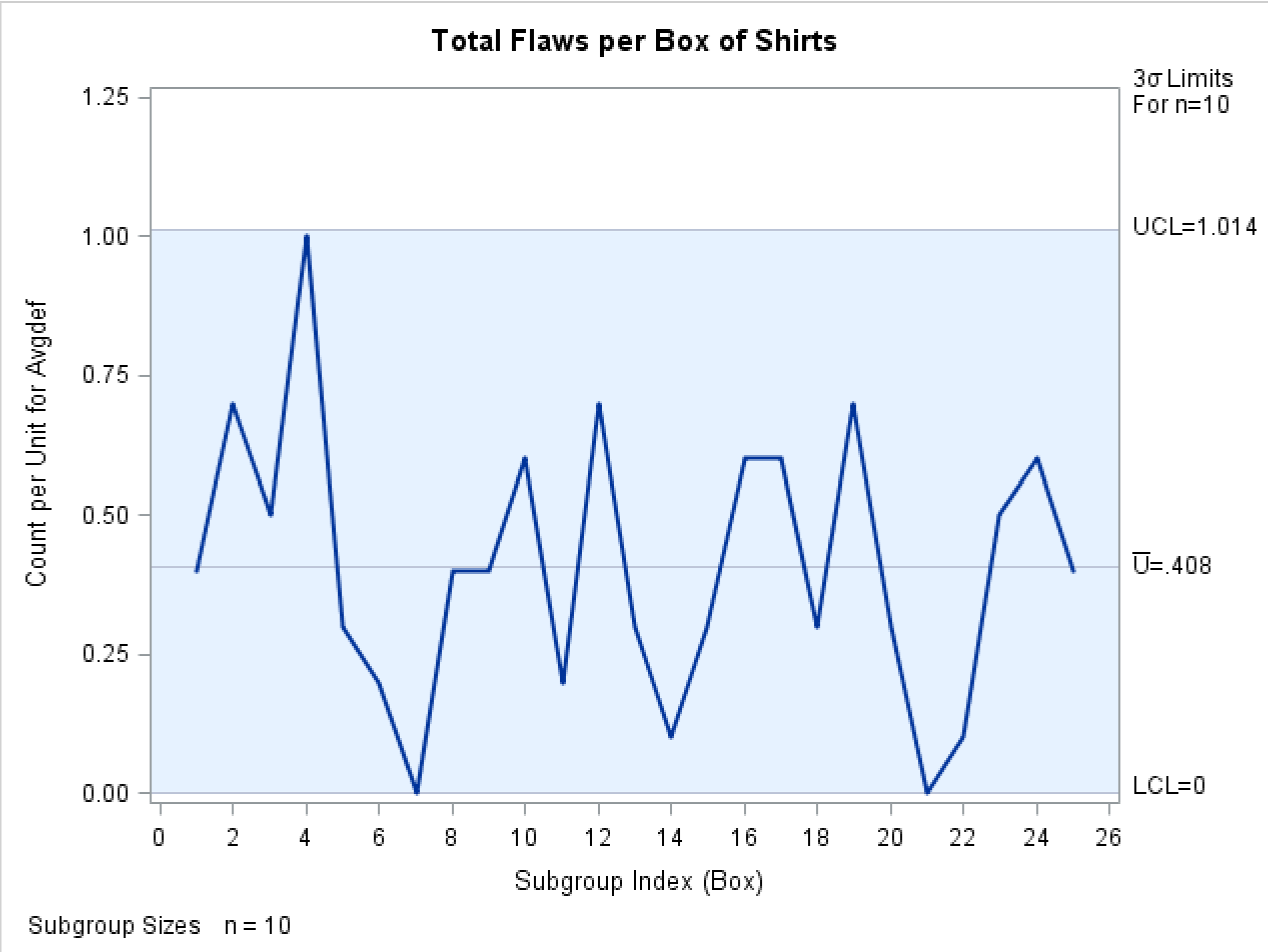
To create a *u* chart for the average number of flaws per shirt in each box, you can specify Shirts as a HISTORY= data set.

```
ods graphics on;
title 'Total Flaws per Box of Shirts';
proc shewhart history=Shirts;
   uchart Avgdef*Box / odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the *u* chart is created by using ODS Graphics instead of traditional graphics.

Note that Avgdef is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the SAS variables AvgdefU and AvgdefN. The suffix characters *U* and *N* indicate *number of nonconformities per unit* and *sample size*, respectively. This naming convention enables you to specify two variables in the HISTORY= data set with a single name, which is referred to as the *process*. The name Box, specified after the asterisk, is the name of the *subgroup-variable*. The *u* chart is shown in Figure 17.92.

Figure 17.92 u Chart for Boxes of Shirts (ODS Graphics)



In general, a HISTORY= input data set used with the UCHART statement must contain the following variables:

- subgroup variable
- subgroup number of nonconformities per unit variable
- subgroup sample size variable

Furthermore, the names of the nonconformities per unit and sample size variables must begin with the *process* name specified in the UCHART statement and end with the special suffix characters *U* and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. Suppose that, instead of the variables AvgdefU and AvgdefN, the data set Shirts contained the variables Shirtdef and Sizes. The following statements temporarily rename Shirtdef and Sizes to AvgdefU and AvgdefN:

```
proc shewhart
   history=Shirts (rename=(Shirtdef = AvgdefU
                           Sizes    = AvgdefN ));
   uchart Avgdef*Box;
run;
```

For more information, see "HISTORY= Data Set" on page 1784.

Saving Nonconformities per Unit

NOTE: See *u Chart Examples* in the SAS/QC Sample Library.

In this example, the UCHART statement is used to create a summary data set containing the number of nonconformities per unit. This data set can be read later by the SHEWHART procedure (as in the preceding example).

A department store receives boxes of shirts containing 10, 25, or 50 shirts. Each box is inspected, and the total number of defects per box is recorded. The following statements create a SAS data set named Shirts2, which contains the total defects per box for 20 boxes:

```
data Shirts2;
   input Box Flaws nShirts @@;
   datalines;
 1  3  10     2  8  10     3 15  25     4 20  25
 5  9  25     6  1  10     7  1  10     8 21  50
 9  3  10    10  7  10    11  1  10    12 21  25
13  9  25    14  3  25    15 12  50    16 18  50
17  7  10    18  4  10    19  8  10    20  4  10
;
```

A partial listing of Shirts2 is shown in Figure 17.93.

Figure 17.93 The Data Set Shirts2

Number of Shirt Flaws per Box

Box	AvgdefU	AvgdefN
1	0.4	10
2	0.7	10
3	0.5	10
4	1.0	10
5	0.3	10

The variable Box contains the box number, the variable Flaws contains the number of flaws in each box, and the variable nShirts contains the number of shirts in each box. To evaluate the quality of the shirts, you should report the average number of defects per shirt. The following statements create a data set containing the number of flaws per shirt and the number of shirts per box:

```
proc shewhart data=Shirts2;
   uchart Flaws*Box / subgroupn   = nShirts
                      outhistory = Shirthist
                      nochart;
run;
```

The SUBGROUPN= option names the variable in the DATA= data set whose values specify the number of inspection units per subgroup. The OUTHISTORY= option names an output data set containing the number of nonconformities per inspection unit and the number of inspection units per subgroup. A partial listing of Shirthist is shown in Figure 17.94.

Figure 17.94 The Data Set Shirthist

Average Defects Per Tee Shirt

Box	FlawsU	FlawsN
1	0.30	10
2	0.80	10
3	0.60	25
4	0.80	25
5	0.36	25

There are three variables in the data set Shirthist.

- Box contains the subgroup index.
- FlawsU contains the numbers of nonconformities per inspection unit.
- FlawsN contains the subgroup sample sizes.

Note that the variables containing the numbers of nonconformities per inspection unit and subgroup sample sizes are named by adding the suffix characters *U* and *N* to the *process* Flaws specified in the UCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1780.

Syntax: UCHART Statement

The basic syntax for the UCHART statement is as follows:

UCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

UCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=**'*character*' > / < *options* > **;**

You can use any number of UCHART statements in the SHEWHART procedure. The components of the UCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If numbers of nonconformities per subgroup are read from a DATA= data set, *process* must be the name of the variable containing the numbers of nonconformities. For an example, see "Creating u Charts from Defect Count Data" on page 1756.
- If numbers of nonconformities per unit and numbers of inspection units per subgroup are read from a HISTORY= data set, *process* must be the common prefix of the appropriate variables in the HISTORY= data set. For an example, see "Creating u Charts from Nonconformities per Unit" on page 1762.
- If numbers of nonconformities per item, numbers of inspection units per subgroup, and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1759.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct *u* charts for Defects and Flaws:

```
proc shewhart data=Measures;
   uchart (Defects Flaws)*Sample / subgroupn=50;
run;
```

Note that when data are read from a DATA= data set with the UCHART statement, the SUBGROUPN= option (which specifies the number of inspection units per subgroup) is required.

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding UCHART statement, Sample is the subgroup variable. For details, see "Subgroup Variables" on page 1871.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2026 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the number of nonconformities per unit.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2025 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create a *u* chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   uchart Defects*Sample='*' / subgroupn=100;
run;
```

options

enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options", which follows, lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1946 describes each option in detail.

Summary of Options

The following tables list the UCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1946.

Table 17.59 UCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	requests probability limits for chart
LIMITN=	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
PROBLIMITS=	requests probability limits at discrete values
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
ACTUALALPHA	displays the actual probability of a point being outside the control limits in the control limits legend
CINFILL=	specifies color for area inside control limits
CLIMITS=	specifies color of control limits, central line, and related labels
LCLLABEL=	specifies label for lower control limit
LIMLABSUBCHAR=	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	specifies line type for control limits
NDECIMAL=	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line

Table 17.59 *continued*

Option	Description
NOLCL	suppresses display of lower control limit
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit
UCLLABEL=	specifies label for upper control limit
USYMBOL=	specifies label for central line
WLIMITS=	specifies width for control limits and central line
Standard Value Options	
TYPE=	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
U0=	specifies known average number of nonconformities per unit
Options for Plotting and Labeling Points	
ALLLABEL=	labels every point on *u* chart
CLABEL=	specifies color for labels
CCONNECT=	specifies color for line segments that connect points on chart
CFRAMELAB=	specifies fill color for frame around labeled points
CNEEDLES=	specifies color for needles that connect points to central line
COUT=	specifies color for portions of line segments that connect points outside control limits
COUTFILL=	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	specifies angle at which labels are drawn
LABELFONT=	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=	labels points outside control limits
SYMBOLLEGEND=	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=	specifies width of needles

Table 17.59 *continued*

Option	Description
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	computes zone widths independently above and below center line
NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	suppresses tests across *phase* boundaries
TESTS=	specifies tests for special causes
TEST2RUN=	specifies length of pattern for Test 2
TEST3RUN=	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL=	provides labels for points where test is positive
TESTLABEL*n*=	specifies label for *n*th test for special causes
TESTNMETHOD=	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=	enables tests for special causes to be reset
WESTGARD=	requests that Westgard rules be applied
ZONELABELS	adds labels A, B, and C to zone lines
ZONES	adds lines delineating zones A, B, and C
ZONEVALPOS=	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	specifies color for labels indicating points where test is positive
CTESTSYMBOL=	specifies color for symbol used to plot points where test is positive
CZONES=	specifies color for lines and labels delineating zones A, B, and C
LTESTS=	specifies type of line connecting points where test is positive
LZONES=	specifies line type for lines delineating zones A, B, and C
TESTFONT=	specifies software font for labels at points where test is positive
TESTHEIGHT=	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	specifies plot symbol for points where test is positive
TESTSYMBOLHT=	specifies symbol height for points where test is positive
WTESTS=	specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill colors for frame for plot area

Table 17.59 *continued*

Option	Description
CTEXT=	specifies color for tick mark values and axis labels
DISCRETE	produces horizontal axis for discrete numeric group values
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis
VFORMAT=	specifies format for vertical axis tick mark labels
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=	specifies width of axis lines
Plot Layout Options	
ALLN	plots means for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes

Table 17.59 *continued*

Option	Description
NPANELPOS=	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	specifies number of pages or screens to be used to display chart
ZEROSTD	displays u chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= options
CVREF=	specifies color for lines requested by VREF= options
HREF=	specifies position of reference lines perpendicular to horizontal axis
HREFDATA=	specifies position of reference lines perpendicular to horizontal axis
HREFLABELS=	specifies labels for HREF= lines
HREFLABPOS=	specifies position of HREFLABELS= labels
LHREF=	specifies line type for HREF= lines
LVREF=	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	specifies position of reference lines perpendicular to vertical axis
VREFLABELS=	specifies labels for VREF= lines
VREFLABPOS=	position of VREFLABELS= labels
Grid Options	
CGRID=	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=	specifies line type for grid requested with the ENDGRID option
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme points are clipped
CLIPLEGEND=	specifies text for clipping legend
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features chart

Table 17.59 *continued*

Option	Description
DESCRIPTION=	specifies description of *u* chart's GRSEG catalog entry
FONT=	specifies software font for labels and legends on charts
NAME=	specifies name of *u* chart's GRSEG catalog entry
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	shades areas between control limits and connected points lying outside the limits
STARFILL=	specifies a variable identfying groups of stars filled with different colors
STARS=	specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	specifies the control limit infill transparency
MARKERS	plots subgroup points with markers
NOBLOCKREF	suppresses block and phase reference lines
NOBLOCKREFFILL	suppresses block and phase wall fills
NOFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOPHASEREF	suppresses block and phase reference lines
NOPHASEREFFILL	suppresses block and phase wall fills
NOREF	suppresses block and phase reference lines
NOREFFILL	suppresses block and phase wall fills
NOSTARFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	specifies a graph footnote
ODSFOOTNOTE2=	specifies a secondary graph footnote
ODSLEGENDEXPAND	specifies that legend entries contain all levels observed in the data
ODSTITLE=	specifies a graph title
ODSTITLE2=	specifies a secondary graph title
OUTFILLTRANSPARENCY=	specifies control limit outfill transparency
OVERLAYURL=	specifies URLs to associate with overlay points
PHASEPOS=	specifies vertical position of phase legend

Table 17.59 *continued*

Option	Description
PHASEREFLEVEL=	associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	draws central line labels using a simulated software font
STARTRANSPARENCY=	specifies star fill transparency
URL=	specifies a variable whose values are URLs to be associated with subgroups
Input Data Set Options	
MISSBREAK	specifies that observations with missing values are not to be processed
SUBGROUPN	specifies subgroup sample sizes as constant number *n* or as values of variable in a DATA= data set
Output Data Set Options	
OUTHISTORY=	creates output data set containing subgroup summary statistics
OUTINDEX=	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	creates output data set containing control limits
OUTTABLE=	creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for *block-variable* legend
BLOCKLABTYPE=	specifies text size of *block-variable* legend
BLOCKPOS=	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	specifies fill colors for frames enclosing variable labels in *block-variable* legend

Table 17.59 *continued*

Option	Description
CBLOCKVAR=	specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE=	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	specifies color for STARCIRCLES= circles
CSTARFILL=	specifies color for filling stars
CSTAROUT=	specifies outline color for stars exceeding inner or outer circles
CSTARS=	specifies color for outlines of stars
LSTARCIRCLES=	specifies line types for STARCIRCLES= circles
LSTARS=	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	specifies reference circles for stars
STARINRADIUS=	specifies inner radius of stars
STARLABEL=	specifies vertices to be labeled
STARLEGEND=	specifies style of legend for star vertices
STARLEGENDLAB=	specifies label for STARLEGEND= legend
STAROUTRADIUS=	specifies outer radius of stars
STARSPECS=	specifies method used to standardize vertex variables
STARSTART=	specifies angle for first vertex
STARTYPE=	specifies graphical style of star
STARVERTICES=	superimposes star at each point on chart
WSTARCIRCLES=	specifies width of STARCIRCLES= circles
WSTARS=	specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	specifies colors for overlay line segments
COVERLAY=	specifies colors for overlay plots
COVERLAYCLIP=	specifies color for clipped points on overlays
LOVERLAY=	specifies line types for overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=	specifies variables to overlay on chart

Table 17.59 *continued*

Option	Description
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
OVERLAYHTML=	specifies links to associate with overlay points
OVERLAYID=	specifies labels for overlay points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols for overlays
OVERLAYSYMHT=	specifies symbol heights for overlays
WOVERLAY=	specifies widths of overlay line segments
Options for Interactive Control Charts	
HTML=	specifies a variable whose values create links to be associated with subgroups
HTML_LEGEND=	specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	specifies plot character for clipped points
CONNECTCHAR=	specifies character used to form line segments that connect points on chart
HREFCHAR=	specifies line character for HREF= lines
SYMBOLCHARS=	specifies characters indicating *symbol-variable*
TESTCHAR=	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	specifies line character for VREF= lines
ZONECHAR=	specifies character for lines that delineate zones for tests for special causes

Details: UCHART Statement

Constructing Charts for Nonconformities per Unit (u Charts)

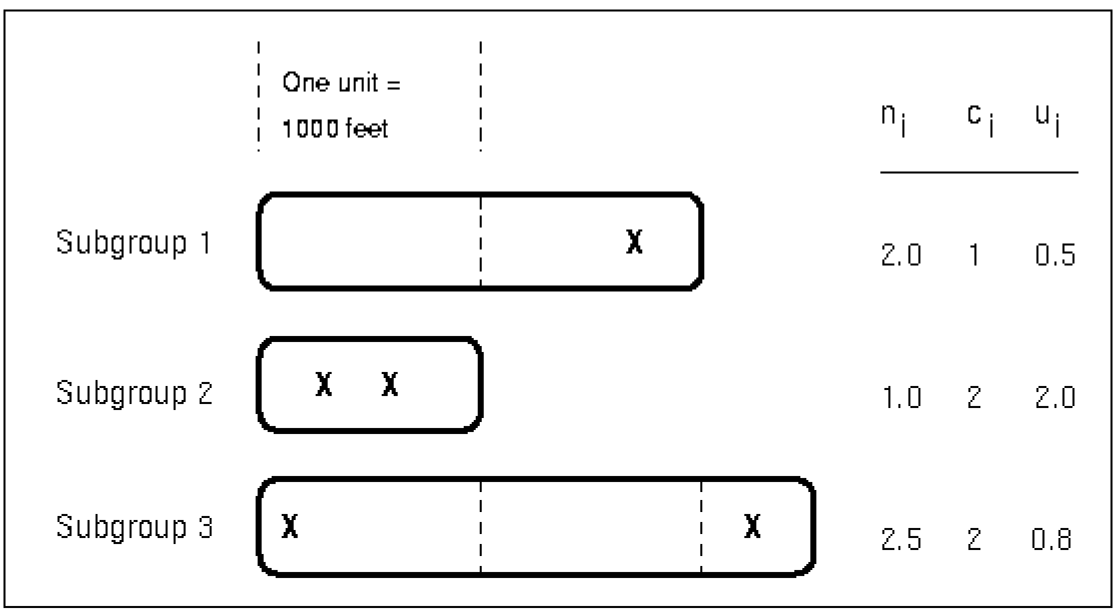
The following notation is used in this section:

u	expected number of nonconformities per unit produced by process
u_i	number of nonconformities per unit in the ith subgroup. In general, $u_i = c_i/n_i$.
c_i	total number of nonconformities in the ith subgroup
n_i	number of inspection units in the ith subgroup
$\bar{u}$	average number of nonconformities per unit taken across subgroups. The quantity $\bar{u}$ is computed as a weighted average: $$\bar{u} = \frac{n_1u_1 + \cdots + n_Nu_N}{n_1 + \cdots + n_N} = \frac{c_1 + \cdots + c_N}{n_1 + \cdots + n_N}$$
N	number of subgroups
χ^2_ν	has a central χ^2 distribution with ν degrees of freedom

Plotted Points

Each point on a u chart indicates the number of nonconformities per unit (u_i) in a subgroup. For example, Figure 17.95 displays three sections of pipeline that are inspected for defective welds (indicated by an X). Each section represents a *subgroup* composed of a number of *inspection units*, which are 1000-foot-long sections. The number of units in the ith subgroup is denoted by n_i, which is the subgroup sample size.

Figure 17.95 Terminology for c Charts and u Charts



The *number of nonconformities* in the ith subgroup is denoted by c_i. The *number of nonconformities per unit* in the ith subgroup is denoted by $u_i = c_i/n_i$. In Figure 17.95, the number of defective welds per unit in the third subgroup is $u_3 = 2/2.5 = 0.8$.

A u chart plots the quantity u_i for the ith subgroup. A c chart plots the quantity c_i for the ith subgroup (see "CCHART Statement: SHEWHART Procedure" on page 1436). An advantage of a u chart is that the value of the central line at the ith subgroup does not depend on n_i. This is not the case for a c chart, and consequently, a u chart is often preferred when the number of units n_i is not constant across subgroups.

Central Line

On a u chart, the central line indicates an estimate of u, which is computed as $\bar{u}$ by default. If you specify a known value (u_0) for u, the central line indicates the value of u_0.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of u_i above and below the central line. The default limits are computed with k = 3 (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that u_i exceeds the limits

The lower and upper control limits, LCLU and UCLU, respectively, are given by

$$\begin{aligned}\text{LCLU} &= \max\left(\bar{u} - k\sqrt{\bar{u}/n_i}\ ,0\right)\\ \text{UCLU} &= \bar{u} + k\sqrt{\bar{u}/n_i}\end{aligned}$$

The limits vary with n_i.

The upper probability limit UCLU for u_i can be determined using the fact that

$$\begin{aligned}P\{u_i > \text{UCLU}\} &= 1 - P\{u_i \le \text{UCLU}\}\\ &= 1 - P\{c_i \le n_i\text{UCLU}\}\\ &= 1 - P\{\chi^2_{2(n_i\times\text{UCLU}+1)} \ge 2n_i\bar{u}\}\end{aligned}$$

The limit UCLU is then calculated by setting

$$1 - P\{\chi^2_{2(n_i\times\text{UCLU}+1)} \ge 2n_i\bar{u}\} = \alpha/2$$

and solving for UCLU.

Likewise, the lower probability limit LCLU for u_i can be determined using the fact that

$$\begin{aligned}P\{u_i < \text{LCLU}\} &= P\{u_i \le \text{LCLU} - 1\}\\ &= P\{c_i \le n_i\text{LCLU} - 1\}\\ &= P\{\chi^2_{2(n_i\times(\text{LCLU}-1)+1)} > 2n_i\bar{u}\}\\ &= P\{\chi^2_{2(n_i\text{LCLU}} > 2n_i\bar{u}\}\end{aligned}$$

The limit LCLU is then calculated by setting

$$P\{\chi^2_{2(n_i\text{LCLU})} > 2n_i\bar{u}\} = \alpha/2$$

and solving for LCLU. For more information, refer to Johnson, Kotz, and Kemp (1992). This assumes that the process is in statistical control and that c_i has a Poisson distribution. Note that the probability limits vary with n_i and are asymmetric around the central line. If a standard value u_0 is available for u, replace $\bar{u}$ with u_0 in the formulas for the control limits.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify u_0 with the U0= option or with the variable _U_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 17.60 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLU	lower control limit for number of nonconformities per unit
LIMITN	sample size associated with the control limits
SIGMAS	multiple (k) of standard error of u_i
SUBGRP	*subgroup-variable* specified in the UCHART statement
TYPE	type (estimate or standard value) of _U_
_U_	value of central line of u chart ($\bar{u}$ or u_0)
UCLU	upper control limit for number of nonconformities per unit
VAR	*process* specified in the UCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value 'V' is assigned to the variables _LCLU_, _UCLU_, and _LIMITN_.
2. If the limits are defined in terms of a multiple k of the standard error of u_i, the value of _ALPHA_ is computed as $P\{u_i < \text{\_LCLU\_}\} + P\{u_i > \text{\_UCLU\_}\}$, provided that n_i is a constant. Otherwise, _ALPHA_ is assigned the special missing value 'V'.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $(\text{\_UCLU\_} - \text{\_U\_})/\sqrt{\text{\_U\_}/\text{\_LIMITN\_}}$, provided that n_i is a constant. Otherwise, _SIGMAS_ is assigned the special missing value *V*.

4. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the UCHART statement. For an example, see “Saving Control Limits” on page 1759.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup number of nonconformities per unit variable named by *process* suffixed with *U*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the UCHART statement. For example, consider the following statements:

```
proc shewhart data=Fabric;
   uchart (Flaws nDefects)*lot / outhistory=Summary
                                 subgroupn = 10;
run;
```

The data set Summary contains the variables Lot, FlawsU, FlawsN, nDefectsU, and nDefectsN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see “Saving Nonconformities per Unit” on page 1765. Note that an OUTHISTORY= data set created with the UCHART statement can be used as a HISTORY= data set by either the CCHART statement or the UCHART statement.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on *u* chart
LCLU	lower control limit for number of nonconformities per unit
LIMITN	nominal sample size associated with the control limits
SIGMAS	multiple (*k*) of the standard error associated with the control limits
subgroup	values of the subgroup variable
SUBU	subgroup number of nonconformities per unit
SUBN	subgroup sample size
TESTS	tests for special causes signaled on *u* chart
_U_	average number of nonconformities per unit
UCLU	upper control limit for number of nonconformities per unit
VAR	*process* specified in the UCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= option, respectively, or with the corresponding variables in a LIMITS= data set).

2. The variable _TESTS_ is saved if you specify the TESTS= option. The *k*th character of a value of _TESTS_ is *k* if Test *k* is positive at that subgroup. For example, if you request the first four tests (the ones appropriate for *u* charts) and Tests 2 and 4 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, a 4 for the fourth character, and blanks for the other six characters.

3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1759.

ODS Tables

The following table summarizes the ODS tables that you can request with the UCHART statement.

Table 17.61 ODS Tables Produced with the UCHART Statement

Table Name	Description	Options
UCHART	*u* chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*). **NOTE:** In SAS/QC 13.1 the SHEWHART procedure does not support the creation of graphs that are editable with the ODS Graphics Editor.

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. UCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1946.

When ODS Graphics is in effect, the UCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 17.62.

Table 17.62 ODS Graphics Produced by the UCHART Statement

ODS Graph Name	Plot Description
UChart	*u* chart

See Chapter 3, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read defect counts for subgroup samples from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the UCHART statement must be a SAS variable in the data set. This variable provides the defect count (number of nonconformities) for subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, specified in the UCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. The data set should contain one observation per subgroup. When you use a DATA= data set with the UCHART statement, the SUBGROUPN= option (which specifies the number of inspection units per subgroup) is required. Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2031).

For an example of a DATA= data set, see "Creating u Charts from Defect Count Data" on page 1756.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   uchart Defects*Lot / subgroupn = 10;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLU_, _U_, and _UCLU_, which specify the control limits
- the variable _U_, which is used to calculate the control limits (see "Control Limits" on page 1778)

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE' and 'STANDARD'.

- BY variables are required if specified with a BY statement.

For an example, see “Reading Preestablished Control Limits” on page 1761.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART procedure or to read output data sets created with SAS summarization procedures.

A HISTORY= data set used with the UCHART statement must contain the following variables:

- *subgroup-variable*
- subgroup number of nonconformities per unit variable for each *process*
- subgroup sample size variable (number of units per subgroup) for each *process*

The names of the variables containing the number of nonconformities per unit and subgroup sample sizes must be the *process* name concatenated with the special suffix characters *U* and *N*, respectively. For example, consider the following statements:

```
proc shewhart history=Summary;
   uchart (Flaws nDefects)*Lot;
run;
```

The data set Summary must include the variables Lot, FlawsU, FlawsN, nDefectsU, and nDefectsN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (see “Displaying Stratification in Phases” on page 2031 for an example).

For an example of a HISTORY= data set, see “Creating u Charts from Nonconformities per Unit” on page 1762.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure or to create your own TABLE= data set. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2096.

The following table lists the variables required in a TABLE= data set used with the UCHART statement:

Table 17.63 Variables Required in a TABLE= Data Set

Variable	Description
LCLU	lower control limit for nonconformities per unit
LIMITN	nominal sample size associated with the control limits
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBU	subgroup number of nonconformities per unit
_U_	average number of nonconformities per unit
UCLU	upper control limit for nonconformities per unit

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1759.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup defects per unit variable
Vertical	TABLE=	_SUBU_

For an example, see "Labeling Axes" on page 2063.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: UCHART Statement

This section provides advanced examples of the UCHART statement.

Example 17.31: Applying Tests for Special Causes

NOTE: See *u Chart-Applying Tests for Special Causes* in the SAS/QC Sample Library.

This example illustrates how you can apply tests for special causes to make *u* charts more sensitive to special causes of variation.

A textile company inspects rolls of fabric for defects. The rolls are one meter wide and 30 meters long. The following statements create a SAS data set named Fabric3, which contains the number of fabric defects for 20 rolls of fabric:

```
data Fabric3;
   input Roll Defects @@;
   datalines;
 1   6    2   9    3  14    4  17
 5   3    6   8    7   9    8   2
 9  14   10   1   11   3   12   5
13   6   14   9   15  10   16  12
17  11   18   4   19   9   20   4
;
```

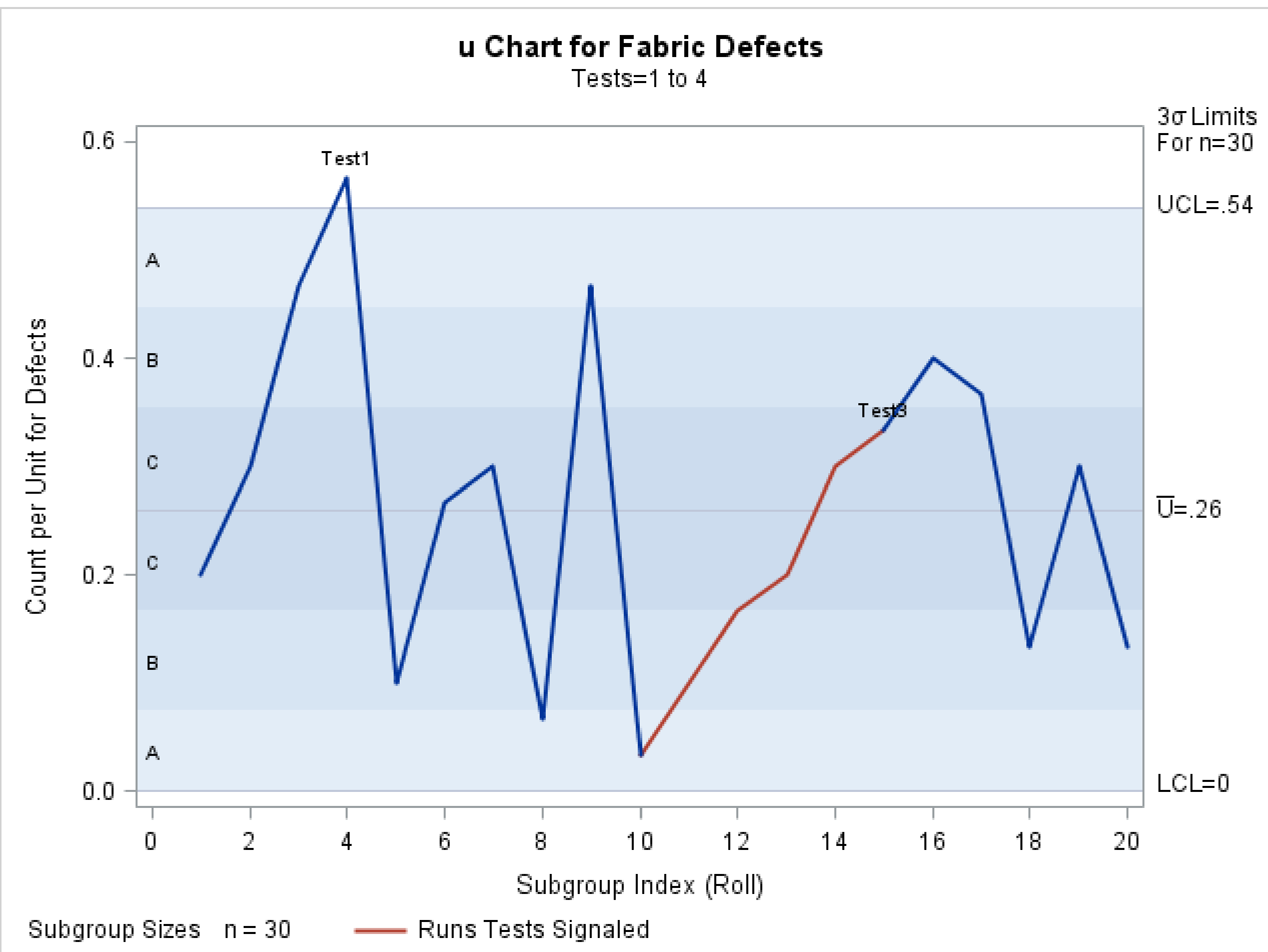
The following statements create a *u* chart and tabulate the information on the chart. The chart and tables are shown in Output 17.31.1 and Output 17.31.2.

```
ods graphics on;
title1 'u Chart for Fabric Defects';
title2 'Tests=1 to 4';
proc shewhart data=Fabric3;
   uchart Defects*Roll / subgroupn = 30
                         tests     = 1 to 4
                         odstitle  = title
                         odstitle2 = title2
                         tabletests
                         zonelabels;
run;
```

The TESTS= option requests Tests 1, 2, 3, and 4, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2073. Only Tests 1, 2, 3, and 4 are recommended for u charts. The ZONELABELS option requests the zone lines, which are used to define the tests, and displays labels for the zones. The TABLETESTS option requests a table of the values of u_i and the control limits, together with a column indicating the subgroups at which the tests are positive.

Output 17.31.1 and Output 17.31.2 indicate that Test 1 is positive for Roll 4 and Test 3 is positive at Roll 15.

Output 17.31.1 Tests for Special Causes Displayed on u Chart



Output 17.31.2 Tabular Form of u Chart

u Chart for Fabric Defects
Tests=1 to 4

The SHEWHART Procedure

u Chart Summary for Defects

3 Sigma Limits with n=30 for Count per Unit

Roll	Subgroup Sample Size	Lower Limit	Subgroup Count per Unit	Upper Limit	Special Tests Signaled
1	30.0000	0	0.20000000	0.53928480	
2	30.0000	0	0.30000000	0.53928480	
3	30.0000	0	0.46666667	0.53928480	
4	30.0000	0	0.56666667	0.53928480	1
5	30.0000	0	0.10000000	0.53928480	
6	30.0000	0	0.26666667	0.53928480	
7	30.0000	0	0.30000000	0.53928480	
8	30.0000	0	0.06666667	0.53928480	
9	30.0000	0	0.46666667	0.53928480	
10	30.0000	0	0.03333333	0.53928480	
11	30.0000	0	0.10000000	0.53928480	
12	30.0000	0	0.16666667	0.53928480	
13	30.0000	0	0.20000000	0.53928480	
14	30.0000	0	0.30000000	0.53928480	
15	30.0000	0	0.33333333	0.53928480	3
16	30.0000	0	0.40000000	0.53928480	
17	30.0000	0	0.36666667	0.53928480	
18	30.0000	0	0.13333333	0.53928480	
19	30.0000	0	0.30000000	0.53928480	
20	30.0000	0	0.13333333	0.53928480	

Example 17.32: Specifying a Known Expected Number of Nonconformities

NOTE: See *u Chart-Known Expected Number of Nonconformities* in the SAS/QC Sample Library.

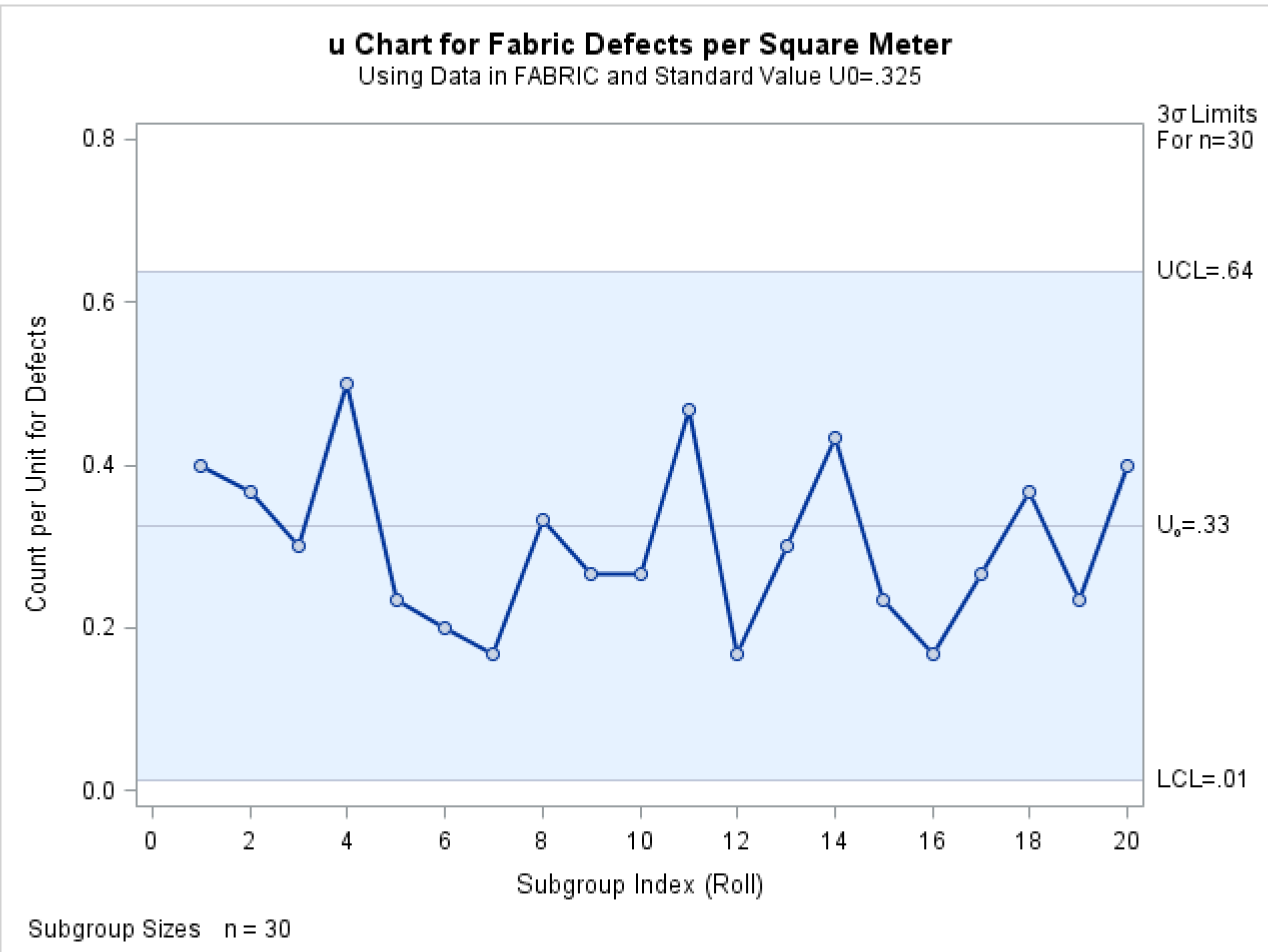
This example illustrates how you can create a u chart based on a known (standard) value u_0 for the expected number of nonconformities per unit.

A u chart is used to monitor the number of defects per square meter of fabric. The defect counts are provided as values of the variable Defects in the data set Fabric (see "Creating u Charts from Defect Count Data" on page 1756). Based on previous testing, it is known that $u_0 = 0.325$. The following statements create a u chart with control limits derived from this value:

```
ods graphics on;
title  'u Chart for Fabric Defects per Square Meter';
title2 'Using Data in FABRIC and Standard Value U0=.325';
proc shewhart data=Fabric;
   uchart Defects*Roll / subgroupn = 30
                         u0        = 0.325
                         usymbol   = u0
                         odstitle  = title
                         odstitle2 = title2
                         markers;
run;
```

The chart is shown in Output 17.32.1. The U0= option specifies u_0, and the USYMBOL= option requests a label for the central line indicating that the line represents a standard value.

Output 17.32.1 A u Chart with Standard Value u_0



Since all the points lie within the control limits, the process is in statistical control.

Alternatively, you can specify u_0 as the value of the variable _U_ in a LIMITS= data set, as follows:

```
data Tlimits;
   length _subgrp_ _var_ _type_ $8;
   _u_      = .325;
   _limitn_ = 30;
   _type_   = 'STANDARD';
   _subgrp_ = 'Roll';
   _var_    = 'Defects';

proc shewhart data=Fabric limits=Tlimits;
   uchart Defects*Roll / subgroupn=30
                         usymbol  =u0;
run;
```

The chart produced by these statements is identical to the one in Output 17.32.1. For further details, see “LIMITS= Data Set” on page 1783.

Example 17.33: Creating u Charts for Varying Numbers of Units

NOTE: See *u Charts-Varying Number of Inspection Units* in the SAS/QC Sample Library.

In the fabric manufacturing process described in “Creating u Charts from Defect Count Data” on page 1756, each roll of fabric is 30 meters long, and an inspection unit is defined as one square meter. Thus, there are 30 inspection units in each subgroup sample. Suppose now that the length of each piece of fabric varies. The following statements create a SAS data set (Fabrics2) that contains the number of fabric defects and size (in square meters) of 25 pieces of fabric:

```
data Fabrics2;
   input Roll Defects Squaremeters @@;
   datalines;
 1  7 30.0   2 11 27.6   3 15 30.4   4  6 34.8   5 11 26.0
 6 15 28.6   7  5 28.0   8 10 30.2   9  8 28.2  10  3 31.4
11  3 30.3  12 14 27.8  13  3 27.0  14  9 30.0  15  7 32.1
16  6 34.8  17  7 26.5  18  5 30.0  19 14 31.3  20 13 31.6
21 11 29.4  22  6 28.6  23  6 27.5  24  9 32.6  25 11 31.7
;
```

A partial listing of Fabrics2 is shown in Output 17.33.1.

Output 17.33.1 The Data Set Fabrics2

Number of Fabric Defects

Roll	Defects	Squaremeters
1	7	30.0
2	11	27.6
3	15	30.4
4	6	34.8
5	11	26.0

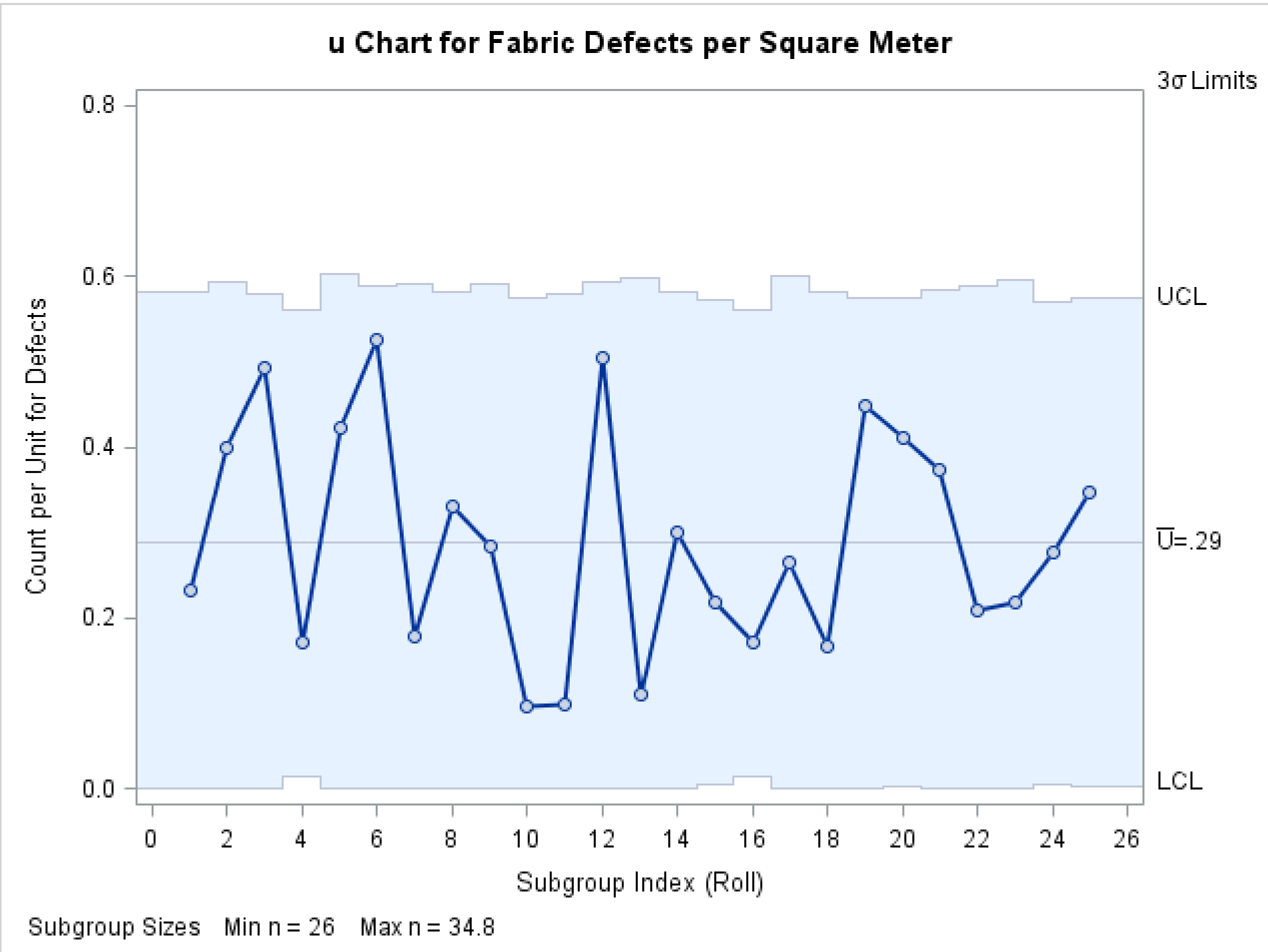
The variable Roll contains the roll number, the variable Defects contains the number of defects in each piece of fabric, and the variable Squaremeters contains the size of each piece.

The following statements request a *u* chart for the number of defects per square meter:

```
ods graphics on;
title 'u Chart for Fabric Defects per Square Meter';
proc shewhart data=Fabrics2;
   uchart Defects*Roll / subgroupn = Squaremeters
                         outlimits = Fablimits
                         odstitle  = title
                         markers;
run;
```

The *u* chart is shown in Output 17.33.2, and the data set Fablimits is listed in Output 17.33.3.

Output 17.33.2 A *u* Chart with Varying Number of Units per Subgroup



Note that the control limits vary with the number of units per subgroup (subgroup sample size). The legend in the lower left corner indicates the minimum and maximum subgroup sample sizes.

Output 17.33.3 The Control Limits Data Set Fablimits

Control Limits for Fabric Defects

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLU_	_U_	_UCLU_
Defects	Roll	ESTIMATE	V	V	3	V	0.28805	V

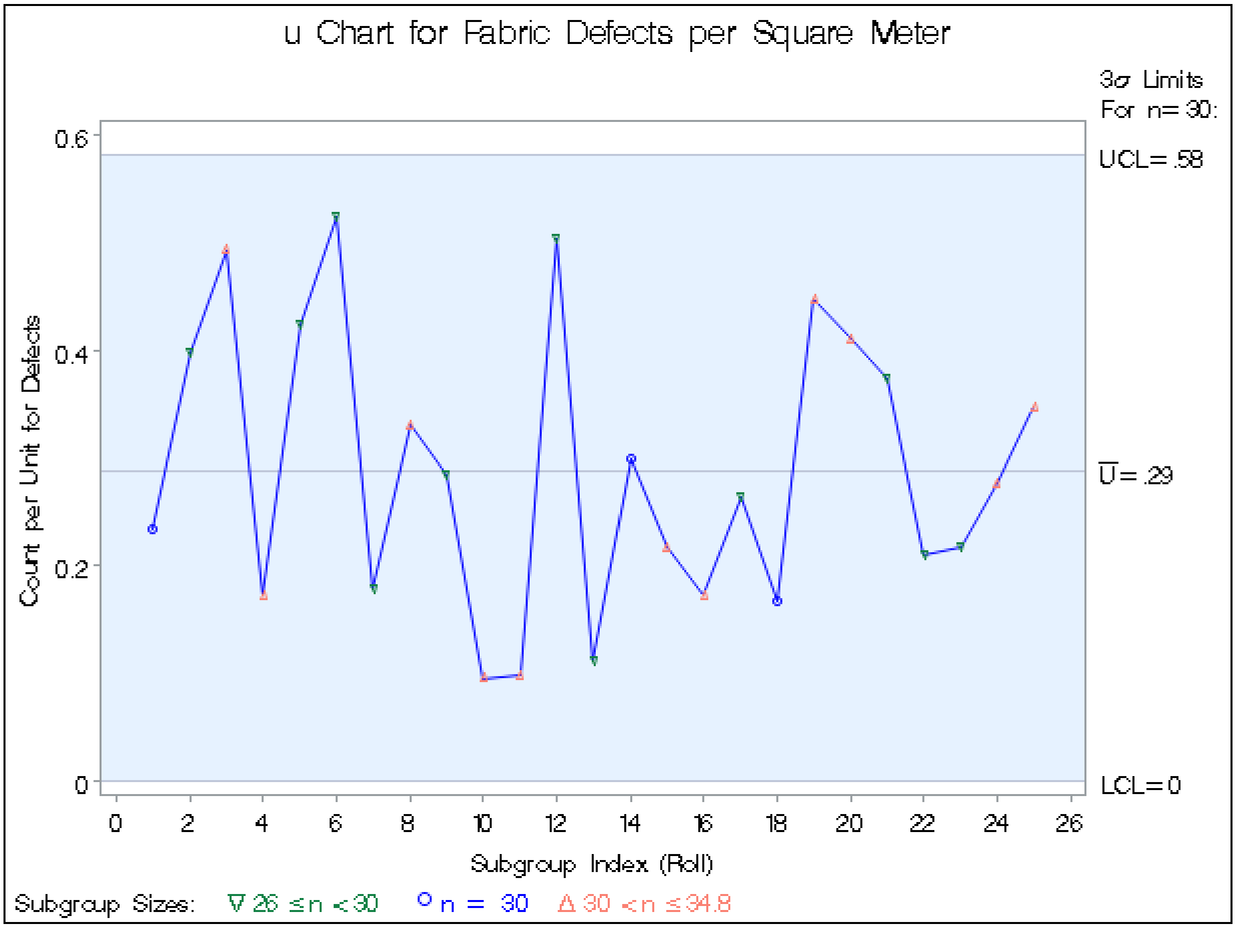
Output 17.33.3 shows that the variables _LIMITN_, _ALPHA_, _LCLU_, and _UCLU_ have the special missing value *V*, indicating that these variables vary with the sample size.

The following statements request a *u* chart with a fixed sample size of 30.0 for the control limits. In other words, the control limits are computed as if each piece of fabric were 30 meters long.

```
ods graphics off;
symbol1 c=blue v=circle;
symbol2 c=vig;
symbol3 c=salmon;
title 'u Chart for Fabric Defects per Square Meter';
proc shewhart data=Fabrics2;
   uchart Defects*Roll / subgroupn = Squaremeters
                         outlimits = Fablimits2
                         limitn    = 30
                         alln
                         nmarkers;
run;
```

The ALLN option specifies that points are to be displayed for all subgroups, regardless of their sample size. By default, when you specify the LIMITN= option, only points for subgroups whose sample size matches the LIMITN= value are displayed. The NMARKERS option requests special symbols that identify points for which the subgroup sample size differs from the nominal sample size of 30. The chart is shown in Output 17.33.4.

Output 17.33.4 Control Limits Based on Fixed Subgroup Sample Size



In Output 17.33.4, no points lie outside the control limits, indicating that the process is in control. However, you should be careful when interpreting charts that use a nominal sample size, since the fixed control limits based on this value are only an approximation. Output 17.33.5 lists the data set Fablimits2, which contains the fixed control limits displayed in Output 17.33.4.

Output 17.33.5 The Fixed Control Limits Data Set Fablimits2

Fixed Control Limits for Fabric Defects

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLU_	_U_	_UCLU_
Defects	Roll	ESTIMATE	30	.002444992	3	0	0.28805	0.58201

XCHART Statement: SHEWHART Procedure

Overview: XCHART Statement

The XCHART statement creates an $\bar{X}$ chart for subgroup means, which is used to analyze the central tendency of a process.

You can use options in the XCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted means or as probability limits
- tabulate subgroup sample sizes, subgroup means, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes and subgroup means in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify one of several methods for estimating the process standard deviation
- specify whether subgroup standard deviations or subgroup ranges are used to estimate the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- create a secondary chart that displays a time trend removed from the data (see "Displaying Trends in Process Data" on page 2054)
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the chart more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing $\bar{X}$ charts with the XCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.

- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 3, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

NOTE: When working with variables data, you should analyze the variability of the process as well as its central tendency. You can use the XRCHART statement or the XSCHART statement in the SHEWHART procedure for this purpose.

Getting Started: XCHART Statement

This section introduces the XCHART statement with simple examples that illustrate the most commonly used options. Complete syntax for the XCHART statement is presented in the section "Syntax: XCHART Statement" on page 1806, and advanced examples are given in the section "Examples: XCHART Statement" on page 1829.

Creating Charts for Means from Raw Data

NOTE: See *Mean (X-BAR) Chart Examples* in the SAS/QC Sample Library.

Subgroup samples of five parts are taken from the manufacturing process at regular intervals, and the width of a critical gap in each part is measured in millimeters. The following statements create a SAS data set named Partgaps, which contains the gap width measurements for 21 samples:

```
data Partgaps;
   input Sample @;
   do i=1 to 5;
      input Partgap @;
      output;
   end;
   drop i;
   label Partgap='Gap Width'
         Sample ='Sample Index';
   datalines;
 1 255 270 268 290 267
 2 260 240 265 262 263
 3 238 236 260 250 256
 4 260 242 281 254 263
 5 268 260 279 289 269
 6 270 249 265 253 263
 7 280 260 256 256 243
 8 229 266 250 243 252
 9 250 270 245 273 262
10 248 258 247 266 256
11 280 251 252 270 287
12 245 253 243 279 245
13 268 260 289 275 273
14 264 286 275 271 279
15 271 257 263 247 247
```

```
16 291 250 273 265 266
17 228 253 240 260 264
18 270 260 269 245 276
19 259 257 246 271 257
20 252 244 230 266 248
21 254 251 239 233 263
;
```

A partial listing of Partgaps is shown in Figure 17.96.

Figure 17.96 Partial Listing of the Data Set Partgaps

The Data Set PARTGAPS

Sample	Partgap
1	255
1	270
1	268
1	290
1	267
2	260
2	240
2	265
2	262
2	263

The data set Partgaps is said to be in "strung-out" form, since each observation contains the sample number and gap width measurement for a single part. The first five observations contain the gap widths for the first sample, the second five observations contain the gap widths for the second sample, and so on. Because the variable Sample classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable Partgap contains the gap width measurements and is referred to as the *process variable* (or *process* for short).

The within-subgroup variability of the gap widths is known to be stable. You can use an $\bar{X}$ chart to determine whether their mean level is in control. The following statements create the $\bar{X}$ chart shown in Figure 17.97:

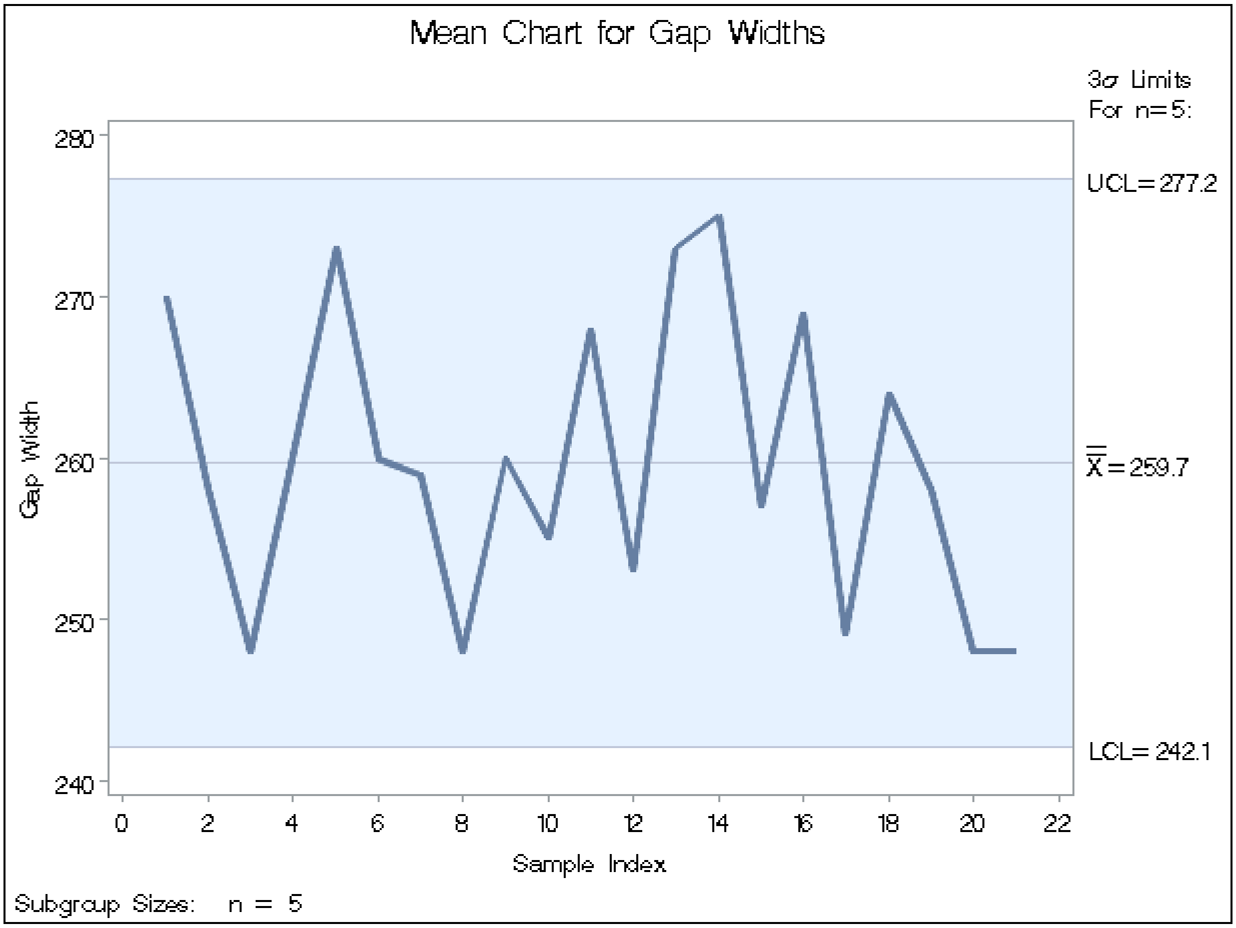
```
ods graphics off;
title 'Mean Chart for Gap Widths';
proc shewhart data=Partgaps;
   xchart Partgap*Sample;
run;
```

This example illustrates the basic form of the XCHART statement. After the keyword XCHART, you specify the *process* to analyze (in this case, Partgap) followed by an asterisk and the *subgroup-variable* (Sample). The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Each point on the $\bar{X}$ chart represents the average (mean) of the measurements for a particular sample. For instance, the mean plotted for the first sample is

$$\frac{255 + 270 + 268 + 290 + 267}{5} = 270$$

Figure 17.97 $\bar{X}$ Chart for Gap Width Data (Traditional Graphics)



Since all of the subgroup means lie within the control limits, it can be concluded that the mean level of the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 17.65. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1804.

For computational details, see "Constructing Charts for Means" on page 1817. For details on reading raw measurements, see "DATA= Data Set" on page 1823.

Creating Charts for Means from Subgroup Summary Data

NOTE: See *Mean (X-BAR) Chart Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create $\bar{X}$ charts using raw data (process measurements). However, in many applications, the data are provided as subgroup summary statistics. This example illustrates how you can use the XCHART statement with data of this type.

The following data set (Parts) provides the data from the preceding example in summarized form:

```
data Parts;
   input Sample PartgapX PartgapR;
   PartgapN=5;
   label PartgapX='Mean of Gap Width'
         Sample  ='Sample Index';
   datalines;
 1  270  35
 2  258  25
 3  248  24
 4  260  39
 5  273  29
 6  260  21
 7  259  37
 8  248  37
 9  260  28
10  255  19
11  268  36
12  253  36
13  273  29
14  275  22
15  257  24
16  269  41
17  249  36
18  264  31
19  258  25
20  248  36
21  248  30
;
```

A partial listing of Parts is shown in Figure 17.98. There is exactly one observation for each subgroup (note that the subgroups are still indexed by Sample). The variable PartgapX contains the subgroup means, the variable PartgapR contains the subgroup ranges, and the variable PartgapN contains the subgroup sample sizes (these are all five).

Figure 17.98 The Summary Data Set Parts

The Data Set PARTS

Sample	PartgapX	PartgapR	PartgapN
1	270	35	5
2	258	25	5
3	248	24	5
4	260	39	5
5	273	29	5

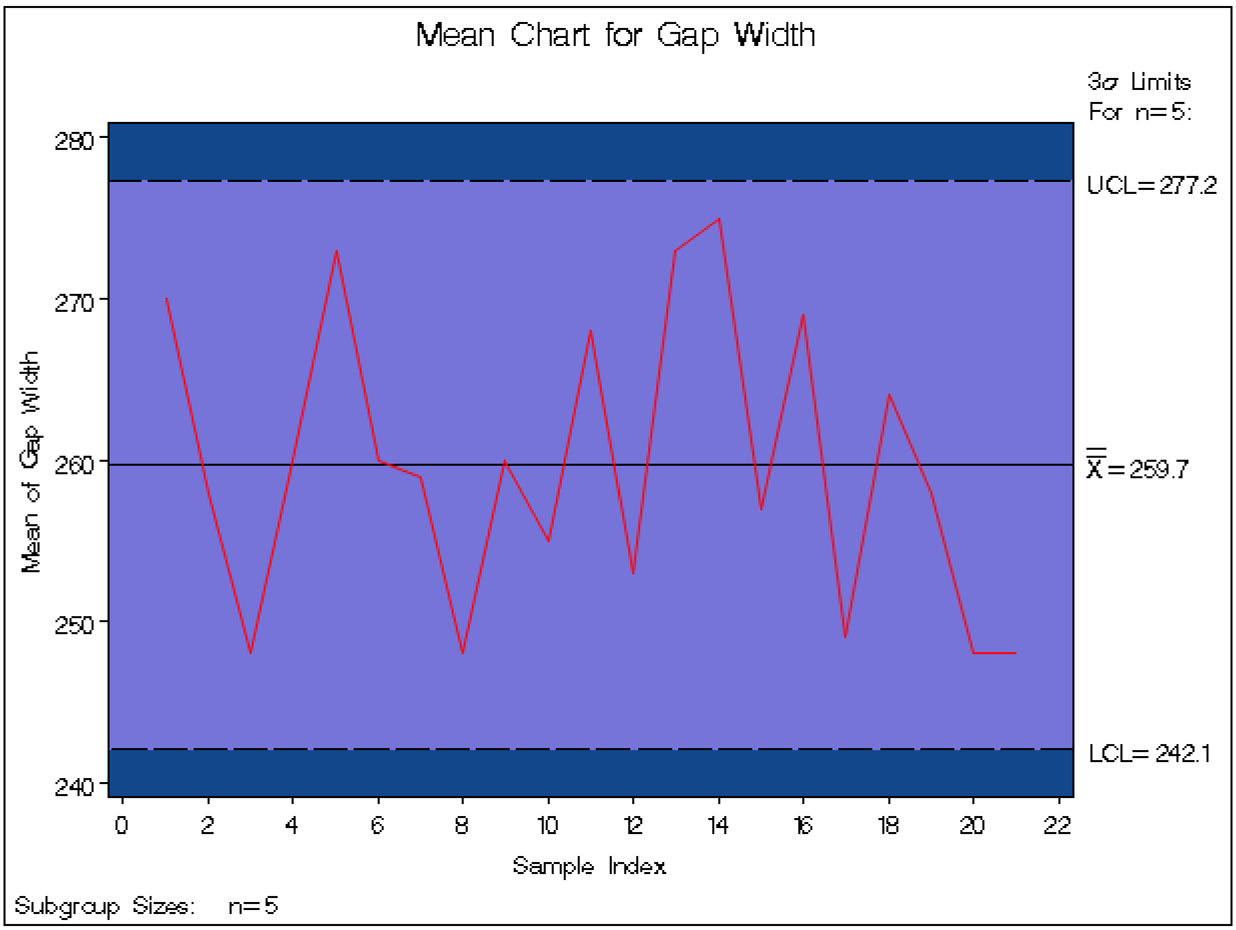
You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext=swiss;
title 'Mean Chart for Gap Width';
proc shewhart history=Parts;
   xchart Partgap*Sample / cframe   = vigb
                           cinfill  = vlib
                           cconnect = red;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the XCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting $\bar{X}$ chart is shown in Figure 17.99.

Note that Partgap is *not* the name of a SAS variable in the data set but is, instead, the common prefix for the names of the three SAS variables PartgapX, PartgapR, and PartgapN. The suffix characters *X*, *R*, and *N* indicate *mean*, *range*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in a HISTORY= data set with a single name (Partgap), which is referred to as the *process*. The name Sample specified after the asterisk is the name of the *subgroup-variable*.

Figure 17.99 $\bar{X}$ Chart from the Summary Data Set Parts (Traditional Graphics with NOGSTYLE)



In general, a HISTORY= input data set used with the XCHART statement must contain the following variables:

- subgroup variable
- subgroup mean variable
- either a subgroup range variable or a subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of the subgroup mean, range (or standard deviation), and sample size variables must begin with the *process* name specified in the XCHART statement and end with the special suffix characters *X*, *R* (or *S*), and *N*, respectively. If the names do not follow this convention, you can use the RENAME option in the PROC SHEWHART statement to rename the variables for the duration of the SHEWHART procedure step (see "Creating Charts for Means and Ranges from Summary Data" on page 1841).

If you specify the STDDEVIATIONS option in the XCHART statement, the HISTORY= data set must contain a subgroup standard deviation variable; otherwise, the HISTORY= data set must contain a subgroup range variable. The STDDEVIATIONS option specifies that the estimate of the process standard deviation σ is to be calculated from subgroup standard deviations rather than subgroup ranges. For example, in the following statements, the data set Parts2 must contain a subgroup standard deviation variable named PartgapS:

```
title 'Mean Chart for Gap Width';
proc shewhart history=Parts2;
   xchart Partgap*Sample='*' / stddeviations;
run;
```

Options such as STDDEVIATIONS are specified after the slash (/) in the XCHART statement. A complete list of options is presented in the section "Syntax: XCHART Statement" on page 1806.

In summary, the interpretation of *process* depends on the input data set.

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1824.

Saving Summary Statistics

NOTE: See *Mean (X-BAR) Chart Examples* in the SAS/QC Sample Library.

In this example, the XCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Partgaps and create a summary data set named Gaphist:

```
proc shewhart data=Partgaps;
   xchart Partgap*Sample / outhistory = Gaphist
                           nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the chart, which would be identical to the chart in Figure 17.97.

Figure 17.100 contains a partial listing of Gaphist.

Figure 17.100 The Summary Data Set Gaphist

Summary Data Set for Gap Widths

Sample	PartgapX	PartgapR	PartgapN
1	270	35	5
2	258	25	5
3	248	24	5
4	260	39	5
5	273	29	5

There are four variables in the data set Gaphist.

- Sample contains the subgroup index.
- PartgapX contains the subgroup means.
- PartgapR contains the subgroup ranges.
- PartgapN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *R*, and *N* to the *process* Partgap specified in the XCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

If you specify the STDDEVIATIONS option, the OUTHISTORY= data set includes a subgroup standard deviation variable rather than a subgroup range variable, as demonstrated by the following statements:

```
proc shewhart data=Partgaps;
   xchart Partgap*Sample / outhistory = Gaphist2
                           stddeviations
                           nochart;
run;
```

Figure 17.101 contains a partial listing of Gaphist2.

Figure 17.101 The Summary Data Set Gaphist2

Summary Data Set with Subgroup Standard Deviations

Sample	PartgapX	PartgapS	PartgapN
1	270	12.6293	5
2	258	10.2225	5
3	248	10.6771	5
4	260	14.2302	5
5	273	11.2027	5

The variable PartgapS, which contains the subgroup standard deviations, is named by adding the suffix character *S* to the *process* Partgap.

For more information, see "OUTHISTORY= Data Set" on page 1820.

Saving Control Limits

NOTE: See *Mean (X-BAR) Chart Examples* in the SAS/QC Sample Library.

You can save the control limits for an $\bar{X}$ chart in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1804) or modify the limits with a DATA step program.

The following statements read measurements from the data set Partgaps (see "Creating Charts for Means from Raw Data" on page 1795) and save the control limits displayed in Figure 17.97 in a data set named Gaplim:

```
proc shewhart data=Partgaps;
   xchart Partgap*Sample / outlimits = Gaplim
                           nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the chart. The data set Gaplim is listed in Figure 17.102.

Figure 17.102 The Data Set Gaplim Containing Control Limit Information

Control Limits for Gap Width Measurements

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
Partgap	Sample	ESTIMATE	5	.002699796	3	242.087	259.667	277.246

LCLR	_R_	_UCLR_	_STDDEV_
0	30.4762	64.4419	13.1028

The data set Gaplim contains one observation with the limits for *process* Partgap. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the means, and the variable _MEAN_ contains the central line. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*.

The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values.

The variables _LCLR_, _R_, and _UCLR_ are not used to create $\bar{X}$ charts, but they are included so the data set Gaplim can be used to create an *R* chart; see "XRCHART Statement: SHEWHART Procedure" on page 1837. If you specify the STDDEVIATIONS option in the XCHART statement, the variables _LCLS_, _S_, and _UCLS_ are included in the OUTLIMITS= data set. These variables can be used to create an *s* chart; see "XSCHART Statement: SHEWHART Procedure" on page 1886. For more information, see "OUTLIMITS= Data Set" on page 1818.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Partgaps;
   xchart Partgap*Sample / outtable=Gaptable
                           nochart;
run;
```

The data set Gaptable is listed in Figure 17.103.

Figure 17.103 The Data Set Gaptable

Summary Statistics and Control Limit Information

VAR	Sample	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_STDDEV_	_EXLIM_
Partgap	1	3	5	5	242.087	270	259.667	277.246	13.1028	
Partgap	2	3	5	5	242.087	258	259.667	277.246	13.1028	
Partgap	3	3	5	5	242.087	248	259.667	277.246	13.1028	
Partgap	4	3	5	5	242.087	260	259.667	277.246	13.1028	
Partgap	5	3	5	5	242.087	273	259.667	277.246	13.1028	
Partgap	6	3	5	5	242.087	260	259.667	277.246	13.1028	
Partgap	7	3	5	5	242.087	259	259.667	277.246	13.1028	
Partgap	8	3	5	5	242.087	248	259.667	277.246	13.1028	
Partgap	9	3	5	5	242.087	260	259.667	277.246	13.1028	
Partgap	10	3	5	5	242.087	255	259.667	277.246	13.1028	
Partgap	11	3	5	5	242.087	268	259.667	277.246	13.1028	
Partgap	12	3	5	5	242.087	253	259.667	277.246	13.1028	
Partgap	13	3	5	5	242.087	273	259.667	277.246	13.1028	
Partgap	14	3	5	5	242.087	275	259.667	277.246	13.1028	
Partgap	15	3	5	5	242.087	257	259.667	277.246	13.1028	
Partgap	16	3	5	5	242.087	269	259.667	277.246	13.1028	
Partgap	17	3	5	5	242.087	249	259.667	277.246	13.1028	
Partgap	18	3	5	5	242.087	264	259.667	277.246	13.1028	
Partgap	19	3	5	5	242.087	258	259.667	277.246	13.1028	
Partgap	20	3	5	5	242.087	248	259.667	277.246	13.1028	
Partgap	21	3	5	5	242.087	248	259.667	277.246	13.1028	

This data set contains one observation for each subgroup sample. The variables _SUBX_ and _SUBN_ contain the subgroup means and sample sizes. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits, and the variable _MEAN_ contains the central line. The variables _VAR_ and Sample contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1821.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Gaptable and display an $\bar{X}$ chart (not shown here) identical to the chart in Figure 17.97:

```
title 'Mean Chart for Gap Widths';
proc shewhart table=Gaptable;
   xchart Partgap*Sample;
label _SUBX_ = 'Gap Width';
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2096).

For more information, see "TABLE= Data Set" on page 1825.

Reading Preestablished Control Limits

NOTE: See *Mean (X-BAR) Chart Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Gaplim saved control limits computed from the measurements in Partgaps. This example shows how these limits can be applied to new data provided in the following data set:

```
data Gaps2;
   input Sample @;
   do i=1 to 5;
      input Partgap @;
      output;
   end;
   drop i;
   datalines;
22 287 265 248 263 271
23 267 253 285 251 271
24 249 252 277 269 241
25 243 248 263 282 261
26 287 266 256 278 242
27 251 262 243 274 245
28 256 245 244 243 272
29 262 247 252 277 266
30 244 269 263 278 261
31 245 264 246 242 273
32 272 257 277 265 241
33 251 249 240 260 261
34 289 277 275 273 261
35 267 286 275 261 272
36 266 256 247 255 241
37 291 267 267 252 262
38 258 245 264 245 281
39 277 267 241 272 244
40 252 267 272 245 252
41 243 241 245 263 248
;
```

The following statements create an $\bar{X}$ chart for the data in Gaps2 using the control limits in Gaplim:

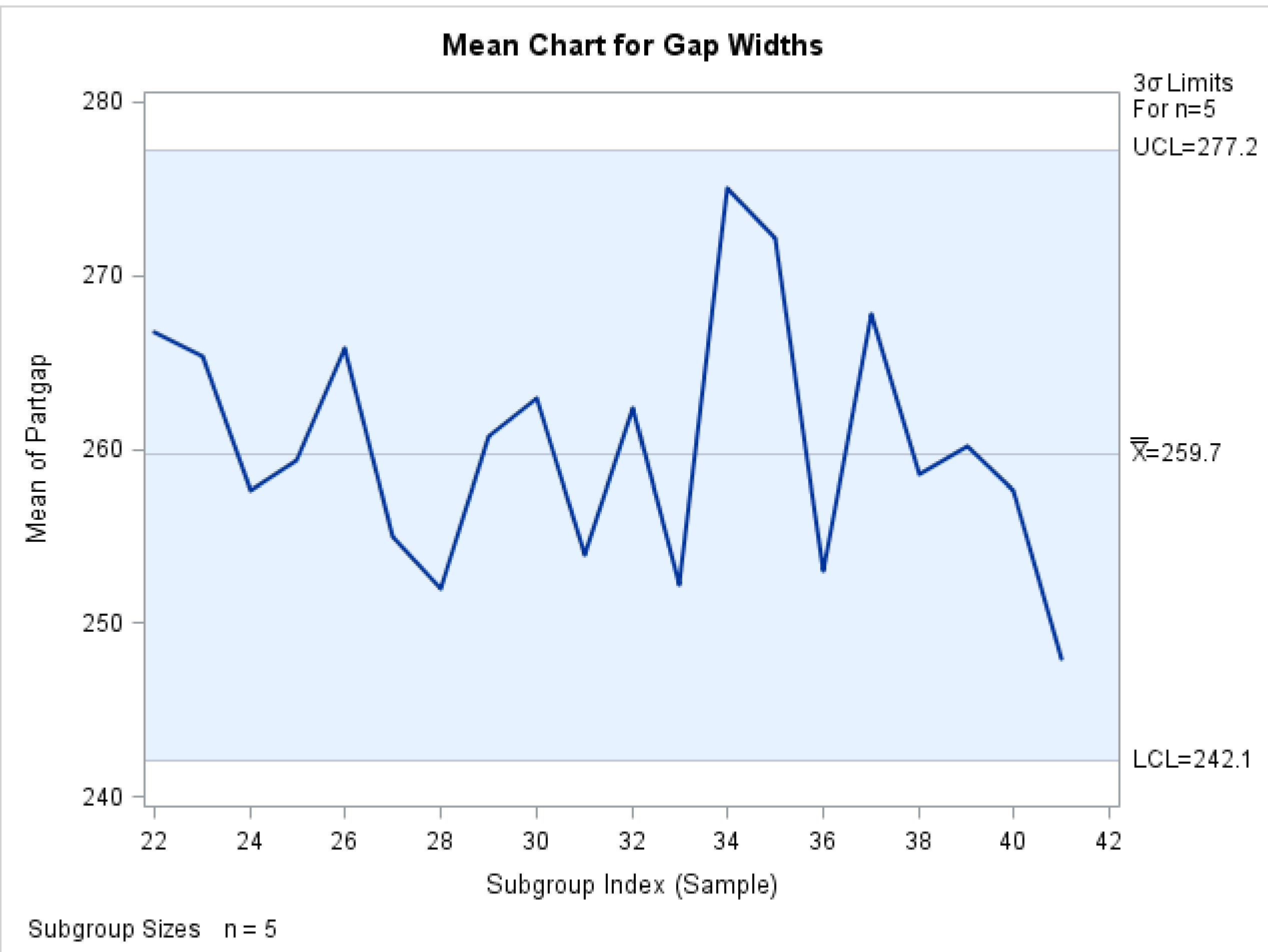
```
ods graphics on;
title 'Mean Chart for Gap Widths';
proc shewhart data=Gaps2 limits=Gaplim;
   xchart Partgap*Sample / odstitle=title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the $\bar{X}$ chart is created using ODS Graphics instead of traditional graphics. The chart is shown in Figure 17.104.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Partgap
- the value of _SUBGRP_ matches the *subgroup-variable* name Sample

Figure 17.104 $\bar{X}$ Chart for Second Set of Gap Width Data (ODS Graphics)



The chart indicates that the process is in control, since all the means lie within the control limits.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1823 for details concerning the variables that you must provide.

Syntax: XCHART Statement

The basic syntax for the XCHART statement is as follows:

XCHART *process* $*$ *subgroup-variable* **;**

The general form of this syntax is as follows:

XCHART *processes* $*$ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=***'character'* > / < *options* > **;**

You can use any number of XCHART statements in the SHEWHART procedure. The components of the XCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Means from Raw Data" on page 1795.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Means from Subgroup Summary Data" on page 1797.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1802.

A *process* is required. If you specify more than one process, enclose the list in parentheses. For example, the following statements request distinct $\bar{X}$ charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   xchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding XCHART statement, Day is the subgroup variable. For details, see "Subgroup Variables" on page 1871.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See the section "Displaying Stratification in Blocks of Observations" on page 2026 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the means.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See the section "Displaying Stratification in Levels of a Classification Variable" on page 2025 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create an $\bar{X}$ chart using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   xchart Weight*Day='*';
run;
```

options

enhance the appearance of the chart, request additional analyses, save results in data sets, and so on. The section "Summary of Options", which follows, lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1946 describes each option in detail.

Summary of Options

The following table lists the XCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1946.

Table 17.64 XCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	requests probability limits for chart
LIMITN=	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	specifies width of control limits in terms of multiple *k* of standard error of plotted means

Table 17.64 *continued*

Option	Description
Options for Displaying Control Limits	
CINFILL=	specifies color for area inside control limits
CLIMITS=	specifies color of control limits, central line, and related labels
LCLLABEL=	specifies label for lower control limit
LIMLABSUBCHAR=	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	specifies line type for control limits
NDECIMAL=	specifies number of digits to right of decimal place in default labels for control limits and central line
NOCTL	suppresses display of central line
NOLCL	suppresses display of lower control limit
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit
UCLLABEL=	specifies label for upper control limit
WLIMITS=	specifies width for control limits and central line
XSYMBOL=	specifies label for central line
Process Mean and Standard Deviation Options	
MU0=	specifies known value of μ_0 for process mean μ
SIGMA0=	specifies known value σ_0 for process standard deviation σ
SMETHOD=	specifies method for estimating process standard deviation σ
STDDEVIATIONS	specifies that estimate of process standard deviation σ is to be calculated from subgroup standard deviations
TYPE=	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	labels every point on $\bar{X}$ chart
ALLLABEL2=	labels every point on trend chart
CLABEL=	specifies color for labels
CCONNECT=	specifies color for line segments that connect points on chart
CFRAMELAB=	specifies fill color for frame around labeled points
CNEEDLES=	specifies color for needles that connect points to central line
COUT=	specifies color for portions of line segments that connect points outside control limits

Table 17.64 *continued*

Option	Description
COUTFILL=	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	specifies angle at which labels are drawn
LABELFONT=	specifies software font for labels (alias for the TEST-FONT= option)
LABELHEIGHT=	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
NOTRENDCONNECT	suppresses line segments that connect points on trend chart
OUTLABEL=	labels points outside control limits
SYMBOLLEGEND=	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=	specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	computes zone widths independently above and below center line
NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	suppresses tests across *phase* boundaries
TESTS=	specifies tests for special causes
TEST2RUN=	specifies length of pattern for Test 2
TEST3RUN=	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL=	provides labels for points where test is positive
TESTLABEL*n*=	specifies label for *n*th test for special causes
TESTNMETHOD=	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=	enables tests for special causes to be reset
WESTGARD=	requests that Westgard rules be applied
ZONELABELS	adds labels A, B, and C to zone lines
ZONES	adds lines delineating zones A, B, and C
ZONEVALPOS=	specifies position of ZONEVALUES labels
ZONEVALUES	labels zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	specifies color for labels indicating points where test is positive

Table 17.64 *continued*

Option	Description
CTESTSYMBOL=	specifies color for symbol used to plot points where test is positive
CZONES=	specifies color for lines and labels delineating zones A, B, and C
LTESTS=	specifies type of line connecting points where test is positive
LZONES=	specifies line type for lines delineating zones A, B, and C
TESTFONT=	specifies software font for labels at points where test is positive
TESTHEIGHT=	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	specifies plot symbol for points where test is positive
TESTSYMBOLHT=	specifies symbol height for points where test is positive
WTESTS=	specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill colors for frame for plot area
CTEXT=	specifies color for tick mark values and axis labels
DISCRETE	produces horizontal axis for discrete numeric group values
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis of $\bar{X}$ chart
VAXIS2=	specifies major tick mark values for vertical axis of trend chart
VFORMAT=	specifies format for primary vertical axis tick mark labels

Table 17.64 *continued*

Option	Description
VFORMAT2=	specifies format for secondary vertical axis tick mark labels
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=	specifies width of axis lines
Plot Layout Options	
ALLN	plots means for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
TOTPANELS=	specifies number of pages or screens to be used to display chart
TRENDVAR=	specifies list of trend variables
YPCT1=	specifies length of vertical axis on $\bar{X}$ chart as a percentage of sum of lengths of vertical axes for $\bar{X}$ and trend charts
ZEROSTD	displays $\bar{X}$ chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= and HREF2= options
CVREF=	specifies color for lines requested by VREF= and VREF2= options
HREF=	specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2=	specifies position of reference lines perpendicular to horizontal axis on trend chart

Table 17.64 *continued*

Option	Description
HREFDATA=	specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2DATA=	specifies position of reference lines perpendicular to horizontal axis on trend chart
HREFLABELS=	specifies labels for HREF= lines
HREF2LABELS=	specifies labels for HREF2= lines
HREFLABPOS=	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	specifies line type for HREF= and HREF2= lines
LVREF=	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	specifies position of reference lines perpendicular to vertical axis on $\bar{X}$ chart
VREF2=	specifies position of reference lines perpendicular to vertical axis on trend chart
VREFLABELS=	specifies labels for VREF= lines
VREF2LABELS=	specifies labels for VREF2= lines
VREFLABPOS=	position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=	specifies line type for grid requested with the ENDGRID option
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme points are clipped
CLIPLEGEND=	specifies text for clipping legend
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features to $\bar{X}$ chart
ANNOTATE2=	specifies annotate data set that adds features to trend chart
DESCRIPTION=	specifies description of $\bar{X}$ chart's GRSEG catalog entry
FONT=	specifies software font for labels and legends on charts
NAME=	specifies name of $\bar{X}$ chart's GRSEG catalog entry

Table 17.64 *continued*

Option	Description
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option
WTREND=	specifies width of line segments connecting points on trend chart
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	shades areas between control limits and connected points lying outside the limits
STARFILL=	specifies a variable identfying groups of stars filled with different colors
STARS=	specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	specifies the control limit infill transparency
MARKERS	plots subgroup points with markers
NOBLOCKREF	suppresses block and phase reference lines
NOBLOCKREFFILL	suppresses block and phase wall fills
NOFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOPHASEREF	suppresses block and phase reference lines
NOPHASEREFFILL	suppresses block and phase wall fills
NOREF	suppresses block and phase reference lines
NOREFFILL	suppresses block and phase wall fills
NOSTARFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	specifies a graph footnote
ODSFOOTNOTE2=	specifies a secondary graph footnote
ODSLEGENDEXPAND	specifies that legend entries contain all levels observed in the data
ODSTITLE=	specifies a graph title
ODSTITLE2=	specifies a secondary graph title
OUTFILLTRANSPARENCY=	specifies control limit outfill transparency
OVERLAYURL=	specifies URLs to associate with overlay points
OVERLAY2URL=	specifies URLs to associate with overlay points on secondary chart

Table 17.64 *continued*

Option	Description
PHASEPOS=	specifies vertical position of phase legend
PHASEREFLEVEL=	associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	draws central line labels using a simulated software font
STARTRANSPARENCY=	specifies star fill transparency
URL=	specifies a variable whose values are URLs to be associated with subgroups
URL2=	specifies a variable whose values are URLs to be associated with subgroups on secondary chart
Input Data Set Options	
MISSBREAK	specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	creates output data set containing subgroup summary statistics
OUTINDEX=	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	creates output data set containing control limits
OUTTABLE=	creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	specifies α value and type for computing capability index confidence limits
LSL=	specifies list of lower specification limits
TARGET=	specifies list of target values
USL=	specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for *block-variable* legend

Table 17.64 *continued*

Option	Description
BLOCKLABTYPE=	specifies text size of *block-variable* legend
BLOCKPOS=	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE=	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	specifies color for STARCIRCLES= circles
CSTARFILL=	specifies color for filling stars
CSTAROUT=	specifies outline color for stars exceeding inner or outer circles
CSTARS=	specifies color for outlines of stars
LSTARCIRCLES=	specifies line types for STARCIRCLES= circles
LSTARS=	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	specifies reference circles for stars
STARINRADIUS=	specifies inner radius of stars
STARLABEL=	specifies vertices to be labeled
STARLEGEND=	specifies style of legend for star vertices
STARLEGENDLAB=	specifies label for STARLEGEND= legend
STAROUTRADIUS=	specifies outer radius of stars
STARSPECS=	specifies method used to standardize vertex variables
STARSTART=	specifies angle for first vertex
STARTYPE=	specifies graphical style of star
STARVERTICES=	superimposes star at each point on $\bar{X}$ chart
WSTARCIRCLES=	specifies width of STARCIRCLES= circles
WSTARS=	specifies width of STARVERTICES= stars

Table 17.64 *continued*

Option	Description
Overlay Options	
CCOVERLAY=	specifies colors for primary chart overlay line segments
CCOVERLAY2=	specifies colors for secondary chart overlay line segments
COVERLAY=	specifies colors for primary chart overlay plots
COVERLAY2=	specifies colors for secondary chart overlay plots
COVERLAYCLIP=	specifies color for clipped points on overlays
LOVERLAY=	specifies line types for primary chart overlay line segments
LOVERLAY2=	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=	specifies variables to overlay on primary chart
OVERLAY2=	specifies variables to overlay on secondary chart
OVERLAY2HTML=	specifies links to associate with secondary chart overlay points
OVERLAY2ID=	specifies labels for secondary chart overlay points
OVERLAY2SYM=	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
OVERLAYHTML=	specifies links to associate with primary chart overlay points
OVERLAYID=	specifies labels for primary chart overlay points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols for primary chart overlays
OVERLAYSYMHT=	specifies symbol heights for primary chart overlays
WOVERLAY=	specifies widths of primary chart overlay line segments
WOVERLAY2=	specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	specifies a variable whose values create links to be associated with subgroups
HTML2=	specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	specifies plot character for clipped points
CONNECTCHAR=	specifies character used to form line segments that connect points on chart
HREFCHAR=	specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	specifies characters indicating *symbol-variable*

Table 17.64 *continued*

Option	Description
TESTCHAR=	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	specifies line character for VREF= and VREF2= lines
ZONECHAR=	specifies character for lines that delineate zones for tests for special causes

Details: XCHART Statement

Constructing Charts for Means

The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	mean of measurements in *i*th subgroup
R_i	range of measurements in *i*th subgroup
n_i	sample size of *i*th subgroup
N	number of subgroups
$\bar{\bar{X}}$	weighted average of subgroup means
z_p	100*p*th percentile of the standard normal distribution

Plotted Points

Each point on an $\bar{X}$ chart indicates the value of a subgroup mean ($\bar{X}_i$). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the value plotted for this subgroup is

$$\bar{X}_{10} = \frac{12 + 15 + 19 + 16 + 14}{5} = 15.2$$

Central Line

By default, the central line on an $\bar{X}$ chart indicates an estimate for μ, which is computed as

$$\hat{\mu} = \bar{\bar{X}} = \frac{n_1 \bar{X}_1 + \cdots + n_N \bar{X}_N}{n_1 + \cdots + n_N}$$

If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard error of $\bar{X}_i$ above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that $\bar{X}_i$ exceeds the limits

The following table provides the formulas for the limits:

Table 17.65 Limits for $\bar{X}$ Charts

Control Limits
LCL = lower limit = $\overline{\overline{X}} - k\hat{\sigma}/\sqrt{n_i}$ UCL = upper limit = $\overline{\overline{X}} + k\hat{\sigma}/\sqrt{n_i}$
Probability Limits
LCL = lower limit = $\overline{\overline{X}} - z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$ UCL = upper limit = $\overline{\overline{X}} + z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$

Note that the limits vary with n_i. If standard values μ_0 and σ_0 are available for μ and σ, respectively, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 17.65.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables can be saved:

Table 17.66 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLR	lower control limit for subgroup range
LCLS	lower control limit for subgroup standard deviation
LCLX	lower control limit for subgroup mean
LIMITN	sample size associated with the control limits
LSL	lower specification limit
MEAN	process mean ($\overline{\overline{X}}$ or μ_0)
_R_	value of central line on R chart
_S_	value of central line on s chart
SIGMAS	multiple (k) of standard error of $\bar{X}_i$
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the XCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLR	upper control limit for subgroup range
UCLS	upper control limit for subgroup standard deviation
UCLX	upper control limit for subgroup mean
USL	upper specification limit
VAR	*process* specified in the XCHART statement

Notes:

1. The variables _LCLS_, _S_, and _UCLS_ are included if you specify the STDDEVIATIONS option; otherwise, the variables _LCLR_, _R_, and _UCLR_ are included. These variables are not used to create $\bar{X}$ charts, but they enable the OUTLIMITS= data set to be used as a LIMITS= data set with the BOXCHART, MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements.

2. If the control limits vary with subgroup sample size, the special missing value 'V' is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLR_, _R_, _UCLR_, _LCLS_, _S_, and _UCLS_.

3. If the limits are defined in terms of a multiple k of the standard error of $\bar{X}_i$, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.

4. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.

5. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are

included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1874 for computational details.

6. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the XCHART statement. For an example, see "Saving Control Limits" on page 1802.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables can be saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup sample size variable named by *process* suffixed with *N*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup standard deviation variable named by *process* suffixed with *S*

A subgroup standard deviation variable is included if you specify the STDDEVIATIONS option; otherwise, a subgroup range variable is included.

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the XCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   xchart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthX, WidthR, WidthN, DiameterX, DiameterR, and DiameterN. The variables WidthR and DiameterR are included, since the STDDEVIATIONS option is not specified. If you specified the STDDEVIATIONS option, the data set Summary would contain the variables WidthS and DiameterS rather than WidthR and DiameterR.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1800.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables can be saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on $\bar{X}$ chart
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBN	subgroup sample size
SUBX	subgroup mean
TESTS	tests for special causes signaled on $\bar{X}$ chart
UCLX	upper control limit for mean
VAR	*process* specified in the XCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)
- _TREND_ (if the TRENDVAR= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= option, respectively, or with the corresponding variables in a LIMITS= data set).
2. The variable _TESTS_ is saved if you specify the TESTS= option. The kth character of a value of _TESTS_ is k if Test k is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.
3. The variables _EXLIM_ and _TESTS_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1802.

ODS Tables

The following table summarizes the ODS tables that you can request with the XCHART statement.

Table 17.67 ODS Tables Produced with the XCHART Statement

Table Name	Description	Options
XCHART	$\bar{X}$ chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*). **NOTE:** In SAS/QC 13.1 the SHEWHART procedure does not support the creation of graphs that are editable with the ODS Graphics Editor.

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. XCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1946.

When ODS Graphics is in effect, the XCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 17.68.

Table 17.68 ODS Graphics Produced by the XCHART Statement

ODS Graph Name	Plot Description
XChart	$\bar{X}$ chart

See Chapter 3, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the XCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the XCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the *i*th subgroup contains n_i items, there should be n_i consecutive observations for which the value of the *subgroup-variable* is the index of the *i*th subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a DATA= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) with the READPHASES= option (for an example, see the section "Displaying Stratification in Phases" on page 2031).

For an example of a DATA= data set, see "Creating Charts for Means from Raw Data" on page 1795.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   xchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set; see Table 17.66. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLX_, _MEAN_, and _UCLX_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 17.65

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1804.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedures or to read output data sets created with SAS summarization procedures, such as the MEANS procedure.

A HISTORY= data set used with the XCHART statement must contain the following:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup sample size variable for each *process*
- either a subgroup range variable or subgroup standard deviation variable for each *process*

If you specify the STDDEVIATIONS option, the subgroup standard deviation variable must be included; otherwise, the subgroup range variable must be included.

The names of the subgroup mean, subgroup range or subgroup standard deviation, and subgroup sample size variables must be the *process* name concatenated with the suffix characters *X*, *R* or *S*, and *N*, respectively.

For example, consider the following statements:

```
proc shewhart history=Summary;
   xchart (Weight Yieldstrength)*Batch;
run;
```

The data set Summary must include the variables Batch, WeightX, WeightR, WeightN, YieldstrengthX, YieldstrengthR, and YieldstrengthN. If the STDDEVIATIONS option were specified in the preceding XCHART statement, it would be necessary for Summary to include the variables Batch, WeightX, WeightS, WeightN, YieldstrengthX, YieldstrengthS, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see the section "Displaying Stratification in Phases" on page 2031 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Means from Subgroup Summary Data" on page 1797.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2096.

The following table lists the variables required in a TABLE= data set used with the XCHART statement:

Table 17.69 Variables Required in a TABLE= Data Set

Variable	Description
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBX	subgroup mean
UCLX	upper control limit for mean

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1802.

Methods for Estimating the Standard Deviation

When control limits are computed from the input data, three methods (referred to as default, MVLUE, and RMSDF) are available for estimating the process standard deviation σ. The method depends on whether you specify the STDDEVIATIONS option. If you specify this option, σ is estimated using subgroup standard deviations; otherwise, σ is estimated using subgroup ranges.

For an illustration of the methods, see Example 17.35.

Default Method Based on Subgroup Ranges

If you do not specify the STDDEVIATIONS option, the default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the ith subgroup.

$$R_i = \max_{1 \leq j \leq n_i} (x_{ij}) - \min_{1 \leq j \leq n_i} (x_{ij})$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

Default Method Based on Subgroup Standard Deviations

If you specify the STDDEVIATIONS option, the default estimate for σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \cdots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the ith subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1} \sum_{j=1}^{n_i} (x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the ith subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method Based on Subgroup Ranges

If you do not specify the STDDEVIATIONS option and you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The unbiasing factor $d_3(n_i)$ is defined so that, if the observations are normally distributed, the expected value of σ_{R_i} is $d_3(n_i)\sigma$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

MVLUE Method Based on Subgroup Standard Deviations

If you specify the STDDEVIATIONS option and SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). This estimate is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

RMSDF Method Based on Subgroup Standard Deviations

If you specify the STDDEVIATIONS option and SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

where $n = n_1 + \cdots + n_N - (N - 1)$. The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications, it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Default Method Based on Individual Measurements

When each subgroup sample contains a single observation ($n_i \equiv 1$), the process standard deviation σ is estimated as $\hat{\sigma} = \bar{R}/d_2(2)$, where $\bar{R}$ is the average of the moving ranges of consecutive measurements taken in pairs. This is the method used to estimate σ for individual measurements and moving range charts. See "Methods for Estimating the Standard Deviation" on page 1504.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical	DATA=	*process*
Vertical	HISTORY=	subgroup mean variable
Vertical	TABLE=	_SUBX_

If you specify the TRENDVAR= option, you can provide distinct labels for the vertical axes of the $\bar{X}$ and trend charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the $\bar{X}$ chart, and the second part labels the vertical axis of the trend chart.

For example, the following sets of statements specify the label *Residual Mean* for the vertical axis of the $\bar{X}$ chart and the label *Fitted Mean* for the vertical axis of the trend chart:

```
proc shewhart data=Toolwear;
   xchart Diameter*Hour / split    = '/'
                          trendvar = Fitted ;
   label Diameter = 'Residual Mean/Fitted Mean';
run;

proc shewhart history=Regdata;
   xchart Diameter*Hour / split    = '/'
                          trendvar = Fitted;
   label DiameterX = 'Residual Mean/Fitted Mean';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: XCHART Statement

This section provides advanced examples of the XCHART statement.

Example 17.34: Applying Tests for Special Causes

NOTE: See *Mean Chart-Tests for Special Causes Applied* in the SAS/QC Sample Library.

This example illustrates how you can apply tests for special causes to make $\bar{X}$ charts more sensitive to special causes of variation.

The following statements create an $\bar{X}$ chart for the gap width measurements in the data set Parts in "Creating Charts for Means from Subgroup Summary Data" on page 1797 and tabulate the results:

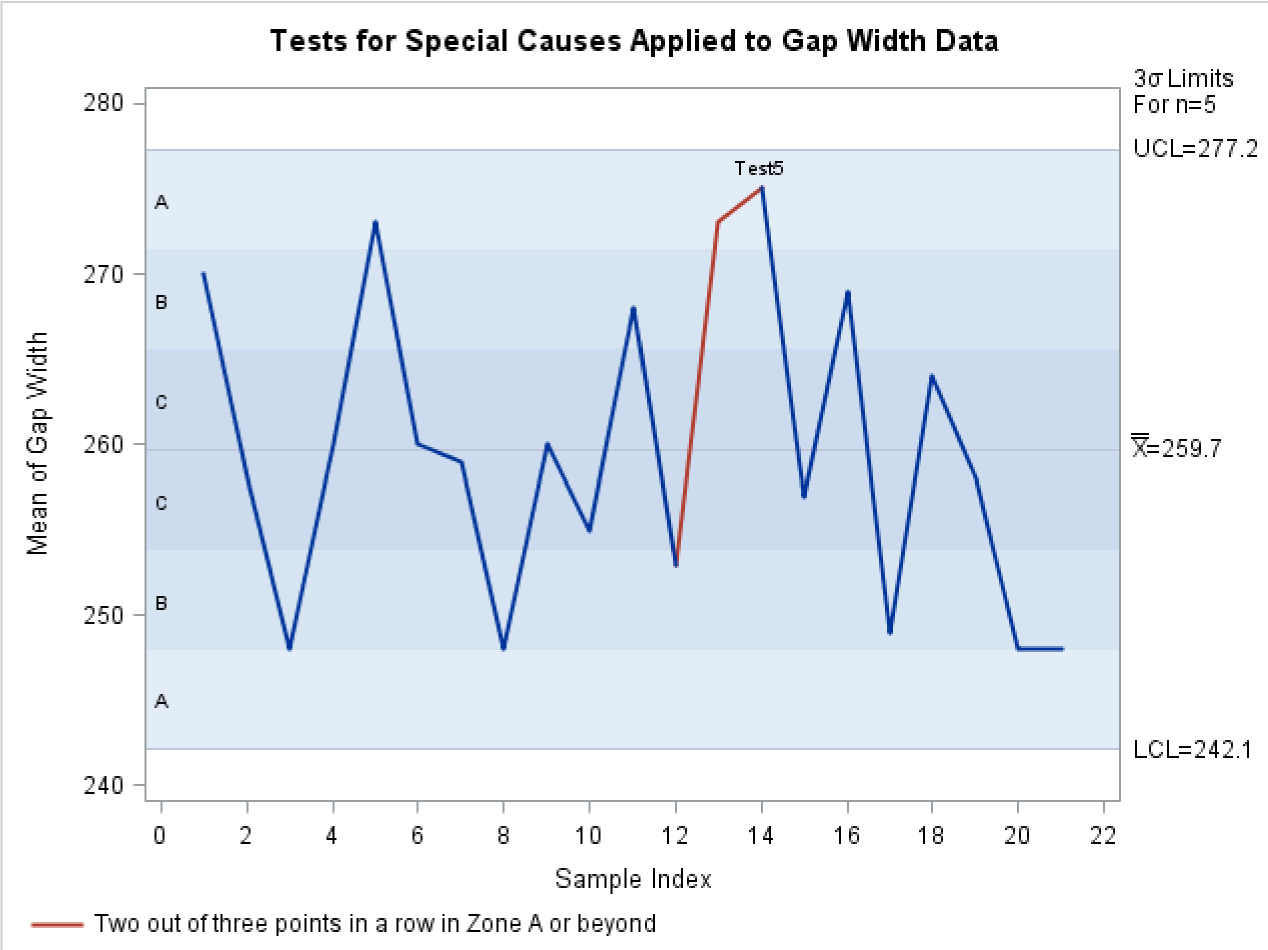
```
ods graphics on;
title 'Tests for Special Causes Applied to Gap Width Data';
proc shewhart history=Parts;
   xchart Partgap*Sample/ tests    = 1 to 5
                          odstitle = title
                          tabletests
                          nolegend
                          tablecentral
                          tablelegend
                          zonelabels;
run;
```

The $\bar{X}$ chart is shown in Output 17.34.1 and the printed output is shown in Output 17.34.2. The TESTS= requests Tests 1, 2, 3, 4, and 5, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2073. The TABLECENTRAL option requests a table of the subgroup means, control limits, and central line. The TABLETESTS option adds a column indicating which subgroups tested positive for special causes, and the TABLELEGEND option adds a legend describing the tests that were signaled.

The ZONELABELS option displays zone lines and zone labels on the chart. The zones are used to define the tests. The NOLEGEND option suppresses the subgroup sample size legend that is displayed by default in the lower left corner of the chart.

Output 17.34.1 and Output 17.34.2 indicate that Test 5 was positive at sample 14, signaling a possible shift in the mean of the process.

Output 17.34.1 Tests for Special Causes Displayed on an $\bar{X}$ Chart



Output 17.34.2 Tabular Form of $\bar{X}$ Chart

Tests for Special Causes Applied to Gap Width Data

The SHEWHART Procedure

Means Chart Summary for Partgap

3 Sigma Limits with n=5 for Mean

Sample	Subgroup Sample Size	Lower Limit	Subgroup Mean	Average Mean	Upper Limit	Special Tests Signaled
1	5	242.08741	270.00000	259.66667	277.24592	
2	5	242.08741	258.00000	259.66667	277.24592	
3	5	242.08741	248.00000	259.66667	277.24592	
4	5	242.08741	260.00000	259.66667	277.24592	
5	5	242.08741	273.00000	259.66667	277.24592	
6	5	242.08741	260.00000	259.66667	277.24592	
7	5	242.08741	259.00000	259.66667	277.24592	
8	5	242.08741	248.00000	259.66667	277.24592	
9	5	242.08741	260.00000	259.66667	277.24592	
10	5	242.08741	255.00000	259.66667	277.24592	
11	5	242.08741	268.00000	259.66667	277.24592	
12	5	242.08741	253.00000	259.66667	277.24592	
13	5	242.08741	273.00000	259.66667	277.24592	
14	5	242.08741	275.00000	259.66667	277.24592	5
15	5	242.08741	257.00000	259.66667	277.24592	
16	5	242.08741	269.00000	259.66667	277.24592	
17	5	242.08741	249.00000	259.66667	277.24592	
18	5	242.08741	264.00000	259.66667	277.24592	
19	5	242.08741	258.00000	259.66667	277.24592	
20	5	242.08741	248.00000	259.66667	277.24592	
21	5	242.08741	248.00000	259.66667	277.24592	

Test Descriptions	
Test 5	Two out of three points in a row in Zone A or beyond

Example 17.35: Estimating the Process Standard Deviation

NOTE: See *Estimating the Process Standard Deviation* in the SAS/QC Sample Library.

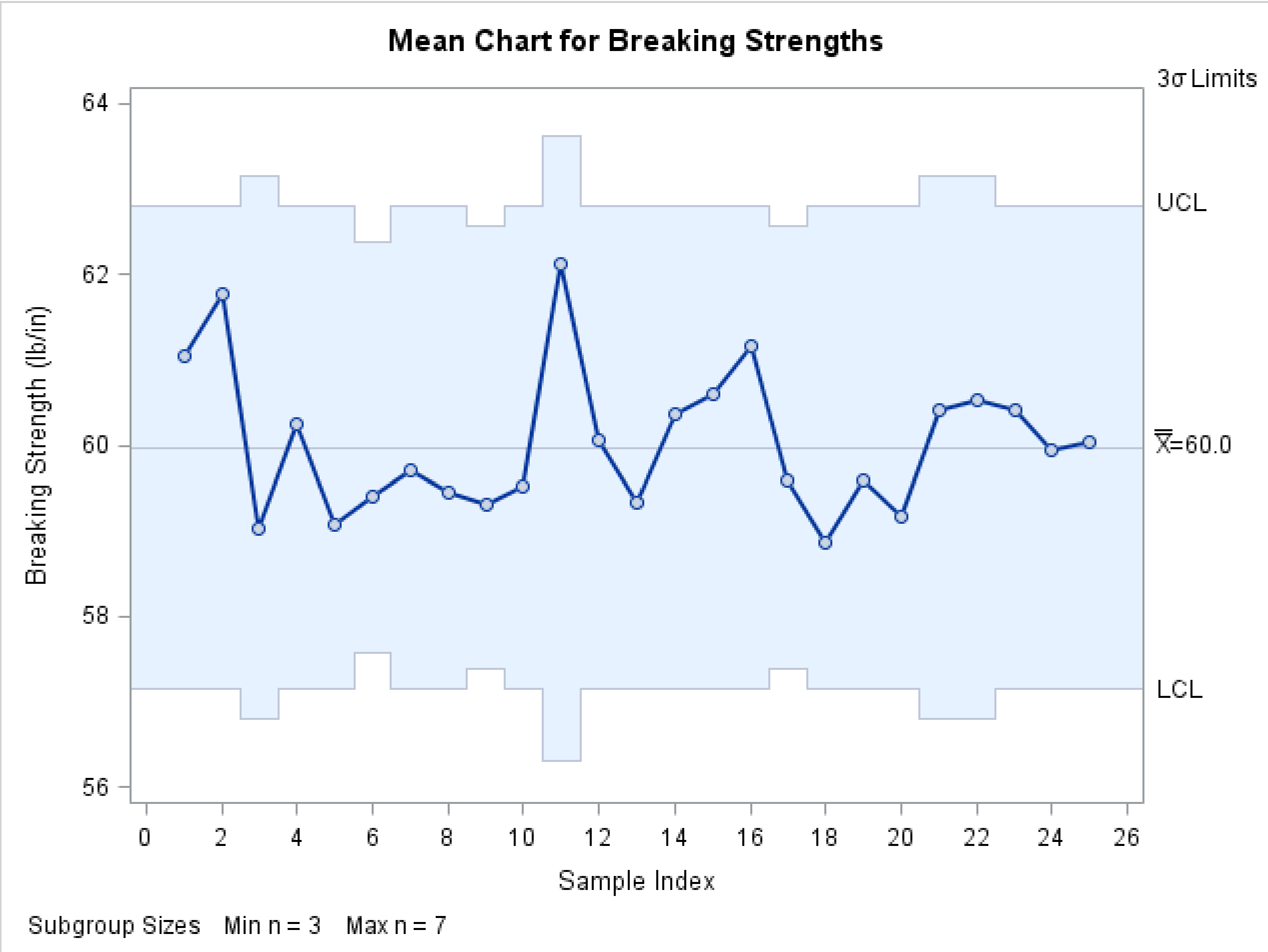
The following data set (Wire) contains breaking strength measurements recorded in pounds per inch for 25 samples from a metal wire manufacturing process. The subgroup sample sizes vary between 3 and 7.

```
data Wire;
   input Sample Size @;
   do i=1 to Size;
      input Breakstrength @@;
      output;
   end;
   drop i Size;
   label Breakstrength  ='Breaking Strength (lb/in)'
         Sample ='Sample Index';
   datalines;
 1  5 60.6 62.3 62.0 60.4 59.9
 2  5 61.9 62.1 60.6 58.9 65.3
 3  4 57.8 60.5 60.1 57.7
 4  5 56.8 62.5 60.1 62.9 58.9
 5  5 63.0 60.7 57.2 61.0 53.5
 6  7 58.7 60.1 59.7 60.1 59.1 57.3 60.9
 7  5 59.3 61.7 59.1 58.1 60.3
 8  5 61.3 58.5 57.8 61.0 58.6
 9  6 59.5 58.3 57.5 59.4 61.5 59.6
10  5 61.7 60.7 57.2 56.5 61.5
11  3 63.9 61.6 60.9
12  5 58.7 61.4 62.4 57.3 60.5
13  5 56.8 58.5 55.7 63.0 62.7
14  5 62.1 60.6 62.1 58.7 58.3
15  5 59.1 60.4 60.4 59.0 64.1
16  5 59.9 58.8 59.2 63.0 64.9
17  6 58.8 62.4 59.4 57.1 61.2 58.6
18  5 60.3 58.7 60.5 58.6 56.2
19  5 59.2 59.8 59.7 59.3 60.0
20  5 62.3 56.0 57.0 61.8 58.8
21  4 60.5 62.0 61.4 57.7
22  4 59.3 62.4 60.4 60.0
23  5 62.4 61.3 60.5 57.7 60.2
24  5 61.2 55.5 60.2 60.4 62.4
25  5 59.0 66.1 57.7 58.5 58.9
;
```

The following statements request an $\bar{X}$ chart, shown in Output 17.35.1, for the breaking strength measurements:

```
ods graphics on;
title 'Mean Chart for Breaking Strengths';
proc shewhart data=Wire;
   xchart Breakstrength*Sample / odstitle = title
                                 markers;
run;
```

Output 17.35.1 $\bar{X}$ Chart with Varying Subgroup Sample Sizes



Note that the control limits vary with the subgroup sample size. The sample size legend in the lower left corner displays the minimum and maximum subgroup sample sizes.

By default, the control limits are 3σ limits estimated from the data. You can use the STDDEVIATIONS option and the SMETHOD= option to specify how the estimate of the process standard deviation σ is to be computed, as illustrated by the following statements:

```
proc shewhart data=Wire;
   xchart Breakstrength*Sample / outlimits=Wirelim1
                                 outindex ='Default-Ranges'
                                 nochart;
   xchart Breakstrength*Sample / outlimits=Wirelim2
                                 stddeviations
                                 outindex ='Default-Stds'
                                 nochart;
   xchart Breakstrength*Sample / outlimits=Wirelim3
                                 smethod  =mvlue
                                 outindex ='MVLUE  -Ranges'
                                 nochart;
   xchart Breakstrength*Sample / outlimits=Wirelim4
                                 stddeviations
```

```
                              smethod  =mvlue
                              outindex ='MVLUE  -Stds'
                              nochart;
   xchart Breakstrength*Sample / outlimits=Wirelim5
                              stddeviations
                              smethod  =rmsdf
                              outindex ='RMSDF  -Stds'
                              nochart;
run;
```

The STDDEVIATIONS option specifies that the estimate is to be calculated from subgroup standard deviations rather than subgroup ranges, the default. The SMETHOD= option specifies the method for estimating σ. The default method estimates σ as an unweighted average of subgroup estimates of σ. Specifying SMETHOD=MVLUE requests a minimum variance linear unbiased estimate, and specifying SMETHOD=RMSDF requests a weighted root-mean-square estimate. For details, see "Methods for Estimating the Standard Deviation" on page 1826.

The variable _STDDEV_ in each OUTLIMITS= data set contains the estimate of σ. The OUTINDEX= option specifies the value of the variable _INDEX_ in the OUTLIMITS= data set and is used here to identify the estimation method.

The following statements merge the five OUTLIMITS= data sets into a single data set, which is listed in Output 17.35.2:

```
data Wlimits;
   set Wirelim1 Wirelim2 Wirelim3 Wirelim4 Wirelim5;
   keep _index_ _stddev_;
run;
```

Output 17.35.2 The Data Set WLIMITS

Estimates of the Process Standard Deviation

INDEX	_STDDEV_
Default-Ranges	2.11146
Default-Stds	2.15453
MVLUE -Ranges	2.11240
MVLUE -Stds	2.14790
RMSDF -Stds	2.17479

The $\bar{X}$ chart shown in Output 17.35.1 uses the default estimate listed first in Output 17.35.2 ($\sigma = 2.11146$). In this case, there is very little difference in the five estimates, since the sample sizes do not differ greatly. In general, the MVLUE's are recommended with large sample sizes ($n_i \geq 10$).

Example 17.36: Plotting OC Curves for Mean Charts

NOTE: See *Plotting OC Curves for Mean Charts* in the SAS/QC Sample Library.

This example uses the GPLOT procedure and the DATA step function PROBNORM to plot operating characteristic (OC) curves for $\bar{X}$ charts with 3σ limits. An OC curve is plotted for each of the subgroup samples sizes 1, 2, 3, 4, and 16. Refer to page 226 in Montgomery (1996). Each curve plots the probability β of not detecting a shift of magnitude $\nu\sigma$ in the process mean as a function of ν. The value of β is computed using the following formula:

$$\begin{aligned}\beta &= P\{LCL \leq \bar{X}_i \leq UCL\} \\ &= \Phi(3 - \nu\sqrt{n}) - \Phi(-3 - \nu\sqrt{n})\end{aligned}$$

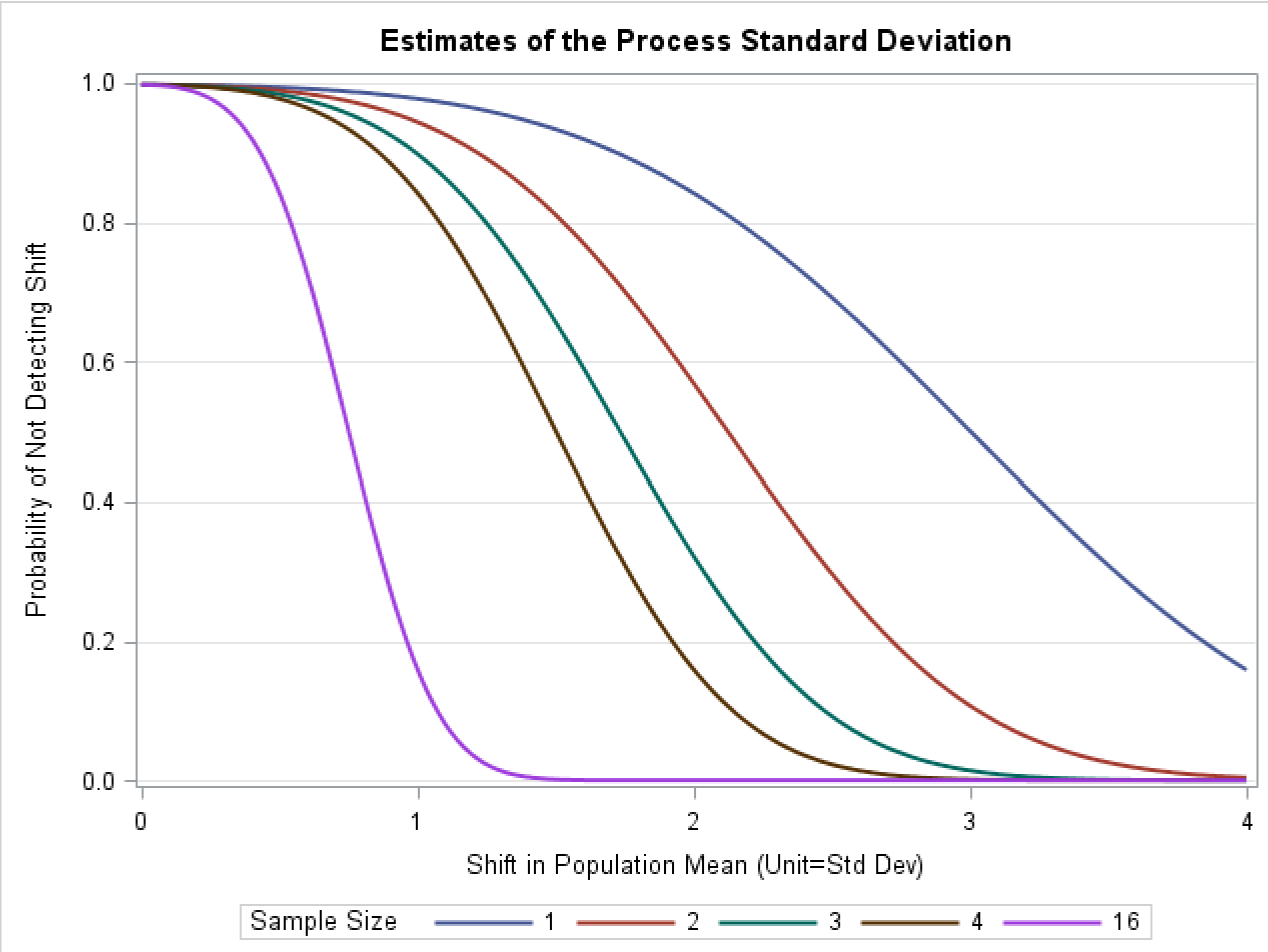
The following statements compute β (the variable BETA) as a function of ν (the variable NU). The variable nSample contains the sample size.

```
data oc;
   keep prob nSample t plot2;
   plot2=.;
   do nSample=1, 2, 3, 4, 16;
      do j=0 to 400;
         t=j/100;
         prob=probnorm( 3-t*sqrt(nSample)) -
              probnorm(-3-t*sqrt(nSample));
         output;
      end;
   end;
   label t   ='Shift in Population Mean (Unit=Std Dev)'
        prob='Probability of Not Detecting Shift';
run;
```

The following statements use the GPLOT procedure to display the OC curves shown in Output 17.36.1:

```
proc sgplot data=oc;
   series x=t y=prob /
      group=nSample lineattrs=(pattern=solid thickness=2);
   yaxis grid;
   label nSample='Sample Size';
run;
```

Output 17.36.1 OC Curves for Different Subgroup Sample Sizes



Example 17.37: Computing Process Capability Indices

You can save process capability indices in an OUTLIMITS= data set if you provide specification limits with the LSL= and USL= options. This is illustrated by the following statements:

```
title 'Control Limits and Capability Indices';
proc shewhart data=Partgaps;
   xchart Partgap*Sample / outlimits = Gaplim2
                           usl       = 270
                           lsl       = 240
                           nochart;
run;
```

The data set Gaplim2 is listed in Output 17.37.1.

Output 17.37.1 Data Set Gaplim2 Containing Control Limit Information

Control Limits with Capability Indices for Gap Width Measurements

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_	_LCLR_
Partgap	Sample	ESTIMATE	5	.002699796	3	242.087	259.667	277.246	0

_R_	_UCLR_	_STDDEV_	_LSL_	_USL_	_CP_	_CPL_	_CPU_	_CPK_
30.4762	64.4419	13.1028	240	270	0.38160	0.50032	0.26288	0.26288

The variables _CP_, _CPL_, _CPU_, and _CPK_ contain the process capability indices. It is reasonable to compute capability indices in this case, since Figure 17.97 indicates that the process is in statistical control. For more information, see the section "OUTLIMITS= Data Set" on page 1818.

XRCHART Statement: SHEWHART Procedure

Overview: XRCHART Statement

The XRCHART statement creates $\bar{X}$ and R charts for subgroup means and ranges, which are used to analyze the central tendency and variability of a process.

You can use options in the XRCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted means and ranges or as probability limits
- tabulate subgroup sample sizes, subgroup means, subgroup ranges, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes, subgroup means, and subgroup ranges in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a method for estimating the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the charts more readable
- display vertical and horizontal reference lines

- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing $\bar{X}$ and R charts with the XRCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 3, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: XRCHART Statement

This section introduces the XRCHART statement with simple examples illustrating commonly used options. Complete syntax for the XRCHART statement is presented in the section "Syntax: XRCHART Statement" on page 1850, and advanced examples are given in the section "Examples: XRCHART Statement" on page 1877.

Creating Charts for Means and Ranges from Raw Data

NOTE: See *Mean and Range (X-Bar and R) Charts* in the SAS/QC Sample Library.

In the manufacture of silicon wafers, batches of five wafers are sampled, and their diameters are measured in millimeters. The following statements create a SAS data set named Wafers, which contains the measurements for 25 batches:

```
data Wafers;
   input Batch @;
   do i=1 to 5;
      input Diameter @;
      output;
   end;
   drop i;
   datalines;
 1  35.00 34.99 34.99 34.98 35.00
 2  35.01 34.99 34.99 34.98 35.00
 3  34.99 35.00 35.00 35.00 35.00
 4  35.01 35.00 34.99 34.99 35.00
 5  35.00 34.99 34.98 34.99 35.00
 6  34.99 34.99 35.00 35.00 35.00
 7  35.01 34.98 35.00 35.00 34.99
 8  35.00 35.00 34.99 34.98 34.99
 9  34.99 34.98 34.98 35.01 35.00
10  34.99 35.00 35.01 34.99 35.01
11  35.01 35.00 35.00 34.98 34.99
12  34.99 34.99 35.00 34.98 35.01
```

```
13   35.01 34.99 34.98 34.99 34.99
14   35.00 35.00 34.99 35.01 34.99
15   34.98 34.99 34.99 34.98 35.00
16   34.99 35.00 35.00 35.01 35.00
17   34.98 34.98 34.99 34.99 34.98
18   35.01 35.02 35.00 34.98 35.00
19   34.99 34.98 35.00 34.99 34.98
20   34.99 35.00 35.00 34.99 34.99
21   35.00 34.99 34.99 34.98 35.00
22   35.00 35.00 35.01 35.00 35.00
23   35.02 35.00 34.98 35.02 35.00
24   35.00 35.00 34.99 35.01 34.98
25   34.99 34.99 34.99 35.00 35.00
;
```

The following statements use the PRINT procedure to list the data set Wafers. A portion of this listing is shown in Figure 17.105.

```
title 'The Data Set Wafers';
proc print data=Wafers noobs;
run;
```

Figure 17.105 Partial Listing of the Data Set Wafers

The Data Set Wafers

Batch	Diameter
1	35.00
1	34.99
1	34.99
1	34.98
1	35.00
2	35.01
2	34.99
2	34.99
2	34.98
2	35.00
3	34.99
3	35.00
3	35.00
3	35.00
3	35.00

The data set Wafers is said to be in "strung-out" form since each observation contains the batch number and diameter measurement for a single wafer. The first five observations contain the diameters for the first batch, the second five observations contain the diameters for the second batch, and so on. Because the variable Batch classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable Diameter contains the wafer diameter measurements and is referred to as the *process variable* (or *process* for short).

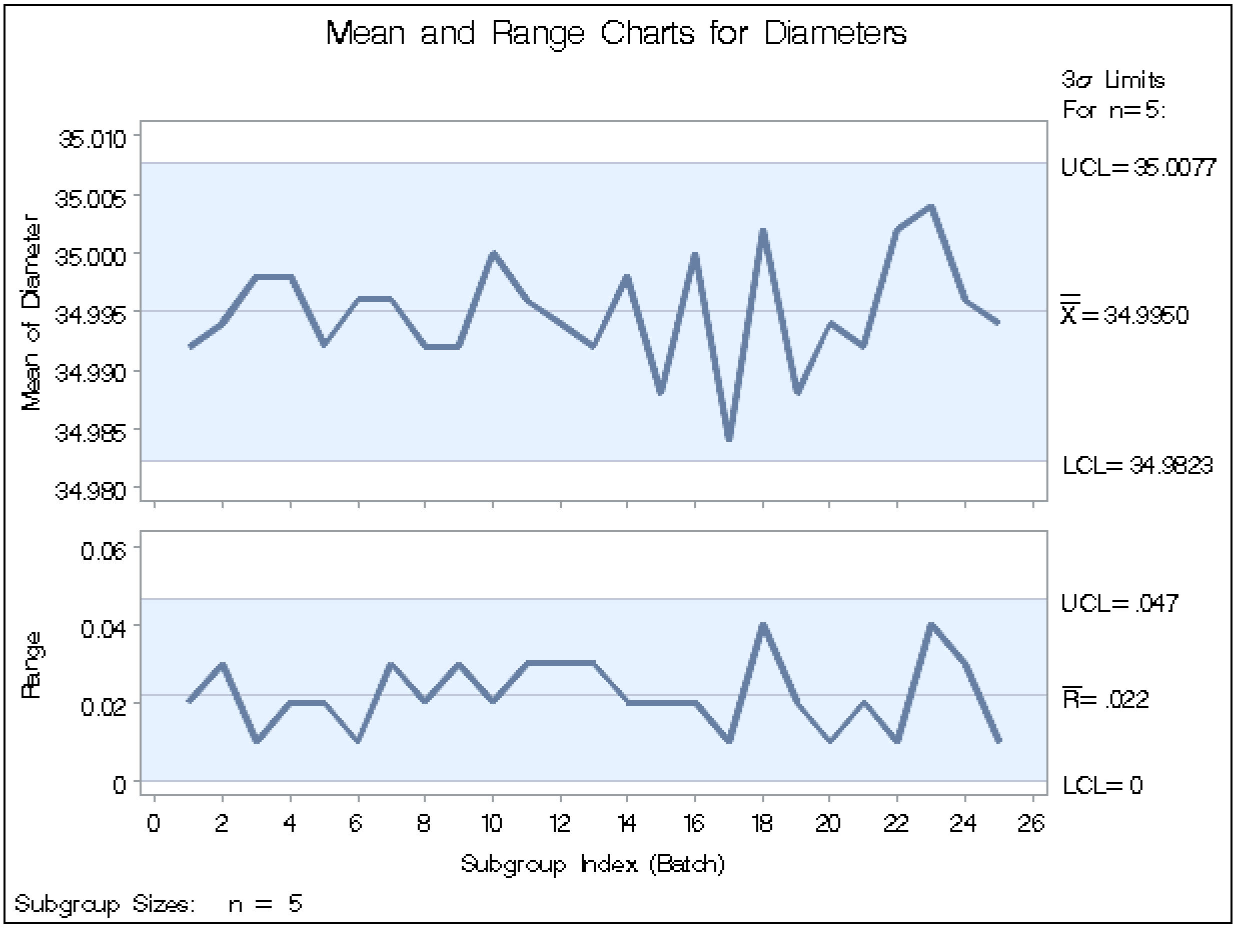
You can use $\bar{X}$ and R charts to determine whether the manufacturing process is in control. The following statements create the $\bar{X}$ and R charts shown in Figure 17.106:

```
ods graphics off;
title 'Mean and Range Charts for Diameters';
proc shewhart data=Wafers;
   xrchart Diameter*Batch;
run;
```

This example illustrates the basic form of the XRCHART statement. After the keyword XRCHART, which specifies the type of control chart to display, you specify the *process* to analyze (in this case, Diameter) followed by an asterisk and the *subgroup-variable* (Batch).

The input data set is specified with the DATA= option in the PROC SHEWHART statement. By default, traditional graphics output is produced, and its appearance is governed by the style in effect for any given ODS destination. Note that the STATISTICAL style is used throughout this book. See Chapter 3, "SAS/QC Graphics," for a discussion of alternatives for producing graphics with SAS/QC procedures.

Figure 17.106 $\bar{X}$ and R Charts for Wafer Diameter Data (Traditional Graphics)



Each point on the $\bar{X}$ chart represents the average (mean) of the measurements for a particular batch. For instance, the mean plotted for the first batch is

$$\frac{35.00 + 34.99 + 34.99 + 34.98 + 35.00}{5} = 34.992$$

Each point on the *R* chart represents the range of the measurements for a particular batch. For instance, the range plotted for the first batch is 35.00 – 34.98 = 0.02.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 17.71. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1848.

Since all the points lie within the control limits, it can be concluded that the process is in statistical control. For computational details, see "Constructing Charts for Means and Ranges" on page 1862. For more details on reading raw data, see "DATA= Data Set" on page 1868.

Creating Charts for Means and Ranges from Summary Data

NOTE: See *Mean and Range (X-Bar and R) Charts* in the SAS/QC Sample Library.

The previous example illustrates how you can create $\bar{X}$ and *R* charts based on raw data (process measurements). However, in many applications, the data are provided as subgroup means and ranges. This example illustrates how you can use the XRCHART statement with data of this type.

The following data set (Wafersum) provides the data from the preceding example in summarized form:

```
data Wafersum;
   input Batch DiameterX DiameterR;
   DiameterN = 5;
   datalines;
 1  34.992  0.02
 2  34.994  0.03
 3  34.998  0.01
 4  34.998  0.02
 5  34.992  0.02
 6  34.996  0.01
 7  34.996  0.03
 8  34.992  0.02
 9  34.992  0.03
10  35.000  0.02
11  34.996  0.03
12  34.994  0.03
13  34.992  0.03
14  34.998  0.02
15  34.988  0.02
16  35.000  0.02
17  34.984  0.01
18  35.002  0.04
19  34.988  0.02
20  34.994  0.01
21  34.992  0.02
22  35.002  0.01
23  35.004  0.04
24  34.996  0.03
25  34.994  0.01
;
```

A partial listing of the data set Wafersum is shown in Figure 17.107.

Figure 17.107 Partial Listing of the Summary Data Set Wafersum

Summary Data Set for Wafer Diameters

Batch	DiameterX	DiameterR	DiameterN
1	34.992	0.02	5
2	34.994	0.03	5
3	34.998	0.01	5
4	34.998	0.02	5
5	34.992	0.02	5

In this data set, there is exactly one observation for each subgroup (note that the subgroups are still indexed by Batch). The variable DiameterX contains the subgroup means, the variable DiameterR contains the subgroup ranges, and the variable DiameterN contains the subgroup sample sizes (these are all equal to five).

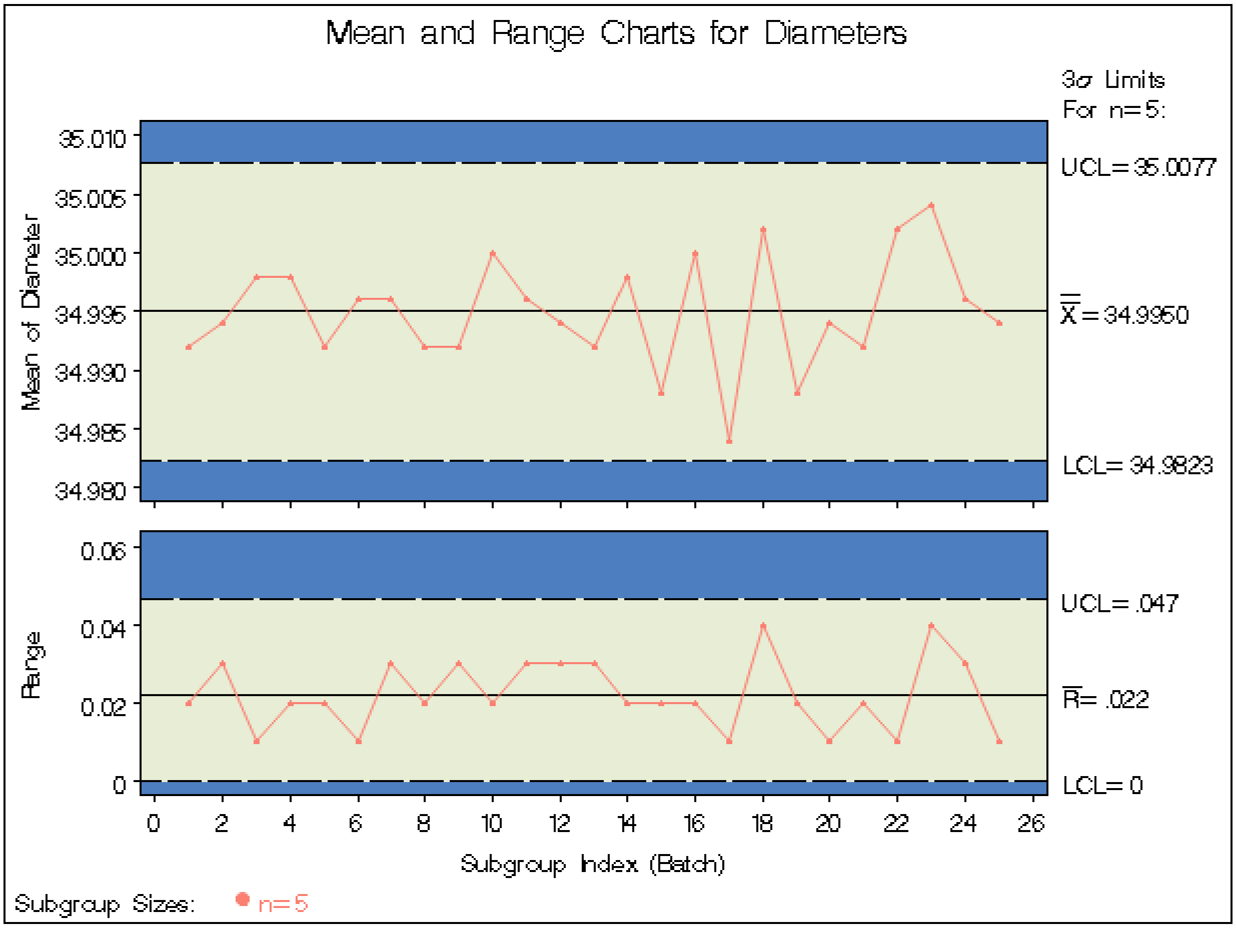
You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext=swiss;
title 'Mean and Range Charts for Diameters';
symbol value = dot color = salmon;
proc shewhart history=Wafersum;
   xrchart Diameter*Batch / cframe   = bigb
                            cinfill  = ywh
                            cconnect = salmon
                            coutfill = yellow;
run;
options gstyle;
```

Note that Diameter is *not* the name of a SAS variable in the data set Wafersum but is, instead, the common prefix for the names of the three SAS variables DiameterX, DiameterR, and DiameterN. The suffix characters *X*, *R*, and *N* indicate *mean*, *range*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in the HISTORY= data set with a single name (Diameter), which is referred to as the *process*. The name Batch specified after the asterisk is the name of the *subgroup-variable*.

The NOGSTYLE system option causes the XRCHART statement to ignore ODS styles when producing traditional graphics. Instead, global statements (such as GOPTIONS, SYMBOL, and AXIS statements) and XRCHART options, specified after the slash (/) in the XRCHART statement, control the appearance of the charts. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. A complete list of options is presented in the section "Syntax: XRCHART Statement" on page 1850. For more information about the GOPTIONS and SYMBOL statements, see *SAS/GRAPH: Reference*. The resulting charts are shown in Figure 17.108.

Figure 17.108 $\bar{X}$ and R Charts from Summary Data (Traditional Graphics with NOGSTYLE)



In general, a HISTORY= input data set used with the XRCHART statement must contain the following variables:

- subgroup variable
- subgroup mean variable
- subgroup range variable
- subgroup sample size variable

Furthermore, the names of the subgroup mean, range, and sample size variables must begin with the *process* name specified in the XRCHART statement and end with the special suffix characters *X*, *R*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. Suppose that, instead of the variables DiameterX, DiameterR, and DiameterN, the data set Wafersum contained summary variables named means, ranges, and sizes. The following statements would temporarily rename means, ranges, and sizes to DiameterX, DiameterR, and DiameterN, respectively:

```
proc shewhart
   history=Wafersum (rename=(means  = DiameterX
                             ranges = DiameterR
                             sizes  = DiameterN ));
   xrchart Diameter*Batch=;
run;
```

In summary, the interpretation of *process* depends on the input data set:

- If raw data are read by using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read by using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see "HISTORY= Data Set" on page 1869.

Saving Summary Statistics

NOTE: See *Mean and Range (X-Bar and R) Charts* in the SAS/QC Sample Library.

In this example, the XRCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Wafers and create a summary data set named Waferhist:

```
proc shewhart data=Wafers;
   xrchart Diameter*Batch / outhistory = Waferhist
                            nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of charts. Figure 17.109 contains a partial listing of Waferhist.

Figure 17.109 Partial Listing of the Summary Data Set Waferhist

Summary Data Set for Wafer Diameters

Batch	DiameterX	DiameterR	DiameterN
1	34.992	0.02	5
2	34.994	0.03	5
3	34.998	0.01	5
4	34.998	0.02	5
5	34.992	0.02	5

There are four variables in the data set Waferhist:

- Batch contains the subgroup index.
- DiameterX contains the subgroup means.
- DiameterR contains the subgroup ranges.
- DiameterN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *R*, and *N* to the *process* Diameter specified in the XRCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1865.

Saving Control Limits

NOTE: See *Mean and Range (X-Bar and R) Charts* in the SAS/QC Sample Library.

You can save the control limits for $\bar{X}$ and *R* charts in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1848) or modify the limits with a DATA step program.

The following statements read measurements from the data set Wafers (see "Creating Charts for Means and Ranges from Raw Data" on page 1838) and save the control limits displayed in Figure 17.106 in Waferlim:

```
proc shewhart data=Wafers;
   xrchart Diameter*Batch / outlimits = Waferlim
                            nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set Waferlim is listed in Figure 17.110.

Figure 17.110 The Data Set Waferlim Containing Control Limit Information

Control Limits for Wafer Diameters

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
Diameter	Batch	ESTIMATE	5	.002699796	3	34.9823	34.9950	35.0077

LCLR	_R_	_UCLR_	_STDDEV_
0	0.022	0.046519	.009458586

The data set Waferlim contains one observation with the limits for *process* Diameter. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the $\bar{X}$ chart. The variables _LCLR_ and _UCLR_ contain the lower and upper control limits for the *R* chart. The variable _MEAN_ contains the central line for the $\bar{X}$ chart, and the variable _R_ contains the central line for the *R* chart. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable* names. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values.

You can save process capability indices in an OUTLIMITS= data set if you provide specification limits with the LSL= and USL= options. This is illustrated by the following statements:

```
proc shewhart data=Wafers;
   xrchart Diameter*Batch / outlimits = Waferlim2
                            usl       = 35.03
                            lsl       = 34.97
                            nochart;
run;
```

The data set Waferlim2 is listed in Figure 17.111.

Figure 17.111 The Data Set Waferlim2 Containing Process Capability Indices

Control Limits and Capability Indices

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_	_LCLR_
Diameter	Batch	ESTIMATE	5	.002699796	3	34.9823	34.9950	35.0077	0

_R_	_UCLR_	_STDDEV_	_LSL_	_USL_	_CP_	_CPL_	_CPU_	_CPK_
0.022	0.046519	.009458586	34.97	35.03	1.05724	0.87962	1.23486	0.87962

The variables _CP_, _CPL_, _CPU_, and _CPK_ contain the process capability indices. It is reasonable to compute capability indices, since Figure 17.106 indicates that the wafer process is in statistical control. However, it is recommended that you also check for normality of the data. You can use the CAPABILITY procedure for this purpose.

For more information, see "OUTLIMITS= Data Set" on page 1864.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Wafers;
   xrchart Diameter*Batch / outtable=Wafertab
                            nochart;
run;
```

The data set Wafertab is listed in Figure 17.112.

Figure 17.112 The Data Set Wafertab

Summary Statistics and Control Limit Information

VAR	Batch	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_STDDEV_
Diameter	1	3	5	5	34.9823	34.992	34.9950	35.0077	.009458586
Diameter	2	3	5	5	34.9823	34.994	34.9950	35.0077	.009458586
Diameter	3	3	5	5	34.9823	34.998	34.9950	35.0077	.009458586
Diameter	4	3	5	5	34.9823	34.998	34.9950	35.0077	.009458586
Diameter	5	3	5	5	34.9823	34.992	34.9950	35.0077	.009458586
Diameter	6	3	5	5	34.9823	34.996	34.9950	35.0077	.009458586
Diameter	7	3	5	5	34.9823	34.996	34.9950	35.0077	.009458586
Diameter	8	3	5	5	34.9823	34.992	34.9950	35.0077	.009458586
Diameter	9	3	5	5	34.9823	34.992	34.9950	35.0077	.009458586
Diameter	10	3	5	5	34.9823	35.000	34.9950	35.0077	.009458586
Diameter	11	3	5	5	34.9823	34.996	34.9950	35.0077	.009458586
Diameter	12	3	5	5	34.9823	34.994	34.9950	35.0077	.009458586
Diameter	13	3	5	5	34.9823	34.992	34.9950	35.0077	.009458586
Diameter	14	3	5	5	34.9823	34.998	34.9950	35.0077	.009458586
Diameter	15	3	5	5	34.9823	34.988	34.9950	35.0077	.009458586
Diameter	16	3	5	5	34.9823	35.000	34.9950	35.0077	.009458586
Diameter	17	3	5	5	34.9823	34.984	34.9950	35.0077	.009458586
Diameter	18	3	5	5	34.9823	35.002	34.9950	35.0077	.009458586
Diameter	19	3	5	5	34.9823	34.988	34.9950	35.0077	.009458586
Diameter	20	3	5	5	34.9823	34.994	34.9950	35.0077	.009458586
Diameter	21	3	5	5	34.9823	34.992	34.9950	35.0077	.009458586
Diameter	22	3	5	5	34.9823	35.002	34.9950	35.0077	.009458586

EXLIM	_LCLR_	_SUBR_	_R_	_UCLR_	_EXLIMR_
	0	0.02	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.01	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.01	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.01	0.022	0.046519	
	0	0.04	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.01	0.022	0.046519	
	0	0.02	0.022	0.046519	
	0	0.01	0.022	0.046519	

Figure 17.112 *continued*

Summary Statistics and Control Limit Information

VAR	Batch	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_STDDEV_
Diameter	23	3	5	5	34.9823	35.004	34.9950	35.0077	.009458586
Diameter	24	3	5	5	34.9823	34.996	34.9950	35.0077	.009458586
Diameter	25	3	5	5	34.9823	34.994	34.9950	35.0077	.009458586

EXLIM	_LCLR_	_SUBR_	_R_	_UCLR_	_EXLIMR_
	0	0.04	0.022	0.046519	
	0	0.03	0.022	0.046519	
	0	0.01	0.022	0.046519	

This data set contains one observation for each subgroup sample. The variables _SUBX_, _SUBR_, and _SUBN_ contain the subgroup means, subgroup ranges, and subgroup sample sizes. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the $\bar{X}$ chart. The variables _LCLR_ and _UCLR_ contain the lower and upper control limits for the R chart. The variable _MEAN_ contains the central line of the $\bar{X}$ chart, and the variable _R_ contains the central line of the R chart. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1866.

An OUTTABLE= data set can be read later as a TABLE= data set. For example, the following statements read Wafertab and display $\bar{X}$ and R charts identical to those in Figure 17.106:

```
title 'Mean and Range Charts for Diameters';
proc shewhart table=Wafertab;
   xrchart Diameter*Batch;
run;
```

Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2096).

For more information, see "TABLE= Data Set" on page 1870.

Reading Preestablished Control Limits

NOTE: See *Mean and Range (X-Bar and R) Charts* in the SAS/QC Sample Library.

In a previous example, the OUTLIMITS= data set saved control limits computed from the measurements in Wafers. This example shows how these limits can be applied to new data provided in the following data set:

```
data Wafers2;
   input Batch @;
   do i=1 to 5;
      input Diameter @;
      output;
   end;
   drop i;
   datalines;
26 34.99 34.99 35.00 34.99 35.00
27 34.99 35.01 34.98 34.98 34.97
28 35.00 34.99 34.99 34.99 35.01
29 34.98 34.96 34.98 34.98 34.99
30 34.98 35.00 34.98 34.98 34.99
31 35.00 35.00 34.99 35.01 35.01
32 35.00 34.99 34.98 34.98 35.00
33 34.98 35.00 34.99 35.00 35.01
34 35.00 34.97 35.00 34.99 35.01
35 34.99 34.99 34.98 34.99 34.98
36 35.01 34.98 34.99 34.99 35.00
37 35.01 34.99 34.97 34.98 35.00
38 34.98 34.99 35.00 34.98 35.00
39 34.99 34.99 34.99 34.99 35.01
40 34.99 35.01 35.00 35.01 34.99
41 34.99 35.00 34.99 34.98 34.99
42 35.00 34.99 34.98 34.99 35.00
43 34.99 34.98 34.98 34.99 34.99
44 35.00 35.00 34.98 35.00 34.99
45 34.99 34.99 35.00 34.99 34.99
;
```

The following statements use the control limits in Waferlim to create $\bar{X}$ and R charts for the data in Wafers2:

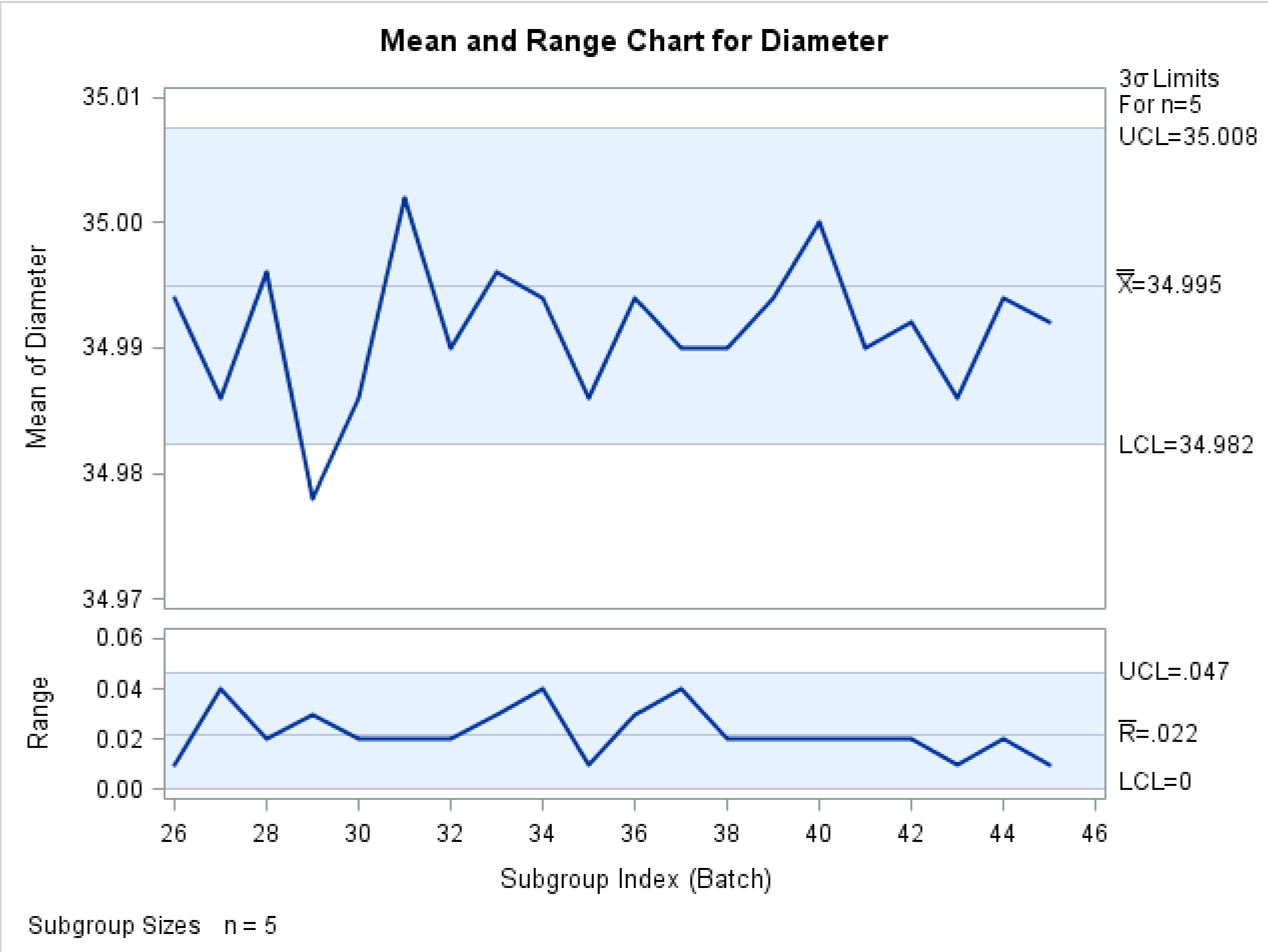
```
ods graphics on;
proc shewhart data=Wafers2 limits=Waferlim;
   xrchart Diameter*Batch;
run;
```

The ODS GRAPHICS ON statement specifies that the $\bar{X}$ and R charts are produced using ODS Graphics. The LIMITS= option in the PROC SHEWHART statement specifies the data set containing the control limits. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name Diameter
- the value of _SUBGRP_ matches the *subgroup-variable* name Batch

The charts are shown in Figure 17.113.

Figure 17.113 $\bar{X}$ and R Charts for Second Set of Wafer Data (ODS Graphics)



Note that the mean diameter of the 29th batch lies below the lower control limit in the $\bar{X}$ chart, signaling a special cause of variation.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1868 for details concerning the variables that you must provide.

Syntax: XRCHART Statement

The basic syntax for the XRCHART statement is as follows:

XRCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

XRCHART *processes* ∗ *subgroup-variable* < **(***block-variables***)** >
< **=***symbol-variable* | **=**'*character*' > / < *options* > **;**

You can use any number of XRCHART statements in the SHEWHART procedure. The components of the XRCHART statement are described as follows.

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If raw data are read from a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Means and Ranges from Raw Data" on page 1838.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Means and Ranges from Summary Data" on page 1841.
- If summary data and control limits are read from a TABLE= data set, *process* must be a value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1845.

A *process* is required. If you specify more than one *process*, enclose the list in parentheses. For example, the following statements request distinct $\bar{X}$ and R charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   xrchart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding XRCHART statement, Day is the subgroup variable. For details, see "Subgroup Variables" on page 1871.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2026 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the means and ranges.

- Distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2025 for an example.
- If you specify the LINEPRINTER option in the PROC SHEWHART statement, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.

character

specifies a plotting character for charts produced with the LINEPRINTER option. For example, the following statements use an asterisk (*) to plot the points on the $\bar{X}$ and R charts:

```
proc shewhart data=Values lineprinter;
   xrchart Weight*Day='*';
run;
```

options

enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The section "Summary of Options", which follows, lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1946 describes each option in detail.

Summary of Options

The following tables list the XRCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1946.

Table 17.70 XRCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	requests probability limits for chart
LIMITN=	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	specifies width of control limits in terms of multiple *k* of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	specifies color for area inside control limits
CLIMITS=	specifies color of control limits, central line, and related labels
LCLLABEL=	specifies label for lower control limit on $\bar{X}$ chart
LCLLABEL2=	specifies label for lower control limit on *R* chart
LIMLABSUBCHAR=	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	specifies line type for control limits
NDECIMAL=	specifies number of digits to right of decimal place in default labels for control limits and central line on $\bar{X}$ chart
NDECIMAL2=	specifies number of digits to right of decimal place in default labels for control limits and central line on *R* chart

Table 17.70 *continued*

Option	Description
NOCTL	suppresses display of central line on $\bar{X}$ chart
NOCTL2	suppresses display of central line on R chart
NOLCL	suppresses display of lower control limit on $\bar{X}$ chart
NOLCL2	suppresses display of lower control limit on R chart
NOLIMIT0	suppresses display of zero lower control limit on R chart
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit on $\bar{X}$ chart
NOUCL2	suppresses display of upper control limit on R chart
RSYMBOL=	specifies label for central line on R chart
UCLLABEL=	specifies label for upper control limit on $\bar{X}$ chart
UCLLABEL2=	specifies label for upper control limit on R chart
WLIMITS=	specifies width for control limits and central line
XSYMBOL=	specifies label for central line on $\bar{X}$ chart
Process Mean and Standard Deviation Options	
MU0=	specifies known value of μ_0 for process mean μ
SIGMA0=	specifies known value σ_0 for process standard deviation σ
SMETHOD=	specifies method for estimating process standard deviation σ
TYPE=	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	labels every point on $\bar{X}$ chart
ALLLABEL2=	labels every point on R chart
CLABEL=	specifies color for labels
CCONNECT=	specifies color for line segments that connect points on chart
CFRAMELAB=	specifies fill color for frame around labeled points
CNEEDLES=	specifies color for needles that connect points to central line
COUT=	specifies color for portions of line segments that connect points outside control limits
COUTFILL=	specifies color for shading areas between the connected points and control limits outside the limits
LABELANGLE=	specifies angle at which labels are drawn
LABELFONT=	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	specifies height of labels (alias for the TESTHEIGHT= option)

Table 17.70 *continued*

Option	Description
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=	labels points outside control limits on $\bar{X}$ chart
OUTLABEL2=	labels points outside control limits on *R* chart
SYMBOLLEGEND=	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=	specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	computes zone widths independently above and below center line
NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	suppresses tests across *phase* boundaries
TESTS=	specifies tests for special causes for the $\bar{X}$ chart
TESTS2=	specifies tests for special causes for the *R* chart
TEST2RESET=	enables tests for special causes to be reset for the *R* chart
TEST2RUN=	specifies length of pattern for Test 2
TEST3RUN=	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL=	provides labels for points where test is positive
TESTLABEL*n*=	specifies label for *n*th test for special causes
TESTNMETHOD=	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=	enables tests for special causes to be reset
WESTGARD=	requests that Westgard rules be applied to the $\bar{X}$ chart
ZONELABELS	adds labels A, B, and C to zone lines for $\bar{X}$ chart
ZONE2LABELS	adds labels A, B, and C to zone lines for *R* chart
ZONES	adds lines to $\bar{X}$ chart delineating zones A, B, and C
ZONES2	adds lines to *R* chart delineating zones A, B, and C
ZONEVALPOS=	specifies position of ZONEVALUES labels
ZONEVALUES	labels $\bar{X}$ chart zone lines with their values
ZONE2VALUES	labels *R* zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	specifies color for boxes enclosing labels indicating points where test is positive
CTESTS=	specifies color for labels indicating points where test is positive
CTESTSYMBOL=	specifies color for symbol used to plot points where test is positive
CZONES=	specifies color for lines and labels delineating zones A, B, and C

Table 17.70 *continued*

Option	**Description**
LTESTS=	specifies type of line connecting points where test is positive
LZONES=	specifies line type for lines delineating zones A, B, and C
TESTFONT=	specifies software font for labels at points where test is positive
TESTHEIGHT=	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	specifies plot symbol for points where test is positive
TESTSYMBOLHT=	specifies symbol height for points where test is positive
WTESTS=	specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill colors for frame for plot area
CTEXT=	specifies color for tick mark values and axis labels
DISCRETE	produces horizontal axis for discrete numeric group values
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default to R chart
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis of $\bar{X}$ chart
VAXIS2=	specifies major tick mark values for vertical axis of R chart
VFORMAT=	specifies format for primary vertical axis tick mark labels
VFORMAT2=	specifies format for secondary vertical axis tick mark labels

Table 17.70 *continued*

Option	Description
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=	specifies width of axis lines
Plot Layout Options	
ALLN	plots means for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of charts
NOCHART2	suppresses creation of R chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
SEPARATE	displays $\bar{X}$ and R charts on separate screens or pages
TOTPANELS=	specifies number of pages or screens to be used to display chart
YPCT1=	specifies length of vertical axis on $\bar{X}$ chart as a percentage of sum of lengths of vertical axes for $\bar{X}$ and R charts
ZEROSTD	displays $\bar{X}$ chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= and HREF2= options
CVREF=	specifies color for lines requested by VREF= and VREF2= options
HREF=	specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2=	specifies position of reference lines perpendicular to horizontal axis on R chart

Table 17.70 *continued*

Option	Description
HREFDATA=	specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2DATA=	specifies position of reference lines perpendicular to horizontal axis on R chart
HREFLABELS=	specifies labels for HREF= lines
HREF2LABELS=	specifies labels for HREF2= lines
HREFLABPOS=	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	specifies line type for HREF= and HREF2= lines
LVREF=	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	specifies position of reference lines perpendicular to vertical axis on $\bar{X}$ chart
VREF2=	specifies position of reference lines perpendicular to vertical axis on R chart
VREFLABELS=	specifies labels for VREF= lines
VREF2LABELS=	specifies labels for VREF2= lines
VREFLABPOS=	position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=	specifies line type for grid requested with the ENDGRID option
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme points are clipped
CLIPLEGEND=	specifies text for clipping legend
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features to $\bar{X}$ chart
ANNOTATE2=	specifies annotate data set that adds features to R chart
DESCRIPTION=	specifies description of $\bar{X}$ chart's GRSEG catalog entry
DESCRIPTION2=	specifies description of R chart's GRSEG catalog entry
FONT=	specifies software font for labels and legends on charts

Table 17.70 *continued*

Option	Description
NAME=	specifies name of $\bar{X}$ chart's GRSEG catalog entry
NAME2=	specifies name of R chart's GRSEG catalog entry
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	shades areas between control limits and connected points lying outside the limits
STARFILL=	specifies a variable identfying groups of stars filled with different colors
STARS=	specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	specifies the control limit infill transparency
MARKERS	plots subgroup points with markers
NOBLOCKREF	suppresses block and phase reference lines
NOBLOCKREFFILL	suppresses block and phase wall fills
NOFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOPHASEREF	suppresses block and phase reference lines
NOPHASEREFFILL	suppresses block and phase wall fills
NOREF	suppresses block and phase reference lines
NOREFFILL	suppresses block and phase wall fills
NOSTARFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	specifies a graph footnote
ODSFOOTNOTE2=	specifies a secondary graph footnote
ODSLEGENDEXPAND	specifies that legend entries contain all levels observed in the data
ODSTITLE=	specifies a graph title
ODSTITLE2=	specifies a secondary graph title
OUTFILLTRANSPARENCY=	specifies control limit outfill transparency
OVERLAYURL=	specifies URLs to associate with overlay points
OVERLAY2URL=	specifies URLs to associate with overlay points on secondary chart

Table 17.70 *continued*

Option	Description
PHASEPOS=	specifies vertical position of phase legend
PHASEREFLEVEL=	associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	draws central line labels using a simulated software font
STARTRANSPARENCY=	specifies star fill transparency
URL=	specifies a variable whose values are URLs to be associated with subgroups
URL2=	specifies a variable whose values are URLs to be associated with subgroups on secondary chart
Input Data Set Options	
MISSBREAK	specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	creates output data set containing subgroup summary statistics
OUTINDEX=	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	creates output data set containing control limits
OUTTABLE=	creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive
Specification Limit Options	
CIINDICES	specifies α value and type for computing capability index confidence limits
LSL=	specifies list of lower specification limits
TARGET=	specifies list of target values
USL=	specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for *block-variable* legend

Table 17.70 *continued*

Option	Description
BLOCKLABTYPE=	specifies text size of *block-variable* legend
BLOCKPOS=	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE=	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	specifies color for STARCIRCLES= circles
CSTARFILL=	specifies color for filling stars
CSTAROUT=	specifies outline color for stars exceeding inner or outer circles
CSTARS=	specifies color for outlines of stars
LSTARCIRCLES=	specifies line types for STARCIRCLES= circles
LSTARS=	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	specifies reference circles for stars
STARINRADIUS=	specifies inner radius of stars
STARLABEL=	specifies vertices to be labeled
STARLEGEND=	specifies style of legend for star vertices
STARLEGENDLAB=	specifies label for STARLEGEND= legend
STAROUTRADIUS=	specifies outer radius of stars
STARSPECS=	specifies method used to standardize vertex variables
STARSTART=	specifies angle for first vertex
STARTYPE=	specifies graphical style of star
STARVERTICES=	superimposes star at each point on $\bar{X}$ chart
WSTARCIRCLES=	specifies width of STARCIRCLES= circles
WSTARS=	specifies width of STARVERTICES= stars

Table 17.70 *continued*

Option	Description
Overlay Options	
CCOVERLAY=	specifies colors for primary chart overlay line segments
CCOVERLAY2=	specifies colors for secondary chart overlay line segments
COVERLAY=	specifies colors for primary chart overlay plots
COVERLAY2=	specifies colors for secondary chart overlay plots
COVERLAYCLIP=	specifies color for clipped points on overlays
LOVERLAY=	specifies line types for primary chart overlay line segments
LOVERLAY2=	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=	specifies variables to overlay on primary chart
OVERLAY2=	specifies variables to overlay on secondary chart
OVERLAY2HTML=	specifies links to associate with secondary chart overlay points
OVERLAY2ID=	specifies labels for secondary chart overlay points
OVERLAY2SYM=	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
OVERLAYHTML=	specifies links to associate with primary chart overlay points
OVERLAYID=	specifies labels for primary chart overlay points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols for primary chart overlays
OVERLAYSYMHT=	specifies symbol heights for primary chart overlays
WOVERLAY=	specifies widths of primary chart overlay line segments
WOVERLAY2=	specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	specifies a variable whose values create links to be associated with subgroups
HTML2=	specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	creates an OUTTABLE= data set with additional graphics coordinate data
Options for Line Printer Charts	
CLIPCHAR=	specifies plot character for clipped points
CONNECTCHAR=	specifies character used to form line segments that connect points on chart
HREFCHAR=	specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	specifies characters indicating *symbol-variable*

Table 17.70 *continued*

Option	Description
TESTCHAR=	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	specifies line character for VREF= and VREF2= lines
ZONECHAR=	specifies character for lines that delineate zones for tests for special causes

Details: XRCHART Statement

Constructing Charts for Means and Ranges

The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	mean of measurements in ith subgroup
R_i	range of measurements in ith subgroup
n_i	sample size of ith subgroup
N	number of subgroups
$\overline{\overline{X}}$	weighted average of subgroup means
$d_2(n)$	expected value of the range of n independent normally distributed variables with unit standard deviation
$d_3(n)$	standard error of the range of n independent observations from a normal population with unit standard deviation
z_p	$100 \times p$th percentile of the standard normal distribution
$D_p(n)$	$100 \times p$th percentile of the distribution of the range of n independent observations from a normal population with unit standard deviation

Plotted Points

Each point on the $\bar{X}$ chart indicates the value of a subgroup mean ($\bar{X}_i$). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 14, the mean plotted for this subgroup is

$$\bar{X}_{10} = \frac{12 + 15 + 19 + 16 + 14}{5} = 15.2$$

Each point on the R chart indicates the value of a subgroup range (R_i). For example, the range plotted for the tenth subgroup is $R_{10} = 19 - 12 = 7$.

Central Lines

On an $\bar{X}$ chart, by default, the central line indicates an estimate of μ, which is computed as

$$\hat{\mu} = \overline{\overline{X}} = \frac{n_1 \bar{X}_1 + \cdots + n_N \bar{X}_N}{n_1 + \cdots + n_N}$$

If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

On an R chart, by default, the central line for the ith subgroup indicates an estimate for the expected value of R_i, which is computed as $d_2(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $d_2(n_i)\sigma_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard errors of $\bar{X}_i$ and R_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that $\bar{X}_i$ or R_i exceeds the limits

The following table provides the formulas for the limits.

Table 17.71 Limits for $\bar{X}$ and R Charts

Control Limits	
$\bar{X}$ Chart	LCL = lower limit = $\overline{\overline{X}} - k\hat{\sigma}/\sqrt{n_i}$
	UCL = upper limit = $\overline{\overline{X}} + k\hat{\sigma}/\sqrt{n_i}$
R Chart	LCL = lower limit = $\max(d_2(n_i)\hat{\sigma} - kd_3(n_i)\hat{\sigma}, 0)$
	UCL = upper limit = $d_2(n_i)\hat{\sigma} + kd_3(n_i)\hat{\sigma}$
Probability Limits	
$\bar{X}$ Chart	LCL = lower limit = $\overline{\overline{X}} - z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$
	UCL = upper limit = $\overline{\overline{X}} + z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$
R Chart	LCL = lower limit = $D_{\alpha/2}\hat{\sigma}$
	UCL = upper limit = $D_{1-\alpha/2}\hat{\sigma}$

The formulas for R charts assume that the data are normally distributed. If standard values μ_0 and σ_0 are available for μ and σ, respectively, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 17.71. Note that the limits vary with n_i and that the probability limits for R_i are asymmetric around the central line.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.

- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables are saved.

Table 17.72 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLR	lower control limit for subgroup range
LCLX	lower control limit for subgroup mean
LIMITN	nominal sample size associated with the control limits
LSL	lower specification limit
MEAN	process mean ($\overline{\overline{X}}$ or μ_0)
_R_	value of central line on R chart
SIGMAS	multiple (k) of standard error of $\bar{X}_i$ or R_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the XRCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLR	upper control limit for subgroup range
UCLX	upper control limit for subgroup mean
USL	upper specification limit
VAR	*process* specified in the XRCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value V is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLR_, _R_, and _UCLR_.
2. If the limits are defined in terms of a multiple k of the standard errors of $\bar{X}_i$ and R_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.

3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.

4. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1874 for computational details.

5. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the XRCHART statement. For an example, see "Saving Control Limits" on page 1845.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup range variable named by *process* suffixed with *R*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name containing 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Variables containing subgroup means, ranges, and sample sizes are created for each *process* specified in the XRCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   xrchart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthX, WidthR, WidthN, DiameterX, DiameterR, and DiameterN.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1844.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on $\bar{X}$ chart
EXLIMR	control limit exceeded on *R* chart
LCLR	lower control limit for range
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
_R_	average range
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBN	subgroup sample size
SUBR	subgroup range
SUBX	subgroup mean
TESTS	tests for special causes signaled on $\bar{X}$ chart
TESTS2	tests for special causes signaled on *R* chart
UCLR	upper control limit for range
UCLX	upper control limit for mean
VAR	*process* specified in the XRCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved, depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).

2. The variable _TESTS_ is saved if you specify the TESTS= option. The *k*th character of a value of _TESTS_ is *k* if Test *k* is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.

3. The variable _TESTS2_ is saved if you specify the TESTS2= option.

4. The variables _EXLIM_, _EXLIMR_, _TESTS_, and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1845.

ODS Tables

The following table summarizes the ODS tables that you can request with the XRCHART statement.

Table 17.73 ODS Tables Produced with the XRCHART Statement

Table Name	Description	Options
XRCHART	$\bar{X}$ and R chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*). **NOTE:** In SAS/QC 13.1 the SHEWHART procedure does not support the creation of graphs that are editable with the ODS Graphics Editor.

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. XRCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics and traditional graphics with ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1946.

When ODS Graphics is in effect, the XRCHART statement assigns a name to the graph it creates. You can use this name to reference the graph with ODS statements. The name is listed in Table 17.74.

Table 17.74 ODS Graphics Produced by the XRCHART Statement

ODS Graph Name	Plot Description
XRChart	$\bar{X}$ and R chart

See Chapter 3, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the XRCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the XRCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the ith subgroup contains n_i items, there should be n_i consecutive observations for which the value of the subgroup variable is the index of the ith subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2031).

For an example of a DATA= data set, see "Creating Charts for Means and Ranges from Raw Data" on page 1838.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   xrchart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly by using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLX_, _MEAN_, _UCLX_, _LCLR_, _R_, and _UCLR_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 17.71

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1848.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedure or to read output data sets created with SAS summarization procedures, such as the MEANS procedure.

A HISTORY= data set used with the XRCHART statement must contain the following variables:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup range variable for each *process*
- a subgroup sample size variable for each *process*

The names of the subgroup mean, subgroup range, and subgroup sample size variables must be the *process* name concatenated with the special suffix characters *X*, *R*, and *N*, respectively.

For example, consider the following statements:

```
proc shewhart history=Summary;
   xrchart (Weight Yieldstrength)*Batch;
run;
```

The data set Summary must include the variables Batch, WeightX, WeightR, WeightN, YieldstrengthX, YieldstrengthR, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see "Displaying Stratification in Phases" on page 2031 for an example).

For an example of a HISTORY= data set, see "Creating Charts for Means and Ranges from Summary Data" on page 1841.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure or to read data sets created by other SAS procedures. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in "Specialized Control Charts: SHEWHART Procedure" on page 2096.

Table 17.75 lists the variables required in a TABLE= data set used with the XRCHART statement.

Table 17.75 Variables Required in a TABLE= Data Set

Variable	Description
LCLR	lower control limit for range
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
_R_	average range
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBR	subgroup range
SUBX	subgroup mean
UCLR	upper control limit for range
UCLX	upper control limit for mean

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*

- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes for subgroup means and must be a character variable of length 8.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes for subgroup ranges and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1845.

Subgroup Variables

The values of the *subgroup-variable*, which is specified in the chart statement, indicate how the observations in the input data set (a DATA=, HISTORY=, or TABLE= data set) are arranged into rational subgroups.[9] Typically, the values of the *subgroup-variable* are one of the following:

- *indices* that give the order in which subgroup samples were collected (for example, 1, 2, 3, . . .). An unformatted numeric *subgroup-variable* is appropriate for this situation. For an example that uses this type of *subgroup-variable*, see "Creating Charts for Means and Ranges from Raw Data" on page 1838.
- the *dates* or *times* at which subgroup samples were collected (for example, 01JUN, 02JUN, 03JUN, . . .). A numeric *subgroup-variable* with a SAS date, time, or datetime format is appropriate for this situation. You can optionally associate a format with the *subgroup-variable* by using a FORMAT statement; refer to *SAS Formats and Informats: Reference* for details. For an example that uses this type of *subgroup-variable*, see Example 17.40.
- *labels* that uniquely identify subgroup samples (for example, LOT39, LOTX12, LOT43A). A character *subgroup-variable* (with or without a format) is appropriate for this situation. For an example that uses this type of *subgroup-variable*, see Example 17.38.

The values of the *subgroup-variable* also determine how the horizontal axis of the control chart is scaled and labeled.

The notion of a rational subgroup is fundamental to the application of a Shewhart chart. You should select your subgroups so that if special causes of variation are present, the opportunity for variation within subgroups is minimized while the opportunity for variation between subgroups is maximized. In other words, the conditions within a subgroup should be homogeneous. The reason for this requirement is that the construction of the control limits is based on within-subgroup variability. Refer to Montgomery (1996) and Wheeler and Chambers (1986) for approaches to rational subgrouping.

[9]This discussion also applies to the use of *subgroup-variables* in the CUSUM procedure and the MACONTROL procedure.

The selection of subgroups is both a practical and a statistical issue that requires knowledge of the process and the sampling or measurement procedure. The values of the subgroup-variable should reflect the selection of subgroups and should not be assigned arbitrarily. Incorrect subgrouping or assignment of subgroup-variable values can result in control limits that are too tight or too wide.

If the input data set is a HISTORY= or TABLE= data set, each observation represents a distinct subgroup, and, consequently, the observations within each BY group must have distinct subgroup variable values. Similarly, if the input data set is a DATA= data set and you are using the CCHART, IRCHART, NPCHART, PCHART, or UCHART statement, each observation represents a distinct subgroup, and, consequently, the observations within each BY group must have distinct subgroup variable values. However, if the input data set is a DATA= data set and you are using the BOXCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, or XSCHART statement, subgroups are identified by groups of consecutive observations with identical values of the subgroup-variable.

The order of the observations in the input data set and the scaling of the horizontal axis depend on the type of the subgroup-variable, which can be numeric or character.

Numeric Subgroup Variables

If the subgroup-variable is numeric, the observations must be sorted in increasing order of the values of the subgroup variable. If you use a BY statement, first sort by the BY variables and then by the subgroup variable.

The unformatted values of the subgroup-variable are used to scale the horizontal axis of the control chart, and the formatted values are used to label the major tick marks on the horizontal axis. As a result, the horizontal distance between two points corresponding to consecutive subgroups is proportional to the difference between their unformatted subgroup values.

If a DATE, DATETIME, WEEKDATE, or WORDDATE format is associated with the subgroup variable, the major tick mark labels are split and displayed in two levels to save space. You can override this default with the TURNHLABELS option (which turns the labels vertically) or with tick label options in an AXIS*n* statement specified with the HAXIS= option.

Character Subgroup Variables

If the subgroup-variable is numeric, the order of the observations is not checked. The horizontal axis is scaled so that the subgroups are spaced uniformly. Formatted subgroup variable values are used to label the major tick marks.

You can use a character subgroup variable to avoid gaps between groups of points or time values on a control chart. You can also use a character subgroup variable to create a chart in which the order of the points depends only on the order in which the subgroups are arranged in the input data set.

You should verify the order of the observations in the input data set before you use a character subgroup variable in conjunction with the TESTS= option. With the exception of Test 1, the tests for special causes are applicable only if the subgroups are provided in chronological order. See "Tests for Special Causes: SHEWHART Procedure" on page 2073 for details.

To avoid collision of adjacent tick labels on the horizontal axis, the labels are thinned by default. You can override this default with the TURNHLABELS option or with tick label options in an AXIS*n* statement specified with the HAXIS= option.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, three methods (referred to as default, MVLUE, and MVGRANGE) are available for estimating σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{R_1/d_2(n_1) + \cdots + R_N/d_2(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, and R_i is the sample range of the observations $x_{i1}, \ldots, x_{in_i}$ in the ith subgroup.

$$R_i = \max_{1 \leq j \leq n_i}(x_{ij}) - \min_{1 \leq j \leq n_i}(x_{ij})$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$. The unbiasing factor $d_2(n_i)$ is defined so that, if the observations are normally distributed, the expected value of R_i is $d_2(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). The MVLUE is a weighted average of N unbiased estimates of σ of the form $R_i/d_2(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{f_1 R_1/d_2(n_1) + \cdots + f_N R_N/d_2(n_N)}{f_1 + \cdots + f_N}$$

where

$$f_i = \frac{[d_2(n_i)]^2}{[d_3(n_i)]^2}$$

A subgroup range R_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The unbiasing factor $d_3(n_i)$ is defined so that, if the observations are normally distributed, the expected value of σ_{R_i} is $d_3(n_i)\sigma$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

MVGRANGE Method

If you specify SMETHOD=MVGRANGE, σ is estimated by using a moving range of subgroup averages. This is appropriate for constructing control charts for means when the jth measurement in the ith subgroup can be modeled as $x_{ij} = \sigma_B \omega_i + \sigma_W \epsilon_{ij}$, where σ_B^2 is the between-subgroup variance, σ_W^2 is the within-subgroup variance, the ω_i are independent with zero mean and unit variance, and the ω_i are independent of the ϵ_{ij}.

The estimate for σ is

$$\hat{\sigma} = \bar{R}/d_2(n)$$

where $\bar{R}$ is the average of the moving ranges, n is the number of consecutive subgroup averages used to compute each moving range, and the unbiasing factor $d_2(n)$ is defined so that if the subgroup averages are normally distributed, the expected value of R_i is

$$E(R_i) = d_2(n_i)\sigma$$

This method is appropriate for constructing the three-way control chart that is advocated for this situation by Wheeler (1995). A three-way control chart is useful when sampling, or *within-group* variation is not the only source of variation, as discussed in "Multiple Components of Variation" on page 2106. A three-way control chart comprises a chart of subgroup means, a moving range chart of the subgroup means, and a chart of subgroup ranges. When you specify the SMETHOD=MVGRANGE option, the XRCHART statement produces the appropriate charts of subgroup means and subgroup ranges.

Capability Indices

This section provides formulas for process capability indices, which are saved in the OUTLIMITS= data set when you use the LSL= and USL= options to provide lower and upper specification limits (LSL and USL, respectively) for the *process*. The estimate $\hat{\sigma}$ is computed as described in the previous section, "Methods for Estimating the Standard Deviation" on page 1873

The Index C_p

The process capability index C_p is computed as

$$C_p = (USL - LSL)/6\hat{\sigma}$$

If you do not specify both LSL and USL, the variable _CP_ is assigned a missing value.

The Index CPL

The process capability index CPL is computed as

$$CPL = (\overline{\overline{X}} - LSL)/3\hat{\sigma}$$

If you do not specify LSL, the variable _CPL_ is assigned a missing value.

The Index CPU

The process capability index CPU is computed as

$$CPU = (USL - \overline{\overline{X}})/3\hat{\sigma}$$

If you do not specify USL, the variable _CPU_ is assigned a missing value.

The Index C_{pk}

The process capability index C_{pk} is computed as

$$C_{pk} = \min(USL - \overline{\overline{X}}, \overline{\overline{X}} - LSL)/3\hat{\sigma}$$

If you specify only USL, the index C_{pk} is computed as

$$C_{pk} = (USL - \overline{\overline{X}})/3\hat{\sigma}$$

and if you specify only LSL, the index C_{pk} is computed as

$$C_{pk} = (\overline{\overline{X}} - LSL)/3\hat{\sigma}$$

The Index C_{pm}

The process capability index C_{pm} is computed as

$$C_{pm} = \frac{\min(T - LSL, USL - T)}{3\sqrt{\hat{\sigma}^2 + (\overline{\overline{X}} - T)^2}}$$

where T is the target value specified with the TARGET= option.

When a single specification limit (SL) and target are specified, C_{pm} is computed as

$$C_{pm} = \frac{|T - SL|}{3\sqrt{\hat{\sigma}^2 + (\overline{\overline{X}} - T)^2}}$$

You can also use the CAPABILITY procedure to compute a variety of capability indices. The SHEWHART procedure and the CAPABILITY procedure use the same formulas to calculate the indices, but they use different estimates for the process standard deviation σ.

- The SHEWHART procedure calculates $\hat{\sigma}$ from subgroup estimates of σ. For details, see the previous section, "Methods for Estimating the Standard Deviation."
- The CAPABILITY procedure calculates $\hat{\sigma}$ as the sample standard deviation of the entire sample. For details, see the section "Standard Deviation" on page 220.

Regardless of which method you use, you should verify that the process is in statistical control before interpreting the indices, and you should verify that the data are normally distributed. The CAPABILITY procedure provides a variety of statistical and graphical tests for checking normality.

Some references use different notation and names for capability indices. For example, the manual ASQC Automotive Division/AIAG (1990) uses the term "process capability indices" for the indices listed in this section, and it uses the term "process performance indices" for the indices computed by the CAPABILITY procedure.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	Input Data Set	Variable
Horizontal	all	*subgroup-variable*
Vertical ($\bar{X}$ chart)	DATA=	*process*
Vertical ($\bar{X}$ chart)	HISTORY=	subgroup mean variable
Vertical ($\bar{X}$ chart)	TABLE=	_SUBX_

You can specify distinct labels for the vertical axes of the $\bar{X}$ and R charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the $\bar{X}$ chart, and the second part labels the vertical axis of the R chart.

For example, the following sets of statements specify the label "Avg Diameter in mm" for the vertical axis of the $\bar{X}$ chart and the label "Range in mm" for the vertical axis of the R chart:

```
proc shewhart data=Wafers;
   xrchart Diameter*Batch / split = '/' ;
   label Diameter = 'Avg Diameter in mm/Range in mm';
run;

proc shewhart history=Wafersum;
   xrchart Diameter*Batch / split = '/' ;
   label DiameterX = 'Avg Diameter in mm/Range in mm';
run;

proc shewhart table=Wafertab;
   xrchart Diameter*Batch / split = '/' ;
   label _SUBX_ = 'Avg Diameter in mm/Range in mm';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: XRCHART Statement

This section provides advanced examples that use the XRCHART statement.

Example 17.38: Applying Tests for Special Causes

NOTE: See *Mean and Range Charts-Tests for Special Causes* in the SAS/QC Sample Library.

This example illustrates how you can apply tests for special causes to make $\bar{X}$ and R charts more sensitive to special causes of variation.

The weight of a roll of tape is measured before and after an adhesive is applied. The difference in weight represents the amount of adhesive applied to the tape during the coating process. The following data set contains the average and the range of the adhesive amounts for 21 samples of five rolls:

```
data Tape;
   input Sample $ WeightX WeightR;
   WeightN=5;
   label WeightX = 'Average Adhesive Amount'
         Sample  = 'Sample Code';
   datalines;
C9  1270  35
C4  1258  25
A7  1248  24
A1  1260  39
A5  1273  29
D3  1260  21
D6  1259  37
D1  1240  37
R4  1260  28
H7  1255  19
H2  1268  36
H6  1253  36
P4  1273  29
P9  1275  22
J7  1257  24
J2  1269  41
J3  1249  36
B2  1264  31
G4  1258  25
G6  1248  36
G3  1248  30
;
```

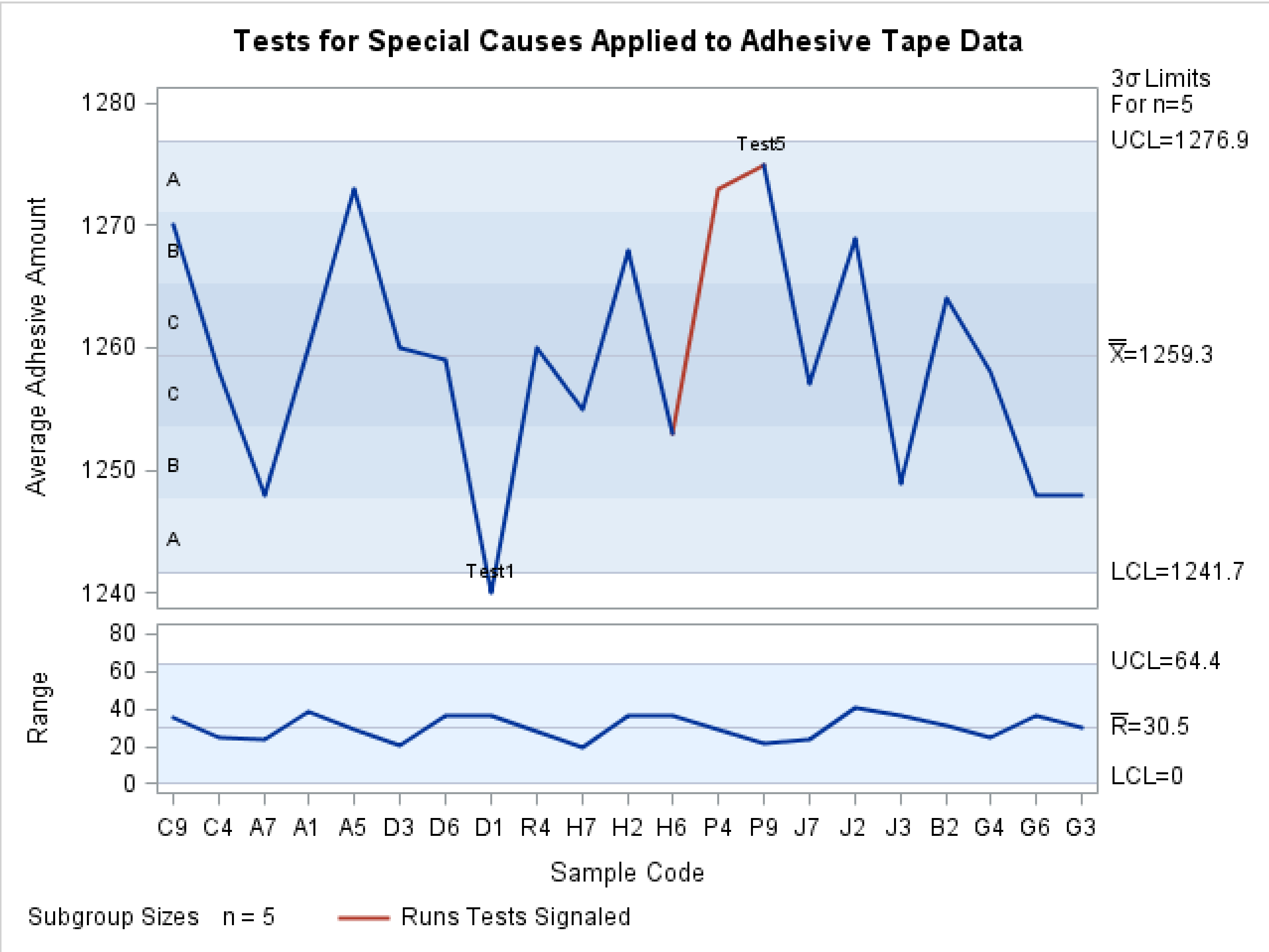
The following statements create $\bar{X}$ and R charts, apply several tests to the $\bar{X}$ chart, and tabulate the results:

```
title 'Tests for Special Causes Applied to Adhesive Tape Data';
ods graphics on;
proc shewhart history=Tape;
   xrchart Weight*Sample / tests    = 1 to 5
                           odstitle = title
                           tabletests
                           zonelabels;
run;
```

The charts are shown in Output 17.38.1, and the table is shown in Output 17.38.2. The TESTS= option requests Tests 1, 2, 3, 4, and 5, which are described in "Tests for Special Causes: SHEWHART Procedure" on page 2073. The TABLETESTS option requests a basic table of subgroup statistics and control limits with a column indicating which subgroups tested positive for special causes.

The ZONELABELS option displays zone lines and zone labels on the $\bar{X}$ chart. The zones are used to define the tests.

Output 17.38.1 Tests for Special Causes Displayed on $\bar{X}$ and R Charts



Output 17.38.1 and Output 17.38.2 indicate that Test 1 is positive at sample D1 and Test 5 is positive at sample P9. Test 1 detects one point beyond Zone A (outside the control limits), and Test 5 detects two out of three points in a row in Zone A or beyond.

Output 17.38.2 Tabular Form of $\bar{X}$ and R Charts

Tests for Special Causes Applied to Adhesive Tape Data

The SHEWHART Procedure

Means and Ranges Chart Summary for Weight									
		3 Sigma Limits with n=5 for Mean				3 Sigma Limits with n=5 for Range			
Sample	Subgroup Sample Size	Lower Limit	Subgroup Mean	Upper Limit	Special Tests Signaled	Lower Limit	Subgroup Range	Upper Limit
C9	5	1241.7065	1270.0000	1276.8650		0	35.000000	64.441879
C4	5	1241.7065	1258.0000	1276.8650		0	25.000000	64.441879
A7	5	1241.7065	1248.0000	1276.8650		0	24.000000	64.441879
A1	5	1241.7065	1260.0000	1276.8650		0	39.000000	64.441879
A5	5	1241.7065	1273.0000	1276.8650		0	29.000000	64.441879
D3	5	1241.7065	1260.0000	1276.8650		0	21.000000	64.441879
D6	5	1241.7065	1259.0000	1276.8650		0	37.000000	64.441879
D1	5	1241.7065	1240.0000	1276.8650	1	0	37.000000	64.441879
R4	5	1241.7065	1260.0000	1276.8650		0	28.000000	64.441879
H7	5	1241.7065	1255.0000	1276.8650		0	19.000000	64.441879
H2	5	1241.7065	1268.0000	1276.8650		0	36.000000	64.441879
H6	5	1241.7065	1253.0000	1276.8650		0	36.000000	64.441879
P4	5	1241.7065	1273.0000	1276.8650		0	29.000000	64.441879
P9	5	1241.7065	1275.0000	1276.8650	5	0	22.000000	64.441879
J7	5	1241.7065	1257.0000	1276.8650		0	24.000000	64.441879
J2	5	1241.7065	1269.0000	1276.8650		0	41.000000	64.441879
J3	5	1241.7065	1249.0000	1276.8650		0	36.000000	64.441879
B2	5	1241.7065	1264.0000	1276.8650		0	31.000000	64.441879
G4	5	1241.7065	1258.0000	1276.8650		0	25.000000	64.441879
G6	5	1241.7065	1248.0000	1276.8650		0	36.000000	64.441879
G3	5	1241.7065	1248.0000	1276.8650		0	30.000000	64.441879

Example 17.39: Specifying Standard Values for the Process Mean and Standard Deviation

NOTE: See *X-bar and R CHARTS-Specifying Standard Values* in the SAS/QC Sample Library.

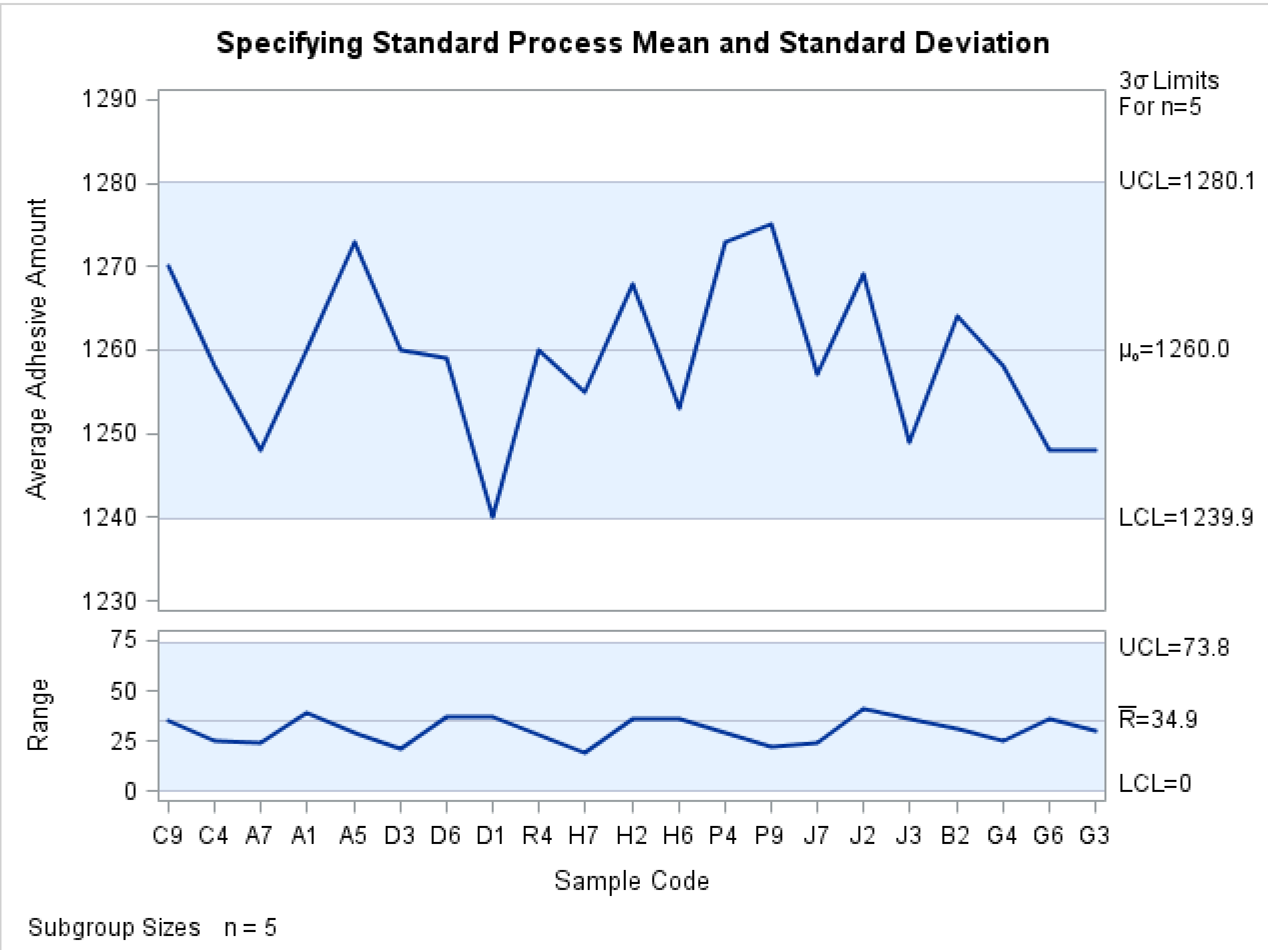
By default, the XRCHART statement estimates the process mean (μ) and standard deviation (σ) from the data, as in the previous example. However, there are applications in which standard values (μ_0 and σ_0) are available based, for instance, on previous experience or extensive sampling. You can specify these values with the MU0= and SIGMA0= options.

For example, suppose it is known that the adhesive coating process introduced in the previous example has a mean of 1260 and standard deviation of 15. The following statements specify these standard values:

```
ods graphics on;
title 'Specifying Standard Process Mean and Standard Deviation';
proc shewhart history=Tape;
   xrchart Weight*Sample / mu0      = 1260
                           sigma0   = 15
                           xsymbol  = mu0
                           odstitle = title;
run;
```

The XSYMBOL= option specifies the label for the central line on the $\bar{X}$ chart. The resulting $\bar{X}$ and R charts are shown in Output 17.39.1.

Output 17.39.1 Specifying Standard Values with MU0= and SIGMA0=



The central lines and control limits for both charts are determined by using μ_0 and σ_0 (see the equations in Table 17.71). Output 17.39.1 indicates that the process is in statistical control.

You can also specify μ_0 and σ_0 with the variables _MEAN_ and _STDDEV_ in a LIMITS= data set, as illustrated by the following statements:

```
data Tapelim;
   length _var_ _subgrp_ _type_ $8;
   _var_    = 'Weight';
   _subgrp_ = 'Sample';
   _type_   = 'STANDARD';
   _limitn_ = 5;
   _mean_   = 1260;
   _stddev_ = 15;

proc shewhart history=Tape limits=Tapelim;
   xrchart Weight*Sample / xsymbol=mu0;
run;
```

The variables _VAR_ and _SUBGRP_ are required, and their values must match the *process* and *subgroup-variable*, respectively, specified in the XRCHART statement. The bookkeeping variable _TYPE_ is not required, but it is recommended to indicate that the variables _MEAN_ and _STDDEV_ provide standard values rather than estimated values.

The resulting charts (not shown here) are identical to those shown in Output 17.39.1.

Example 17.40: Working with Unequal Subgroup Sample Sizes

NOTE: See *X-bar and R Charts with Varying Sample Sizes* in the SAS/QC Sample Library.

The following data set (Wire) contains breaking strength measurements recorded in pounds per inch for 25 samples from a metal wire manufacturing process. The subgroup sample sizes vary between 3 and 7.

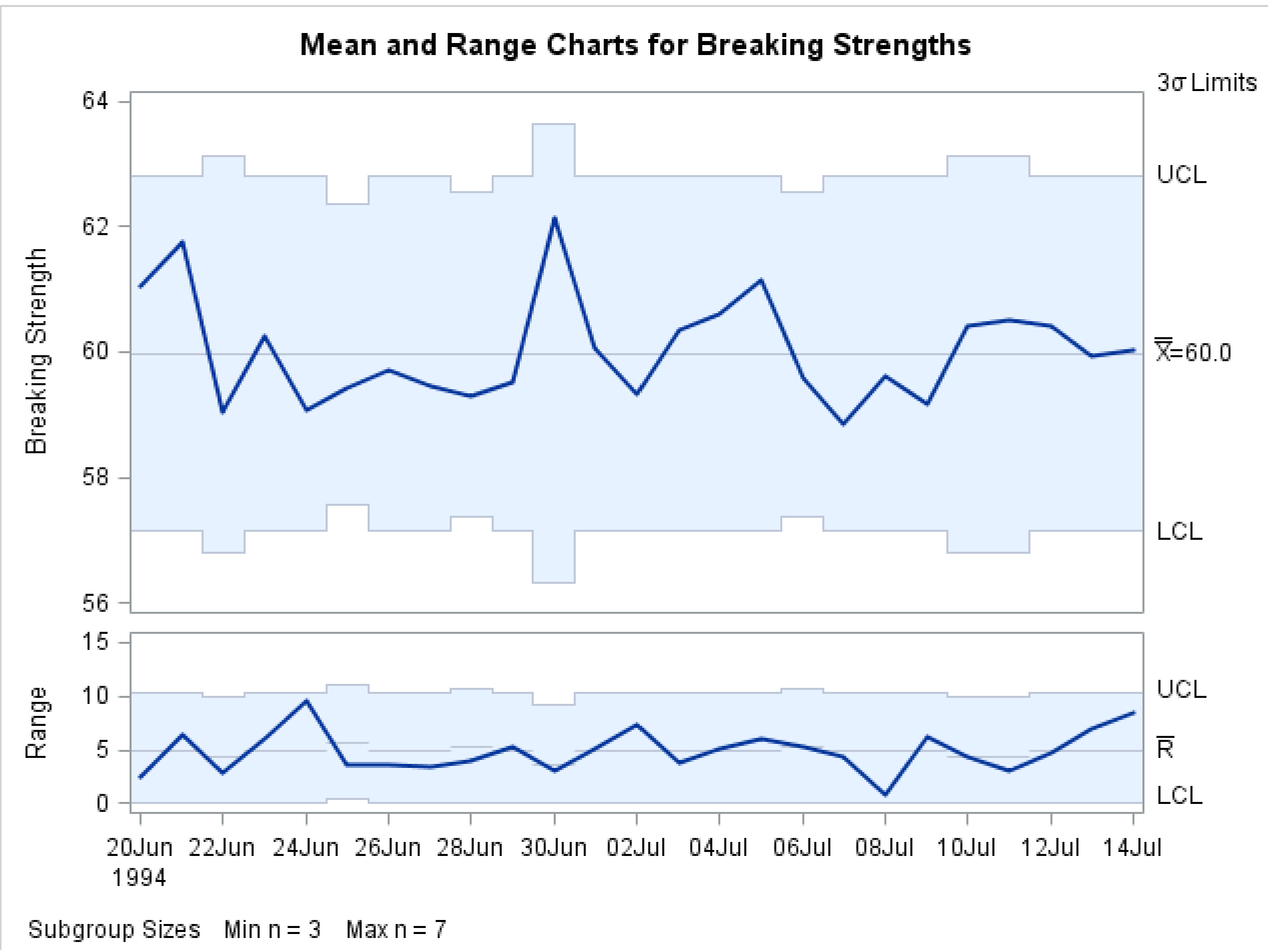
```
data Wire;
   input Day size @;
   informat Day date7.;
   format Day date7.;
   do i=1 to size;
      input Breakstrength @@;
      output;
   end;
   drop i size;
   label Breakstrength = 'Breaking Strength';
   datalines;
20JUN94 5 60.6 62.3 62.0 60.4 59.9
21JUN94 5 61.9 62.1 60.6 58.9 65.3
22JUN94 4 57.8 60.5 60.1 57.7
23JUN94 5 56.8 62.5 60.1 62.9 58.9
24JUN94 5 63.0 60.7 57.2 61.0 53.5
25JUN94 7 58.7 60.1 59.7 60.1 59.1 57.3 60.9
26JUN94 5 59.3 61.7 59.1 58.1 60.3
27JUN94 5 61.3 58.5 57.8 61.0 58.6
28JUN94 6 59.5 58.3 57.5 59.4 61.5 59.6
29JUN94 5 61.7 60.7 57.2 56.5 61.5
30JUN94 3 63.9 61.6 60.9
01JUL94 5 58.7 61.4 62.4 57.3 60.5
02JUL94 5 56.8 58.5 55.7 63.0 62.7
03JUL94 5 62.1 60.6 62.1 58.7 58.3
```

```
04JUL94 5 59.1 60.4 60.4 59.0 64.1
05JUL94 5 59.9 58.8 59.2 63.0 64.9
06JUL94 6 58.8 62.4 59.4 57.1 61.2 58.6
07JUL94 5 60.3 58.7 60.5 58.6 56.2
08JUL94 5 59.2 59.8 59.7 59.3 60.0
09JUL94 5 62.3 56.0 57.0 61.8 58.8
10JUL94 4 60.5 62.0 61.4 57.7
11JUL94 4 59.3 62.4 60.4 60.0
12JUL94 5 62.4 61.3 60.5 57.7 60.2
13JUL94 5 61.2 55.5 60.2 60.4 62.4
14JUL94 5 59.0 66.1 57.7 58.5 58.9
;
```

The following statements request $\bar{X}$ and R charts, shown in Output 17.40.1, for the strength measurements:

```
ods graphics on;
title 'Mean and Range Charts for Breaking Strengths';
proc shewhart data=Wire;
   xrchart Breakstrength*Day / nohlabel
                               odstitle = title;
run;
```

Output 17.40.1 $\bar{X}$ and R Charts with Varying Subgroup Sample Sizes



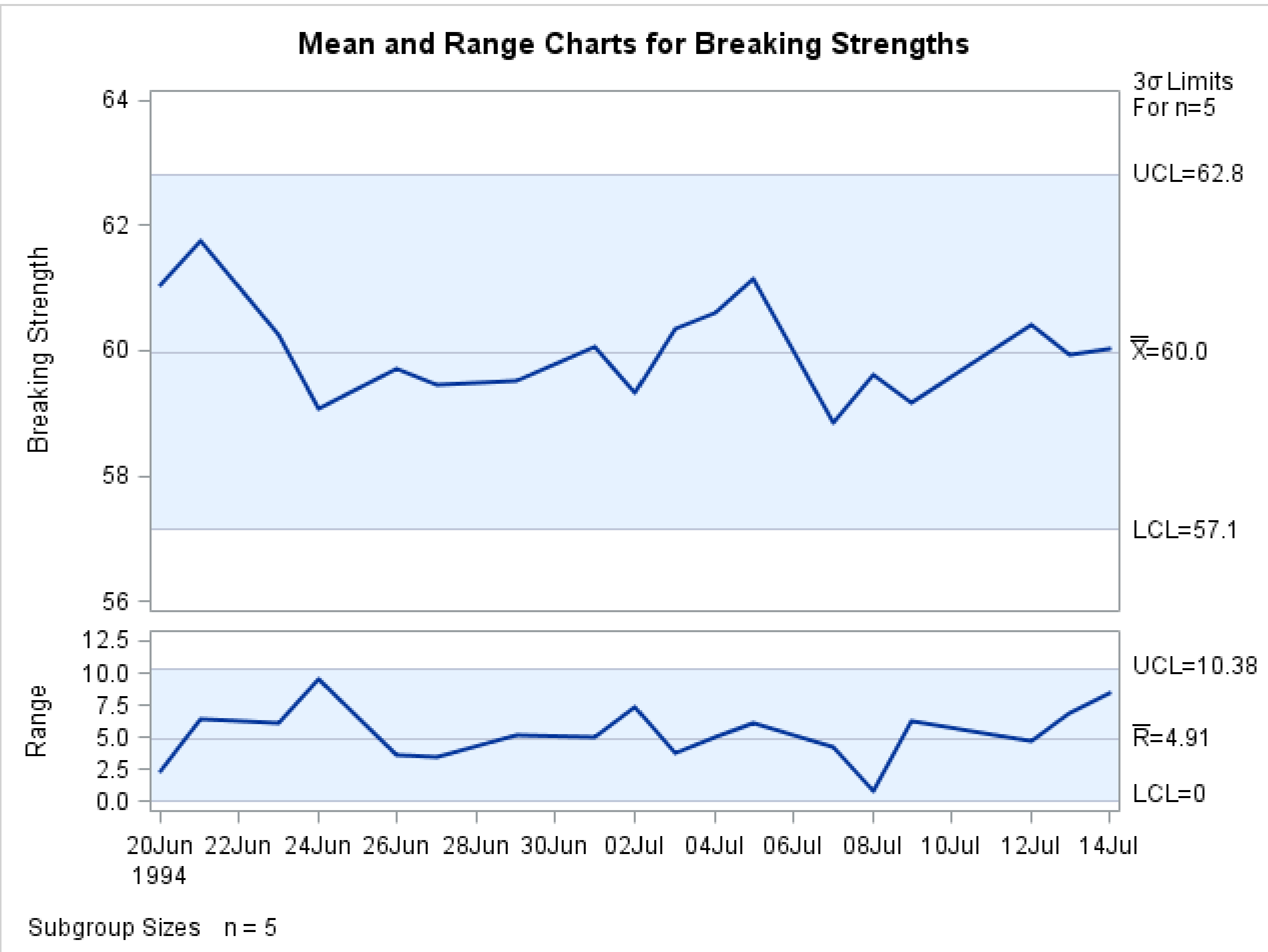
Note that the central line on the *R* chart and the control limits on both charts vary with the subgroup sample size. The sample size legend in the lower-left corner displays the minimum and maximum subgroup sample sizes.

The XRCHART statement provides various options for working with unequal subgroup sample sizes. For example, you can use the LIMITN= option to specify a fixed (nominal) sample size for computing control limits, as illustrated by the following statements:

```
title 'Mean and Range Charts for Breaking Strengths';
proc shewhart data=Wire;
   xrchart Breakstrength*Day / nohlabel
                               odstitle = title
                               limitn   = 5;
run;
```

The resulting charts are shown in Output 17.40.2.

Output 17.40.2 Control Limits Based on Fixed Sample Size

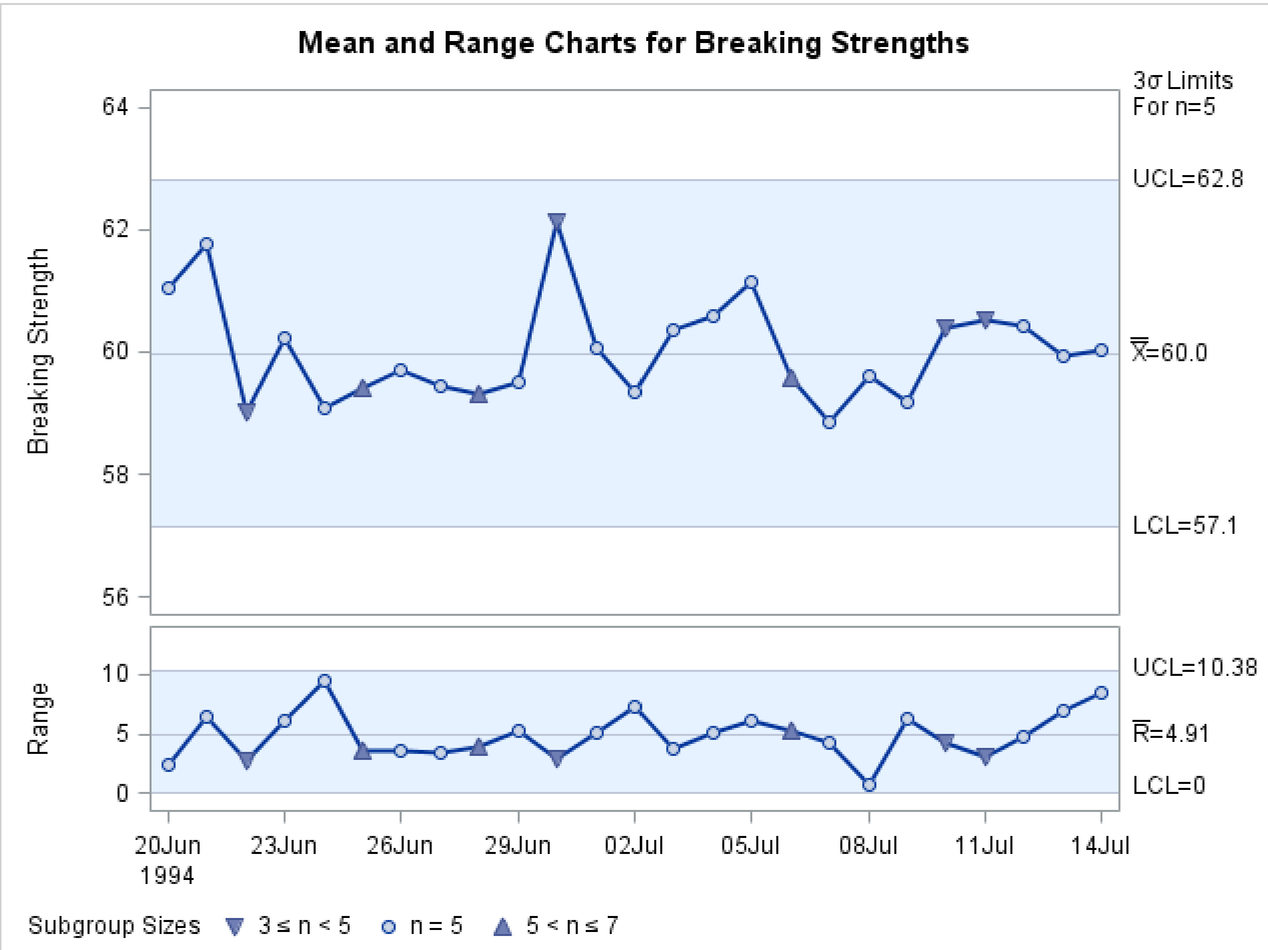


Note that the only points displayed on the chart are those corresponding to subgroups whose sample sizes match the nominal sample size of five. To plot points for all subgroups (regardless of subgroup sample size), you can specify the ALLN option, as follows:

```
title 'Mean and Range Charts for Breaking Strengths';
proc shewhart data=Wire;
   xrchart Breakstrength*Day / nohlabel
                               odstitle = title
                               limitn   = 5
                               alln
                               nmarkers;
run;
```

The charts are shown in Output 17.40.3. The NMARKERS option requests special symbols to identify points for which the subgroup sample size differs from the nominal sample size.

Output 17.40.3 Displaying All Subgroups Regardless of Sample Size



You can use the SMETHOD= option to determine how the process standard deviation σ is to be estimated when the subgroup sample sizes vary. The default method computes $\hat{\sigma}$ as an unweighted average of subgroup estimates of σ. Specifying SMETHOD=MVLUE requests an estimate that assigns greater weight to estimates of σ from subgroups with larger sample sizes. For more information, see "Methods for Estimating the Standard Deviation" on page 1873.

The following statements apply both methods:

```
proc shewhart data=Wire;
   xrchart Breakstrength*Day / outlimits = Wlim1
                               outindex  = 'Default'
                               nochart;
   xrchart Breakstrength*Day / smethod   = mvlue
                               outlimits = Wlim2
                               outindex  = 'MVLUE'
                               nochart;
run;

data Wlimits;
   set Wlim1 Wlim2;
run;
```

The data set Wlimits is listed in Output 17.40.4.

Output 17.40.4 Listing of the Data Set Wlimits

The WLIMITS Data Set

VAR	_SUBGRP_	_INDEX_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_
Breakstrength	Day	Default	ESTIMATE	V	.002699796	3	V
Breakstrength	Day	MVLUE	ESTIMATE	V	.002699796	3	V

MEAN	_UCLX_	_LCLR_	_R_	_UCLR_	_STDDEV_
59.9766	V	V	V	V	2.11146
59.9766	V	V	V	V	2.11240

The variables in an OUTLIMITS= data set whose values vary with subgroup sample size are assigned the special missing value *V*. Consequently, the control limit variables (_LCLX_, _UCLX_, _LCLR_, and _UCLR_), as well as the variables _R_ and _LIMITN_, have this value.

XSCHART Statement: SHEWHART Procedure

Overview: XSCHART Statement

The XSCHART statement creates $\bar{X}$ and s charts for subgroup means and standard deviations, which are used to analyze the central tendency and variability of a process.

You can use options in the XSCHART statement to

- compute control limits from the data based on a multiple of the standard error of the plotted means and standard deviations or as probability limits
- tabulate subgroup sample sizes, subgroup means, subgroup standard deviations, control limits, and other information
- save control limits in an output data set
- save subgroup sample sizes, subgroup means, and subgroup standard deviations in an output data set
- read preestablished control limits from a data set
- apply tests for special causes (also known as runs tests and Western Electric rules)
- specify a method for estimating the process standard deviation
- specify a known (standard) process mean and standard deviation for computing control limits
- display distinct sets of control limits for data from successive time phases
- add block legends and symbol markers to reveal stratification in process data
- superimpose stars at points to represent related multivariate factors
- clip extreme points to make the charts more readable
- display vertical and horizontal reference lines
- control axis values and labels
- control layout and appearance of the chart

You have three alternatives for producing $\bar{X}$ and s charts with the XSCHART statement:

- ODS Graphics output is produced if ODS Graphics is enabled, for example by specifying the ODS GRAPHICS ON statement prior to the PROC statement.
- Otherwise, traditional graphics are produced by default if SAS/GRAPH® is licensed.
- Legacy line printer charts are produced when you specify the LINEPRINTER option in the PROC statement.

See Chapter 3, "SAS/QC Graphics," for more information about producing these different kinds of graphs.

Getting Started: XSCHART Statement

This section introduces the XSCHART statement with simple examples that illustrate commonly used options. Complete syntax for the XSCHART statement is presented in the section "Syntax: XSCHART Statement" on page 1897, and advanced examples are given in the section "Examples: XSCHART Statement" on page 1921.

Creating Charts for Means and Standard Deviations from Raw Data

NOTE: See *Mean and Standard Deviation Charts Examples* in the SAS/QC Sample Library.

A petroleum company uses a turbine to heat water into steam, which is then pumped into the ground to make oil less viscous and easier to extract. This process occurs 20 times daily, and the amount of power (in kilowatts) used to heat the water to the desired temperature is recorded. The following statements create a SAS data set named Turbine, which contains the power output measurements for 20 days:

```
data Turbine;
   informat Day date7.;
   format Day date5.;
   input Day @;
   do i=1 to 10;
      input KWatts @;
      output;
   end;
   drop i;
   datalines;
04JUL94 3196 3507 4050 3215 3583 3617 3789 3180 3505 3454
04JUL94 3417 3199 3613 3384 3475 3316 3556 3607 3364 3721
05JUL94 3390 3562 3413 3193 3635 3179 3348 3199 3413 3562
05JUL94 3428 3320 3745 3426 3849 3256 3841 3575 3752 3347
06JUL94 3478 3465 3445 3383 3684 3304 3398 3578 3348 3369
06JUL94 3670 3614 3307 3595 3448 3304 3385 3499 3781 3711
07JUL94 3448 3045 3446 3620 3466 3533 3590 3070 3499 3457

   ... more lines ...

23JUL94 3756 3145 3571 3331 3725 3605 3547 3421 3257 3574
;
```

A partial listing of Turbine is shown in Figure 17.114.

Figure 17.114 Partial Listing of the Data Set Turbine

Kilowatt Power Output Data

Obs	Day	KWatts
1	04JUL	3196
2	04JUL	3507
3	04JUL	4050
4	04JUL	3215
5	04JUL	3583
6	04JUL	3617
7	04JUL	3789
8	04JUL	3180
9	04JUL	3505
10	04JUL	3454
11	04JUL	3417
12	04JUL	3199
13	04JUL	3613
14	04JUL	3384
15	04JUL	3475
16	04JUL	3316
17	04JUL	3556
18	04JUL	3607
19	04JUL	3364
20	04JUL	3721
21	05JUL	3390
22	05JUL	3562
23	05JUL	3413
24	05JUL	3193
25	05JUL	3635

The data set is said to be in "strung-out" form since each observation contains the day and power output for a single heating. The first 20 observations contain the power outputs for the first day, the second 20 observations contain the power outputs for the second day, and so on. Because the variable Day classifies the observations into rational subgroups, it is referred to as the *subgroup-variable*. The variable KWatts contains the power output measurements and is referred to as the *process variable* (or *process* for short).

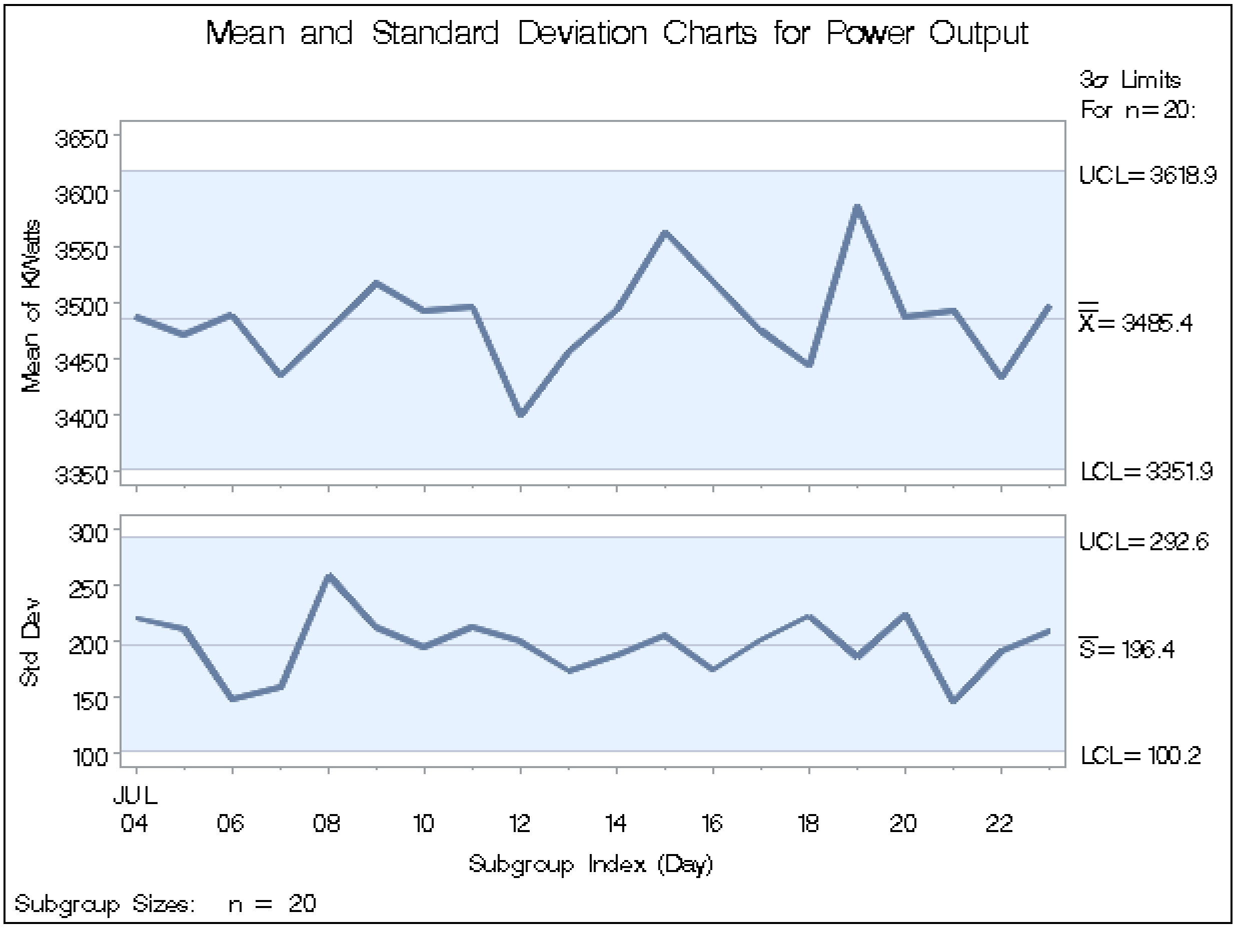
You can use $\bar{X}$ and s charts to determine whether the heating process is in control. The following statements create the $\bar{X}$ and s charts shown in Figure 17.115:

```
ods graphics off;
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart data=Turbine;
   xschart KWatts*Day ;
run;
```

This example illustrates the basic form of the XSCHART statement. After the keyword XSCHART, you specify the *process* to analyze (in this case KWatts), followed by an asterisk and the *subgroup-variable* (Day).

The input data set is specified with the DATA= option in the PROC SHEWHART statement.

Figure 17.115 $\bar{X}$ and s Charts for Power Output Data (Traditional Graphics)



Each point on the $\bar{X}$ chart represents the mean of the measurements for a particular day. For instance, the mean plotted for the first day is $(3196 + 3507 + \cdots + 3721)/20 = 3487.4$.

Each point on the s chart represents the standard deviation of the measurements for a particular day. For instance, the standard deviation plotted for the first day is

$$\sqrt{\frac{(3196 - 3487.4)^2 + (3507 - 3487.4)^2 + \cdots + (3721 - 3487.4)^2}{19}} = 220.26$$

Since all the points lie within the control limits, it can be concluded that the process is in statistical control.

By default, the control limits shown are 3σ limits estimated from the data; the formulas for the limits are given in Table 17.77. You can also read control limits from an input data set; see "Reading Preestablished Control Limits" on page 1896.

For computational details, see "Constructing Charts for Means and Standard Deviations" on page 1909. For more details on reading raw data, see "DATA= Data Set" on page 1915.

Creating Charts for Means and Standard Deviations from Summary Data

NOTE: See *Mean and Standard Deviation Charts Examples* in the SAS/QC Sample Library.

The previous example illustrates how you can create $\bar{X}$ and s charts using raw data (process measurements). However, in many applications the data are provided as subgroup summary statistics. This example illustrates how you can use the XSCHART statement with data of this type.

The following data set (Oilsum) provides the data from the preceding example in summarized form:

```
data Oilsum;
   input Day KWattsX KWattsS KWattsN;
   informat Day date7. ;
   format Day date5. ;
   label Day='Date of Measurement';
   datalines;
04JUL94 3487.40 220.260 20
05JUL94 3471.65 210.427 20
06JUL94 3488.30 147.025 20
07JUL94 3434.20 157.637 20
08JUL94 3475.80 258.949 20
09JUL94 3518.10 211.566 20
10JUL94 3492.65 193.779 20
11JUL94 3496.40 212.024 20
12JUL94 3398.50 199.201 20
13JUL94 3456.05 173.455 20
14JUL94 3493.60 187.465 20
15JUL94 3563.30 205.472 20
16JUL94 3519.05 173.676 20
17JUL94 3474.20 200.576 20
18JUL94 3443.60 222.084 20
19JUL94 3586.35 185.724 20
20JUL94 3486.45 223.474 20
21JUL94 3492.90 145.267 20
22JUL94 3432.80 190.994 20
23JUL94 3496.90 208.858 20
;
```

A partial listing of Oilsum is shown in Figure 17.116.

Figure 17.116 The Summary Data Set Oilsum

Summary Data Set for Power Output

Day	KWattsX	KWattsS	KWattsN
04JUL	3487.40	220.260	20
05JUL	3471.65	210.427	20
06JUL	3488.30	147.025	20
07JUL	3434.20	157.637	20
08JUL	3475.80	258.949	20

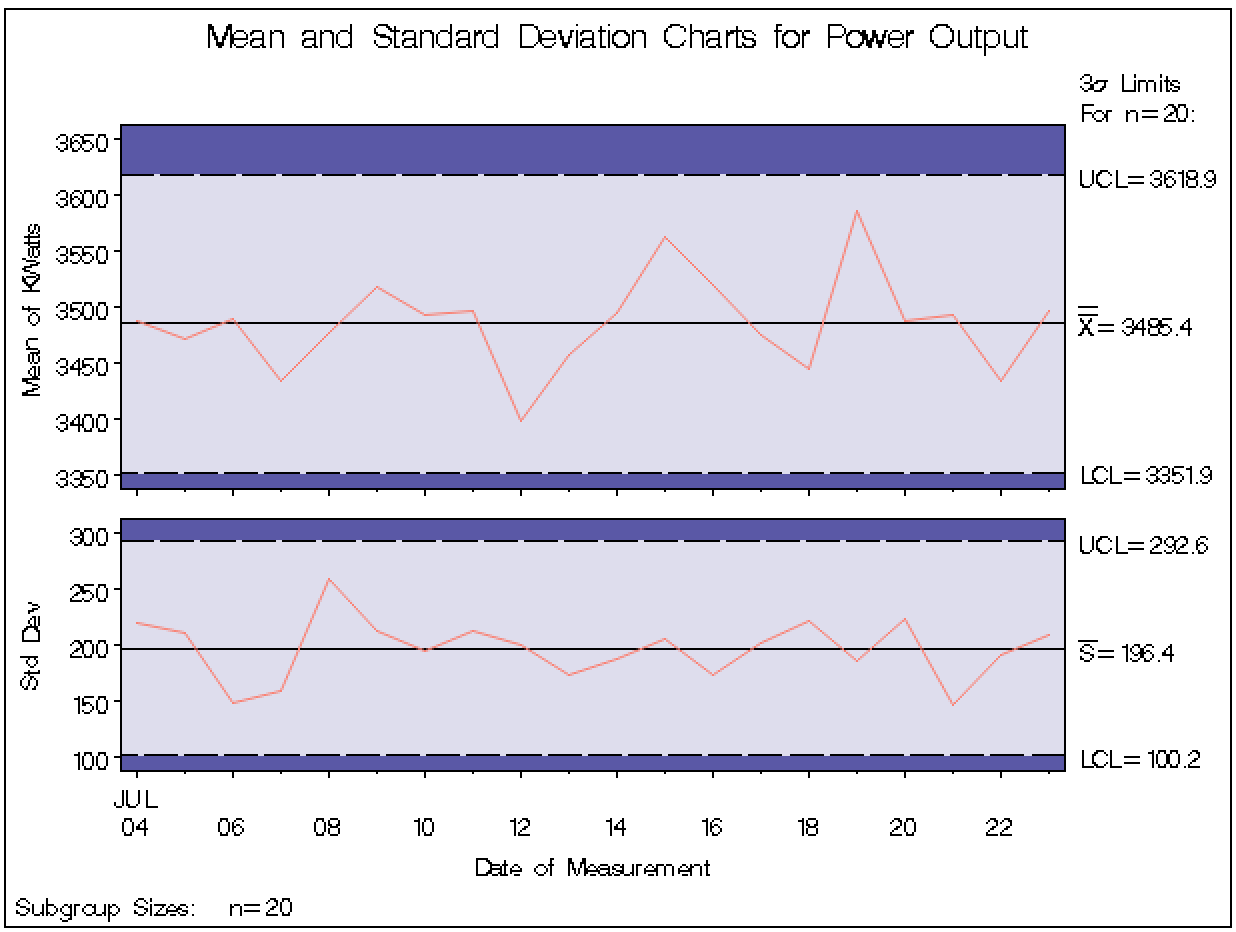
There is exactly one observation for each subgroup (note that the subgroups are still indexed by Day). The variable KWattsX contains the subgroup means, the variable KWattsS contains the subgroup standard deviations, and the variable KWattsN contains the subgroup sample sizes (which are all 20). You can read this data set by specifying it as a HISTORY= data set in the PROC SHEWHART statement, as follows:

```
options nogstyle;
goptions ftext=swiss;
symbol color = salmon h = .8;
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart history=Oilsum;
   xschart KWatts*Day / cframe   = lib
                        cinfill  = bwh
                        cconnect = salmon;
run;
options gstyle;
```

The NOGSTYLE system option causes ODS styles not to affect traditional graphics. Instead, the SYMBOL statement and XSCHART statement options control the appearance of the graph. The GSTYLE system option restores the use of ODS styles for traditional graphics produced subsequently. The resulting $\bar{X}$ and s charts are shown in Figure 17.117.

Note that KWatts is *not* the name of a SAS variable in the data set Oilsum but is, instead, the common prefix for the names of the three SAS variables KWattsX, KWattsS, and KWattsN. The suffix characters X, S, and N indicate *mean*, *standard deviation*, and *sample size*, respectively. Thus, you can specify three subgroup summary variables in the HISTORY= data set with a single name (KWatts), which is referred to as the *process*. The name Day specified after the asterisk is the name of the *subgroup-variable*.

Figure 17.117 $\bar{X}$ and s Charts for Power Output Data (Traditional Graphics with NOGSTYLE)



In general, a HISTORY= input data set used with the XSCHART statement must contain the following variables:

- subgroup variable
- subgroup mean variable
- subgroup standard deviation variable
- subgroup sample size variable

Furthermore, the names of the subgroup mean, standard deviation, and sample size variables must begin with the *process* name specified in the XSCHART statement and end with the special suffix characters *X*, *S*, and *N*, respectively. If the names do not follow this convention, you can use the RENAME option to rename the variables for the duration of the SHEWHART procedure step. For an illustration, see Example 17.42.

In summary, the interpretation of *process* depends on the input data set:

- If raw data are read using the DATA= option (as in the previous example), *process* is the name of the SAS variable containing the process measurements.
- If summary data are read using the HISTORY= option (as in this example), *process* is the common prefix for the names of the variables containing the summary statistics.

For more information, see “HISTORY= Data Set” on page 1916.

Saving Summary Statistics

NOTE: See *Mean and Standard Deviation Charts Examples* in the SAS/QC Sample Library.

In this example, the XSCHART statement is used to create a summary data set that can be read later by the SHEWHART procedure (as in the preceding example). The following statements read measurements from the data set Turbine (see “Creating Charts for Means and Standard Deviations from Raw Data” on page 1887) and create a summary data set named Turbhist:

```
proc shewhart data=Turbine;
   xschart KWatts*Day / outhistory = Turbhist
                        nochart;
run;
```

The OUTHISTORY= option names the output data set, and the NOCHART option suppresses the display of the charts, which would be identical to those in Figure 17.115. Options such as OUTHISTORY= and NOCHART are specified after the slash (/) in the XSCHART statement. A complete list of options is presented in the section “Syntax: XSCHART Statement” on page 1897.

Figure 17.118 contains a partial listing of Turbhist.

Figure 17.118 The Summary Data Set Turbhist

Summary Data Set for Power Output

Day	KWattsX	KWattsS	KWattsN
04JUL	3487.40	220.260	20
05JUL	3471.65	210.427	20
06JUL	3488.30	147.025	20
07JUL	3434.20	157.637	20
08JUL	3475.80	258.949	20

There are four variables in the data set Turbhist.

- Day contains the subgroup index.
- KWattsX contains the subgroup means.
- KWattsS contains the subgroup standard deviations.
- KWattsN contains the subgroup sample sizes.

Note that the summary statistic variables are named by adding the suffix characters *X*, *S*, and *N* to the *process* KWatts specified in the XSCHART statement. In other words, the variable naming convention for OUTHISTORY= data sets is the same as that for HISTORY= data sets.

For more information, see "OUTHISTORY= Data Set" on page 1912.

Saving Control Limits

NOTE: See *Mean and Standard Deviation Charts Examples* in the SAS/QC Sample Library.

You can save the control limits for $\bar{X}$ and s charts in a SAS data set; this enables you to apply the control limits to future data (see "Reading Preestablished Control Limits" on page 1896) or modify the limits with a DATA step program.

The following statements read measurements from the data set Turbine (see "Creating Charts for Means and Standard Deviations from Raw Data" on page 1887) and save the control limits displayed in Figure 17.115 in a data set named Turblim:

```
proc shewhart data=Turbine;
   xschart KWatts*Day / outlimits=Turblim
                        nochart;
run;
```

The OUTLIMITS= option names the data set containing the control limits, and the NOCHART option suppresses the display of the charts. The data set Turblim is listed in Figure 17.119.

Figure 17.119 The Data Set Turblim Containing Control Limit Information

Control Limits for Power Output Data

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
KWatts	Day	ESTIMATE	20	.002699796	3	3351.92	3485.41	3618.90

LCLS	_S_	_UCLS_	_STDDEV_
100.207	196.396	292.584	198.996

The data set Turblim contains one observation with the limits for *process* KWatts. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the $\bar{X}$ chart, and the variables _LCLS_ and _UCLS_ contain the lower and upper control limits for the s chart. The variable _MEAN_ contains the central line for the $\bar{X}$ chart, and the variable _S_ contains the central line for the s chart. The value of _MEAN_ is an estimate of the process mean, and the value of _STDDEV_ is an estimate of the process standard deviation σ. The value of _LIMITN_ is the nominal sample size associated with the control limits, and the value of _SIGMAS_ is the multiple of σ associated with the control limits. The variables _VAR_ and _SUBGRP_ are bookkeeping variables that save the *process* and *subgroup-variable*. The variable _TYPE_ is a bookkeeping variable that indicates whether the values of _MEAN_ and _STDDEV_ are estimates or standard values. For more information, see “OUTLIMITS= Data Set” on page 1911.

You can create an output data set containing both control limits and summary statistics with the OUTTABLE= option, as illustrated by the following statements:

```
proc shewhart data=Turbine;
   xschart KWatts*Day / outtable=Turbtab
                        nochart;
run;
```

The data set Turbtab is listed in Figure 17.120.

Figure 17.120 The OUTTABLE= Data Set Turbtab

Summary Statistics and Control Limit Information

VAR	Day	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_STDDEV_
KWatts	04JUL	3	20	20	3351.92	3487.40	3485.41	3618.90	198.996
KWatts	05JUL	3	20	20	3351.92	3471.65	3485.41	3618.90	198.996
KWatts	06JUL	3	20	20	3351.92	3488.30	3485.41	3618.90	198.996
KWatts	07JUL	3	20	20	3351.92	3434.20	3485.41	3618.90	198.996
KWatts	08JUL	3	20	20	3351.92	3475.80	3485.41	3618.90	198.996
KWatts	09JUL	3	20	20	3351.92	3518.10	3485.41	3618.90	198.996
KWatts	10JUL	3	20	20	3351.92	3492.65	3485.41	3618.90	198.996
KWatts	11JUL	3	20	20	3351.92	3496.40	3485.41	3618.90	198.996
KWatts	12JUL	3	20	20	3351.92	3398.50	3485.41	3618.90	198.996
KWatts	13JUL	3	20	20	3351.92	3456.05	3485.41	3618.90	198.996
KWatts	14JUL	3	20	20	3351.92	3493.60	3485.41	3618.90	198.996
KWatts	15JUL	3	20	20	3351.92	3563.30	3485.41	3618.90	198.996
KWatts	16JUL	3	20	20	3351.92	3519.05	3485.41	3618.90	198.996
KWatts	17JUL	3	20	20	3351.92	3474.20	3485.41	3618.90	198.996
KWatts	18JUL	3	20	20	3351.92	3443.60	3485.41	3618.90	198.996
KWatts	19JUL	3	20	20	3351.92	3586.35	3485.41	3618.90	198.996
KWatts	20JUL	3	20	20	3351.92	3486.45	3485.41	3618.90	198.996
KWatts	21JUL	3	20	20	3351.92	3492.90	3485.41	3618.90	198.996
KWatts	22JUL	3	20	20	3351.92	3432.80	3485.41	3618.90	198.996
KWatts	23JUL	3	20	20	3351.92	3496.90	3485.41	3618.90	198.996

EXLIM	_LCLS_	_SUBS_	_S_	_UCLS_	_EXLIMS_
	100.207	220.260	196.396	292.584	
	100.207	210.427	196.396	292.584	
	100.207	147.025	196.396	292.584	
	100.207	157.637	196.396	292.584	
	100.207	258.949	196.396	292.584	
	100.207	211.566	196.396	292.584	
	100.207	193.779	196.396	292.584	
	100.207	212.024	196.396	292.584	
	100.207	199.201	196.396	292.584	
	100.207	173.455	196.396	292.584	
	100.207	187.465	196.396	292.584	
	100.207	205.472	196.396	292.584	
	100.207	173.676	196.396	292.584	
	100.207	200.576	196.396	292.584	
	100.207	222.084	196.396	292.584	
	100.207	185.724	196.396	292.584	
	100.207	223.474	196.396	292.584	
	100.207	145.267	196.396	292.584	
	100.207	190.994	196.396	292.584	
	100.207	208.858	196.396	292.584	

The data set Turbtab contains one observation for each subgroup sample. The variables _SUBX_, _SUBS_, and _SUBN_ contain the subgroup means, subgroup standard deviations, and subgroup sample sizes. The variables _LCLX_ and _UCLX_ contain the lower and upper control limits for the $\bar{X}$ chart. The variables _LCLS_ and _UCLS_ contain the lower and upper control limits for the s chart. The variable _MEAN_ contains the central line for the $\bar{X}$ chart. The variable _S_ contains the central line for the s chart. The variables _VAR_ and Batch contain the *process* name and values of the *subgroup-variable*, respectively. For more information, see "OUTTABLE= Data Set" on page 1913.

A data set created with the OUTTABLE= option can be read later as a TABLE= data set. For example, the following statements read Turbtab and display charts (not shown here) identical to those in Figure 17.115:

```
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart table=Turbtab;
   xschart KWatts*Day;
run;
```

Because the SHEWHART procedure simply displays the information in a TABLE= data set, you can use TABLE= data sets to create specialized control charts (see "Specialized Control Charts: SHEWHART Procedure" on page 2096). For more information, see "TABLE= Data Set" on page 1917.

Reading Preestablished Control Limits

NOTE: See *Mean and Standard Deviation Charts Examples* in the SAS/QC Sample Library.

In the previous example, the OUTLIMITS= data set Turblim saved control limits computed from the measurements in Turbine. This example shows how these limits can be applied to new data. The following statements create $\bar{X}$ and s charts for new measurements in a data set named Turbine2 (not listed here) using the control limits in Turblim:

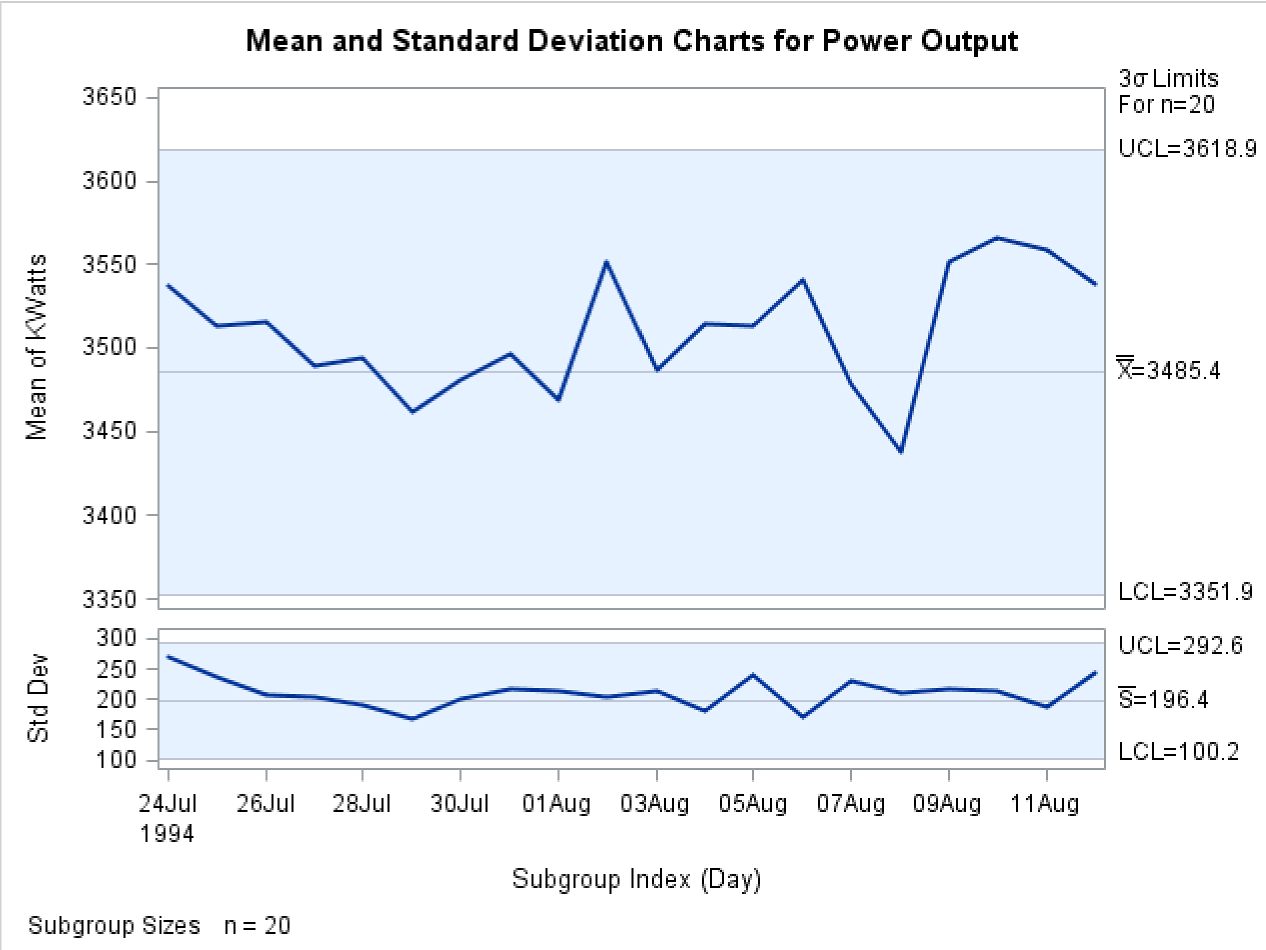
```
ods graphics on;
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart data=Turbine2 limits=Turblim;
   xschart KWatts*Day / odstitle = title;
run;
```

The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the $\bar{X}$ and s charts are created by using ODS Graphics instead of traditional graphics. The charts are shown in Figure 17.121.

The LIMITS= option in the PROC SHEWHART statement specifies the data set containing preestablished control limit information. By default, this information is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches the *process* name KWatts
- the value of _SUBGRP_ matches the *subgroup-variable* name Day

Figure 17.121 $\bar{X}$ and s Charts for Second Set of Power Outputs (ODS Graphics)



The means and standard deviations lie within the control limits, indicating that the heating process is still in statistical control.

In this example, the LIMITS= data set was created in a previous run of the SHEWHART procedure. You can also create a LIMITS= data set with the DATA step. See "LIMITS= Data Set" on page 1915 for details concerning the variables that you must provide.

Syntax: XSCHART Statement

The basic syntax for the XSCHART statement is as follows:

XSCHART *process* ∗ *subgroup-variable* **;**

The general form of this syntax is as follows:

XSCHART *processes* ∗ *subgroup-variable* < **(***block-variables* | **=**'*character*' > / < *options* > **;**

You can use any number of XSCHART statements in the SHEWHART procedure. The components of the XSCHART statement are described as follows:

process

processes

identify one or more processes to be analyzed. The specification of *process* depends on the input data set specified in the PROC SHEWHART statement.

- If the raw data are read using a DATA= data set, *process* must be the name of the variable containing the raw measurements. For an example, see "Creating Charts for Means and Standard Deviations from Raw Data" on page 1887.
- If summary data are read from a HISTORY= data set, *process* must be the common prefix of the summary variables in the HISTORY= data set. For an example, see "Creating Charts for Means and Standard Deviations from Summary Data" on page 1890.
- If summary data and control limits are read from a TABLE= data set, *process* must be the value of the variable _VAR_ in the TABLE= data set. For an example, see "Saving Control Limits" on page 1893.

A *process* is required. If more than one *process* is specified, enclose the list in parentheses. For example, the following statements request distinct $\bar{X}$ and s charts for Weight, Length, and Width:

```
proc shewhart data=Measures;
   xschart (Weight Length Width)*Day;
run;
```

subgroup-variable

is the variable that identifies subgroups in the data. The *subgroup-variable* is required. In the preceding XSCHART statement, Day is the subgroup variable. For details, see "Subgroup Variables" on page 1871.

block-variables

are optional variables that group the data into blocks of consecutive subgroups. The blocks are labeled in a legend, and each *block-variable* provides one level of labels in the legend. See "Displaying Stratification in Blocks of Observations" on page 2026 for an example.

symbol-variable

is an optional variable whose levels (unique values) determine the symbol marker or character used to plot the means and standard deviations.

- If you produce a line printer chart, an 'A' is displayed for the points corresponding to the first level of the *symbol-variable*, a 'B' is displayed for the points corresponding to the second level, and so on.
- If you produce traditional graphics, distinct symbol markers are displayed for points corresponding to the various levels of the *symbol-variable*. You can specify the symbol markers with SYMBOL*n* statements. See "Displaying Stratification in Levels of a Classification Variable" on page 2025 for an example.

character

specifies a plotting character for line printer charts. For example, the following statements create $\bar{X}$ and s charts using an asterisk (*) to plot the points:

```
proc shewhart data=Values lineprinter;
   xschart Weight*Day='*';
run;
```

options

enhance the appearance of the charts, request additional analyses, save results in data sets, and so on. The "Summary of Options" on page 1899 section, which follows, lists all options by function. "Dictionary of Options: SHEWHART Procedure" on page 1946 describes each option in detail.

Summary of Options

The following tables list the XSCHART statement options by function. For complete descriptions, see "Dictionary of Options: SHEWHART Procedure" on page 1946.

Table 17.76 XSCHART Statement Options

Option	Description
Options for Specifying Control Limits	
ALPHA=	requests probability limits for chart
LIMITN=	specifies either nominal sample size for fixed control limits or varying limits
NOREADLIMITS	computes control limits for each *process* from the data rather than a LIMITS= data set (SAS 6.10 and later releases)
READALPHA	reads _ALPHA_ instead of _SIGMAS_ from a LIMITS= data set
READINDEX=	reads control limits for each *process* from a LIMITS= data set
READLIMITS	reads single set of control limits for each *process* from a LIMITS= data set (SAS 6.09 and earlier releases)
SIGMAS=	specifies width of control limits in terms of multiple k of standard error of plotted means
Options for Displaying Control Limits	
CINFILL=	specifies color for area inside control limits
CLIMITS=	specifies color of control limits, central line, and related labels
LCLLABEL=	specifies label for lower control limit on $\bar{X}$ chart
LCLLABEL2=	specifies label for lower control limit on s chart
LIMLABSUBCHAR=	specifies a substitution character for labels provided as quoted strings; the character is replaced with the value of the control limit
LLIMITS=	specifies line type for control limits

Table 17.76 *continued*

Option	Description
NDECIMAL=	specifies number of digits to right of decimal place in default labels for control limits and central line on $\bar{X}$ chart
NDECIMAL2=	specifies number of digits to right of decimal place in default labels for control limits and central line on s chart
NOCTL	suppresses display of central line on $\bar{X}$ chart
NOCTL2	suppresses display of central line on s chart
NOLCL	suppresses display of lower control limit on $\bar{X}$ chart
NOLCL2	suppresses display of lower control limit on s chart
NOLIMIT0	suppresses display of zero lower control limit on s chart
NOLIMITLABEL	suppresses labels for control limits and central line
NOLIMITS	suppresses display of control limits
NOLIMITSFRAME	suppresses default frame around control limit information when multiple sets of control limits are read from a LIMITS= data set
NOLIMITSLEGEND	suppresses legend for control limits
NOUCL	suppresses display of upper control limit on $\bar{X}$ chart
NOUCL2	suppresses display of upper control limit on s chart
SSYMBOL=	specifies label for central line on s chart
UCLLABEL=	specifies label for upper control limit on $\bar{X}$ chart
UCLLABEL2=	specifies label for upper control limit on s chart
WLIMITS=	specifies width for control limits and central line
XSYMBOL=	specifies label for central line on $\bar{X}$ chart
Process Mean and Standard Deviation Options	
MU0=	specifies known value of μ_0 for process mean μ
SIGMA0=	specifies known value σ_0 for process standard deviation σ
SMETHOD=	specifies method for estimating process standard deviation σ
TYPE=	identifies parameters as estimates or standard values and specifies value of _TYPE_ in the OUTLIMITS= data set
Options for Plotting and Labeling Points	
ALLLABEL=	labels every point on $\bar{X}$ chart
ALLLABEL2=	labels every point on s chart
CLABEL=	specifies color for labels
CCONNECT=	specifies color for line segments that connect points on chart
CFRAMELAB=	specifies fill color for frame around labeled points
CNEEDLES=	specifies color for needles that connect points to central line
COUT=	specifies color for portions of line segments that connect points outside control limits
COUTFILL=	specifies color for shading areas between the connected points and control limits outside the limits

Table 17.76 *continued*

Option	Description
LABELANGLE=	specifies angle at which labels are drawn
LABELFONT=	specifies software font for labels (alias for the TESTFONT= option)
LABELHEIGHT=	specifies height of labels (alias for the TESTHEIGHT= option)
NEEDLES	connects points to central line with vertical needles
NOCONNECT	suppresses line segments that connect points on chart
OUTLABEL=	labels points outside control limits on $\bar{X}$ chart
OUTLABEL2=	labels points outside control limits on s chart
SYMBOLLEGEND=	specifies LEGEND statement for levels of *symbol-variable*
SYMBOLORDER=	specifies order in which symbols are assigned for levels of *symbol-variable*
TURNALL|TURNOUT	turns point labels so that they are strung out vertically
WNEEDLES=	specifies width of needles
Options for Specifying Tests for Special Causes	
INDEPENDENTZONES	computes zone widths independently above and below center line
NO3SIGMACHECK	enables tests to be applied with control limits other than 3σ limits
NOTESTACROSS	suppresses tests across *phase* boundaries
TESTS=	specifies tests for special causes for the $\bar{X}$ chart
TESTS2=	specifies tests for special causes for the s chart
TEST2RESET=	enables tests for special causes to be reset for the s chart
TEST2RUN=	specifies length of pattern for Test 2
TEST3RUN=	specifies length of pattern for Test 3
TESTACROSS	applies tests across *phase* boundaries
TESTLABEL=	provides labels for points where test is positive
TESTLABEL*n*=	specifies label for *n*th test for special causes
TESTNMETHOD=	applies tests to standardized chart statistics
TESTOVERLAP	performs tests on overlapping patterns of points
TESTRESET=	enables tests for special causes to be reset
WESTGARD=	requests that Westgard rules be applied to the $\bar{X}$ chart
ZONELABELS	adds labels A, B, and C to zone lines for $\bar{X}$ chart
ZONE2LABELS	adds labels A, B, and C to zone lines for s chart
ZONES	adds lines to $\bar{X}$ chart delineating zones A, B, and C
ZONES2	adds lines to s chart delineating zones A, B, and C
ZONEVALPOS=	specifies position of ZONEVALUES labels
ZONEVALUES	labels $\bar{X}$ chart zone lines with their values
ZONE2VALUES	labels s zone lines with their values
Options for Displaying Tests for Special Causes	
CTESTLABBOX=	specifies color for boxes enclosing labels indicating points where test is positive

Table 17.76 *continued*

Option	Description
CTESTS=	specifies color for labels indicating points where test is positive
CTESTSYMBOL=	specifies color for symbol used to plot points where test is positive
CZONES=	specifies color for lines and labels delineating zones A, B, and C
LTESTS=	specifies type of line connecting points where test is positive
LZONES=	specifies line type for lines delineating zones A, B, and C
TESTFONT=	specifies software font for labels at points where test is positive
TESTHEIGHT=	specifies height of labels at points where test is positive
TESTLABBOX	requests that labels for points where test is positive be positioned so that do not overlap
TESTSYMBOL=	specifies plot symbol for points where test is positive
TESTSYMBOLHT=	specifies symbol height for points where test is positive
WTESTS=	specifies width of line connecting points where test is positive
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill colors for frame for plot area
CTEXT=	specifies color for tick mark values and axis labels
DISCRETE	produces horizontal axis for discrete numeric group values
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric subgroup variable
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on horizontal axis
NOTRUNC	suppresses vertical axis truncation at zero applied by default to s chart
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
SPLIT=	specifies splitting character for axis labels
TURNHLABELS	requests horizontal axis labels that are strung out vertically

Table 17.76 *continued*

Option	Description
VAXIS=	specifies major tick mark values for vertical axis of $\bar{X}$ chart
VAXIS2=	specifies major tick mark values for vertical axis of s chart
VFORMAT=	specifies format for primary vertical axis tick mark labels
VFORMAT2=	specifies format for secondary vertical axis tick mark labels
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis for primary chart
VZERO2	forces origin to be included in vertical axis for secondary chart
WAXIS=	specifies width of axis lines
Plot Layout Options	
ALLN	plots means for all subgroups
BILEVEL	creates control charts using half-screens and half-pages
EXCHART	creates control charts for a process only when exceptions occur
INTERVAL=	natural time interval between consecutive subgroup positions when time, date, or datetime format is associated with a numeric subgroup variable
MAXPANELS=	maximum number of pages or screens for chart
NMARKERS	requests special markers for points corresponding to sample sizes not equal to nominal sample size for fixed control limits
NOCHART	suppresses creation of charts
NOCHART2	suppresses creation of s chart
NOFRAME	suppresses frame for plot area
NOLEGEND	suppresses legend for subgroup sample sizes
NPANELPOS=	specifies number of subgroup positions per panel on each chart
REPEAT	repeats last subgroup position on panel as first subgroup position of next panel
SEPARATE	displays $\bar{X}$ and s charts on separate screens or pages
TOTPANELS=	specifies number of pages or screens to be used to display chart
YPCT1=	specifies length of vertical axis on $\bar{X}$ chart as a percentage of sum of lengths of vertical axes for $\bar{X}$ and s charts
ZEROSTD	displays $\bar{X}$ chart regardless of whether $\hat{\sigma} = 0$
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= and HREF2= options

Table 17.76 *continued*

Option	Description
CVREF=	specifies color for lines requested by VREF= and VREF2= options
HREF=	specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2=	specifies position of reference lines perpendicular to horizontal axis on s chart
HREFDATA=	specifies position of reference lines perpendicular to horizontal axis on $\bar{X}$ chart
HREF2DATA=	specifies position of reference lines perpendicular to horizontal axis on s chart
HREFLABELS=	specifies labels for HREF= lines
HREF2LABELS=	specifies labels for HREF2= lines
HREFLABPOS=	specifies position of HREFLABELS= and HREF2LABELS= labels
LHREF=	specifies line type for HREF= and HREF2= lines
LVREF=	specifies line type for VREF= and VREF2= lines
NOBYREF	specifies that reference line information in a data set applies uniformly to charts created for all BY groups
VREF=	specifies position of reference lines perpendicular to vertical axis on $\bar{X}$ chart
VREF2=	specifies position of reference lines perpendicular to vertical axis on s chart
VREFLABELS=	specifies labels for VREF= lines
VREF2LABELS=	specifies labels for VREF2= lines
VREFLABPOS=	position of VREFLABELS= and VREF2LABELS= labels
Grid Options	
CGRID=	specifies color for grid requested with GRID or ENDGRID option
ENDGRID	adds grid after last plotted point
GRID	adds grid to control chart
LENDGRID=	specifies line type for grid requested with the ENDGRID option
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme points are clipped
CLIPLEGEND=	specifies text for clipping legend
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points

Table 17.76 *continued*

Option	Description
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features to $\bar{X}$ chart
ANNOTATE2=	specifies annotate data set that adds features to s chart
DESCRIPTION=	specifies description of $\bar{X}$ chart's GRSEG catalog entry
DESCRIPTION2=	specifies description of s chart's GRSEG catalog entry
FONT=	specifies software font for labels and legends on charts
NAME=	specifies name of $\bar{X}$ chart's GRSEG catalog entry
NAME2=	specifies name of s chart's GRSEG catalog entry
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option
Options for Producing Graphs Using ODS Styles	
BLOCKVAR=	specifies one or more variables whose values define colors for filling background of *block-variable* legend
CFRAMELAB	draws a frame around labeled points
COUT	draw portions of line segments that connect points outside control limits in a contrasting color
CSTAROUT	specifies that portions of stars exceeding inner or outer circles are drawn using a different color
OUTFILL	shades areas between control limits and connected points lying outside the limits
STARFILL=	specifies a variable identfying groups of stars filled with different colors
STARS=	specifies a variable identfying groups of stars whose outlines are drawn with different colors
Options for ODS Graphics	
BLOCKREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
INFILLTRANSPARENCY=	specifies the control limit infill transparency
MARKERS	plots subgroup points with markers
NOBLOCKREF	suppresses block and phase reference lines
NOBLOCKREFFILL	suppresses block and phase wall fills
NOFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOPHASEREF	suppresses block and phase reference lines
NOPHASEREFFILL	suppresses block and phase wall fills
NOREF	suppresses block and phase reference lines
NOREFFILL	suppresses block and phase wall fills
NOSTARFILLLEGEND	suppresses legend for levels of a STARFILL= variable
NOTRANSPARENCY	disables transparency in ODS Graphics output
ODSFOOTNOTE=	specifies a graph footnote
ODSFOOTNOTE2=	specifies a secondary graph footnote
ODSLEGENDEXPAND	specifies that legend entries contain all levels observed in the data

Table 17.76 *continued*

Option	Description
ODSTITLE=	specifies a graph title
ODSTITLE2=	specifies a secondary graph title
OUTFILLTRANSPARENCY=	specifies control limit outfill transparency
OVERLAYURL=	specifies URLs to associate with overlay points
OVERLAY2URL=	specifies URLs to associate with overlay points on secondary chart
PHASEPOS=	specifies vertical position of phase legend
PHASEREFLEVEL=	associates phase and block reference lines with either innermost or the outermost level
PHASEREFTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
REFFILLTRANSPARENCY=	specifies the wall fill transparency for blocks and phases
SIMULATEQCFONT	draws central line labels using a simulated software font
STARTRANSPARENCY=	specifies star fill transparency
URL=	specifies a variable whose values are URLs to be associated with subgroups
URL2=	specifies a variable whose values are URLs to be associated with subgroups on secondary chart
Input Data Set Options	
MISSBREAK	specifies that observations with missing values are not to be processed
Output Data Set Options	
OUTHISTORY=	creates output data set containing subgroup summary statistics
OUTINDEX=	specifies value of _INDEX_ in the OUTLIMITS= data set
OUTLIMITS=	creates output data set containing control limits
OUTTABLE=	creates output data set containing subgroup summary statistics and control limits
Tabulation Options	
NOTE: specifying (EXCEPTIONS) after a tabulation option creates a table for exceptional points only.	
TABLE	creates a basic table of subgroup means, subgroup sample sizes, and control limits
TABLEALL	is equivalent to the options TABLE, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS
TABLECENTRAL	augments basic table with values of central lines
TABLEID	augments basic table with columns for ID variables
TABLELEGEND	augments basic table with legend for tests for special causes
TABLEOUTLIM	augments basic table with columns indicating control limits exceeded
TABLETESTS	augments basic table with a column indicating which tests for special causes are positive

Table 17.76 *continued*

Option	Description
Specification Limit Options	
CIINDICES	specifies α value and type for computing capability index confidence limits
LSL=	specifies list of lower specification limits
TARGET=	specifies list of target values
USL=	specifies list of upper specification limits
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for *block-variable* legend
BLOCKLABTYPE=	specifies text size of *block-variable* legend
BLOCKPOS=	specifies vertical position of *block-variable* legend
BLOCKREP	repeats identical consecutive labels in *block-variable* legend
CBLOCKLAB=	specifies fill colors for frames enclosing variable labels in *block-variable* legend
CBLOCKVAR=	specifies one or more variables whose values are colors for filling background of *block-variable* legend
Phase Options	
CPHASELEG=	specifies text color for *phase* legend
NOPHASEFRAME	suppresses default frame for *phase* legend
OUTPHASE=	specifies value of _PHASE_ in the OUTHISTORY= data set
PHASEBREAK	disconnects last point in a *phase* from first point in next *phase*
PHASELABTYPE=	specifies text size of *phase* legend
PHASELEGEND	displays *phase* labels in a legend across top of chart
PHASELIMITS	labels control limits for each phase, provided they are constant within that phase
PHASEREF	delineates *phases* with vertical reference lines
READPHASES=	specifies *phases* to be read from an input data set
Star Options	
CSTARCIRCLES=	specifies color for STARCIRCLES= circles
CSTARFILL=	specifies color for filling stars
CSTAROUT=	specifies outline color for stars exceeding inner or outer circles
CSTARS=	specifies color for outlines of stars
LSTARCIRCLES=	specifies line types for STARCIRCLES= circles
LSTARS=	specifies line types for outlines of STARVERTICES= stars
STARBDRADIUS=	specifies radius of outer bound circle for vertices of stars
STARCIRCLES=	specifies reference circles for stars
STARINRADIUS=	specifies inner radius of stars
STARLABEL=	specifies vertices to be labeled
STARLEGEND=	specifies style of legend for star vertices
STARLEGENDLAB=	specifies label for STARLEGEND= legend

Table 17.76 *continued*

Option	Description
STAROUTRADIUS=	specifies outer radius of stars
STARSPECS=	specifies method used to standardize vertex variables
STARSTART=	specifies angle for first vertex
STARTYPE=	specifies graphical style of star
STARVERTICES=	superimposes star at each point on $\bar{X}$ chart
WSTARCIRCLES=	specifies width of STARCIRCLES= circles
WSTARS=	specifies width of STARVERTICES= stars
Overlay Options	
CCOVERLAY=	specifies colors for primary chart overlay line segments
CCOVERLAY2=	specifies colors for secondary chart overlay line segments
COVERLAY=	specifies colors for primary chart overlay plots
COVERLAY2=	specifies colors for secondary chart overlay plots
COVERLAYCLIP=	specifies color for clipped points on overlays
LOVERLAY=	specifies line types for primary chart overlay line segments
LOVERLAY2=	specifies line types for secondary chart overlay line segments
NOOVERLAYLEGEND	suppresses legend for overlay plots
OVERLAY=	specifies variables to overlay on primary chart
OVERLAY2=	specifies variables to overlay on secondary chart
OVERLAY2HTML=	specifies links to associate with secondary chart overlay points
OVERLAY2ID=	specifies labels for secondary chart overlay points
OVERLAY2SYM=	specifies symbols for secondary chart overlays
OVERLAY2SYMHT=	specifies symbol heights for secondary chart overlays
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
OVERLAYHTML=	specifies links to associate with primary chart overlay points
OVERLAYID=	specifies labels for primary chart overlay points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols for primary chart overlays
OVERLAYSYMHT=	specifies symbol heights for primary chart overlays
WOVERLAY=	specifies widths of primary chart overlay line segments
WOVERLAY2=	specifies widths of secondary chart overlay line segments
Options for Interactive Control Charts	
HTML=	specifies a variable whose values create links to be associated with subgroups
HTML2=	specifies variable whose values create links to be associated with subgroups on secondary chart
HTML_LEGEND=	specifies a variable whose values create links to be associated with symbols in the symbol legend
WEBOUT=	creates an OUTTABLE= data set with additional graphics coordinate data

Table 17.76 *continued*

Option	Description
Options for Line Printer Charts	
CLIPCHAR=	specifies plot character for clipped points
CONNECTCHAR=	specifies character used to form line segments that connect points on chart
HREFCHAR=	specifies line character for HREF= and HREF2= lines
SYMBOLCHARS=	specifies characters indicating *symbol-variable*
TESTCHAR=	specifies character for line segments that connect any sequence of points for which a test for special causes is positive
VREFCHAR=	specifies line character for VREF= and VREF2= lines
ZONECHAR=	specifies character for lines that delineate zones for tests for special causes

Details: XSCHART Statement

Constructing Charts for Means and Standard Deviations

The following notation is used in this section:

μ	process mean (expected value of the population of measurements)
σ	process standard deviation (standard deviation of the population of measurements)
$\bar{X}_i$	mean of measurements in *i*th subgroup
s_i	standard deviation of the measurements $x_{i1}, \ldots, x_{in_i}$ in the *i*th subgroup

$$s_i = \sqrt{((x_{i1} - \bar{X}_i)^2 + \cdots + (x_{in_i} - \bar{X}_i)^2)/(n_i - 1)}$$

n_i	sample size of *i*th subgroup
N	number of subgroups
$\bar{\bar{X}}$	weighted average of subgroup means
z_p	100*p*th percentile of the standard normal distribution
$c_4(n)$	expected value of the standard deviation of *n* independent normally distributed variables with unit standard deviation
$c_5(n)$	standard error of the standard deviation of *n* independent observations from a normal population with unit standard deviation
$\chi^2_p(n)$	100*p*th percentile ($0 < p < 1$) of the χ^2 distribution with *n* degrees of freedom

Plotted Points

Each point on an $\bar{X}$ chart indicates the value of a subgroup mean ($\bar{X}_i$). For example, if the tenth subgroup contains the values 12, 15, 19, 16, and 13, the mean plotted for this subgroup is

$$\bar{X}_{10} = \frac{12 + 15 + 19 + 16 + 13}{5} = 15$$

Each point on an s chart indicates the value of a subgroup standard deviation (s_i). For example, the standard deviation plotted for the tenth subgroup is

$$s_{10} = \sqrt{((12-15)^2 + (15-15)^2 + (19-15)^2 + (16-15)^2 + (13-15)^2)/4} = 2.739$$

Central Lines

On an $\bar{X}$ chart, by default, the central line indicates an estimate of μ, which is computed as

$$\hat{\mu} = \overline{\overline{X}} = \frac{n_1 \bar{X}_1 + \cdots + n_N \bar{X}_N}{n_1 + \cdots + n_N}$$

If you specify a known value (μ_0) for μ, the central line indicates the value of μ_0.

On the s chart, by default, the central line for the ith subgroup indicates an estimate for the expected value of s_i, which is computed as $c_4(n_i)\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of σ. If you specify a known value (σ_0) for σ, the central line indicates the value of $c_4(n_i)\sigma_0$. Note that the central line varies with n_i.

Control Limits

You can compute the limits in the following ways:

- as a specified multiple (k) of the standard errors of $\bar{X}_i$ and s_i above and below the central line. The default limits are computed with $k = 3$ (these are referred to as 3σ *limits*).
- as probability limits defined in terms of α, a specified probability that $\bar{X}_i$ or s_i exceeds the limits

The following table provides the formulas for the limits:

Table 17.77 Limits for $\bar{X}$ and s Charts

Control Limits	
$\bar{X}$ Chart	LCL = lower limit = $\overline{\overline{X}} - k\hat{\sigma}/\sqrt{n_i}$
	UCL = upper limit = $\overline{\overline{X}} + k\hat{\sigma}/\sqrt{n_i}$
s Chart	LCL = lower limit = $\max(c_4(n_i)\hat{\sigma} - kc_5(n_i)\hat{\sigma}, 0)$
	UCL = upper limit = $c_4(n_i)\hat{\sigma} + kc_5(n_i)\hat{\sigma}$
Probability Limits	
$\bar{X}$ Chart	LCL = lower limit = $\overline{\overline{X}} - z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$
	UCL = upper limit = $\overline{\overline{X}} + z_{\alpha/2}(\hat{\sigma}/\sqrt{n_i})$
s Chart	LCL = lower limit = $\hat{\sigma}\sqrt{\chi^2_{\alpha/2}(n_i-1)/(n_i-1)}$
	UCL = upper limit = $\hat{\sigma}\sqrt{\chi^2_{1-\alpha/2}(n_i-1)/(n_i-1)}$

The formulas for s charts assume that the data are normally distributed. If standard values μ_0 and σ_0 are available for μ and σ, respectively, replace $\overline{\overline{X}}$ with μ_0 and $\hat{\sigma}$ with σ_0 in Table 17.77. Note that the limits vary with n_i and that the probability limits for s_i are asymmetric about the central line.

You can specify parameters for the limits as follows:

- Specify k with the SIGMAS= option or with the variable _SIGMAS_ in a LIMITS= data set.
- Specify α with the ALPHA= option or with the variable _ALPHA_ in a LIMITS= data set.
- Specify a constant nominal sample size $n_i \equiv n$ for the control limits with the LIMITN= option or with the variable _LIMITN_ in a LIMITS= data set.
- Specify μ_0 with the MU0= option or with the variable _MEAN_ in a LIMITS= data set.
- Specify σ_0 with the SIGMA0= option or with the variable _STDDEV_ in a LIMITS= data set.

Output Data Sets

OUTLIMITS= Data Set

The OUTLIMITS= data set saves control limits and control limit parameters. The following variables are saved:

Table 17.78 OUTLIMITS= Data Set

Variable	Description
ALPHA	probability (α) of exceeding limits
_CP_	capability index C_p
CPK	capability index C_{pk}
CPL	capability index CPL
CPM	capability index C_{pm}
CPU	capability index CPU
INDEX	optional identifier for the control limits specified with the OUTINDEX= option
LCLS	lower control limit for subgroup standard deviation
LCLX	lower control limit for subgroup mean
LIMITN	nominal sample size associated with the control limits
LSL	lower specification limit
MEAN	process mean ($\overline{\overline{X}}$ or μ_0)
_S_	value of central line on s chart
SIGMAS	multiple (k) of standard error of $\bar{X}_i$ or s_i
STDDEV	process standard deviation ($\hat{\sigma}$ or σ_0)
SUBGRP	*subgroup-variable* specified in the XSCHART statement
TARGET	target value
TYPE	type (estimate or standard value) of _MEAN_ and _STDDEV_
UCLS	upper control limit for subgroup standard deviation
UCLX	upper control limit for subgroup mean
USL	upper specification limit
VAR	*process* specified in the XSCHART statement

Notes:

1. If the control limits vary with subgroup sample size, the special missing value 'V' is assigned to the variables _LIMITN_, _LCLX_, _UCLX_, _LCLS_, _S_, and _UCLS_.
2. If the limits are defined in terms of a multiple k of the standard errors of $\bar{X}_i$ and s_i, the value of _ALPHA_ is computed as $\alpha = 2(1 - \Phi(k))$, where $\Phi(\cdot)$ is the standard normal distribution function.
3. If the limits are probability limits, the value of _SIGMAS_ is computed as $k = \Phi^{-1}(1 - \alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function.
4. The variables _CP_, _CPK_, _CPL_, _CPU_, _LSL_, and _USL_ are included only if you provide specification limits with the LSL= and USL= options. The variables _CPM_ and _TARGET_ are included if, in addition, you provide a target value with the TARGET= option. See "Capability Indices" on page 1874 for computational details.
5. Optional BY variables are saved in the OUTLIMITS= data set.

The OUTLIMITS= data set contains one observation for each *process* specified in the XSCHART statement. For an example, see "Saving Control Limits" on page 1893.

OUTHISTORY= Data Set

The OUTHISTORY= data set saves subgroup summary statistics. The following variables are saved:

- the *subgroup-variable*
- a subgroup mean variable named by *process* suffixed with *X*
- a subgroup standard deviation variable named by *process* suffixed with *S*
- a subgroup sample size variable named by *process* suffixed with *N*

Given a *process* name that contains 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Subgroup summary variables are created for each *process* specified in the XSCHART statement. For example, consider the following statements:

```
proc shewhart data=Steel;
   xschart (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthX, WidthS, WidthN, DiameterX, DiameterS, and DiameterN. Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*

- ID variables
- _PHASE_ (if the OUTPHASE= option is specified)

For an example of an OUTHISTORY= data set, see "Saving Summary Statistics" on page 1892.

OUTTABLE= Data Set

The OUTTABLE= data set saves subgroup summary statistics, control limits, and related information. The following variables are saved:

Variable	Description
ALPHA	probability (α) of exceeding control limits
EXLIM	control limit exceeded on $\bar{X}$ chart
EXLIMS	control limit exceeded on s chart
LCLS	lower control limit for standard deviation
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
_S_	average standard deviation
SIGMAS	multiple (k) of the standard error associated with control limits
subgroup	values of the subgroup variable
SUBN	subgroup sample size
SUBS	subgroup standard deviation
SUBX	subgroup mean
TESTS	tests for special causes signaled on $\bar{X}$ chart
TESTS2	tests for special causes signaled on s chart
UCLS	upper control limit for standard deviation
UCLX	upper control limit for mean
VAR	*process* specified in the XSCHART statement

In addition, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables
- _PHASE_ (if the READPHASES= option is specified)

Notes:

1. Either the variable _ALPHA_ or the variable _SIGMAS_ is saved depending on how the control limits are defined (with the ALPHA= or SIGMAS= options, respectively, or with the corresponding variables in a LIMITS= data set).

2. The variable _TESTS_ is saved if you specify the TESTS= option. The *k*th character of a value of _TESTS_ is *k* if Test *k* is positive at that subgroup. For example, if you request all eight tests and Tests 2 and 8 are positive for a given subgroup, the value of _TESTS_ has a 2 for the second character, an 8 for the eighth character, and blanks for the other six characters.

3. The variable _TESTS2_ is saved if you specify the TESTS2= option.

4. The variables _EXLIM_, _EXLIMS_, _TESTS_, and _TESTS2_ are character variables of length 8. The variable _PHASE_ is a character variable of length 48. The variable _VAR_ is a character variable whose length is no greater than 32. All other variables are numeric.

For an example, see "Saving Control Limits" on page 1893.

ODS Tables

The following table summarizes the ODS tables that you can request with the XSCHART statement.

Table 17.79 ODS Tables Produced with the XSCHART Statement

Table Name	Description	Options
XSCHART	$\bar{X}$ and *s* chart summary statistics	TABLE, TABLEALL, TABLEC, TABLEID, TABLELEG, TABLEOUT, TABLETESTS
TestDescriptions	descriptions of tests for special causes requested with the TESTS= option for which at least one positive signal is found	TABLEALL, TABLELEG

ODS Graphics

Before you create ODS Graphics output, ODS Graphics must be enabled (for example, by using the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" (Chapter 21, *SAS/STAT User's Guide*). **NOTE:** In SAS/QC 13.1 the SHEWHART procedure does not support the creation of graphs that are editable with the ODS Graphics Editor.

The appearance of a graph produced with ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. XSCHART options used to control the appearance of traditional graphics are ignored for ODS Graphics output. Options for Producing Graphs Using ODS Styles lists options that can be used to control the appearance of graphs produced with ODS Graphics or with traditional graphics using ODS styles. Options for ODS Graphics lists options to be used exclusively with ODS Graphics. Detailed descriptions of these options are provided in "Dictionary of Options: SHEWHART Procedure" on page 1946.

When ODS Graphics is in effect, the XSCHART statement assigns a name to the graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 17.80.

Table 17.80 ODS Graphics Produced by the XSCHART Statement

ODS Graph Name	Plot Description
XSChart	$\bar{X}$ and s chart

See Chapter 3, "SAS/QC Graphics," for more information about ODS Graphics and other methods for producing charts.

Input Data Sets

DATA= Data Set

You can read raw data (process measurements) from a DATA= data set specified in the PROC SHEWHART statement. Each *process* specified in the XSCHART statement must be a SAS variable in the DATA= data set. This variable provides measurements that must be grouped into subgroup samples indexed by the *subgroup-variable*. The *subgroup-variable*, which is specified in the XSCHART statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each *process* and a value for the *subgroup-variable*. If the ith subgroup contains n_i items, there should be n_i consecutive observations for which the value of the subgroup variable is the index of the ith subgroup. For example, if each subgroup contains five items and there are 30 subgroup samples, the DATA= data set should contain 150 observations.

Other variables that can be read from a DATA= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all the observations in a DATA= data set. However, if the DATA= data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (for an example, see "Displaying Stratification in Phases" on page 2031).

For an example of a DATA= data set, see "Creating Charts for Means and Standard Deviations from Raw Data" on page 1887.

LIMITS= Data Set

You can read preestablished control limits (or parameters from which the control limits can be calculated) from a LIMITS= data set specified in the PROC SHEWHART statement. For example, the following statements read control limit information from the data set Conlims:

```
proc shewhart data=Info limits=Conlims;
   xschart Weight*Batch;
run;
```

The LIMITS= data set can be an OUTLIMITS= data set that was created in a previous run of the SHEWHART procedure. Such data sets always contain the variables required for a LIMITS= data set. The LIMITS= data set can also be created directly using a DATA step. When you create a LIMITS= data set, you must provide one of the following:

- the variables _LCLX_, _MEAN_, _UCLX_, _LCLS_, _S_, and _UCLS_, which specify the control limits directly
- the variables _MEAN_ and _STDDEV_, which are used to calculate the control limits according to the equations in Table 17.77

In addition, note the following:

- The variables _VAR_ and _SUBGRP_ are required. These must be character variables whose lengths are no greater than 32.
- The variable _INDEX_ is required if you specify the READINDEX= option; this must be a character variable whose length is no greater than 48.
- The variables _LIMITN_, _SIGMAS_ (or _ALPHA_), and _TYPE_ are optional, but they are recommended to maintain a complete set of control limit information. The variable _TYPE_ must be a character variable of length 8; valid values are 'ESTIMATE', 'STANDARD', 'STDMU', and 'STDSIGMA'.
- BY variables are required if specified with a BY statement.

For an example, see "Reading Preestablished Control Limits" on page 1896.

HISTORY= Data Set

You can read subgroup summary statistics from a HISTORY= data set specified in the PROC SHEWHART statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the SHEWHART, CUSUM, or MACONTROL procedures or to read output data sets created with SAS summarization procedures, such as the MEANS procedure.

A HISTORY= data set used with the XSCHART statement must contain the following variables:

- the *subgroup-variable*
- a subgroup mean variable for each *process*
- a subgroup standard deviation variable for each *process*
- a subgroup sample size variable for each *process*

The names of the subgroup mean, subgroup standard deviation, and subgroup sample size variables must be the *process* name concatenated with the special suffix characters *X*, *S*, and *N*, respectively. For example, consider the following statements:

```
proc shewhart history=Summary;
   xschart (Weight Yieldstrength)*Batch;
run;
```

The data set Summary must include the variables Batch, WeightX, WeightS, WeightN, YieldstrengthX, YieldstrengthS, and YieldstrengthN.

Note that if you specify a *process* name that contains 32 characters, the names of summary variables must be formed from the first 16 characters and the last 15 characters of the *process* name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- _PHASE_ (if the READPHASES= option is specified)
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

By default, the SHEWHART procedure reads all of the observations in a HISTORY= data set. However, if the data set includes the variable _PHASE_, you can read selected groups of observations (referred to as *phases*) by specifying the READPHASES= option (see “Displaying Stratification in Phases” on page 2031 for an example).

For an example of a HISTORY= data set, see “Creating Charts for Means and Standard Deviations from Summary Data” on page 1890.

TABLE= Data Set

You can read summary statistics and control limits from a TABLE= data set specified in the PROC SHEWHART statement. This enables you to reuse an OUTTABLE= data set created in a previous run of the SHEWHART procedure. Because the SHEWHART procedure simply displays the information read from a TABLE= data set, you can use TABLE= data sets to create specialized control charts. Examples are provided in “Specialized Control Charts: SHEWHART Procedure” on page 2096.

The following table lists the variables required in a TABLE= data set used with the XSCHART statement:

Table 17.81 Variables Required in a TABLE= Data Set

Variable	Description
LCLS	lower control limit for standard deviation
LCLX	lower control limit for mean
LIMITN	nominal sample size associated with the control limits
MEAN	process mean
_S_	average standard deviation
subgroup-variable	values of the *subgroup-variable*
SUBN	subgroup sample size
SUBS	subgroup standard deviation
SUBX	subgroup mean
UCLS	upper control limit for standard deviation
UCLX	upper control limit for mean

Other variables that can be read from a TABLE= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables
- _PHASE_ (if the READPHASES= option is specified). This variable must be a character variable whose length is no greater than 48.
- _TESTS_ (if the TESTS= option is specified). This variable is used to flag tests for special causes for subgroup means and must be a character variable of length 8.
- _TESTS2_ (if the TESTS2= option is specified). This variable is used to flag tests for special causes for subgroup standard deviations and must be a character variable of length 8.
- _VAR_. This variable is required if more than one *process* is specified or if the data set contains information for more than one *process*. This variable must be a character variable whose length is no greater than 32.

For an example of a TABLE= data set, see "Saving Control Limits" on page 1893.

Methods for Estimating the Standard Deviation

When control limits are determined from the input data, four methods (referred to as default, MVLUE, MVGRANGE, and RMSDF) are available for estimating σ.

Default Method

The default estimate for σ is

$$\hat{\sigma} = \frac{s_1/c_4(n_1) + \cdots + s_N/c_4(n_N)}{N}$$

where N is the number of subgroups for which $n_i \geq 2$, s_i is the sample standard deviation of the ith subgroup

$$s_i = \sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}(x_{ij} - \bar{X}_i)^2}$$

and

$$c_4(n_i) = \frac{\Gamma(n_i/2)\sqrt{2/(n_i - 1)}}{\Gamma((n_i - 1)/2)}$$

Here, $\Gamma(\cdot)$ denotes the gamma function, and $\bar{X}_i$ denotes the ith subgroup mean. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$. If the observations are normally distributed, then the expected value of s_i is $c_4(n_i)\sigma$. Thus, $\hat{\sigma}$ is the unweighted average of N unbiased estimates of σ. This method is described in the American Society for Testing and Materials (1976).

MVLUE Method

If you specify SMETHOD=MVLUE, a minimum variance linear unbiased estimate (MVLUE) is computed for σ. Refer to Burr (1969, 1976) and Nelson (1989, 1994). This estimate is a weighted average of N unbiased estimates of σ of the form $s_i/c_4(n_i)$, and it is computed as

$$\hat{\sigma} = \frac{h_1 s_1/c_4(n_1) + \cdots + h_N s_N/c_4(n_N)}{h_1 + \cdots + h_N}$$

where

$$h_i = \frac{[c_4(n_i)]^2}{1 - [c_4(n_i)]^2}$$

A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$. The MVLUE assigns greater weight to estimates of σ from subgroups with larger sample sizes, and it is intended for situations where the subgroup sample sizes vary. If the subgroup sample sizes are constant, the MVLUE reduces to the default estimate.

MVGRANGE Method

If you specify SMETHOD=MVGRANGE, σ is estimated using a moving range of subgroup averages. This is appropriate for constructing control charts for means when the jth measurement in the ith subgroup can be modeled as $x_{ij} = \sigma_B \omega_i + \sigma_W \epsilon_{ij}$, where σ_B^2 is the between-subgroup variance, σ_W^2 is the within-subgroup variance, the ω_i are independent with zero mean and unit variance, and the ω_i are independent of the ϵ_{ij}.

The estimate for σ is

$$\hat{\sigma} = \bar{R}/d_2(n)$$

where $\bar{R}$ is the average of the moving ranges, n is the number of consecutive subgroup averages used to compute each moving range, and the unbiasing factor $d_2(n)$ is defined so that if the subgroup averages are normally distributed, the expected value of R_i is

$$E(R_i) = d_2(n_i)\sigma$$

This method is appropriate for constructing a variation on the three-way control chart that is advocated for this situation by Wheeler (1995). A three-way control chart is useful when sampling, or *within-group* variation is not the only source of variation, as discussed in "Multiple Components of Variation" on page 2106. Wheeler's three-way control chart comprises a chart of subgroup means, a moving range chart of the subgroup means, and a chart of subgroup ranges. This variation substitutes a chart of subgroup standard deviations for the chart of subgroup ranges. When you specify the SMETHOD=MVGRANGE option, the XSCHART statement produces the appropriate charts of subgroup means and subgroup standard deviations.

RMSDF Method

If you specify SMETHOD=RMSDF, a weighted root-mean-square estimate is computed for σ:

$$\hat{\sigma} = \frac{\sqrt{(n_1 - 1)s_1^2 + \cdots + (n_N - 1)s_N^2}}{c_4(n)\sqrt{n_1 + \cdots + n_N - N}}$$

where $n = n_1 + \cdots + n_N - (N - 1)$. The weights are the degrees of freedom $n_i - 1$. A subgroup standard deviation s_i is included in the calculation only if $n_i \geq 2$, and N is the number of subgroups for which $n_i \geq 2$.

If the unknown standard deviation σ is constant across subgroups, the root-mean-square estimate is more efficient than the minimum variance linear unbiased estimate. However, in process control applications it is generally not assumed that σ is constant, and if σ varies across subgroups, the root-mean-square estimate tends to be more inflated than the MVLUE.

Axis Labels

You can specify axis labels by assigning labels to particular variables in the input data set, as summarized in the following table:

Axis	**Input Data Set**	**Variable**
Horizontal	all	*subgroup-variable*
Vertical ($\bar{X}$ chart)	DATA=	*process*
Vertical ($\bar{X}$ chart)	HISTORY=	subgroup mean variable
Vertical ($\bar{X}$ chart)	TABLE=	_SUBX_

You can specify distinct labels for the vertical axes of the $\bar{X}$ and s charts by breaking the vertical axis into two parts with a split character. Specify the split character with the SPLIT= option. The first part labels the vertical axis of the $\bar{X}$ chart, and the second part labels the vertical axis of the s chart.

For example, the following sets of statements specify the label *Avg Power Output* for the vertical axis of the $\bar{X}$ chart and the label *Std Deviation* for the vertical axis of the s chart:

```
proc shewhart data=Turbine;
   xschart KWatts*Day / split = '/' ;
   label KWatts = 'Avg Power Output/Std Deviation';
run;

proc shewhart history=Turbhist;
   xschart KWatts*Day / split = '/' ;
   label KWattsX = 'Avg Power Output/Std Deviation';
run;
```

```
proc shewhart table=Turbtab;
   xschart KWatts*Day / split = '/' ;
   label _SUBX_ = 'Avg Power Output/Std Deviation';
run;
```

In this example, the label assignments are in effect only for the duration of the procedure step, and they temporarily override any permanent labels associated with the variables.

Missing Values

An observation read from a DATA=, HISTORY=, or TABLE= data set is not analyzed if the value of the subgroup variable is missing. For a particular process variable, an observation read from a DATA= data set is not analyzed if the value of the process variable is missing. Missing values of process variables generally lead to unequal subgroup sample sizes. For a particular process variable, an observation read from a HISTORY= or TABLE= data set is not analyzed if the values of any of the corresponding summary variables are missing.

Examples: XSCHART Statement

This section provides advanced examples of the XSCHART statement.

Example 17.41: Specifying Probability Limits

NOTE: See *X-Bar and s Charts with Probability Limits* in the SAS/QC Sample Library.

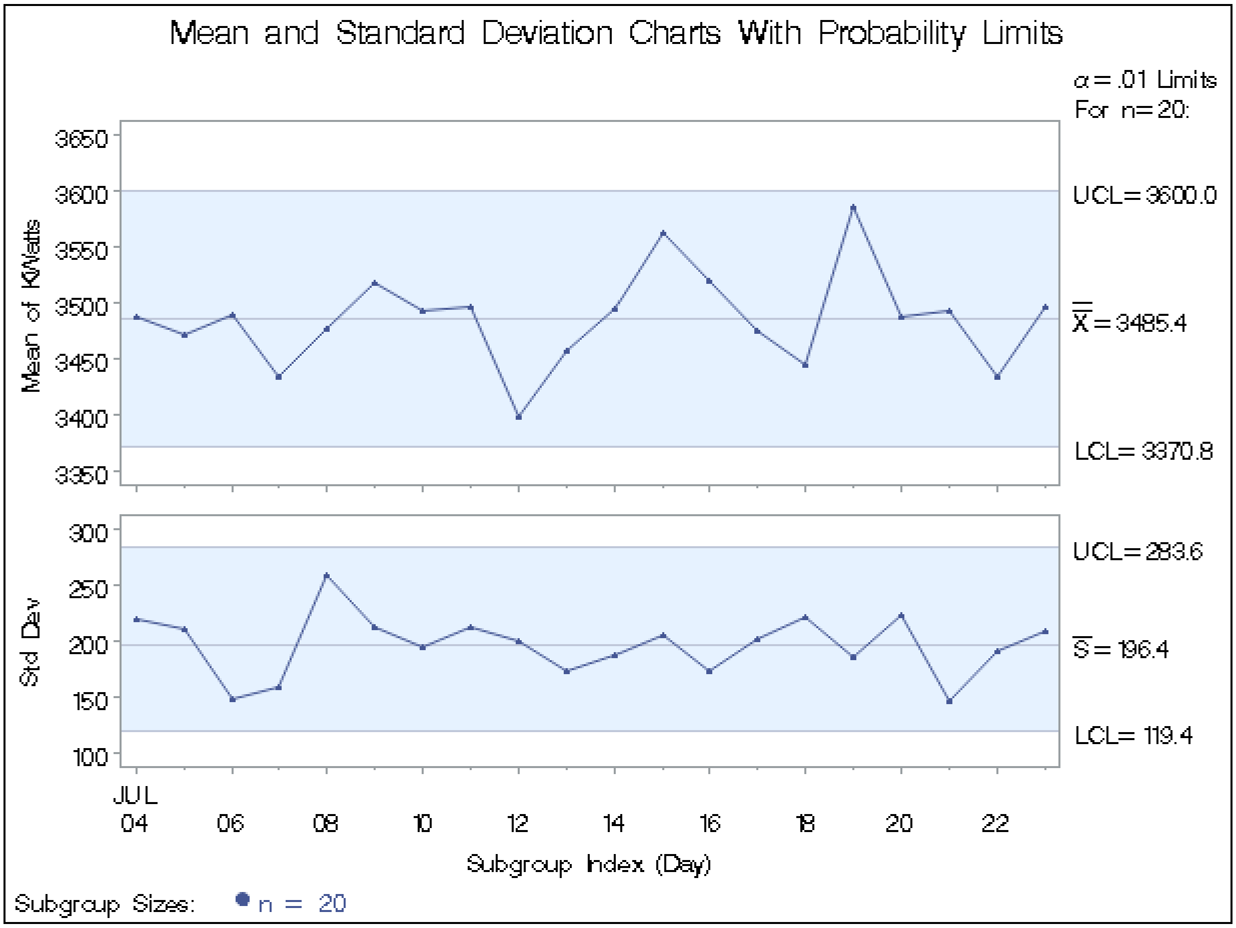
This example illustrates how to create $\bar{X}$ and s charts with probability limits. The following statements read the kilowatt power output measurements from the data set Turbine (see "Creating Charts for Means and Standard Deviations from Raw Data" on page 1887) and create the $\bar{X}$ and s charts shown in Output 17.41.1:

```
ods graphics off;
symbol v=dot h=.8;
title 'Mean and Standard Deviation Charts With Probability Limits';
proc shewhart data=Turbine;
   xschart KWatts*Day  / alpha     = 0.01
                         outlimits = Oillim;
run;
```

The ALPHA= option specifies the probability (α) that a subgroup summary statistic is outside the limits. Here, the limits are computed so that the probability that a subgroup mean or standard deviation is less than its lower limit is $\alpha/2 = 0.005$, and the probability that a subgroup mean or standard deviation is greater than its upper limit is $\alpha/2 = 0.005$. This assumes that the measurements are normally distributed.

The OUTLIMITS= option names an output data set (Oillim) that saves the probability limits. The data set Oillim is shown in Output 17.41.2.

Output 17.41.1 Probability Limits on $\bar{X}$ and s Charts



Output 17.41.2 Probability Limit Information

Mean and Standard Deviation Charts with Probability Limits

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLX_	_MEAN_	_UCLX_
KWatts	Day	ESTIMATE	20	0.01	2.57583	3370.79	3485.41	3600.03

LCLS	_S_	_UCLS_	_STDDEV_
119.432	196.396	283.570	198.996

The variable _ALPHA_ saves the value of α. The value of the variable _SIGMAS_ is computed as $k = \Phi^{-1}(1-\alpha/2)$, where Φ^{-1} is the inverse standard normal distribution function. Note that, in this case, the probability limits for the mean are equivalent to 2.58σ limits.

Since all the points fall within the probability limits, it can be concluded that the process is in statistical control.

Example 17.42: Computing Subgroup Summary Statistics

NOTE: See *Reading Subgroup Summary Data* in the SAS/QC Sample Library.

You can use output data sets from a number of SAS procedures as input data sets for the SHEWHART procedure. In this example, the MEANS procedure is used to create a data set containing subgroup summary statistics, which can be read by the SHEWHART procedure as a HISTORY= data set. The following statements create an output data set named Oilsummeans, which contains subgroup means, standard deviations, and sample sizes for the variable KWatts in the data set Turbine (see "Creating Charts for Means and Standard Deviations from Raw Data" on page 1887):

```
proc means data=Turbine noprint;
   var KWatts;
   by Day;
   output out=Oilsummeans mean=means std=stds n=sizes;
run;
```

A listing of Oilsummeans is shown in Output 17.42.1.

Output 17.42.1 The Data Set Oilsummeans

Summary Statistics for Power Output Data

Day	_TYPE_	_FREQ_	means	stds	sizes
04JUL	0	20	3487.40	220.260	20
05JUL	0	20	3471.65	210.427	20
06JUL	0	20	3488.30	147.025	20
07JUL	0	20	3434.20	157.637	20
08JUL	0	20	3475.80	258.949	20
09JUL	0	20	3518.10	211.566	20
10JUL	0	20	3492.65	193.779	20
11JUL	0	20	3496.40	212.024	20
12JUL	0	20	3398.50	199.201	20
13JUL	0	20	3456.05	173.455	20
14JUL	0	20	3493.60	187.465	20
15JUL	0	20	3563.30	205.472	20
16JUL	0	20	3519.05	173.676	20
17JUL	0	20	3474.20	200.576	20
18JUL	0	20	3443.60	222.084	20
19JUL	0	20	3586.35	185.724	20
20JUL	0	20	3486.45	223.474	20
21JUL	0	20	3492.90	145.267	20
22JUL	0	20	3432.80	190.994	20
23JUL	0	20	3496.90	208.858	20

The variables MEANS, STDS, and SIZES do not follow the naming convention required for HISTORY= data sets (see "HISTORY= Data Set" on page 1916. The following statements temporarily rename these variables to KWattsX, KWattsS, and KWattsN, respectively (the names required when the *process* KWatts is specified in the XSCHART statement):

```
title 'Mean and Standard Deviation Charts for Power Output';
proc shewhart
   history=Oilsummeans (rename=(means  = KWattsX
                                stds   = KWattsS
                                sizes  = KWattsN ));
   xschart KWatts*Day;
run;
```

The resulting charts are identical to the charts in Figure 17.115.

Example 17.43: Analyzing Nonnormal Process Data

NOTE: See *Analyzing Nonnormal Process Data* in the SAS/QC Sample Library.

The standard control limits for s charts (see Table 17.77) are calculated under the assumption that the data are normally distributed. This example illustrates how a transformation to normality can be used in conjunction with $\bar{X}$ and s charts.

The length of a metal brace is measured in centimeters for each of 20 braces sampled daily. Subgroup samples are collected for nineteen days, and the data are analyzed to determine if the manufacturing process is in statistical control.

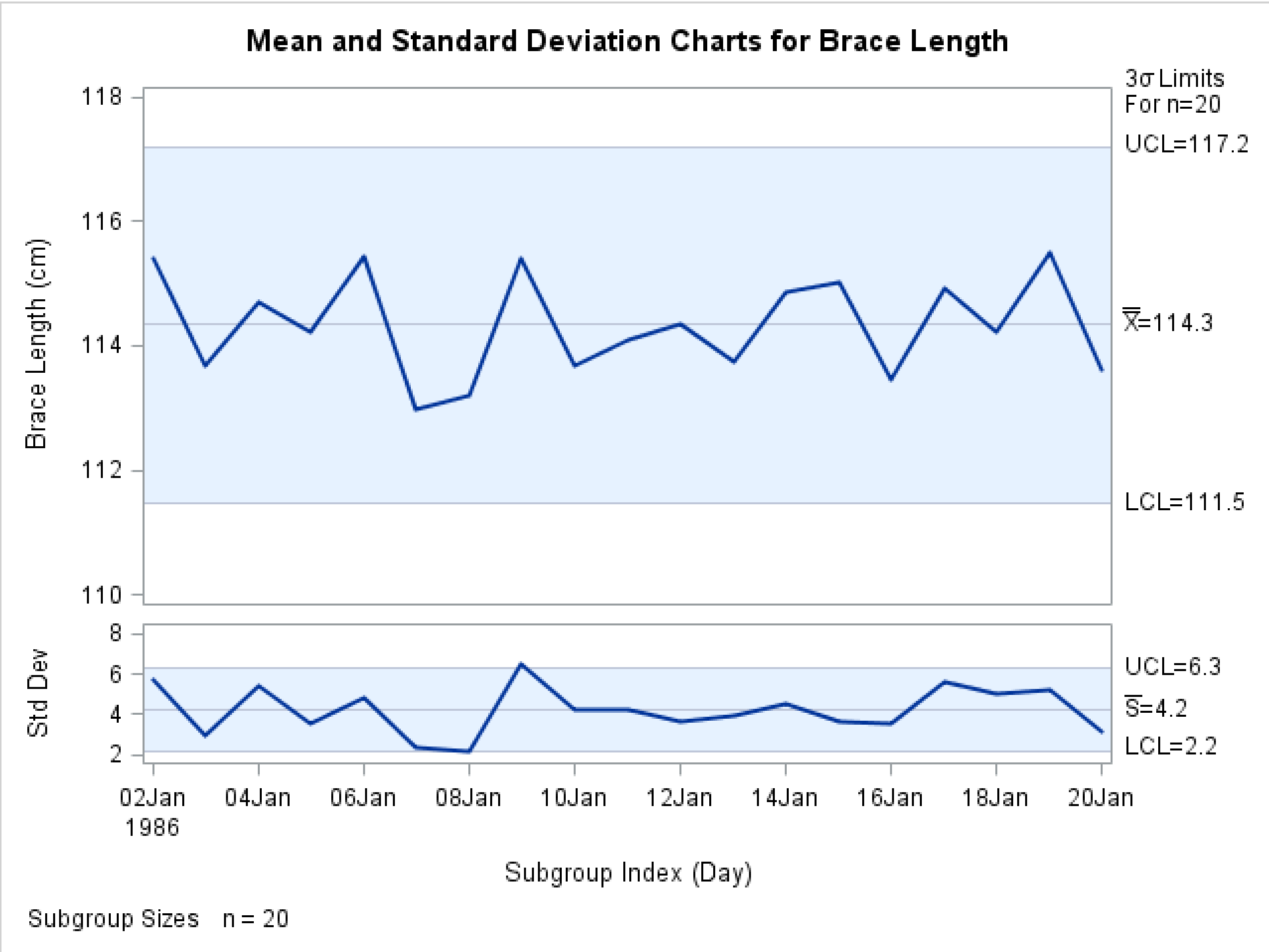
```
data LengthData;
   informat Day date7.;
   format Day date5.;
   label Length='Brace Length (cm)';
   input Day @;
   do i=1 to 5;
      input Length @;
      output;
   end;
   drop i;
   datalines;
02JAN86  113.64  119.60  111.66  111.88  125.29
02JAN86  114.08  115.28  127.84  109.97  109.34
02JAN86  109.65  121.76  112.17  116.01  111.64
02JAN86  112.70  114.43  110.27  114.76  125.89
03JAN86  115.92  113.62  117.52  114.44  118.08
03JAN86  111.13  118.42  112.16  112.25  107.71
03JAN86  110.46  113.78  109.89  114.59  116.98

   ... more lines ...

20JAN86  115.15  112.34  114.99  109.70  111.20
20JAN86  117.81  119.51  109.03  111.61  118.01
20JAN86  113.55  114.78  112.91  111.87  118.54
;
```

The following statements create preliminary $\bar{X}$ and s charts for the lengths:

```
ods graphics on;
title 'Mean and Standard Deviation Charts for Brace Length';
proc shewhart data=LengthData;
   xschart Length*Day / odstitle = title;
run;
```

The charts are shown in Output 17.43.1.

Output 17.43.1 $\bar{X}$ and s Charts



The s chart suggests that the process is not in control, since the standard deviation of the measurements recorded on January 9 exceeds its upper control limit. In addition, a number of other points on the s chart are close to the control limits.

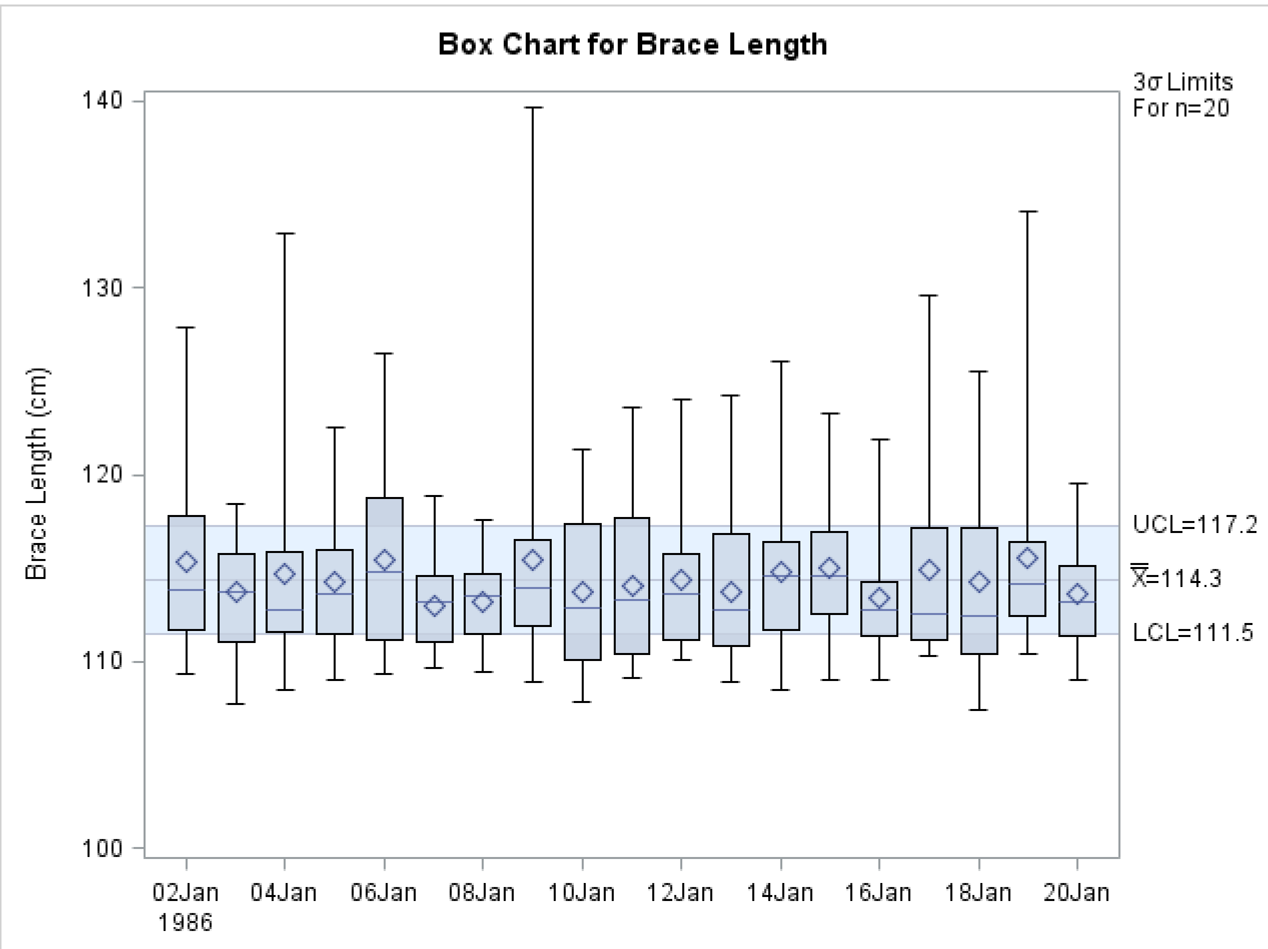
The following statements create a box chart for the lengths (for more information about box charts, see "BOXCHART Statement: SHEWHART Procedure" on page 1370).

```
title 'Box Chart for Brace Length';
proc shewhart data=LengthData;
   boxchart Length*Day / serifs
                         ranges
                         nohlabel
                         nolegend
                         odstitle = title;
run;
```

The chart, shown in Output 17.43.2, reveals that most of the subgroup distributions are skewed to the right. Consequently, the *s* chart shown in Output 17.43.1 should be interpreted with caution, since control limits for *s* charts are based on the assumption that the data are normally distributed.

No special cause for the skewness of the subgroup distributions is discovered. This indicates that the process is in statistical control and that the length distribution is naturally skewed.

Output 17.43.2 Box Chart

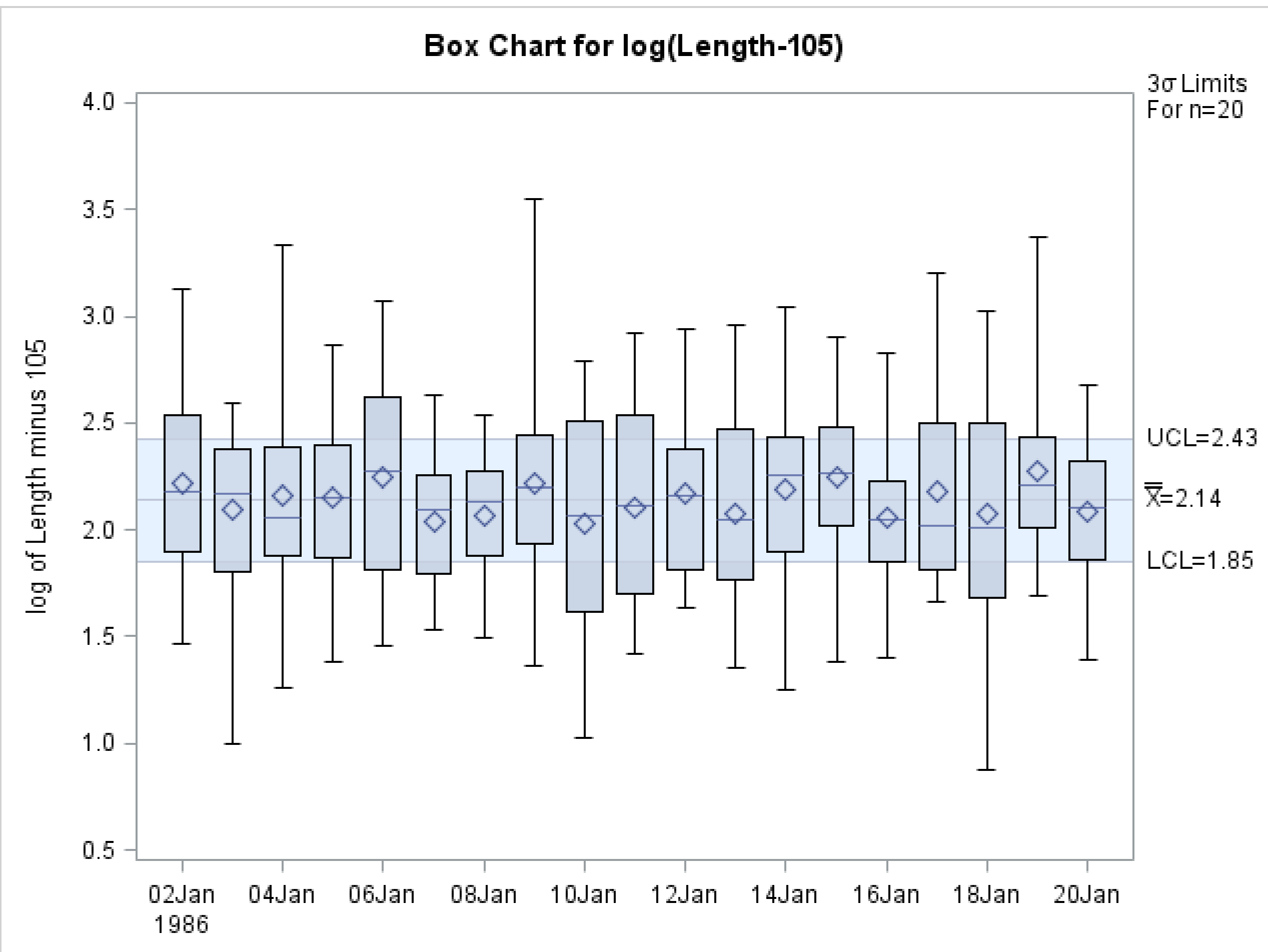


The following statements apply a lognormal transformation to the length measurements and display a box chart for the transformed data:

```
data LengthData;
   set LengthData;
   LogLength=log(Length-105);
label LogLength='log of Length minus 105';
run;

title 'Box Chart for log(Length-105)';
proc shewhart data=LengthData;
   boxchart LogLength*Day / serifs
                            ranges
                            nohlabel
                            nolegend
                            odstitle = title;
run;
```

The chart, shown in Output 17.43.3, indicates that the subgroup distributions of LogLength are approximately normal (this can be verified with goodness-of-fit tests by using the CAPABILITY procedure).

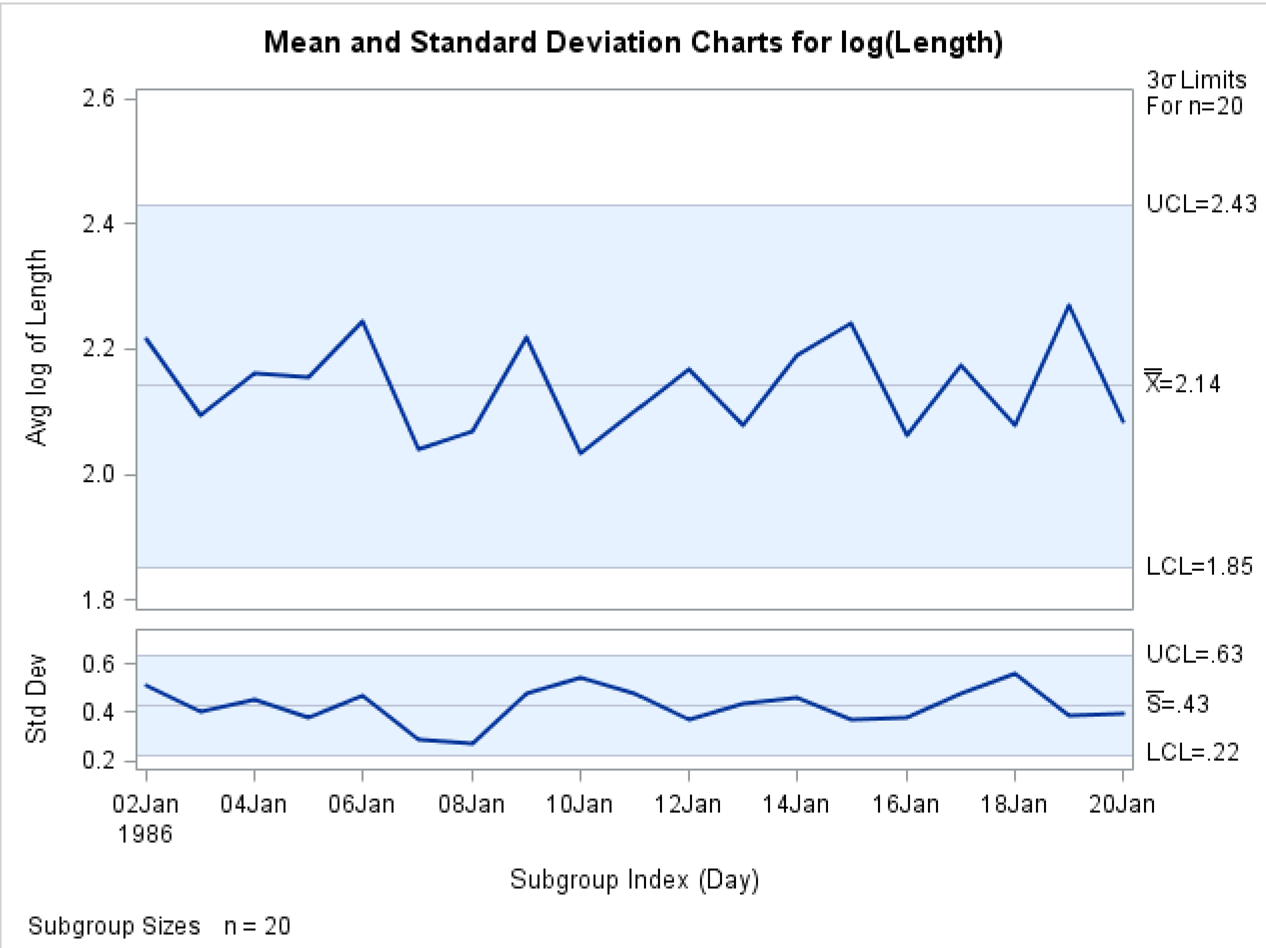
Output 17.43.3 Box Chart for Transformed Data



Finally, $\bar{X}$ and s charts, shown in Output 17.43.4, are created for LogLength. They indicate that the variability and mean level of the transformed lengths are in control.

```
title 'Mean and Standard Deviation Charts for log(Length)';
proc shewhart data=LengthData;
   xschart LogLength*Day / split    = '/'
                           odstitle = title;
   label LogLength='Avg log of Length/Std Dev';
run;
```

Output 17.43.4 $\bar{X}$ and s Charts for Transformed Length



INSET and INSET2 Statements: SHEWHART Procedure

Overview: INSET and INSET2 Statements

The INSET and INSET2 statements enable you to enhance a Shewhart chart by adding a box or table (referred to as an *inset*) of summary statistics directly to the graph. The INSET statement places an inset in a primary Shewhart chart while the INSET2 statement places one in a secondary Shewhart chart. An inset can display statistics calculated by the SHEWHART procedure or arbitrary values provided in a SAS data set.

Note that an INSET or INSET2 statement by itself does not produce a display but must be used in conjunction with a chart statement. Insets are not available with line printer charts, so the INSET and INSET2 statements are not applicable when the LINEPRINTER option is specified in the PROC SHEWHART statement.

You can use options in the INSET and INSET2 statements to

- specify the position of the inset
- specify a header for the inset table
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

The INSET2 statement differs from the INSET statement in only two respects.

1. An INSET2 statement creates an inset within a secondary chart generated by an IRCHART, MRCHART, XRCHART or XSCHART statement or by the TRENDVAR= option. For example, when following an XRCHART statement an INSET statement produces an inset in the $\bar{X}$ chart and an INSET2 statement produces one in the R chart.
2. The INSET statement can be used to place an inset in one of the margins surrounding the plot area, while the INSET2 statement cannot.

Any of the statistics available for display in an inset can be specified with either an INSET or INSET2 statement. Descriptions of the INSET statement in this section also apply to the INSET2 statement except where explicitly noted.

Getting Started: INSET and INSET2 Statements

This section introduces the INSET statement with examples that illustrate commonly used options. Complete syntax for the INSET statement is presented in the section "Syntax: INSET and INSET2 Statements" on page 1934.

Displaying Summary Statistics on a Control Chart

In the manufacture of silicon wafers, batches of five wafers are sampled, and their diameters are measured in millimeters. The following statements create a SAS data set named Wafers, which contains the measurements for 25 batches:

```
data Wafers;
   input Batch @;
   do i=1 to 5;
      input Diameter @;
      output;
   end;
   drop i;
   datalines;
 1  35.00 34.99 34.99 34.98 35.00
```

```
 2  35.01 34.99 34.99 34.98 35.00
 3  34.99 35.00 35.00 35.00 35.00
 4  35.01 35.00 34.99 34.99 35.00
 5  35.00 34.99 34.98 34.99 35.00
 6  34.99 34.99 35.00 35.00 35.00
 7  35.01 34.98 35.00 35.00 34.99
 8  35.00 35.00 34.99 34.98 34.99
 9  34.99 34.98 34.98 35.01 35.00
10  34.99 35.00 35.01 34.99 35.01
11  35.01 35.00 35.00 34.98 34.99
12  34.99 34.99 35.00 34.98 35.01
13  35.01 34.99 34.98 34.99 34.99
14  35.00 35.00 34.99 35.01 34.99
15  34.98 34.99 34.99 34.98 35.00
16  34.99 35.00 35.00 35.01 35.00
17  34.98 34.98 34.99 34.99 34.98
18  35.01 35.02 35.00 34.98 35.00
19  34.99 34.98 35.00 34.99 34.98
20  34.99 35.00 35.00 34.99 34.99
21  35.00 34.99 34.99 34.98 35.00
22  35.00 35.00 35.01 35.00 35.00
23  35.02 35.00 34.98 35.02 35.00
24  35.00 35.00 34.99 35.01 34.98
25  34.99 34.99 34.99 35.00 35.00
;
```

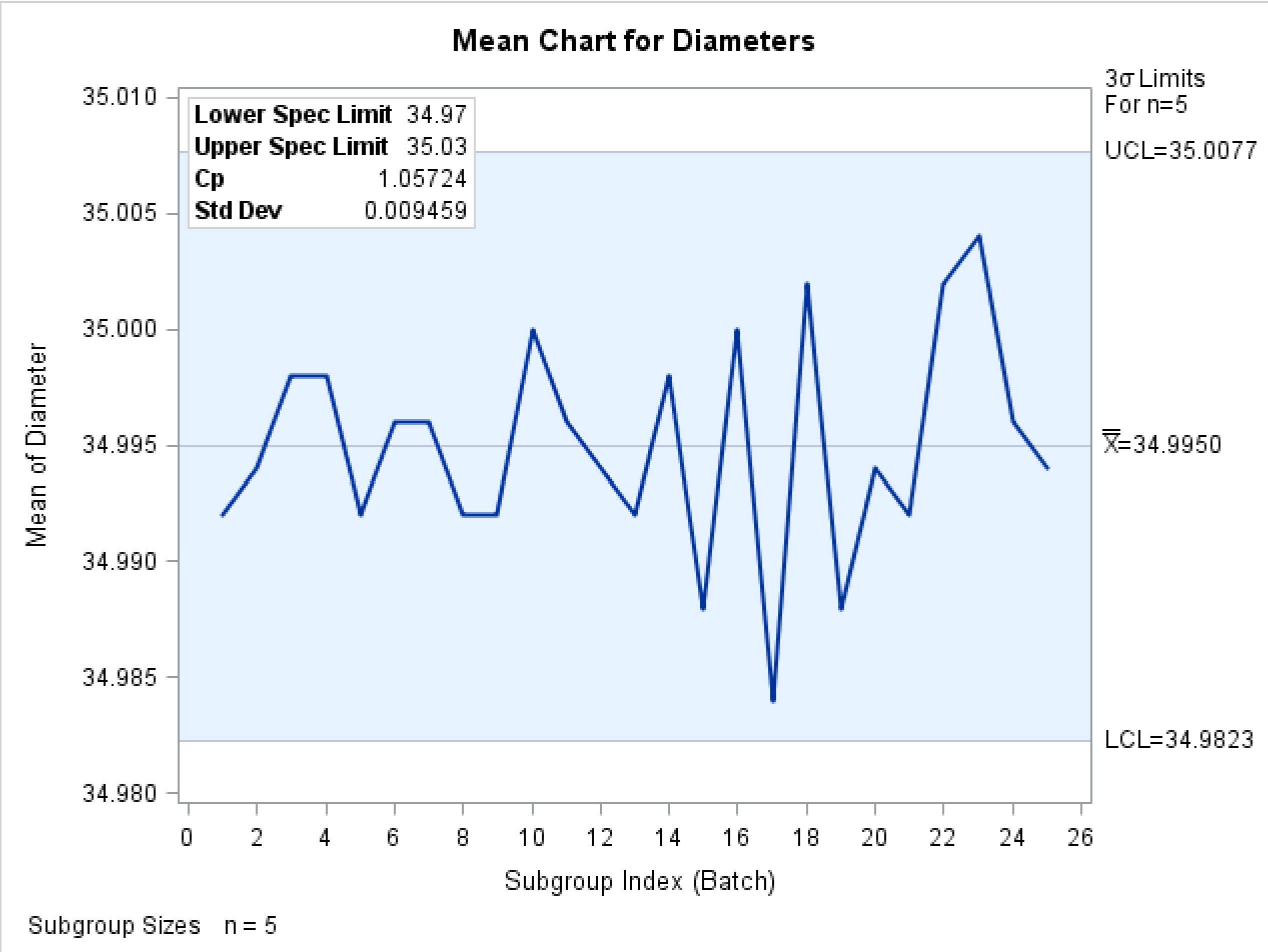
The following statements generate an $\bar{X}$ chart from the Wafers data. Lower and upper specification limits for wafer diameters are given and the process capability index C_p is computed. An INSET statement is used to display the specification limits, the computed value of C_p and the process standard deviation on the chart:

```
ods graphics on;
title 'Mean Chart for Diameters';
proc shewhart data=Wafers;
   xchart Diameter*Batch /
      lsl      = 34.97
      usl      = 35.03
      odstitle = title;
   inset lsl usl cp stddev / height = 3;
run;
```

The resulting $\bar{X}$ chart is displayed in Figure 17.122. The INSET statement immediately follows the chart statement that creates the graphical display (in this case, the XCHART statement). Specify the keywords for inset statistics (such as LSL, USL, CP and STDDEV) immediately after the word INSET. The inset statistics appear in the order in which you specify the keywords. The HEIGHT= option on the INSET statement specifies the text height used to display the statistics in the inset.

A complete list of keywords that you can use with the INSET statement is provided in "Summary of INSET Keywords" on page 1936. Note that the set of keywords available for a particular display depends on both the plot statement that precedes the INSET statement and the options that you specify in the plot statement.

Figure 17.122 An $\bar{X}$ Chart with an Inset



The following examples illustrate options commonly used for enhancing the appearance of an inset.

Formatting Values and Customizing Labels

By default, each inset statistic is identified with an appropriate label, and each numeric value is printed using an appropriate format. However, you may want to provide your own labels and formats. For example, in Figure 17.122 the default format used for C_p and the process standard deviation prints an excessive number of decimal places. The following statements produce $\bar{X}$ and R charts, each with its own inset. The unwanted decimal places are eliminated and the default specification limits labels are replaced with abbreviations:

```
title 'Mean Chart for Diameters';
proc shewhart data=Wafers;
   xrchart Diameter*Batch /
      lsl      = 34.97
      usl      = 35.03
      odstitle = title;
   inset lsl='LSL' usl='USL' / pos = nw;
   inset2 cp (6.4) stddev (6.4) / pos = nw;
run;
```

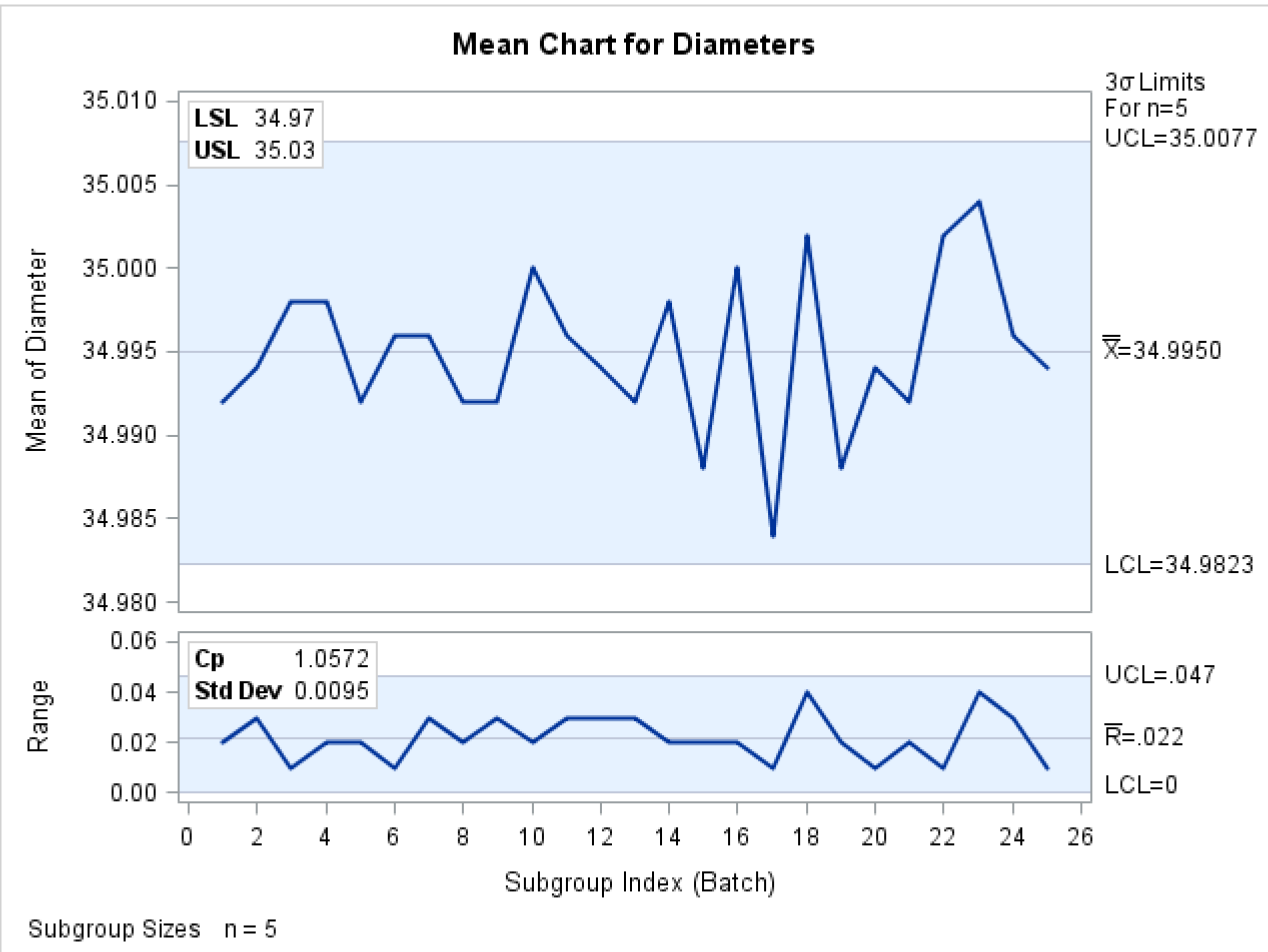
The ODS GRAPHICS ON statement specified before the PROC SHEWHART statement enables ODS Graphics, so the $\bar{X}$ and R charts are created using ODS Graphics instead of traditional graphics. The resulting charts are displayed in Figure 17.123.

You can provide your own label by specifying the keyword for that statistic followed by an equal sign (=) and the label in quotes. The label can have up to 24 characters.

The format 6.4 specified in parentheses after the CP and STDDEV keywords displays those statistics with a field width of six and four decimal places. In general, you can specify any numeric SAS format in parentheses after an inset keyword. You can also specify a format to be used for all the statistics in the INSET statement with the FORMAT= option. For more information about SAS formats, refer to *SAS Formats and Informats: Reference*.

Note that if you specify both a label and a format for a statistic, the label must appear before the format.

Figure 17.123 Formatting Values and Customizing Labels in an Inset



Adding a Header and Positioning the Inset

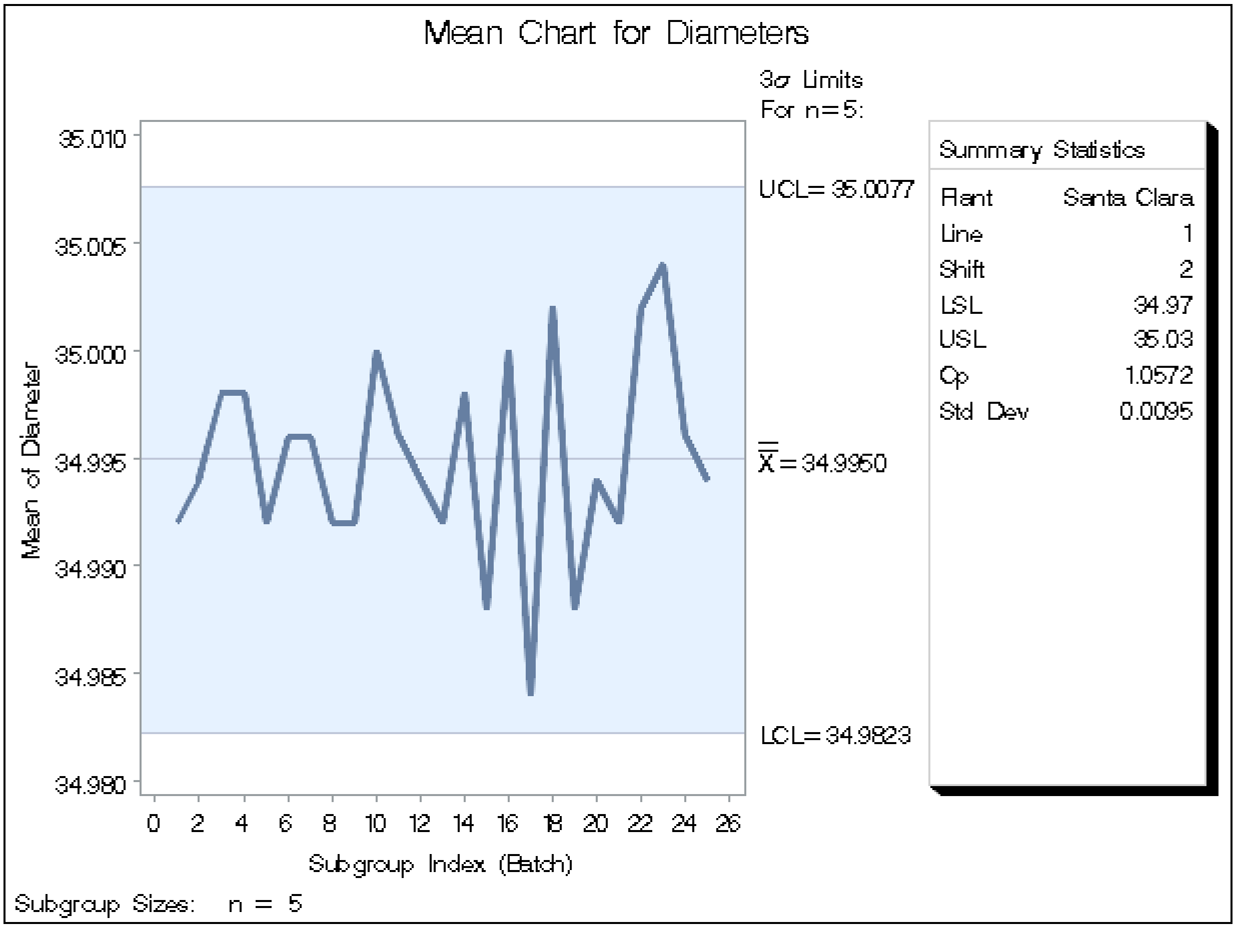
In the previous examples, the insets are displayed in the upper left corners of the plots, the default position for insets added to control charts. You can control the inset position with the POSITION= option. In addition, you can display a header at the top of the inset with the HEADER= option. The following statements create a data set to be used with the INSET DATA= keyword and the chart shown in Figure 17.124:

```
data Location;
   length _LABEL_ $ 10 _VALUE_ $ 12;
   input _LABEL_ _VALUE_ &;
   datalines;
Plant     Santa Clara
Line      1
Shift     2
;

ods graphics off;
title 'Mean Chart for Diameters';
proc shewhart data=Wafers;
   xchart Diameter*Batch /
      lsl = 34.97
      usl = 35.03;
   inset data= Location lsl='LSL' usl='USL' cp (6.4) stddev (6.4) /
      position = rm
      cshadow  = black
      header   = 'Summary Statistics';
run;
```

The header (in this case, *Summary Statistics*) can be up to 40 characters. Note that a relatively long list of inset statistics is requested. Consequently, POSITION=RM is specified to position the inset in the right margin. For more information about positioning, see "Details: INSET and INSET2 Statements" on page 1941. The CSHADOW= option is used to display a drop shadow on this inset. The *options*, such as HEADER=, POSITION= and CSHADOW= are specified after the slash (/) in the INSET statement. For more details on INSET statement options, see "Dictionary of Options" on page 1939.

Note that the contents of the data set LOCATION appear before other statistics in the inset. The position of the DATA= keyword in the keyword list determines the position of the data set's contents in the inset.

Figure 17.124 Adding a Header and Repositioning the Inset



Syntax: INSET and INSET2 Statements

The syntax for the INSET and INSET2 statements is as follows:

INSET *keyword-list* < *options* > **;**

INSET2 *keyword-list* < *options* > **;**

You can use any number of INSET and INSET2 statements in the SHEWHART procedure. However, when ODS Graphics is enabled, at most two insets are displayed inside the primary and secondary plot areas, and at most two are displayed in the chart margins. Each INSET or INSET2 statement produces a separate inset and must follow one of the chart statements. The inset appears on every panel (page) produced by the last chart statement preceding it. The statistics are displayed in the order in which they are specified. The following statements produce a boxplot with two insets and an $\bar{X}$ and R chart with one inset in the $\bar{X}$ chart and one in the R chart.

```
proc shewhart data=Wafers;
   boxchart Diameter * Batch / lsl=34.9 target=35 usl=35.1;
      inset lsl target usl;
      inset cp cpk cpm;
```

```
      xrchart Diameter * Batch;
         inset nmin nmax nout;
         inset2 nlow2 nhigh2;
   run;
```

The statistics displayed in an inset are computed for a specific process variable using observations for the current BY group. For example, in the following statements, there are two process variables (Weight and Diameter) and a BY variable (Location). If there are three different locations (levels of Location), then a total of six $\bar{X}$ charts are produced. The statistics in each inset are computed for a particular variable and location. The labels in the inset are the same for each $\bar{X}$ chart.

```
   proc shewhart data=Axles;
      by Location;
      xchart (Weight Diameter) * Batch / tests=1 to 8;
      inset ntests 1 to 8;
   run;
```

The components of the INSET and INSET2 statements are described as follows.

keyword-list

can include any of the *keywords* listed in "Summary of INSET Keywords" on page 1936. Some *keywords*, such as NTESTS and DATA=, require operands specified immediately after the *keyword*. Also, some inset statistics are available only if you request chart statements and options for which those statistics are calculated. For example,

- the NHIGH2, NLOW2, NTESTS2, LCL2 and UCL2 keywords are available only when a secondary chart is produced with the IRCHART, MRCHART, XRCHART or XSCHART statements.
- the NTESTS *keyword* requires the TESTS= option;
- the NTESTS2 *keyword* requires the TESTS2= option;
- the capability index *keywords* such as CPK all require one or more of the LSL=, USL= and TARGET= options.

By default, inset statistics are identified with appropriate labels, and numeric values are printed using appropriate formats. However, you can provide customized labels and formats. You provide the customized label by specifying the *keyword* for that statistic followed by an equal sign (=) and the label in quotes. Labels can have up to 24 characters. You provide the numeric format in parentheses after the *keyword*. Note that if you specify both a label and a format for a statistic, the label must appear before the format. For an example, see "Formatting Values and Customizing Labels" on page 1931.

options

appear after the slash (/) and control the appearance of the inset. For example, the following INSET statement uses two appearance *options* (POSITION= and CTEXT=):

```
      inset n nmin nmax / position=ne ctext=yellow;
```

The POSITION= option determines the location of the inset, and the CTEXT= option specifies the color of the text of the inset.

See "Summary of Options" on page 1938 for a list of all available *options*, and "Dictionary of Options" on page 1939 for detailed descriptions. Note the difference between *keywords* and *options*; *keywords* specify the information to be displayed in an inset, whereas *options* control the appearance of the inset.

Summary of INSET Keywords

All keywords available with the SHEWHART procedure's INSET and INSET2 statements request a single statistic in an inset, except for the NTESTS, NTESTS2 and DATA= keywords. The NTESTS and NTESTS2 keywords each require a list of indexes specifying the tests for special causes whose counts of positive results are to be displayed:

```
inset ntests 1 2 3 4;
inset ntests2 1 to 4;
```

For each of the requested tests, the number of positive results for the test is displayed in the inset. So if tests 1 through 4 are requested the results occupy four lines in the inset.

The DATA= keyword specifies a SAS data set containing (label, value) pairs to be displayed in an inset. The data set must contain the variables _LABEL_ and _VALUE_. _LABEL_ is a character variable whose values provide labels for inset entries. _VALUE_ can be character or numeric, and provides values displayed in the inset. The label and value from each observation in the DATA= data set occupy one line in the inset. Figure 17.124 shows an inset containing entries from a DATA= data set.

Table 17.82 Summary Statistics

Keyword	Description
DATA=	(label, value) pairs from *SAS-data-set*
LCL	primary chart lower control limit
MEAN	estimated or specified process mean
N	nominal subgroup size
NMIN	minimum subgroup size
NMAX	maximum subgroup size
NOUT	number of subgroups outside control limits on primary chart
NLOW	number of subgroups below lower control limit on primary chart
NHIGH	number of subgroups above upper control limit on primary chart
NTESTS	number of positive results of tests for special causes on primary chart
STDDEV	estimated or specified process standard deviation
UCL	primary chart lower control limit

Table 17.83 Secondary Chart Summary Statistics

Keyword	Description
LCL2	secondary chart lower control limit
MEAN2	mean of subgroup ranges or standard deviations
NOUT2	number of subgroups outside control limits on secondary chart
NLOW2	number of subgroups below lower control limit on secondary chart
NHIGH2	number of subgroups above upper control limit on secondary chart
NTESTS2	number of positive results of tests for special causes on secondary chart
UCL2	secondary chart upper control limit

Table 17.84 Specification Limits

Keyword	Description
LSL	lower specification limit
USL	upper specification limit
TARGET	target value

Table 17.85 Capability Indices and Confidence Limits

Keyword	Description
CIALPHA	α value for computing capability index confidence limits
CP	capability index C_p
CPLCL	lower confidence limit for C_p
CPUCL	upper confidence limit for C_p
CPK	capability index C_{pk}
CPKLCL	lower confidence limit for C_{pk}
CPKUCL	upper confidence limit for C_{pk}
CPL	capability index CPL
CPLLCL	lower confidence limit for CPL
CPLUCL	upper confidence limit for CPL
CPM	capability index C_{pm}
CPMLCL	lower confidence limit for C_{pm}
CPMUCL	upper confidence interval for C_{pm}
CPU	capability index CPU
CPULCL	lower confidence limit for CPU
CPUUCL	upper confidence limit for CPU

You can use the keywords in Table 17.86 only when producing ODS Graphics output. Greek letters are used in the labels for the statistics requested with the UMU and USIGMA keywords.

Table 17.86 Keywords Specific to ODS Graphics Output

Keyword	Description
TESTLEGEND	requests a legend of positive tests for special causes
UMU	estimated or specified process mean
USIGMA	estimated or specified process standard deviation

Summary of Options

The following table lists the INSET and INSET2 statement options. For complete descriptions, see "Dictionary of Options" on page 1939 which follows this section.

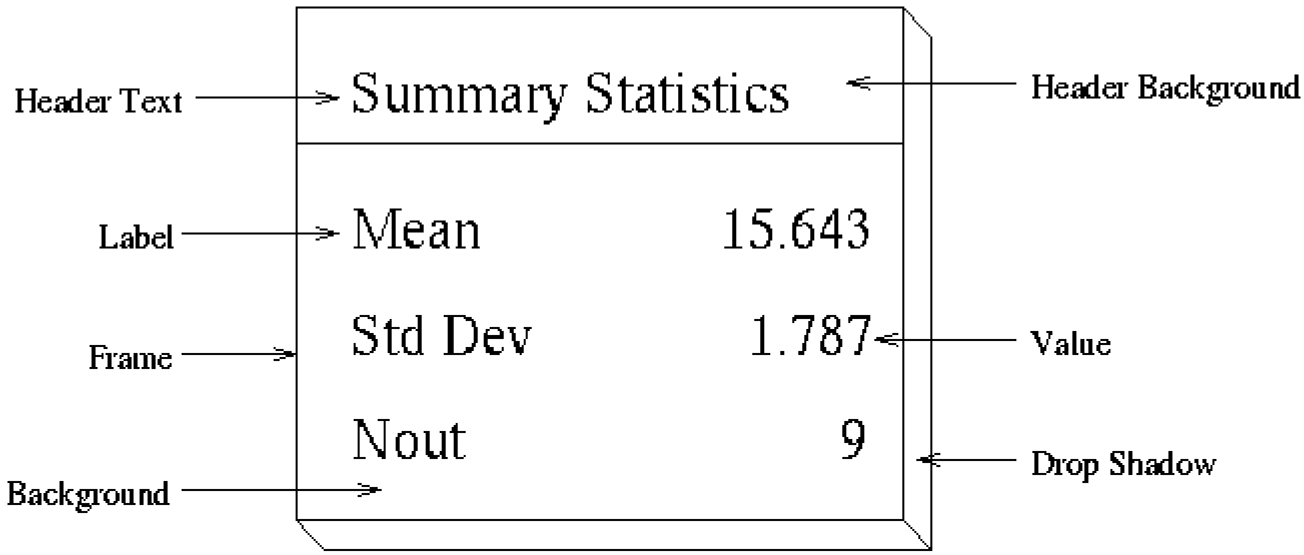
Table 17.87 INSET Options

Option	Description
CFILL=	specifies color of inset background
CFILLH=	specifies color of header background
CFRAME=	specifies color of frame
CHEADER=	specifies color of header text
CSHADOW=	specifies color of drop shadow
CTEXT=	specifies color of inset text
DATA	specifies data units for POSITION=(x, y) coordinates
FONT=	specifies font of text
FORMAT=	specifies format of values in inset
HEADER=	specifies header text
HEIGHT=	specifies height of inset text
NOFRAME	suppresses frame around inset
POSITION=	specifies position of inset
REFPOINT=	specifies reference point of inset positioned with POSITION=(x, y) coordinates

Dictionary of Options

The following sections provide detailed descriptions of options for the INSET and INSET2 statements. Terms used in this section are illustrated in Figure 17.125.

Figure 17.125 The Inset



General Options

You can specify the following options whether you use ODS Graphics or traditional graphics:

DATA

specifies that data coordinates are to be used in positioning the inset with the POSITION= option. The DATA option is available only when you specify POSITION= (x, y), and it must be placed immediately after the coordinates (x, y). For details, see the entry for the POSITION= option or "Positioning the Inset Using Coordinates" on page 1943. See Figure 17.128 for an example.

FORMAT=*format*

specifies a format for all the values displayed in an inset. If you specify a format for a particular statistic, then this format overrides the format you specified with the FORMAT= option.

HEADER= *'string'*

specifies the header text. The *string* cannot exceed 40 characters. If you do not specify the HEADER= option, no header line appears in the inset.

HEIGHT=*value*

HEIGHT=SMALL

specifies the height of the text in in the inset. By default, the GraphLabelText style element determines the size of inset header text and the GraphValueText style element determines the size of text in the body of the inset.

When you produce traditional graphics, you can specify the *height* in screen percent units to be used for text in both the header and the body of the inset.

When you produce ODS Graphics output, you can specify HEIGHT=SMALL to reduce the height of text in the inset. The GraphValueText size is used for the inset header and the GraphDataText size is used in the inset body.

NOFRAME

suppresses the frame drawn around the text.

POSITION=*position*

POS=*position*

determines the position of the inset. The *position* can be a compass point keyword, a margin keyword, or a pair of coordinates (x, y). You can specify coordinates in axis percent units or axis data units. For more information, see "Details: INSET and INSET2 Statements" on page 1941. By default, POSITION=NW, which positions the inset in the upper left (northwest) corner of the display.

NOTE: You cannot specify coordinates with the POSITION= option when producing ODS Graphics output.

REFPOINT=BR | BL | TR | TL

RP=BR | BL | TR | TL

specifies the reference point for an inset that is positioned by a pair of coordinates with the POSITION= option. Use the REFPOINT= option with POSITION= coordinates. The REFPOINT= option specifies which corner of the inset frame you want positioned at coordinates (x, y). The keywords BL, BR, TL, and TR represent bottom left, bottom right, top left, and top right, respectively. See Figure 17.129 for an example. The default is REFPOINT=BL.

If you specify the position of the inset as a compass point or margin keyword, the REFPOINT= option is ignored. For more information, see "Positioning the Inset Using Coordinates" on page 1943.

Options for Traditional Graphics

You can specify the following options only when you produce traditional graphics:

CFILL=*color* | BLANK

specifies the color of the background (including the header background if you do not specify the CFILLH= option).

If you do not specify the CFILL= option, then by default, the background is empty. This means that items that overlap the inset (such as subgroup data points or control limits) show through the inset. If you specify any value for the CFILL= option, then overlapping items no longer show through the inset. Specify CFILL=BLANK to leave the background uncolored and also to prevent items from showing through the inset.

CFILLH=*color*

specifies the color of the header background. By default, if you do not specify a CFILLH= color, the CFILL= color is used.

CFRAME=*color*

specifies the color of the frame. By default, the frame is the same color as the axis of the plot.

CHEADER=*color*
specifies the color of the header text. By default, if you do not specify a CHEADER= color, the CTEXT= color is used.

CSHADOW=*color*
CS=*color*
specifies the color of the drop shadow. See Figure 17.124 for an example. By default, if you do not specify the CSHADOW= option, a drop shadow is not displayed.

CTEXT=*color*
CT=*color*
specifies the color of the text. By default, the inset text color is the same as the other text on the plot.

FONT=*font*
specifies the font used for the text in the inset. By default, the font associated with the GraphLabelText style element is used for inset header and that associated with the GraphValueText style element is used for text in the body of the inset.

Details: INSET and INSET2 Statements

This section provides details on three different methods of positioning the inset using the POSITION= option. With the POSITION= option, you can specify

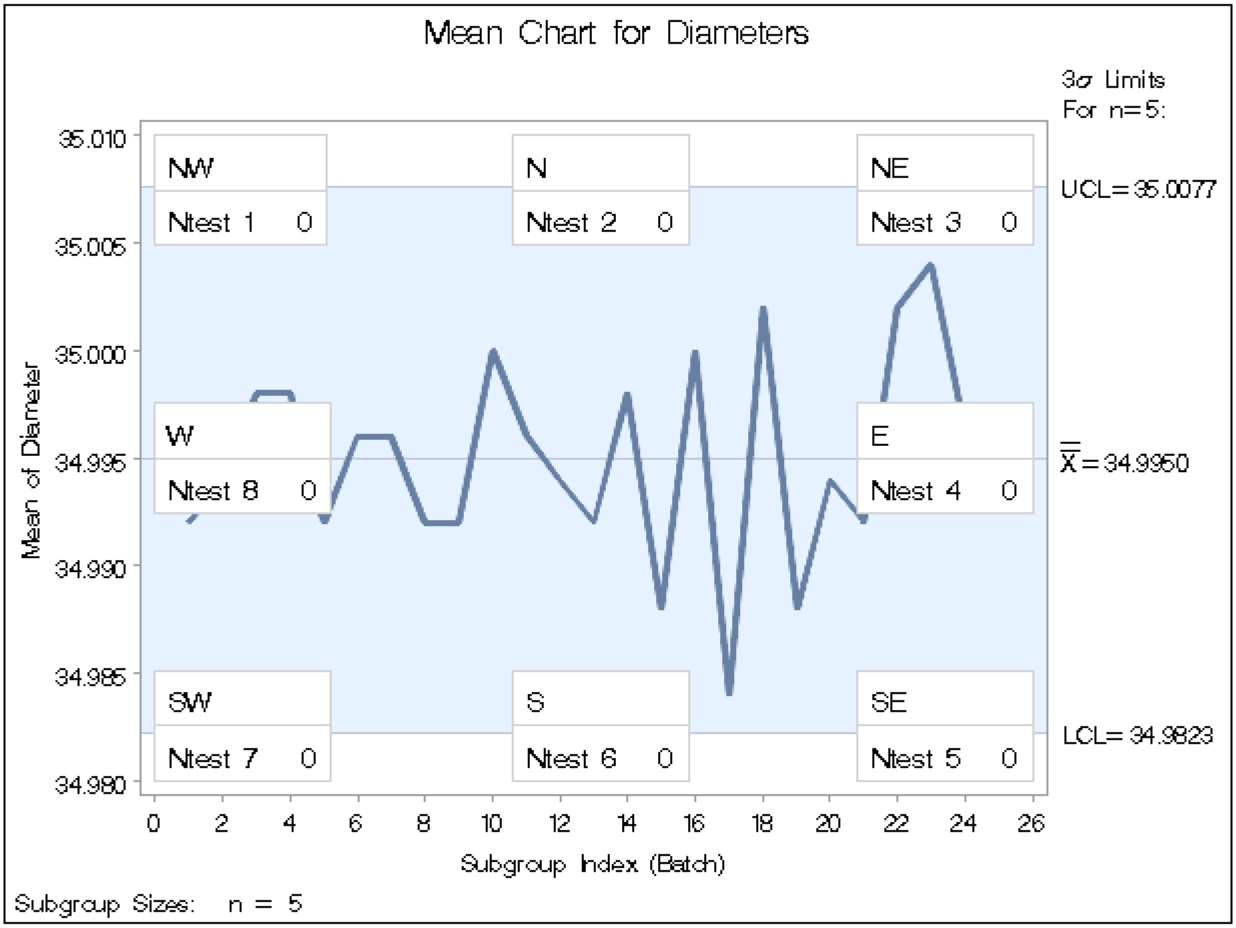
- compass points
- keywords for margin positions
- coordinates in data units or percent axis units

Positioning the Inset Using Compass Points

You can specify the eight compass points N, NE, E, SE, S, SW, W, and NW as keywords for the POSITION= option. The following statements create the display in Figure 17.126, which demonstrates all eight compass positions. The default is NW.

```
ods graphics off;
title 'Mean Chart for Diameters';
proc shewhart data=Wafers;
   xchart Diameter*Batch / tests= 1 to 8;
   inset ntests 1 / height=3 cfill=blank header='NW' pos=nw;
   inset ntests 2 / height=3 cfill=blank header='N ' pos=n ;
   inset ntests 3 / height=3 cfill=blank header='NE' pos=ne;
   inset ntests 4 / height=3 cfill=blank header='E ' pos=e ;
   inset ntests 5 / height=3 cfill=blank header='SE' pos=se;
   inset ntests 6 / height=3 cfill=blank header='S ' pos=s ;
   inset ntests 7 / height=3 cfill=blank header='SW' pos=sw;
   inset ntests 8 / height=3 cfill=blank header='W ' pos=w ;
run;
```

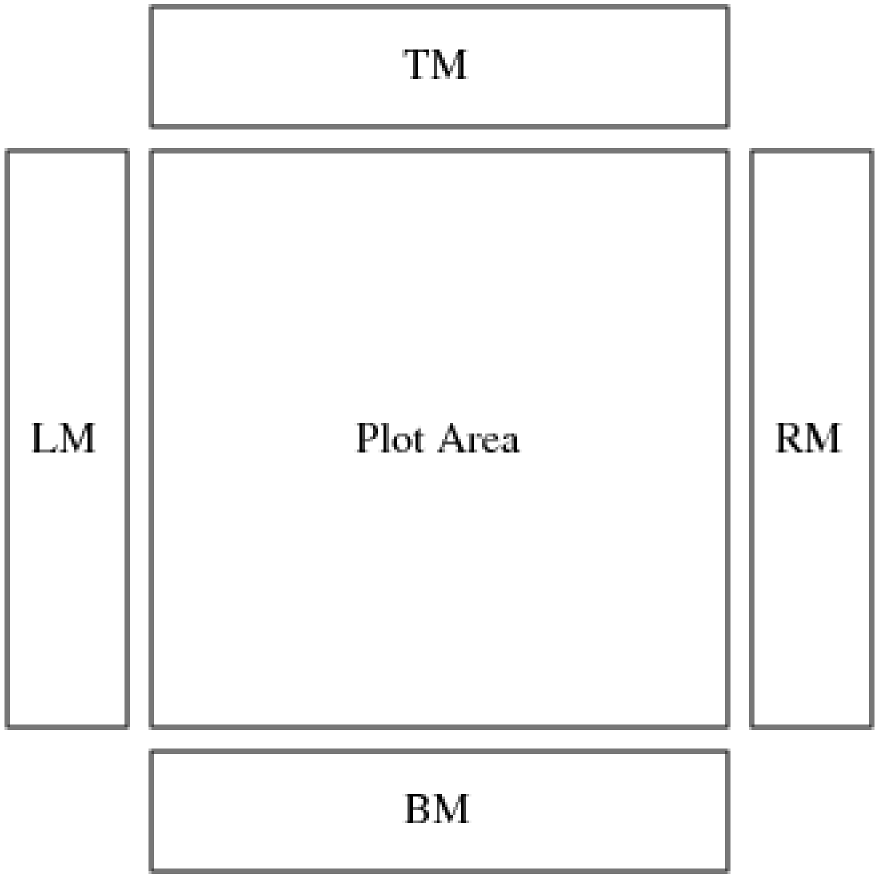
Figure 17.126 Insets Positioned Using Compass Points



Positioning the Inset in the Margins

Using the INSET statement you can also position an inset in one of the four margins surrounding the plot area using the margin keywords LM, RM, TM, or BM, as illustrated in Figure 17.127. The INSET2 statement cannot be used to produce an inset in a margin.

Figure 17.127 Positioning Insets in the Margins



For an example of an inset placed in the right margin, see Figure 17.124. Margin positions are recommended if a large number of statistics are listed in the INSET statement. If you attempt to display a lengthy inset in the interior of the plot, it is likely that the inset will collide with the data display.

Positioning the Inset Using Coordinates

You can also specify the position of the inset with coordinates: POSITION= (x, y). The coordinates can be given in axis percent units (the default) or in axis data units.

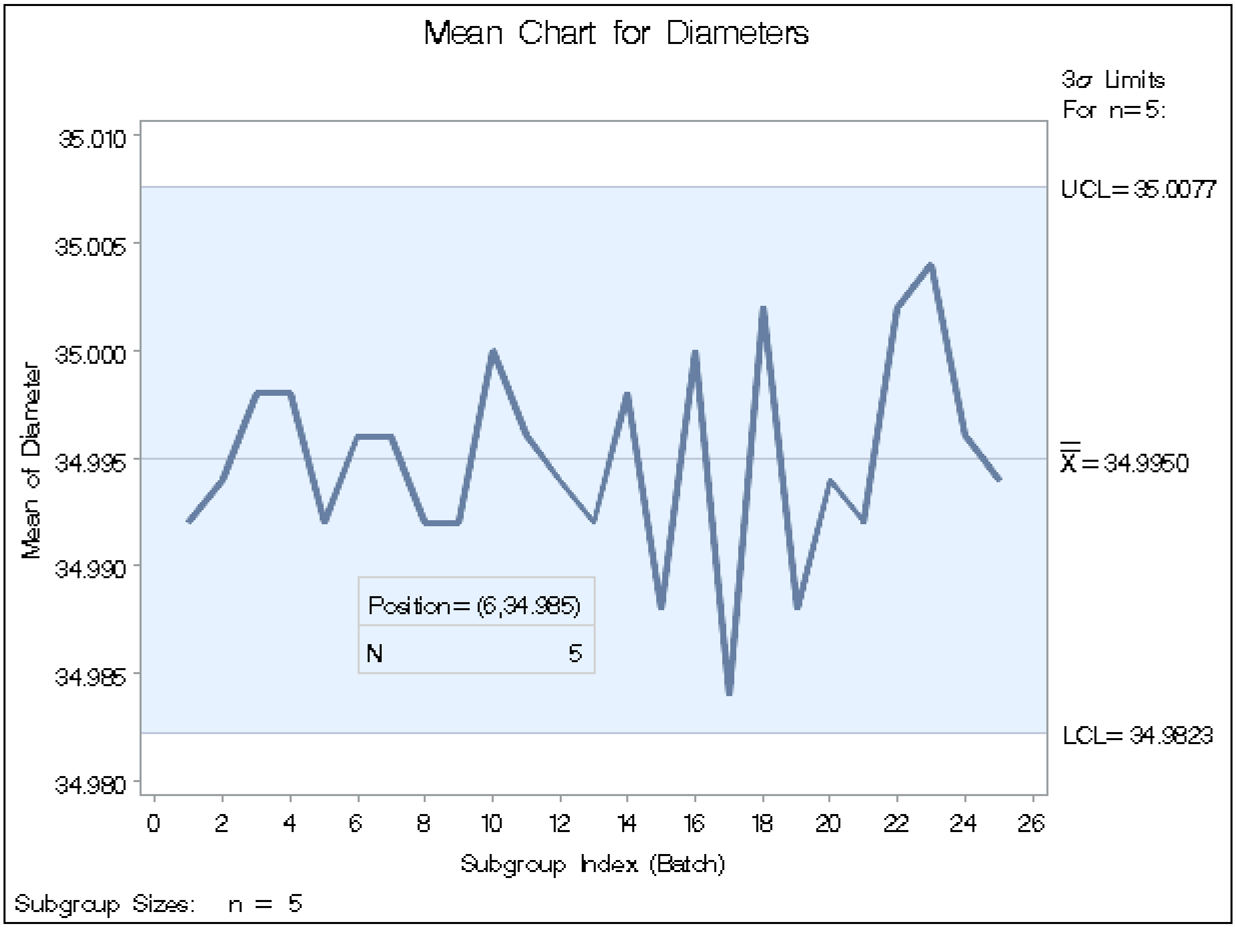
Data Unit Coordinates

If you specify the DATA option immediately following the coordinates, the inset is positioned using axis data units. For example, the following statements place the bottom left corner of the inset at 6 on the horizontal axis and 34.985 on the vertical axis:

```
title 'Mean Chart for Diameters';
proc shewhart data=Wafers;
   xchart Diameter*Batch;
   inset n /
      header   = 'Position=(6,34.985)'
      position = (6,34.985) data;
run;
```

The control chart is displayed in Figure 17.128. By default, the specified coordinates determine the position of the bottom left corner of the inset. You can change this reference point with the REFPOINT= option, as in the next example.

Figure 17.128 Inset Positioned Using Data Unit Coordinates



Axis Percent Unit Coordinates

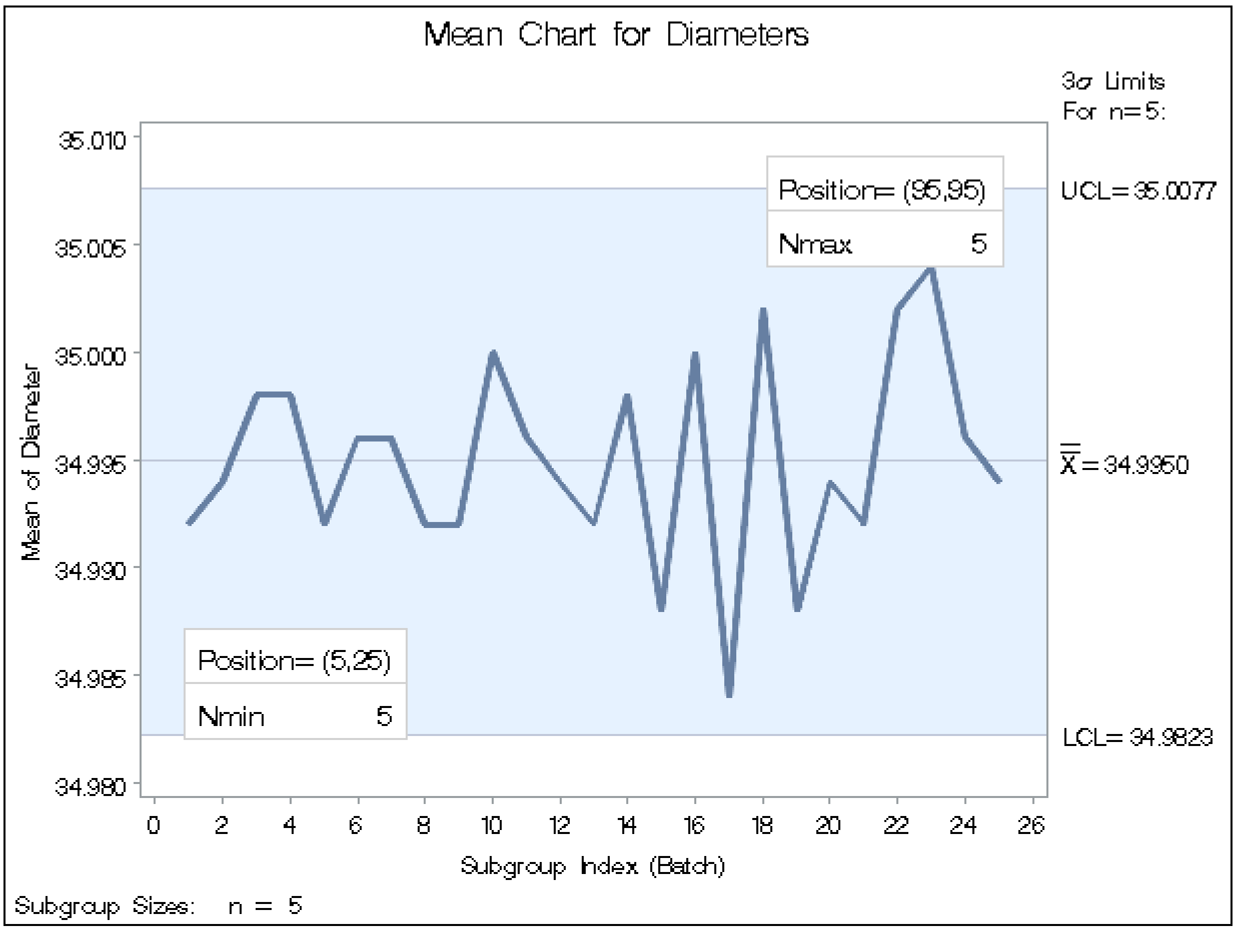
If you do not use the DATA option, the inset is positioned using axis percent units. The coordinates of the bottom left corner of the display are $(0, 0)$, while the upper right corner is $(100, 100)$. For example, the following statements create a $\bar{X}$ chart with two insets, both positioned using coordinates in axis percent units:

```
title 'Mean Chart for Diameters';
proc shewhart data=Wafers;
   xchart Diameter*Batch;
   inset nmin / position = (5,25)
                header   = 'Position=(5,25)'
                height   = 3
                cfill    = blank
                refpoint = tl;
   inset nmax / position = (95,95)
                header   = 'Position=(95,95)'
                height   = 3
```

```
                    cfill    = blank
                    refpoint = tr;
run;
```

The display is shown in Figure 17.129. Notice that the REFPOINT= option is used to determine which corner of the inset is to be placed at the coordinates specified with the POSITION= option. The first inset has REFPOINT=TL, so the top left corner of the inset is positioned 5% of the way across the horizontal axis and 25% of the way up the vertical axis. The second inset has REFPOINT=TR, so the top right corner of the inset is positioned 95% of the way across the horizontal axis and 95% of the way up the vertical axis. Note also that coordinates in axis percent units must be *between* 0 and 100.

Figure 17.129 Inset Positioned Using Axis Percent Unit Coordinates



Dictionary of Options: SHEWHART Procedure

The section provides detailed descriptions of options that you can specify in the following chart statements:

- BOXCHART
- CCHART
- IRCHART
- MCHART
- MRCHART
- NPCHART
- PCHART
- RCHART
- SCHART
- UCHART
- XCHART
- XRCHART
- XSCHART

Options are specified after the slash (/) in a chart statement. For example, to request tests for special causes with an $\bar{X}$ and *R* chart, you can use the TESTS= option as follows:

```
proc shewhart data=Measures;
   xrchart Length*Sample / tests=1 to 4 ;
run;
```

The options described in these sections are listed alphabetically. For tables of options organized by function, see the "Summary of Options" tables in the sections for the various chart statements.

Unless indicated otherwise, the options listed here are available with every chart statement. For statements that create two charts, the term *primary chart* refers to the upper chart (for instance, the $\bar{X}$ chart created with the XRCHART statement), and the term *secondary chart* refers to the lower chart (for instance, the *R* chart created with the XRCHART statement). The term *primary chart* also refers to the single chart created by some statements (for instance, the *p* chart created with the PCHART statement).

The section "General Options" on page 1947 contains descriptions of general chart statement options, which are applicable regardless of the kind of graphics output you produce. The section "Options for ODS Graphics" on page 2005 describes options that apply only when ODS Graphics is enabled. The section "Options for Traditional Graphics" on page 2008 describes options that apply only when producing traditional graphics, as when ODS Graphics is disabled. The section "Options for Legacy Line Printer Charts" on page 2022 contains descriptions of options that apply only to legacy line printer charts, which are produced when the LINEPRINTER option is specified in the PROC SHEWHART statement.

General Options

ACTUALALPHA

requests that the actual probability of a point being outside an attribute chart's probability limits be displayed in the limits legend. This probability is based on the Poisson distribution for *c* and *u* charts; it is based on the binomial distribution for *np* and *p* charts.

Because attribute chart data are discrete, it is not possible in general to compute probability limits so that the probability of a point being outside the limits is α, for any α. Therefore, the specified and actual probabilities are usually different. The actual α is the sum of the probability of a point being below the lower control limit and the probability of a point being above the upper control limit.

This option is available only in the CCHART, NPCHART, PCHART, and UCHART statements. It applies only when you request probability limits by specifying the ALPHA= option and when the probability limits are constant. By default, the α value you specify in the ALPHA= option is displayed in the limits legend.

ALLLABEL=VALUE

ALLLABEL=(*variable*)

labels every point on the primary chart with the value plotted for that subgroup or with the value of *variable* in the input data set.

The *variable* provided in the input data set can be numeric or character. If the *variable* is a character variable, its length cannot exceed 16. For each subgroup of observations, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup. You should use this option with care to avoid cluttering the chart. By default, points are not labeled. Related options are CFRAMELAB=, OUTLABEL=, LABELFONT=, LABELHEIGHT=, and TESTLABEL=, but note that the OUTLABEL= option cannot be specified with the ALLLABEL= option.

ALLLABEL2=VALUE

ALLLABEL2=(*variable*)

labels every point on an *R*, *s*, or trend chart with the value plotted for that subgroup or with the value of *variable* in the input data set.

The *variable* provided in the input data set can be numeric or character. If the *variable* is a character variable, its length cannot exceed 16. For each subgroup of observations, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup. You should use this option with care to avoid cluttering the chart. By default, points are not labeled. Related options are CFRAMELAB=, OUTLABEL2=, LABELFONT=, LABELHEIGHT=, and TESTLABEL2=, but note that the OUTLABEL2= option cannot be specified with the ALLLABEL2= option. The option is available in the IRCHART, MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

ALLN

plots summary statistics for all subgroups, regardless of whether the subgroup sample size equals the nominal control limit sample size *n* specified by the LIMITN= option or the variable _LIMITN_ in the LIMITS= data set. Use the ALLN option in conjunction with the LIMITN= option or the variable _LIMITN_.

The ALLN option is useful in applications where almost all of the subgroups have a common sample size *n*, and you want to display fixed (rather than varying) control limits corresponding to the nominal sample size *n*. The disadvantage of using the ALLN option with widely differing subgroup sample sizes is that the interpretation of the control limits is meaningful only for those subgroups whose sample size is equal to *n*. To request special symbol markers indicating that not all the sample sizes are equal to *n*, use the NMARKERS option in conjunction with the ALLN option.

The ALLN option is not available in the IRCHART statement.

ALPHA=*value*

requests *probability limits*. If you specify ALPHA=α, the control limits are computed so that the probability is α that a subgroup summary statistic exceeds its control limits. This assumes that the process is in statistical control and that the data follow a certain theoretical distribution, which depends on the chart statement. The Poisson distribution is assumed for the CCHART and UCHART statements, and the binomial distribution is assumed for the NPCHART and PCHART statements. The normal distribution is assumed for all other chart statements. For the equations used to compute probability limits, see the "Details" subsection in the section for the chart statement that you are using.

The value of α can range between 0 and 1 for most statements. However, for the MCHART statement, the MRCHART statement, and the BOXCHART statement with the CONTROLSTAT=MEDIAN option, the value of α must be one of the following: 0.001, 0.002, 0.01, 0.02, 0.025, 0.04, 0.05, 0.10, or 0.20.

Note the following:

- As an alternative to specifying ALPHA=α, you can read α from the variable _ALPHA_ in a LIMITS= data set by specifying the READALPHA option. See "Input Data Sets" in the section for the chart statement in which you are interested.
- As an alternative to specifying ALPHA=α (or reading the variable _ALPHA_ from a LIMITS= data set), you can request "$k\sigma$ control limits" by specifying SIGMAS=*k* (or reading the variable _SIGMAS_ from a LIMITS= data set).

If you specify neither the ALPHA= option nor the SIGMAS= option, the procedure computes 3σ control limits by default.

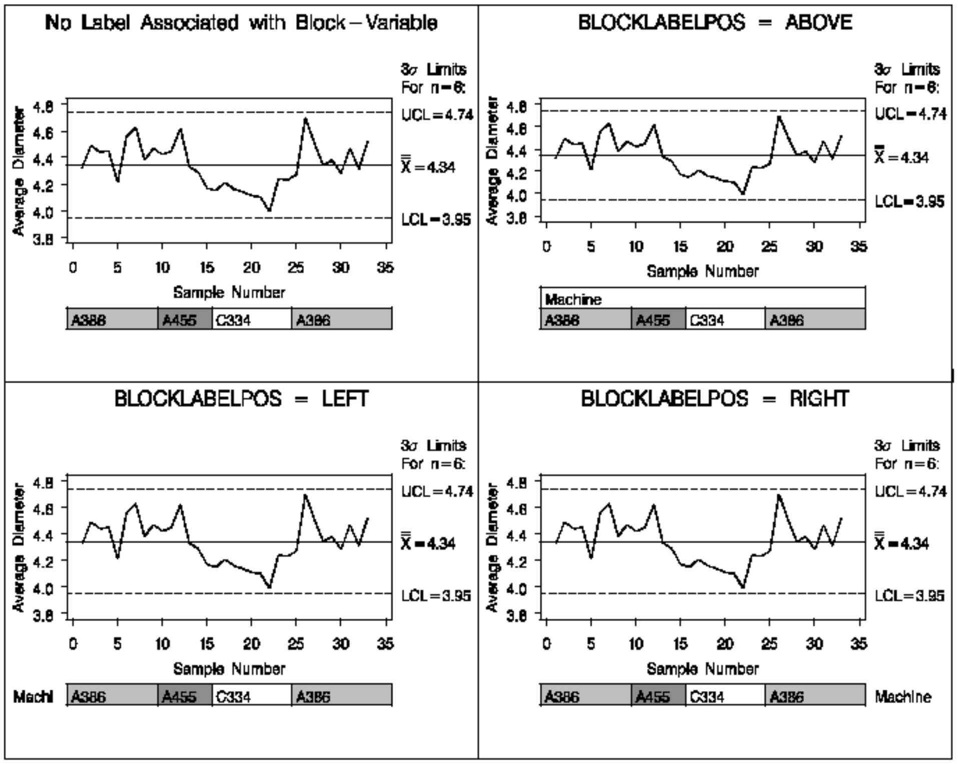
BLOCKLABELPOS=ABOVE | LEFT | RIGHT

specifies the position of a block-variable label in the block legend. You can specify the following keywords, which are illustrated in Figure 17.130:

ABOVE	places the label immediately above the legend
LEFT	places the label to the left of the legend
RIGHT	places the label to the right of the legend

Use the keywords **LEFT** and **RIGHT** with labels that are short enough to fit in the margins on each side of the chart; otherwise, they will be truncated. Use the keyword **RIGHT** only when the legend is below the control chart (BLOCKPOS=3 or BLOCKPOS=4). The default position is **ABOVE**. Related options are BLOCKLABTYPE=, BLOCKREP, BLOCKPOS=, CBLOCKVAR=, and CBLOCKLAB=.

Figure 17.130 Positions for *block-variable* Labels



BLOCKLABTYPE=SCALED | TRUNCATED | ROTATE | ROTATEALL

BLOCKLABTYPE=*height*

specifies how lengthy block variable values are treated when there is insufficient space to display them in the block legend. By default, lengthy values are not displayed.

If you specify the BLOCKLABTYPE=SCALED option, the values are uniformly reduced in height so that they fit. If you specify the BLOCKLABTYPE=TRUNCATED option, lengthy values are truncated on the right until they fit. When producing traditional graphics, you can also specify a text *height* in vertical percent screen units for the values. For ODS Graphics output, you can specify BLOCKLABTYPE=ROTATE to rotate the values of the block variable displayed closest to the chart by 90 degrees, and BLOCKLABTYPE=ROTATEALL to rotate the values of all block variables. Related options are BLOCKLABELPOS=, BLOCKREP, BLOCKPOS=, CBLOCKVAR=, and CBLOCKLAB=.

NOTE: In ODS Graphics output only BLOCKLABTYPE=TRUNCATED is supported.

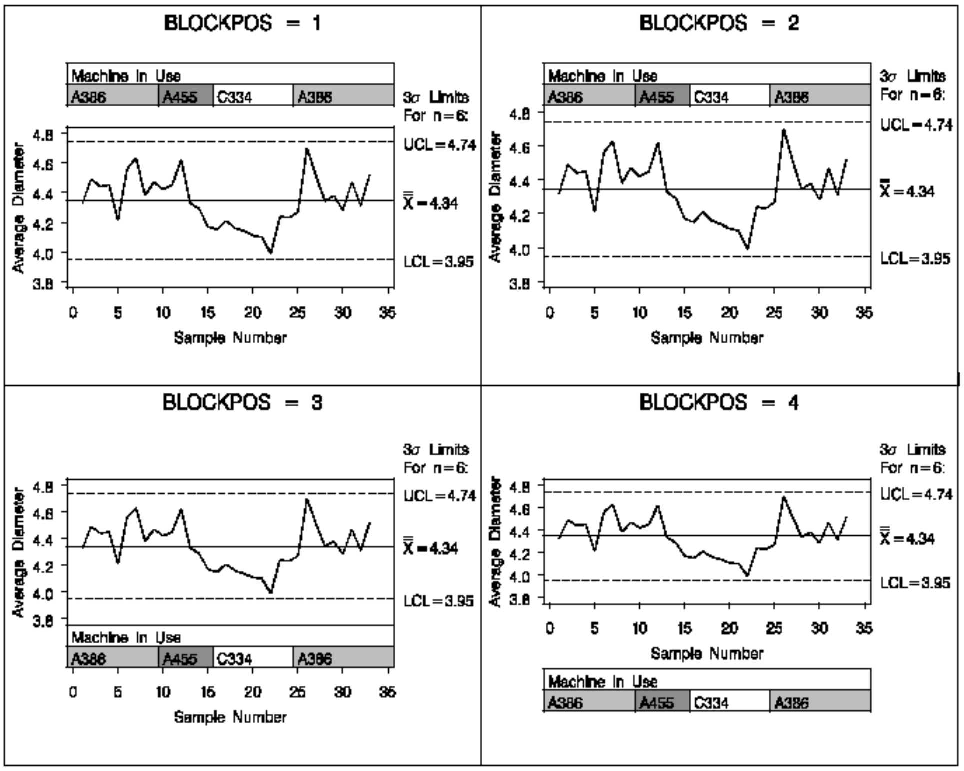
BLOCKPOS=*n*

specifies the vertical position of the legend for the values of the *block-variables* (see "Displaying Stratification in Blocks of Observations" on page 2026). Values of *n* and the corresponding positions are as follows. By default, BLOCKPOS=1.

n	**Legend Position**
1	Top of chart, offset from axis frame
2	Top of chart, immediately above axis frame
3	Bottom of chart, immediately above horizontal axis
4	Bottom of chart, below horizontal axis label

Figure 17.131 illustrates the various positions that can be specified.

Figure 17.131 Positions for *block-variable* Legends



Related options are BLOCKLABELPOS=, BLOCKLABTYPE=, BLOCKREP, CBLOCKVAR=, and CBLOCKLAB=.

BLOCKREP

specifies that block variable values for all subgroups are to be displayed. By default, only the first block variable value in any block is displayed, and repeated block variable values are not displayed. Related options are BLOCKLABELPOS=, BLOCKLABTYPE=, BLOCKPOS=, CBLOCKVAR=, and CBLOCKLAB=. For more information about block variables, see "Displaying Stratification in Blocks of Observations" on page 2026.

BLOCKVAR=*variable* **| (***variable-list***)**

specifies variables whose values are used to assign colors for filling the background of the legend associated with block variables. A list of BLOCKVAR= variables must be enclosed in parentheses. BLOCKVAR= variables are matched with block variables by their order in the respective variable lists. While the values of a CBLOCKVAR= variable are color names, values of a BLOCKVAR= variable are used to group block legends for assigning fill colors from the ODS style. Block legends with the same BLOCKVAR= variable value are filled with the same color.

BOXCONNECT

BOXCONNECT=MEAN | MEDIAN | MAX | MIN | Q1 | Q3

specifies that the points representing subgroup means, medians, maximum values, minimum values, first quartiles or third quartiles in box-and-whisker plots created with the BOXCHART statement are to be connected. If BOXCONNECT is specified without a keyword identifying the points to be connected, subgroup means are connected. By default, no points are connected. The BOXCONNECT option is available only in the BOXCHART statement.

BOXES=*variable*

specifies a variable whose values are used to assign colors for the outlines of box-and-whiskers plots. While the values of a CBOXES= variable are color names, values of the BOXES= variable are used to group box-and-whiskers plots for assigning outline colors from the ODS style. The outlines of box-and-whiskers plots of groups with the same BOXES= variable value are drawn using the same color.

BOXFILL=*variable* **| NONE | EMPTY**

specifies how box-and-whisker plots are filled with colors from the ODS style. You can specify a variable whose values are used to group box-and-whiskers plots for assigning fill colors from the ODS style. Boxes associated with groups having the same BOXFILL= variable value are filled with the same color. You can specify the keyword NONE or EMPTY to produce unfilled boxes. When producing traditional graphics, you can use the CBOXFILL= option to select specific colors for filling the boxes. By default, all boxes are filled with a single color from the ODS style.

BOXSTYLE=*keyword*

specifies the style of the box-and-whisker plots that are displayed for subgroup samples by the BOXCHART statement. You can specify the following keywords:

SKELETAL draws whiskers to the extreme values of the subgroup

SCHEMATIC draws whisker to the most extreme value within or equal to the lower/upper fence

SCHEMATICID labels outliers in schematic box-and-whisker plot

SCHEMATICIDFAR labels far outliers in schematic box-and-whisker plot

POINTS plots the values in a subgroup as points

POINTSJOIN plots the values in a subgroup as points joined with a vertical line

POINTSBOX plots the values in a subgroup as points enclosed in a box

POINTSID labels the points plotted in a subgroup

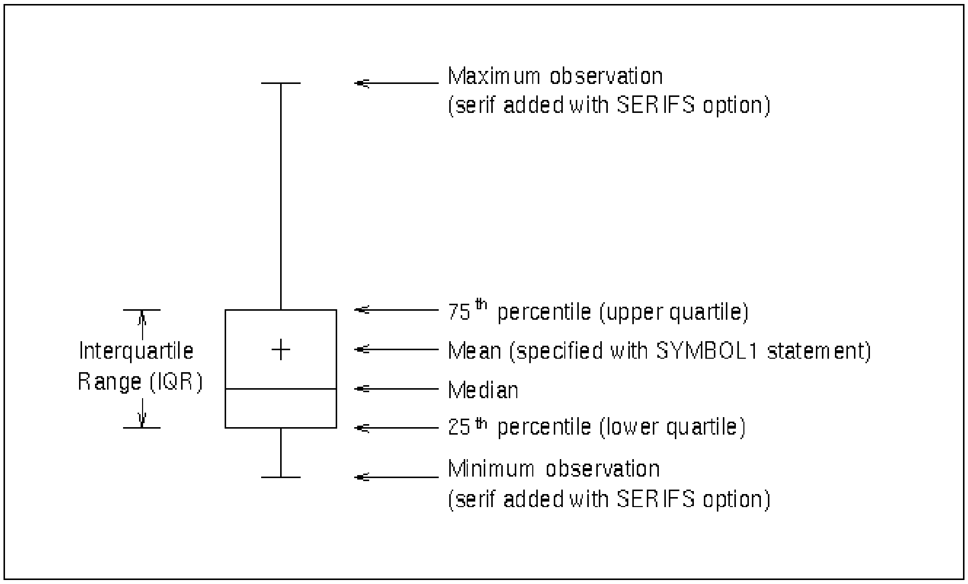
POINTSJOINID labels the points plotted in a subgroup joined by a vertical line

POINTSSCHEMATIC plots the values in a subgroup as points overlaid with a schematic box chart

The **SKELETAL**, **SCHEMATIC**, **SCHEMATICID**, and **SCHEMATICIDFAR** keywords are useful for creating conventional box-and-whisker displays. The keywords POINTS, POINTSJOIN, POINTSBOX, POINTSID, and POINTSJOINID are used to generalize the BOXSTYLE= option and, in particular, to facilitate the creation of so-called “multi-vari” charts, as illustrated in Output 17.7.2 and Output 17.7.3. The keyword POINTSSCHEMATIC combines the POINT and SCHEMATIC boxstyles.

If you specify BOXSTYLE=SKELETAL, the whiskers are drawn from the edges of the box to the extreme values of the subgroup sample. This plot is sometimes referred to as a *skeletal box-and-whisker plot*. By default, the whiskers are drawn without serifs, but you can add serifs with the SERIFS option. Figure 17.132 illustrates the elements of a typical skeletal boxplot.

Figure 17.132 BOXSTYLE= SKELETAL

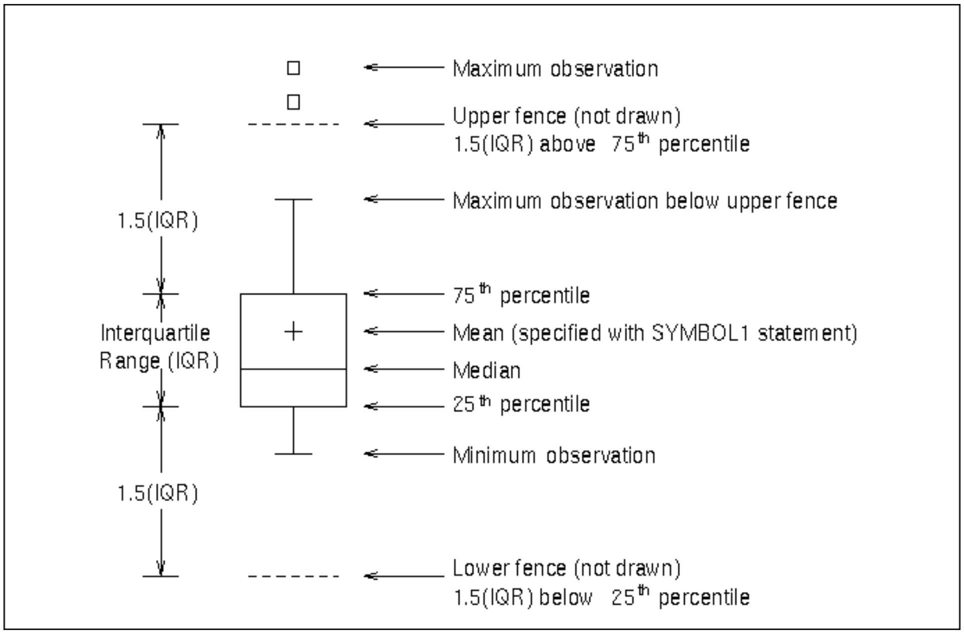


If you specify BOXSTYLE=SCHEMATIC, a whisker is drawn from the upper edge of the box to the largest value less than or equal to the upper fence and from the lower edge of the box to the smallest value greater than or equal to the lower fence. Figure 17.133 illustrates a typical schematic boxplot and the locations of the fences (which are not displayed in actual output). Serifs are added to the whiskers by default. Observations outside the fences are identified with a special symbol; you can specify the shape and color for this symbol with the IDSYMBOL= and IDCOLOR= options. The default symbol is a square. This type of plot corresponds to the *schematic box-and-whisker plot* described in Chapter 2 of Tukey (1977).

If you specify BOXSTYLE=SCHEMATICID, a schematic box-and-whisker plot is displayed in which the value of the first variable listed in the ID statement is used to label the symbol marking each observation outside the upper and lower fences.

If you specify BOXSTYLE=SCHEMATICIDFAR, a schematic box-and-whisker plot is displayed in which the value of the first variable listed in the ID statement is used to label the symbol marking each observation outside the *lower* and *upper far fences*. The lower and upper far fences are located 3×IQR below the 25th percentile and above the 75th percentile, respectively. Observations between the fences and the far fences are identified with a symbol but are not labeled with the ID variable.

Figure 17.133 BOXSTYLE= SCHEMATIC



NOTE: To make side-by-side box charts (as opposed to a control chart with subgroup box plots), you should use the BOXCHART statement with the NOLIMITS option in addition to the BOXSTYLE= option.

If you specify BOXSTYLE=POINTS, all the values in the subgroup sample are plotted as points, and neither a box nor whiskers are drawn. By default, a square plotting symbol is used for the values. You can specify a symbol with the IDSYMBOL= option. You can specify the color of the symbols with the IDCOLOR= option (the default color is the color specified with the CBOXES= option).

If you specify BOXSTYLE=POINTSJOIN, all the values in the subgroup sample are plotted as points joined with a vertical line. Neither a box nor whiskers are drawn. See Output 17.7.2 for an illustration. By default, a square plotting symbol is used for the values. You can specify a symbol with the IDSYMBOL= option, and you can specify the color of the symbol with the IDCOLOR= option. You can specify the color of the vertical line with the CBOXES= option.

If you specify BOXSTYLE=POINTSBOX, all the values in the subgroup sample are plotted as points enclosed in a box. By default, a square plotting symbol is used for the values. You can specify a symbol with the IDSYMBOL= option, and you can specify the color of the symbol with the IDCOLOR= option. You can specify the color of the box with the CBOXES= option, the fill color of the box with the CBOXFILL= option, and the line type of the box with the LBOXES= option.

If you specify BOXSTYLE=POINTSID, all the values in the subgroup sample are plotted using labels specified as the values of the first variable in the ID statement. See Output 17.7.3 for an illustration. It is recommended that you use single-character labels. You can specify a font for the labels with the IDFONT= option. You can specify the height of the labels with the IDHEIGHT= option. You can specify the color of the labels with the IDCTEXT= option.

If you specify BOXSTYLE=POINTSJOINID, all the values in the subgroup sample are plotted using labels specified as the values of the first variable in the ID statement, and the values are joined by a vertical line. It is recommended that you use single-character labels. You can specify a font for the labels with the IDFONT= option. You can specify the height of the labels with the IDHEIGHT= option. You can specify the color of the labels with the IDCTEXT= option, and you can specify the color of the vertical line with the CBOXES= option.

If you specify BOXSTYLE=POINTSSCHEMATIC, a schematic box chart is overlaid with points plotting all observations in the subgroups.

The BOXSTYLE= option is available only in the BOXCHART statement; see Example 17.2. The styles SCHEMATIC, SCHEMATICID, and SCHEMATICIDFAR are available only when the input data set is a DATA= data set. By default, BOXSTYLE= SKELETAL. Related options include BOXWIDTH=, BOXWIDTHSCALE=, IDCOLOR=, and IDSYMBOL=.

Note that the keywords POINTS, POINTSJOIN, POINTSBOX, POINTSID, and POINTSJOINID for the BOXSTYLE= option can be used in conjunction with the CPHASEBOX=, CPHASEBOXFILL=, CPHASEBOXCONNECT=, CPHASEMEANCONNECT=, and PHASEMEANSYMBOL= options to create “multi-vari” displays.

BOXWIDTH=*value*

specifies the width of box-and-whisker plots created with the BOXCHART statement. For traditional graphics, the width is specified in horizontal percent screen units. For ODS Graphics output, the width is specified in pixels. The default width is chosen so that the boxes are as wide as possible without colliding. You should use the BOXWIDTH= option in situations where the number of subgroups per panel is very small and you want to reduce the width. The BOXWIDTH= option is available only in the BOXCHART statement.

BOXWIDTHSCALE=*value*

specifies that the width of box-and-whisker plots created with the BOXCHART statement is to vary proportionately to a particular function of the subgroup sample size *n*. The function is determined by the *value* and is identified on the chart with a legend.

If you specify a positive *value*, the widths are proportional to n^{value}. In particular, if you specify BOXWIDTHSCALE=1, the widths are proportional to the sample size. If you specify BOXWIDTH-SCALE=0.5, the widths are proportional to $\sqrt{n}$, as described by McGill, Tukey, and Larsen (1978). If you specify BOXWIDTHSCALE=0, the widths are proportional to $\log(n)$. See Example 17.4 for an illustration of the BOXWIDTHSCALE= option.

By default, the box widths are constant. The BOXWIDTHSCALE= option is available only in the BOXCHART statement.

CFRAMELAB=*color*

CFRAMELAB

specifies the color for filling rectangles that frame the point labels displayed with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and OUTLABEL2= options. Specify CFRAMELAB with no argument to produced unfilled frames. By default, the points are not framed.

CIINDICES < (< TYPE=*keyword* > < ALPHA=*value* >) >

requests capability index confidence limits based on subgroup summary data, calculated using “effective degrees of freedom” as described by Bissell (1990). These confidence limits are approximate. When you specify the CIINDICES option, the calculated confidence limits are available for display in an inset and are included in the OUTLIMITS= data set, if one is produced.

TYPE=*keyword*

specifies the type of confidence limit. Valid values are LOWER, UPPER and TWOSIDED. The default value is TWOSIDED.

ALPHA=*value*

specifies the default confidence level to compute confidence limits. The percentage for the confidence limits is $(1 - value) * 100$. For example, ALPHA=.05 results in a 95% confidence limit. The default value is .05 and the possible range of values is from 0 to 1.

CINFILL=*color* **| EMPTY | NONE**

specifies the color for the area inside the upper and lower control limits. By default, this area filled with an appropriate color from the ODS style. You can specify the keyword EMPTY or NONE to leave the area between the control limits unfilled. See also the COUTFILL= option.

CLIPFACTOR=*factor*

requests clipping of extreme points on the control chart. The *factor* that you specify determines the extent to which these values are clipped, and it must be greater than one (useful values are in the range 1.5 to 2).

For examples of the CLIPFACTOR= option, see Figure 17.170 and Figure 17.171. The CLIPFACTOR= option should not be used in any statement in which the STARVERTICES= option is also used. Related clipping options are CCLIP=, CLIPCHAR=, CLIPLEGEND=, CLIPLEGPOS=, CLIPSUBCHAR=, and CLIPSYMBOL=.

CLIPLEGEND='*label*'

specifies the *label* for the legend that indicates the number of clipped points when the CLIPFACTOR= option is used. The *label* must be no more than 16 characters and must be enclosed in quotes. For an example, see Figure 17.171.

CLIPSUBCHAR='*character*'

specifies a substitution character (such as #) for the label provided with the CLIPLEGEND= option. The substitution character is replaced with the number of points that are clipped. For example, suppose that the following statements produce a chart in which three extreme points are clipped:

```
proc shewhart data=Pistons;
   xrchart Diameter*Hour /
      clipfactor  = 1.5
      cliplegend  = 'Points clipped=#'
      clipsubchar = '#' ;
run;
```

Then the clipping legend displayed on the chart will be

```
Points clipped=3
```

CONTROLSTAT=MEAN | MEDIAN

specifies whether the control limits displayed in a box chart are computed for subgroup means or for subgroup medians. By default, CONTROLSTAT=MEAN. The CONTROLSTAT= option is available only in the BOXCHART statement.

COUT=*color*

COUT

specifies the color for the plotting symbols and the portions of connecting line segments that lie outside the control limits. Specify COUT with no argument to use an appropriate contrasting color from the ODS style. This option is useful for highlighting out-of-control subgroups.

When ODS Graphics is enabled and the BOXCHART statement or STARVERTICES= option is used, COUT highlights the boxes or stars whose subgroup values fall outside the control limits.

CPHASEBOX=*color*

CPHASEBOX

PHASEBOX

specifies the color for a box that encloses all of the plotted points for a phase (group of consecutive observations that have the same value of the variable _PHASE_). Specify CPHASEBOX or PHASEBOX with no argument to request phase boxes drawn using an appropriate color from the ODS style. By default, an enclosing box is not drawn. This option is available only in the BOXCHART statement.

CPHASEBOXCONNECT=*color*

CPHASEBOXCONNECT

PHASEBOXCONNECT

specifies the color for line segments that connect the vertical edges of adjacent enclosing boxes requested with the CPHASEBOX= option or the CPHASEBOXFILL= option. The vertical coordinates of the attachment points represent the average of the values plotted inside the box. The CPHASEBOXCONNECT= option is an alternative to the CPHASEMEANCONNECT= option. Specify CPHASEBOXCONNECT or PHASEBOXCONNECT with no argument to connect the phase boxes with lines drawn in an appropriate color from the ODS style. This option is available only in the BOXCHART statement.

CPHASEBOXFILL=*color*

CPHASEBOXFILL

PHASEBOXFILL

specifies the fill color for a box that encloses all of the plotted points for a phase. Specify CPHASEBOXFILL or PHASEBOXFILL with no argument to fill the phase boxes with an appropriate color from the ODS style. By default, an enclosing box is not drawn. This option is available only in the BOXCHART statement.

CPHASEMEANCONNECT=*color*

CPHASEMEANCONNECT

PHASEMEANCONNECT

specifies the color for line segments that connect points representing the average of the values plotted within a phase. This option must be used in conjunction with the CPHASEBOX= or CPHASEBOXFILL= options, and it is an alternative to the CPHASEBOXCONNECT= option. The points are centered horizontally within the enclosing boxes. Specify CPHASEMEANCONNECT or PHASEMEANCONNECT with no argument to connect phase means with lines drawn in an appropriate color from the ODS style. This option is available only in the BOXCHART statement.

CSTAROUT=*color*

CSTAROUT

specifies a color for those portions of the outlines of stars (requested with the STARVERTICES= option) that exceed the inner or outer circles. This option applies only with the STARTYPE=RADIAL and STARTYPE=SPOKE options, and it is useful for highlighting extreme values of star vertex variables. Specify CSTAROUT with no argument to use an appropriate contrasting color from the ODS style. See "Displaying Auxiliary Data with Stars" on page 2042.

CSYMBOL='*label***'**

CSYMBOL=C | CBAR | CPM | CPM2 | C0

specifies a label for the central line in a *c* chart. You can use the option in two ways:

- You can specify a quoted *label* of length 16 or less.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol*=*value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

	Symbol Used in	
Keyword	**Graphics**	**Line Printer Charts**
C	C	C
CBAR	$\overline{C}$	$\overline{C}$
CPM	C'	C'
CPM2	C''	C''
C0	C_0	C0

See Example 17.9 for an example. The default keyword is CBAR. The CSYMBOL= option is available only in the CCHART statement.

DATAUNIT=PERCENT | PROPORTION

enables you to use percents or proportions as the values for *processes* when you are using the PCHART or NPCHART statements and reading a DATA= input data set. Specify DATAUNIT=PERCENT to indicate that the values are percents of nonconforming items. Specify DATAUNIT=PROPORTION to indicate that the values are proportions of nonconforming items. Values for percents can range from 0 to 100, while values for proportions can range from 0 to 1. By default, the values of *processes* read from a DATA= data set for PCHART and NPCHART statements are assumed to be numbers (counts) of nonconforming items. The DATAUNIT= option is available only in the NPCHART and PCHART statements.

DISCRETE

specifies that numeric subgroup variable values be treated as discrete values, so that each tick value on the default subgroup axis corresponds to a unique subgroup variable value. By default, a *continuous* subgroup axis is created, and if the subgroup variable values are not evenly spaced, the axis contains ticks with no corresponding subgroup data.

EXCHART

creates a control chart only when exceptions occur, specifically, when the control limits are exceeded or when any of the tests requested with the TESTS= option or the TESTS2= option are positive.

FRONTREF

draws reference lines specified with the HREF= and VREF= options in front of box-and-whiskers plots. By default, reference lines are drawn behind the box-and-whiskers plots and can be obscured by filled boxes.

GRID

adds a grid to the control chart. Grid lines are horizontal and vertical lines positioned at labeled major tick marks, and they cover the length and height of the plotting area.

HAXIS=*values*

HAXIS=AXIS*n*

specifies tick mark values for the horizontal (subgroup) axis. If the subgroup variable is numeric, the *values* must be numeric and equally spaced. Numeric values can be given in an explicit or implicit list. If the subgroup variable is character, *values* must be quoted strings of length 32 or less. If a date, time, or datetime format is associated with a numeric subgroup variable, SAS datetime literals can be used. Examples of HAXIS= lists follow:

```
haxis=0 2 4 6 8 10
haxis=0 to 10 by 2
haxis='LT12A' 'LT12B' 'LT12C' 'LT15A' 'LT15B' 'LT15C'
haxis='20MAY88'D to '20AUG88'D by 7
haxis='01JAN88'D to '31DEC88'D by 30
```

If the subgroup variable is numeric, the HAXIS= list must span the subgroup variable values, and if the subgroup variable is character, the HAXIS= list must include all of the subgroup variable values. You can add subgroup positions to the chart by specifying HAXIS= values that are not subgroup variable values.

If you specify a large number of HAXIS= values, some of these may be thinned to avoid collisions between tick mark labels. To avoid thinning, use one of the following methods:

- Shorten values of the subgroup variable by eliminating redundant characters. For example, if your subgroup variable has values LOT1, LOT2, LOT3, and so on, you can use the SUBSTR function in a DATA step to eliminate "LOT" from each value, and you can modify the horizontal axis label to indicate that the values refer to lots.
- Use the TURNHLABELS option to turn the labels vertically.
- Use the NPANELPOS= option to force fewer subgroup positions per panel.

If you are producing traditional graphics, you can also specify a previously defined AXIS statement with the HAXIS= option.

HOFFSET=*value*

specifies the length of the offset at each end of the horizontal axis. For traditional graphics, the offset is specified in percent screen units. For ODS Graphics output, the offset is specified in pixels. You can eliminate the offset by specifying HOFFSET=0.

HREF=*values*

HREF=*SAS-data-set*

draws reference lines perpendicular to the horizontal (subgroup) axis on the primary chart. You can use this option in the following ways:

- You can specify the *values* for the lines with an HREF= list. If the subgroup variable is numeric, the *values* must be numeric. If the subgroup variable is character, the *values* must be quoted strings of up to 32 characters. If the subgroup variable is formatted, the *values* must be given as internal values.

  Examples of HREF=*values* follow:

  ```
  href=5
  href=5 10 15 20 25 30
  href='Shift 1' 'Shift 2' 'Shift 3'
  ```

- You can specify the values for the lines as the values of a variable named _REF_ in an HREF= data set. The type and length of _REF_ must match those of the *subgroup variable* specified in the chart statement. Optionally, you can provide labels for the lines as values of a variable named _REFLAB_, which must be a character variable of length 16 or less. If you want distinct reference lines to be displayed in charts for different *processes* specified in the chart statement, you must include a character variable of length 32 or less named _VAR_, whose values are the *processes*. If you do not include the variable _VAR_, all of the lines are displayed in all of the charts.

  Each observation in the HREF= data set corresponds to a reference line. If BY variables are used in the input data set (DATA=, HISTORY=, or TABLE=), the same BY variable structure must be used in the HREF= data set unless you specify the NOBYREF option.

Related options are CHREF=, HREFCHAR=, HREFLABELS=, HREFLABPOS=, LHREF=, and NOBYREF.

HREF2=*values*

HREF2=*SAS-data-set*

draws reference lines perpendicular to the horizontal (subgroup) axis on the secondary chart. The conventions for specifying the HREF2= option are identical to those for specifying the HREF= option. Related options are CHREF=, HREFCHAR=, HREF2LABELS=, HREFLABPOS=, LHREF=, and NOBYREF. The HREF2= option is available only in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

HREF2DATA=*SAS-data-set*

draws reference lines perpendicular to the horizontal (subgroup) axis on the secondary chart. The HREF2DATA= option must be used in place of the HREF2= option to specify a data set using the quoted filename notation.

HREF2LABELS=‘*label1***’ . . . ‘***labeln***’**

HREF2LABEL=‘*label1***’ . . . ‘***labeln***’**

HREF2LAB=‘*label1***’ . . . ‘***labeln***’**

specifies labels for the reference lines requested by the HREF2= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters. The HREF2LABELS= option is available only in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

HREFDATA=*SAS-data-set*
draws reference lines perpendicular to the horizontal (subgroup) axis on the primary chart. The HREFDATA= option must be used in place of the HREF= option to specify a data set using the quoted filename notation.

HREFLABELS='*label1*' **. . .** '*labeln*'
HREFLABEL='*label1*' **. . .** '*labeln*'
HREFLAB='*label1*' **. . .** '*labeln*'
specifies labels for the reference lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

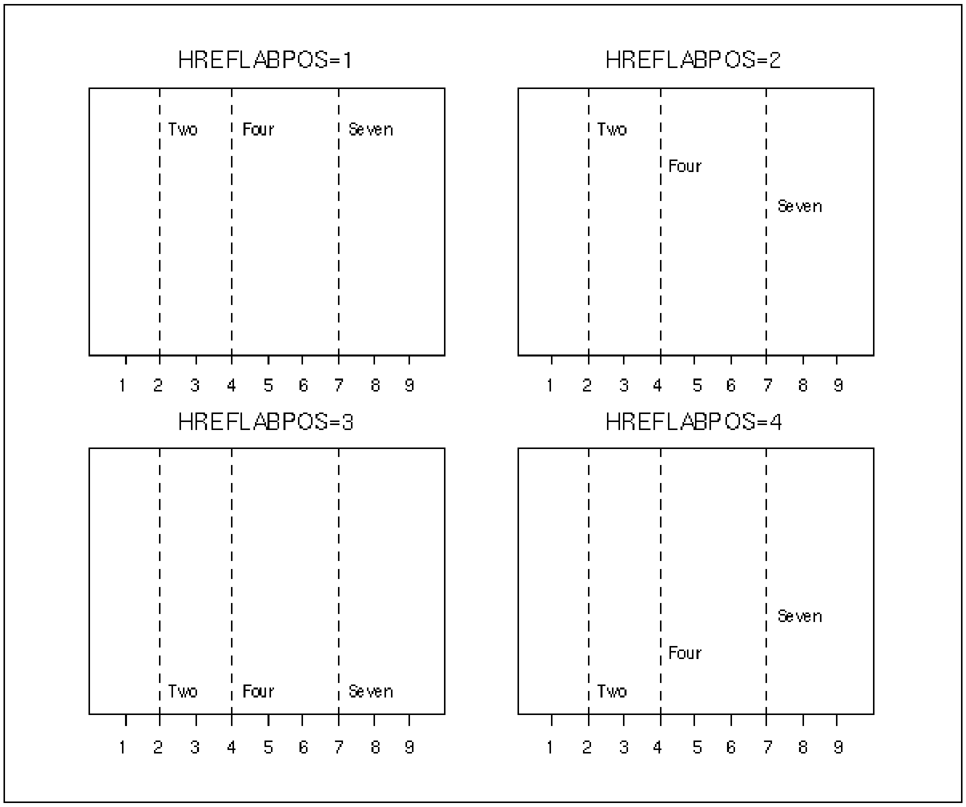
HREFLABPOS=*n*
specifies the vertical position of the HREFLABEL= and HREF2LABEL= labels, as described in the following table. By default, *n* = 2.

n	**Position**
1	along top of subplot area
2	staggered from top to bottom of subplot area
3	along bottom of subplot area
4	staggered from bottom to top of subplot area

Figure 17.134 illustrates label positions for values of the HREFLABPOS= option when the HREF= and HREFLABELS= options are as follows:

```
href       = 2 4 7
hreflabels = 'Two' 'Four' 'Seven'
```

Figure 17.134 Positions for Reference Line Labels



INDEPENDENTZONES

INDEPZONES

specifies that the widths of the zones requested with the ZONES option be computed independently above and below the center line of the chart, so that the width of each zone is one-third of the difference between the process mean and the control limit on its side of the chart. By default, the width of all zones is one-third of the difference between the upper control limits and the process mean, with zones below the center line truncated if necessary. The INDEPENDENTZONES option has no effect when the control limits are symmetric.

INTERVAL=DAY | DTDAY | HOUR | MINUTE | MONTH | QTR | SECOND

specifies the natural time interval between consecutive subgroup positions when a time, date, or datetime format is associated with a numeric subgroup variable. By default, the INTERVAL= option uses the number of subgroup positions per panel that you specify with the NPANELPOS= option. The default time interval keywords for various time formats are shown in the following table.

Format	Default Keyword	Format	Default Keyword
DATE	DAY	MONYY	MONTH
DATETIME	DTDAY	TIME	SECOND
DDMMYY	DAY	TOD	SECOND
HHMM	HOUR	WEEKDATE	DAY
HOUR	HOUR	WORDDATE	DAY
MMDDYY	DAY	YYMMDD	DAY
MMSS	MINUTE	YYQ	QTR

You can use the INTERVAL= option to modify the effect of the NPANELPOS= option, which specifies the number of subgroup positions per panel (screen or page). The INTERVAL= option enables you to match the scale of the horizontal axis to the scale of the subgroup variable without having to associate a different format with the subgroup variable.

For example, suppose your formatted subgroup values span an overall time interval of 100 days and a DATETIME format is associated with the subgroup variable. Since the default interval for the DATETIME format is DTDAY and since NPANELPOS=50 by default, the chart is displayed with two panels (screens or pages).

Now, suppose your data span an overall time interval of 100 hours and a DATETIME format is associated with the subgroup variable. The chart for these data are created in a single panel, but the data occupy only a small fraction of the chart since the scale of the data (hours) does not match that of the horizontal axis (days). If you specify INTERVAL=HOUR, the horizontal axis is scaled for 50 hours, matching the scale of the data, and the chart is displayed with two panels.

INTSTART=*value*

specifies the starting value for a numeric horizontal axis, when a date, time, or datetime format is associated with the subgroup variable. If the value specified is greater than the first subgroup variable value, this option has no effect.

LCLLABEL='*label*'

specifies a label for the lower control limit in the primary chart. The label can be of length 16 or less. Enclose the label in quotes. The default label is of the form *LCL=value* if the control limit has a fixed value; otherwise, the default label is *LCL*. Related options are LCLLABEL2=, UCLLABEL=, and UCLLABEL2=.

LCLLABEL2='*label*'

specifies a label for the lower control limit in the secondary chart. The label can be of length 16 or less. Enclose the label in quotes. The default label is of the form *LCL=value* if the control limit has a fixed value; otherwise, the default label is *LCL*. The LCLLABEL2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements. Related options are LCLLABEL=, UCLLABEL=, and UCLLABEL2=.

LIMITN=*n*

LIMITN=VARYING

specifies either a fixed or varying nominal sample size for the control limits.

If you specify LIMITN=*n*, the control limits are computed for the fixed value *n*, and they do not vary with the subgroup sample sizes. Moreover, subgroup summary statistics are plotted *only* for those subgroups with a sample size equal to *n*. You can specify ALLN in conjunction with LIMITN=*n* to force all of the statistics to be plotted, regardless of subgroup sample size.

If you do not specify LIMITN=*n* and the subgroup sample sizes are constant, the default value of *n* is the constant subgroup sample size.

Depending on the chart statement, there are restrictions on the value of *n* that you can specify with the LIMITN= option. For the MRCHART, RCHART, and XRCHART statements, $2 \le n \le 25$. For the SCHART and XSCHART statements, $n \ge 2$. For the BOXCHART, MCHART, and XCHART statements, $n \ge 1$. If you omit the STDDEVIATIONS option for the MCHART or XCHART statements (or use the RANGES option with the BOXCHART statement) $n < 26$. For the CCHART and UCHART statements, $n > 0$, and *n* can assume fractional values (for all other chart statements, *n* must be a whole number). For the PCHART and NPCHART statements, $n \ge 1$.

For the IRCHART statement, n has a somewhat different interpretation; it specifies the number of consecutive measurements from which the moving ranges are to be computed, and $n \ge 2$. You can think of n as a *pseudo* nominal sample size for the control limits, since the data for an individual measurements and moving range chart are not subgrouped.

Note the difference between the LIMITN= option and the SUBGROUPN= option that is available in the CCHART, NPCHART, PCHART, and UCHART statements. The LIMITN= option specifies a nominal sample size for the *control limits*, whereas the SUBGROUPN= option provides the sample sizes for the *data*.

By default, LIMITN=2 in an IRCHART statement. You cannot specify LIMITN= VARYING in an IRCHART statement. For all other chart statements, LIMITN= VARYING is the default.

The following table identifies the chart features that vary when you use LIMITN= VARYING:

Chart Statement	**Features Affected by LIMITN=VARYING**
BOXCHART	control limits
CCHART	control limits, central line
MCHART	control limits
MRCHART	control limits on both charts, central line on *R* chart
NPCHART	control limits, central line
PCHART	control limits
RCHART	control limits, central line
SCHART	control limits, central line
UCHART	control limits
XCHART	control limits
XRCHART	control limits on both charts, central line on *R* chart
XSCHART	control limits on both charts, central line on *s* chart

NOTE: As an alternative to specifying the LIMITN= option, you can read the nominal control limit sample size from the variable _LIMITN_ in a LIMITS= data set. See “Input Data Sets” in the section for the chart statement in which you are interested.

LIMLABSUBCHAR='*character*'

specifies a substitution character (such as #) for labels provided as quoted strings with the LCLLABEL=, LCLLABEL2=, UCLLABEL=, UCLLABEL2=, CSYMBOL=, NPSYMBOL=, PSYMBOL=, RSYMBOL=, SSYMBOL=, USYMBOL=, and XSYMBOL= options. The substitution character must appear in the label. When the label is displayed on the chart, the character is replaced with the value of the corresponding control limit or center line, provided that this value is constant across subgroups. Otherwise, the default label for a varying control limit or center line is displayed.

LSL=*value-list*

provides lower specification limits used to compute capability indices. If you provide more than one *value*, the number of *values* must match the number of *processes* listed in the chart statement. If you specify only one *value*, it is used for all the *processes*.

The SHEWHART procedure uses the specification limits to compute capability indices, and it saves the limits and indices in the OUTLIMITS= data set. For more information, see "Capability Indices" on page 1874 and "Output Data Sets" in the section for the chart statement in which you are interested. Also see the entry for the USL= option. The LSL= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements.

LTMARGIN=*value*

LTM=*value*

specifies the width of the left marginal area for the plot requested with the LTMPLOT= option. For traditional graphics, the width is specified in horizontal percent screen units. For ODS Graphics output, the width is specified in pixels. The LTMARGIN= option is available only in the IRCHART statement.

LTMPLOT=*keyword*

requests a univariate plot of the control chart statistics that is positioned in the left margin of the control chart. The keywords that you can specify and the associated plots are listed in the following table:

Keyword	Marginal Plot
HISTOGRAM	histogram
DIGIDOT	digidot plot
SKELETAL	skeletal box-and-whisker plot
SCHEMATIC	schematic box-and-whisker plot
SCHEMATICID	schematic box-and-whisker plot with outliers labeled
SCHEMATICIDFAR	schematic box-and-whisker plot with far outliers labeled

NOTE: Digidot plots are not available in ODS Graphics output.

The LTMPLOT= option is available only in the IRCHART statement; see Example 17.13 for an example. Refer to Hunter (1988) for a description of digidot plots, and see the entry for the BOXSTYLE= option for a description of the various box-and-whisker plots. Related options are LTMARGIN=, RTMARGIN=, and RTMPLOT=.

MAXPANELS=*n*

specifies the maximum number of pages or screens for a chart. By default, $n = 20$.

MEDCENTRAL=AVGMEAN | AVGMED | MEDMED

identifies a method for estimating the process mean μ, which is represented by the central line on a median chart. The methods corresponding to each keyword are given in the following table:

Keyword	Method for Estimating Process Mean
AVGMEAN	average of subgroup means
AVGMED	average of subgroup medians
MEDMED	median of subgroup medians

The default keyword is AVGMED. The MEDCENTRAL= option is available only in the MCHART and MRCHART statements and in the BOXCHART statement with the CONTROLSTAT=MEDIAN option.

MISSBREAK

determines how subgroups are formed when observations are read from a DATA= data set and a character *subgroup-variable* is provided. When you specify the MISSBREAK option, observations with missing values of the *subgroup variable* are not processed. Furthermore, the next observation with a nonmissing value of the *subgroup-variable* is treated as the beginning observation of a new subgroup even if this value is identical to the most recent nonmissing subgroup value. In other words, by specifying the option MISSBREAK and by inserting an observation with a missing *subgroup-variable* value into a group of consecutive observations with the same *subgroup-variable* value, you can split the group into two distinct subgroups of observations.

By default, if MISSBREAK is not specified, observations with missing values of the *subgroup variable* are not processed, and all remaining observations with the same consecutive value of the *subgroup-variable* are treated as a single subgroup.

MRRESTART

MRRESTART=*value*

causes the moving range computation on the IRCHART to be restarted when a missing value is encountered. Without the MRRESTART option, a missing value is simply skipped, and the moving range for the next non-missing subgroup is computed using the most recent previous non-missing value. MRRESTART restarts the moving range computation, so only the observations after the missing value are used in subsequent moving range computations. MRRESTART restarts the moving range computation on any missing value; you can also specify MRRESTART=value to restart only on a particular missing value. For example, MRRESTART=R will restart the computation only when the missing value ".R" is encountered.

MU0=*value*

specifies a known (standard) value μ_0 for the process mean μ. By default, μ is estimated from the data. The MU0= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, XCHART, XRCHART, and XSCHART statements.

NOTE: As an alternative to specifying MU0=μ_0, you can read a predetermined value for μ_0 from the variable _MEAN_ in a LIMITS= data set. See "Input Data Sets" in the section for the chart statement in which you are interested.

NDECIMAL=*n*

specifies the number of decimal digits in the default labels for the control limits and the central line in the primary chart. The default is one more than the maximum number of decimal digits in the vertical axis tick mark labels. For example, if the vertical axis tick mark label with the largest number of digits after the decimal point is 110.05, the default is $n = 3$.

NDECIMAL2=*n*

specifies the number of decimal digits in the default labels for the control limits and central line in a secondary chart. The default is one more than the maximum number of decimal digits in the vertical axis tick mark labels. The NDECIMAL2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NEEDLES

connects plotted points to the central line with vertical line segments (needles). See Example 17.19 for an example. By default, adjacent points are connected to one another. The NEEDLES option is available in all chart statements except the BOXCHART statement.

NMARKERS

identifies a plotted subgroup summary statistic with a special symbol marker (character) when the corresponding subgroup sample size is not equal to the nominal control limit sample size n. Specify the nominal control limit sample size n with the LIMITN= option or with the variable _LIMITN_ read from a LIMITS= data set. The following table summarizes the identification:

Sample Size	Graphics Symbol	Line Printer Character
$< n$	▽	L
$> n$	△	G

A legend that explains the symbols is displayed at the bottom of the chart. This legend can be suppressed with the NOLEGEND option.

The NMARKERS option is not available in the IRCHART statement. The NMARKERS option applies only when specified in conjunction with the ALLN option and a fixed nominal control limit sample size provided with the LIMITN= option or the variable _LIMITN_. See Example 17.40 for an illustration.

NO3SIGMACHECK

suppresses the check for 3σ limits when tests for special causes are requested. This enables tests for special causes to be applied when the SIGMAS= option is used to specify control limits other than the default 3σ limits. This option should not be used for standard control chart applications, since the standard tests for special causes assume 3σ limits.

NOBYREF

specifies that the reference line information in an HREF=, HREF2=, VREF=, or VREF2= data set is to be applied uniformly to charts created for all the BY groups in the input data set (DATA=, HISTORY=, or TABLE=). If you specify the NOBYREF option, you do not need to provide BY variables in the reference line data set. By default, you must provide BY variables.

NOCHART

suppresses the creation of the chart. You typically specify the NOCHART option when you are using the procedure to compute control limits and save them in an output data set. You can also use the NOCHART option when you are tabulating results with the TABLE and related options.

In the IRCHART, MRCHART, XRCHART, and XSCHART statements, the NOCHART option suppresses the creation of both the primary and secondary charts. If you are producing traditional graphics and specify the NOCHART option, the chart is not saved in a graphics catalog. To save the chart in a graphics catalog while suppressing the display of the chart, specify the NODISPLAY option in a GOPTIONS statement.

NOCHART2

suppresses the creation of a secondary chart. You typically use this option in the IRCHART statement to create a chart for individual measurements and suppress the accompanying chart for moving ranges. The NOCHART2 option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NOCONNECT

suppresses line segments that connect points on the chart. By default, points are connected except in box charts produced with the BOXCHART statement (see the BOXCONNECT option).

NOCTL

suppresses the display of the central line in a primary chart.

NOCTL2

suppresses the display of the central line in a secondary chart. The NOCTL2 option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NOHLABEL

suppresses the label for the horizontal (subgroup) axis. Use the NOHLABEL option when the meaning of the axis is evident from the tick mark labels, such as when a date format is associated with the subgroup variable.

NOLCL

suppresses the display of the lower control limit in a primary chart.

NOLCL2

suppresses the drawing of the lower control limit in a secondary chart. The NOLCL2 option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NOLEGEND

suppresses the default legend for subgroup sample sizes, which appears by default below the chart. This option also suppresses the legend displayed by the NMARKERS option. Use the NOLEGEND option when the subgroup sample sizes are constant and equal to the control limit sample size, since the control limit sample size is displayed in the upper right corner of the chart.

NOLIMIT0

suppresses the display of a fixed lower control limit if and only if the value of the limit is zero. This option is useful in situations where a lower limit of zero is considered to be uninformative or visually distracting (for instance, on certain p charts or R charts). The NOLIMIT0 option is available with all chart statements except BOXCHART, MCHART, and XCHART. For the IRCHART, MRCHART, XRCHART, and XSCHART statements, the NOLIMIT0 option applies only to the secondary chart.

NOLIMIT1

suppresses the display of a fixed upper control limit on a *p* chart if and only if the value of the control limit is 1 (or 100%), or on an *np* chart if and only if the value of the control limit is *n*. The NOLIMIT1 option is available only in the NPCHART and PCHART statements.

NOLIMITLABEL

suppresses the default labels for the control limits and central lines.

NOLIMITS

suppresses the display of control limits. This option is particularly useful if you are using the BOXCHART statement to create side by side box-and-whisker plots; in this case, you should also use one of the BOXSTYLE= options.

NOLIMITSLEGEND

suppresses the legend for the control limits (for example, *3σ Limits For n=5*), which appears by default in the upper right corner of the chart.

NOOVERLAYLEGEND

suppresses the legend for overlay variables which is displayed by default when the OVERLAY= or OVERLAY2= option is specified.

NOREADLIMITS

specifies that the control limits for each *process* listed in the chart statement *not* be read from the LIMITS= data set specified in the PROC SHEWHART statement. There are two basic methods of displaying control limits: calculating control limits from the data and reading control limits from a LIMITS= data set. If you want control limits calculated from the data, you can do one of the following:

1. Do not specify a LIMITS= data set.
2. If you specify a LIMITS= data set, also specify the NOREADLIMITS option.

Otherwise, if you specify a LIMITS= data set in the PROC SHEWHART statement, the procedure reads control limits from that data set.

The following example illustrates the NOREADLIMITS option:

```
proc shewhart data=Pistons limits=Diamlim;
   xrchart Diameter*Hour;
   xrchart Diameter*Hour / noreadlimits;
run;
```

The first XRCHART statement reads the control limits from the first observation in the data set DIAMLIM for which the variable _VAR_ is equal to 'Diameter' and the variable _SUBGRP_ is equal to 'Hour'. The second XRCHART statement computes the control limits from the measurements in the data set Pistons. Note that the second XRCHART statement is equivalent to the following statements, which are more commonly used:

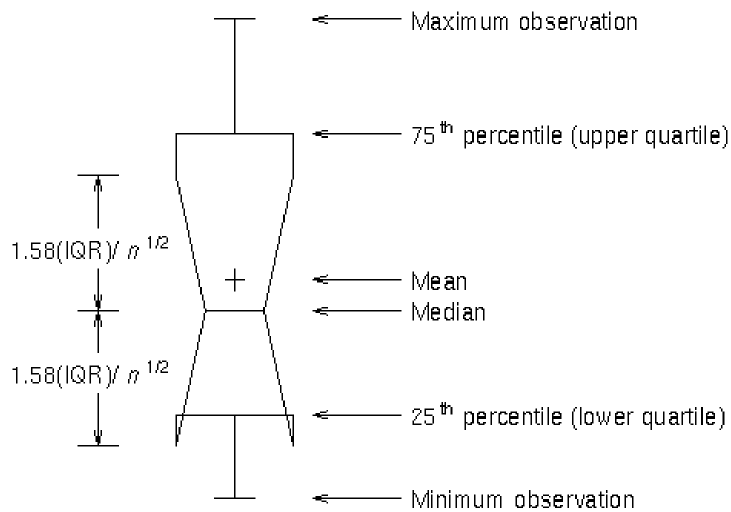
```
proc shewhart data=Pistons;
   xrchart Diameter*Hour;
run;
```

For more information about reading control limits from a LIMITS= data set, see the entry for the READLIMITS option and "Displaying Multiple Sets of Control Limits" on page 2033.

NOTCHES

specifies that box-and-whisker plots created by the BOXCHART statement be notched. The endpoints of the notches are located at the median plus and minus $1.58(\text{IQR}/\sqrt{n})$, where IQR is the interquartile range and n is the subgroup sample size. The medians (central lines) of two box-and-whisker plots are significantly different at approximately the 0.05 level if the corresponding notches do not overlap. Refer to McGill, Tukey, and Larsen (1978). Figure 17.135 illustrates the NOTCHES option. Notice the folding effect at the bottom, which happens when the endpoint of a notch is beyond its corresponding quartile. This situation occurs typically only when the subgroup sample size is small.

Figure 17.135 NOTCHES Option for Box-and-Whisker Plots



The NOTCHES option is also illustrated in Output 17.3.1 and is available only in the BOXCHART statement.

NOTESTACROSS

specifies that tests for special causes requested with the TESTS= or TESTS2= options not be applied across the boundaries of phases (blocks of consecutive subgroups) determined by the READPHASES= option and the variable _PHASE_ in the input data set. With constant control limits, if you specify the READPHASES= option but do not specify the NOTESTACROSS option, tests for special causes are applied without regard to phase boundaries. With varying control limits, tests are applied only within phases by default, and you can use the TESTACROSS option to specify that they be applied across phase boundaries. See "Tests for Special Causes: SHEWHART Procedure" on page 2073.

NOTICKREP

applies to character-valued *subgroup-variables* and specifies that only the first occurrence of repeated, adjacent subgroup values is to be labeled on the horizontal axis.

NOTRENDCONNECT

suppresses line segments that connect points on a trend chart. Points are connected by default. The NOTRENDCONNECT option is available only in the BOXCHART, MCHART, and XCHART statements when the TRENDVAR= option is used.

NOTRUNC

overrides the vertical axis truncation at zero, which is applied by default to *c* charts, moving range charts, *np* charts, *p* charts, *R* charts, *s* charts, and *u* charts. This option is useful if you are creating a customized version of one of these charts and want to replace the plotted statistics and control limits with values read from a TABLE= input data set that can be positive or negative. Do not use the NOTRUNC option in standard control chart applications. This option is not available in the BOXCHART, MCHART, and XCHART statements.

NOUCL

suppresses the display of the upper control limit in a primary chart.

NOUCL2

suppresses the display of the upper control limit in a secondary chart. The NOUCL2 option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

NPANELPOS=*n*

NPANEL=*n*

specifies the number of subgroup positions per panel on each chart. A *panel* is defined as a screen or page (or a half-screen or half-page if you are also using the BILEVEL option). You typically specify the NPANELPOS= option to display more points on a panel than the default number, which is $n = 50$ for all chart statements except the BOXCHART statement, for which the default is $n = 20$.

You can specify a positive or negative number for *n*. The absolute value of *n* must be at least 5. If *n* is positive, the number of positions is adjusted so that it is approximately equal to *n* and so that all panels display approximately the same number of subgroup positions. If *n* is negative, no balancing is done, and each panel (except possibly the last) displays approximately $|n|$ positions. In this case, the approximation is due only to axis scaling.

You can use the INTERVAL= option to change the effect of the NPANELPOS= option when a date or time format is associated with the *subgroup-variable*. The INTERVAL= option enables you to match the scale of the horizontal axis to the scale of the subgroup variable without having to associate a different format with the subgroup variable.

NPSYMBOL='*label*'

NPSYMBOL=NP | NPBAR | NPPM | NPPM2 | NP0

specifies a label for the central line in an *np* chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters in length.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol*=*value*, where *symbol* is one of the symbols given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Used in Graphics	Symbol Used in Line Printer Charts
NP	NP	NP
NPBAR	$\overline{\text{NP}}$	$\overline{\text{NP}}$
NPPM	NP′	NP′
NPPM2	NP″	NP″
NP0	NP_0	NP0

The default keyword is NPBAR. The NPSYMBOL= option is available only in the NPCHART statement.

OUTBOX=*SAS-data-set*
creates an output data set that contains subgroup summary statistics, control limits, and outlier values for a box chart. An OUTBOX= data set is the only type of summary data set produced by the SHEWHART procedure from which you can reconstruct a schematic box chart. The OUTBOX= option is available only in the BOXCHART statement. See "OUTBOX= Data Set" on page 1402 for details.

OUTFILL
COUTFILL
fills the areas outside the control limits that lie between the connected points and the control limits and are bounded by connecting lines. The areas are filled with an appropriate contrasting color from the ODS style. This option is useful for highlighting out-of-control points.

OUTHISTORY=*SAS-data-set*
creates an output data set that contains the subgroup summary statistics. You can use an OUTHISTORY= data set as a HISTORY= input data set in a subsequent run of the procedure. You cannot request an OUTHISTORY= data set if the input data set is a TABLE= data set. See "Output Data Sets" in the section for the chart statement in which you are interested. A related option is OUTPHASE=.

OUTINDEX='*label*'
specifies the value of the _INDEX_ variable in the OUTLIMITS= output data set. This is a bookkeeping variable that provides information identifying the control limits saved in the data set. See "Output Data Sets" in the section for the chart statement in which you are interested.

The *label* can be up to 128 characters and should be enclosed in quotes. You should use a label that uniquely identifies the control limits. For example, you might specify OUTINDEX='April 1-15' to indicate that the limits were computed from data collected during the first half of April.

The OUTINDEX= option is intended to be used in conjunction with the OUTLIMITS= option. The _INDEX_ variable is created only if you specify the OUTINDEX= option. If you specify the OUTINDEX= option and do not specify the name of the OUTLIMITS= data set with the OUTLIMITS= option, the procedure creates an OUTLIMITS= data set whose name is of the form WORK.DATA*n*.

NOTE: You cannot use the OUTINDEX= and READINDEXES= options in the same chart statement.

OUTLABEL=VALUE

OUTLABEL=(*variable*)

labels each point that falls outside the control limits on the primary chart with the value plotted for that subgroup or with the value of *variable* in the input data set.

The *variable* provided in the input data set can be numeric or character. If the *variable* is a character variable, it can be up to 16 characters. For each subgroup of observations whose summary statistic falls outside the control limits, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup. By default, points are not labeled. The OUTLABEL= option takes precedence over the TESTLABEL= option when TESTS=1 is specified. You cannot specify both the OUTLABEL= and ALLLABEL= options.

OUTLABEL2=VALUE

OUTLABEL2=(*variable*)

labels each point that falls outside the control limits on an *R* or *s* chart with the value plotted for that subgroup or with the value of *variable* in the input data set.

The *variable* provided in the input data set can be numeric or character. If the *variable* is a character variable, its length cannot exceed 16. For each subgroup of observations whose summary statistic falls outside the control limits, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup. By default, points are not labeled. The OUTLABEL2= option takes precedence over the TESTLABEL2= option when TESTS2=1 is specified. You cannot specify both the OUTLABEL2= and ALLLABEL2= options. The OUTLABEL2= option is available only in the IRCHART, MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements.

OUTLIMITS=*SAS-data-set*

creates an output data set that saves the control limits. You can use an OUTLIMITS= data set as an input LIMITS= data set in a subsequent run of the procedure. See "Output Data Sets" in the section for the chart statement in which you are interested. A related option is OUTINDEX=.

OUTPHASE='*label*'

specifies the value of the _PHASE_ variable in the OUTHISTORY= data set. This is a bookkeeping variable that provides information identifying the summary statistics saved in the data set. See "Output Data Sets" in the section for the chart statement in which you are interested.

You should use the OUTPHASE= option if you create OUTHISTORY= data sets at different stages (phases) for the same *processes* and concatenate the data sets to build a master historical data set. The _PHASE_ variable then identifies the block of observations that corresponds to each phase.

The *label* can be up to 128 characters and should be enclosed in quotes. You should use a *label* that uniquely identifies the saved data. For example, you might specify OUTPHASE='April 1-15' to indicate that the data were collected during the first half of April.

The _PHASE_ variable is created only if you specify the OUTPHASE= option. If you specify the OUTPHASE= option and do not specify the name of the OUTHISTORY= data set with the OUTHISTORY= option, the procedure creates an OUTHISTORY= data set whose name is of the form WORK.DATA*n*.

OUTTABLE=*SAS-data-set*

creates an output SAS data set that saves the information plotted on the chart, including the subgroup variable values and their corresponding summary statistics and control limits.

You can use the OUTTABLE= data set to create a customized report with the reporting procedures and methods described in *Base SAS Procedures Guide*. You can also use an OUTTABLE= data set as a TABLE= input data set in a subsequent run of the procedure. See “Output Data Sets” in the section for the chart statement in which you are interested.

OVERLAY=(*variable-list***)**

specifies variables to be overlaid on the primary control chart. A point is plotted for each overlay variable at each subgroup for which it has a non-missing value. The value of a particular overlay variable should be the same for each observation in the input data set with a given value of the subgroup variable. If values differ within a subgroup, the first value appearing in that subgroup is used. The OVERLAY= option cannot be specified with the STARVERTICES= option.

OVERLAY2=(*variable-list***)**

specifies variables to be overlaid on a secondary control chart. A point is plotted for each overlay variable at each subgroup for which it has a non-missing value. The value of a particular overlay variable should be the same for each observation in the input data set with a given value of the subgroup variable. If values differ within a subgroup, the first value appearing in that subgroup is used. The OVERLAY2= option cannot be specified with the STARVERTICES= option.

OVERLAY2ID=(*variable-list***)**

specifies variables whose formatted values are used to label points on secondary chart overlays. Variables in the OVERLAY2ID= list are matched with variables in the corresponding positions in the OVERLAY2= list. The value of the OVERLAY2ID= variable should be the same for each observation with a given value of the subgroup variable.

OVERLAYID=(*variable-list***)**

specifies variables whose formatted values are used to label points on primary chart overlays. Variables in the OVERLAYID= list are matched with variables in the corresponding positions in the OVERLAY= list. The value of the OVERLAYID= variable should be the same for each observation with a given value of the subgroup variable.

OVERLAYLEGLAB=‘*label***’**

specifies the label displayed to the left of the legend for overlays requested with the OVERLAY= or OVERLAY2= option. The label can be up to 16 characters and must be enclosed in quotes.

P0=*value*

specifies a known (standard) value p_0 for the proportion of nonconforming items produced by the process. By default, p_0 is estimated from the data. The P0= option is available only in the NPCHART and PCHART statements.

NOTE: As an alternative to specifying P0=p_0, you can read a predetermined value for p_0 from the variable _P_ in a LIMITS= data set. See “Input Data Sets” in the section for the chart statement in which you are interested.

PAGENUM=‘*string*’
specifies the form of the label used for pagination.

The *string* must be no longer than 16 characters, and it must include one or two occurrences of the substitution character #. The first # is replaced with the page number, and the optional second # is replaced with the total number of pages.

The PAGENUM= option is useful when you are working with a large number of subgroups, resulting in multiple pages of output. For example, suppose that each of the following XRCHART statements produces multiple pages:

```
proc shewhart data=Pistons;
   xrchart Diameter*Hour / pagenum='Page #';
   xrchart Diameter*Hour / pagenum='Page # of #';
   xrchart Diameter*Hour / pagenum='#/#';
run;
```

The third page produced by the first statement would be labeled *Page 3*. The third page produced by the second statement would be labeled *Page 3 of 5*. The third page produced by the third statement would be labeled *3/5*.

By default, no page number is displayed.

PAGENUMPOS=TL | TR | BL | BR | TL100 | TR100 | BL0 | BR0
specifies where to position the page number requested with the PAGENUM= option. The keywords TL, TR, BL, and BR correspond to the positions top left, top right, bottom left, and bottom right, respectively. You can use the TL100 and TR100 keywords to ensure that the page number appears at the very top of a page when a title is displayed. The BL0 and BR0 keywords ensure that the page number appears at the very bottom of a page when footnotes are displayed. The default keyword is BR.

PCTLDEF=*index*
specifies one of five definitions used to calculate percentiles in the construction of box-and-whisker plots requested with the BOXCHART statement. The *index* can be 1, 2, 3, 4, or 5. The five corresponding percentile definitions are discussed in “Percentile Definitions” on page 1414. The default is 5. The PCTLDEF= option is available only in the BOXCHART statement.

PHASEBREAK
specifies that the last point in a phase (defined as a block of consecutive subgroups with the same value of the _PHASE_ variable) is not to be connected to the first point in the next phase. By default, the points are connected.

PHASELABTYPE=SCALED | TRUNCATED
PHASELABTYPE=*height*
specifies how lengthy _PHASE_ variable values are displayed when there is insufficient space in the legend requested with the PHASELEGEND option. By default, lengthy values are not displayed.

If you specify PHASELABTYPE=SCALED, the values are uniformly reduced in height so that they fit. If you specify PHASELABTYPE=TRUNCATED, lengthy values are truncated on the right until they fit. When producing traditional graphics, you can also specify a text *height* in vertical percent screen units for the values. Related options are PHASELEGEND and PHASEREF.

NOTE: In ODS Graphics output only PHASELABTYPE=TRUNCATED is supported.

PHASELEGEND

PHASELEG

identifies the phases requested with the READPHASES= option in a legend across the top of the chart. Related options are PHASELABTYPE= and PHASEREF.

PHASELIMITS

specifies that the control limits and center line be labeled for each phase specified with the READ-PHASES= option, providing the limits are constant within that phase.

PHASEMEANSYMBOL=*symbol*

PHASEMEAN

specifies a symbol marker for the average of the values plotted within a phase. Specify PHASEMEAN without an argument to plot the phase average in ODS Graphics output. This option is available only in the BOXCHART statement.

PHASEREF

delineates the phases specified with the READPHASES= option with reference lines drawn vertically. Related options are PHASELABTYPE= and PHASELEGEND.

PHASEVARLABEL

displays the label associated with the variable _PHASE_ above the phase values in the phase legend. If there is no label associated with _PHASE_, or if the PHASELEGEND option is not specified, PHASEVARLABEL has no effect.

PHASEVALSEP

displays vertical lines separating phase values in the phase legend. If the PHASELEGEND option is not specified, PHASEVALSEP has no effect.

PROBLIMITS=DISCRETE

requests that discrete-valued probability limits be computed for attribute charts. This option is available only in the CCHART, NPCHART, PCHART, and UCHART statements, and it applies only when you request probability limits by specifying the ALPHA= option.

The possible values for the discrete probability limits are the same as for the subgroup values that are plotted on the control chart. For c and np charts these are integer values; for p and u charts these are multiples of $1/n$, where n is the subgroup sample size. Because attribute chart data are discrete, it is not possible in general to compute probability limits so that the probability of a point being outside the limits is α, for any arbitrary α.

The c and u charts are based on the Poisson distribution, which has the probability function

$$g(x) = \frac{\mu^x e^{-\mu}}{x!}, x = 0, 1, 2, \ldots$$

and the cumulative distribution function

$$G(x) = \sum_{i=0}^{x} g(i) = e^{-\mu} \sum_{i=0}^{x} \frac{\mu^i}{i!}, x = 0, 1, 2, \ldots$$

The np and p charts are based on the binomial distribution, which has the probability function

$$g(x) = \binom{n}{x} p^x (1-p)^{n-x}, x = 0, 1, 2, \ldots$$

and the cumulative distribution function

$$G(x) = \sum_{i=0}^{x} g(i) = \sum_{i=0}^{x} \binom{n}{i} p^i (1-p)^{n-i}, x = 0, 1, 2, \ldots$$

For *c* and *np* charts, the discrete lower control limit x_L is the smallest integer such that

$$G(x_L) \geq 1 - \alpha/2$$

and the discrete upper control limit x_U is the smallest integer such that

$$G(x_U) > \alpha/2$$

For *p* and *u* charts, the discrete lower control limit x_L is the smallest multiple of $1/n$ such that

$$G(nx_L) \geq 1 - \alpha/2$$

and the discrete upper control limit x_U is the smallest multiple of $1/n$ such that

$$G(nx_U) > \alpha/2$$

You can specify the ACTUALALPHA option to display the actual probability (instead of the probability you specify in the ALPHA= option) of a point being outside an attribute chart's probability limits.

PSYMBOL='*label*'

PSYMBOL=P | PBAR | PPM | PPM2 | P0

specifies a label for the central line in a *p* chart. You can use the option in the following ways:

- Specify a quoted *label* up to 16 characters.
- Specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol*=*value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

	Symbol Used in	
Keyword	**Graphics**	**Line Printer Charts**
P	P	P
PBAR	$\overline{P}$	$\overline{P}$
PPM	P′	P′
PPM2	P″	P″
P0	P_0	P0

The default keyword is PBAR. The PSYMBOL= option is available only in the PCHART statement.

RANGES

estimates the process standard deviation for a boxplot using subgroup ranges. By default, the process standard deviation for a boxplot is estimated from the subgroup standard deviations.

READALPHA

specifies that the variable _ALPHA_, rather than the variable _SIGMAS_, be read from a LIMITS= data set when both variables are available in the data set. Thus, the limits displayed are probability limits. If you do not specify the READALPHA option, then _SIGMAS_ is read by default. For details, see "Input Data Sets" in the section for the chart statement in which you are interested.

READINDEX=*value-list* **| ALL**

READINDEXES=*value-list* **| ALL**

READINDICES=*value-list* **| ALL**

reads one or more sets of control limits from a LIMITS= data set (specified in the PROC SHEWHART statement) for each *process* listed in the chart statement. The *i*th set of control limits for a particular *process* is read from the first observation in the LIMITS= data set for which

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup variable*
- the value of _INDEX_ matches *value*

The *values* can be up to 128 characters and must be enclosed in quotes.

NOTE: You cannot use the READINDEX= and OUTINDEX= options in the same chart statement. Also, the READLIMITS and READINDEX= options are alternatives to each other. If the LIMITS= data set contains more than one set of control limits for the same *process*, you should use the READINDEX= option.

You can display distinct sets of control limits (read from a LIMITS= data set) with data for various *phases* (read from blocks of observations in the input data set) by using the READINDEXES= and READPHASES= options together. See the entry for the READPHASES= option.

For more information about multiple sets of control limits and about the keyword ALL, see "Displaying Multiple Sets of Control Limits" on page 2033.

READLIMITS

specifies that the control limits are read from a LIMITS= data set specified in the PROC SHEWHART statement.[10] The control limits for each *process* listed in the chart statement are to be read from the first observation in the LIMITS= data set where

- the value of _VAR_ matches *process*
- the value of _SUBGRP_ matches the *subgroup variable*

The use of the READLIMITS option depends on the release of SAS/QC software that you are using.

- **In SAS 6.10 and later releases, the READLIMITS option is not necessary.** To read control limits as described previously, you simply specify a LIMITS= data set. However, even though the READLIMITS option is redundant, it continues to function as in earlier releases. Consequently, the following two XRCHART statements are equivalent:

[10] For details about computing control limits from the data, see the entry for the NOREADLIMITS option.

```
proc shewhart data=Pistons limits=Diamlim;
   xrchart Diameter*Hour;
   xrchart Diameter*Hour / readlimits;
run;
```

If the LIMITS= data set contains more than one set of control limits for the same *process*, you should use the READINDEX= option.

- **In SAS 6.09 and earlier releases, you must specify the READLIMITS option to read control limits as described previously.** If you specify a LIMITS= data set without specifying the READLIMITS option (or the READINDEX= option), the control limits are computed from the data. Consequently, the following two XRCHART statements are **not** equivalent:

```
proc shewhart data=Pistons limits=diamlim;
   xrchart Diameter*Hour;  /* limits computed from data */
   xrchart Diameter*Hour /
      readlimits;          /* limits read from DIAMLIM  */
run;
```

The READLIMITS and READINDEX= options are alternatives to each other.

You can use the READLIMITS and READPHASES= options together. In this case, the control limits are read as described previously, and the data plotted on the chart are those selected by the READPHASES= option.

READPHASES=*value-list* **| ALL**

READPHASE=*value-list* **| ALL**

selects blocks of consecutive observations to be read from the input data set. You can use the READPHASES= option only if

- the input data set contains a _PHASE_ variable
- the _PHASE_ variable is a character variable of no more than 128 characters

The READPHASES= option selects those observations whose _PHASE_ value matches one of the *values* specified in the *value-list*. The block of consecutive observations identified by the *i*th *value* is referred to as the *i*th *phase*. The *values* can be up to 128 characters and must be enclosed in quotes. List the *values* in the same order that they appear as values of the variable _PHASE_ in the input data set.

With the READPHASES= option you can

- create control charts that label blocks of data corresponding to multiple time *phases*. See the PHASELEGEND, PHASEREF, and CFRAME= options.
- create *historical control charts* that display distinct sets of control limits for different *phases*. This also requires a LIMITS= data set and the READINDEXES= option.

If the subgroup variable is numeric, the values of the subgroup variable should be contiguous from one block of observations to the next. Otherwise, there may be a gap in the control chart between the last point in one phase and the first point in the next phase. If you read a data set that contains multiple observations for each subgroup, the value of _PHASE_ must be constant within the subgroup.

You can display distinct sets of control limits (read from a LIMITS= data set) with data for various *phases* by using the READINDEX= and READPHASES= options together. For example, consider the flange width data in the HISTORY= data set Flange and the LIMITS= data set Flangelim. A partial listing of Flange is given in Figure 17.136 (for a complete listing of Flange, see Figure 17.149). The complete listing of Flangelim is given in Figure 17.137.

```
proc print data=Flange;
   var _phase_ Day Sample FlangewidthX FlangewidthR FlangewidthN;
run;
```

Figure 17.136 Listing of the HISTORY= Data Set Flange

Mean Chart for Diameters

Obs	_phase_	Day	Sample	FlangewidthX	FlangewidthR	FlangewidthN
1	Production	08FEB90	6	0.97360	0.06247	5
2	Production	09FEB90	7	1.00486	0.11478	5
3	Production	10FEB90	8	1.00251	0.13537	5
4	Production	11FEB90	9	0.95509	0.08378	5
5	Production	12FEB90	10	1.00348	0.09993	5
6	Production	15FEB90	11	1.02566	0.06766	5
7	Production	16FEB90	12	0.97053	0.07608	5
8	Production	17FEB90	13	0.94713	0.10170	5
9	Production	18FEB90	14	1.00377	0.04875	5
10	Production	19FEB90	15	0.99604	0.08242	5
11	Change 1	22FEB90	16	0.99218	0.09787	5
12	Change 1	23FEB90	17	0.99526	0.02017	5
13	Change 1	24FEB90	18	1.02235	0.10541	5
14	Change 1	25FEB90	19	0.99950	0.11476	5
15	Change 1	26FEB90	20	0.99271	0.05395	5
16	Change 1	01MAR90	21	0.98695	0.03833	5
17	Change 1	02MAR90	22	1.00969	0.06183	5
18	Change 1	03MAR90	23	0.98791	0.05836	5
19	Change 1	04MAR90	24	1.00170	0.05243	5
20	Change 1	05MAR90	25	1.00412	0.04815	5
21	Change 2	08MAR90	26	1.00261	0.05604	5
22	Change 2	09MAR90	27	0.99553	0.02818	5
23	Change 2	10MAR90	28	1.01463	0.05558	5
24	Change 2	11MAR90	29	0.99812	0.03648	5
25	Change 2	12MAR90	30	1.00047	0.04309	5
26	Change 2	15MAR90	31	0.99714	0.03689	5
27	Change 2	16MAR90	32	0.98642	0.04809	5
28	Change 2	17MAR90	33	0.98891	0.07777	5
29	Change 2	18MAR90	34	1.00087	0.06409	5
30	Change 2	19MAR90	35	1.00863	0.02649	5

```
proc print data=Flangelim;
   var _index_ _var_ _subgrp_ _type_ _limitn_ _alpha_ _sigmas_
       _lclx_ _mean_ _uclx_ _lclr_ _r_ _uclr_ _stddev_;
run;
```

Figure 17.137 Listing of the LIMITS= Data Set Flangelim

Mean Chart for Diameters

Obs	_index_	_var_	_subgrp_	_type_	_limitn_	_alpha_	_sigmas_	_lclx_	_mean_
1	Change 1	Flangewidth	Sample	ESTIMATE	5	.0026998	3	0.96167	0.99924
2	Production	Flangewidth	Sample	ESTIMATE	5	.0026998	3	0.93792	0.98827
3	Start	Flangewidth	Sample	ESTIMATE	5	.0026998	3	0.87088	0.96803

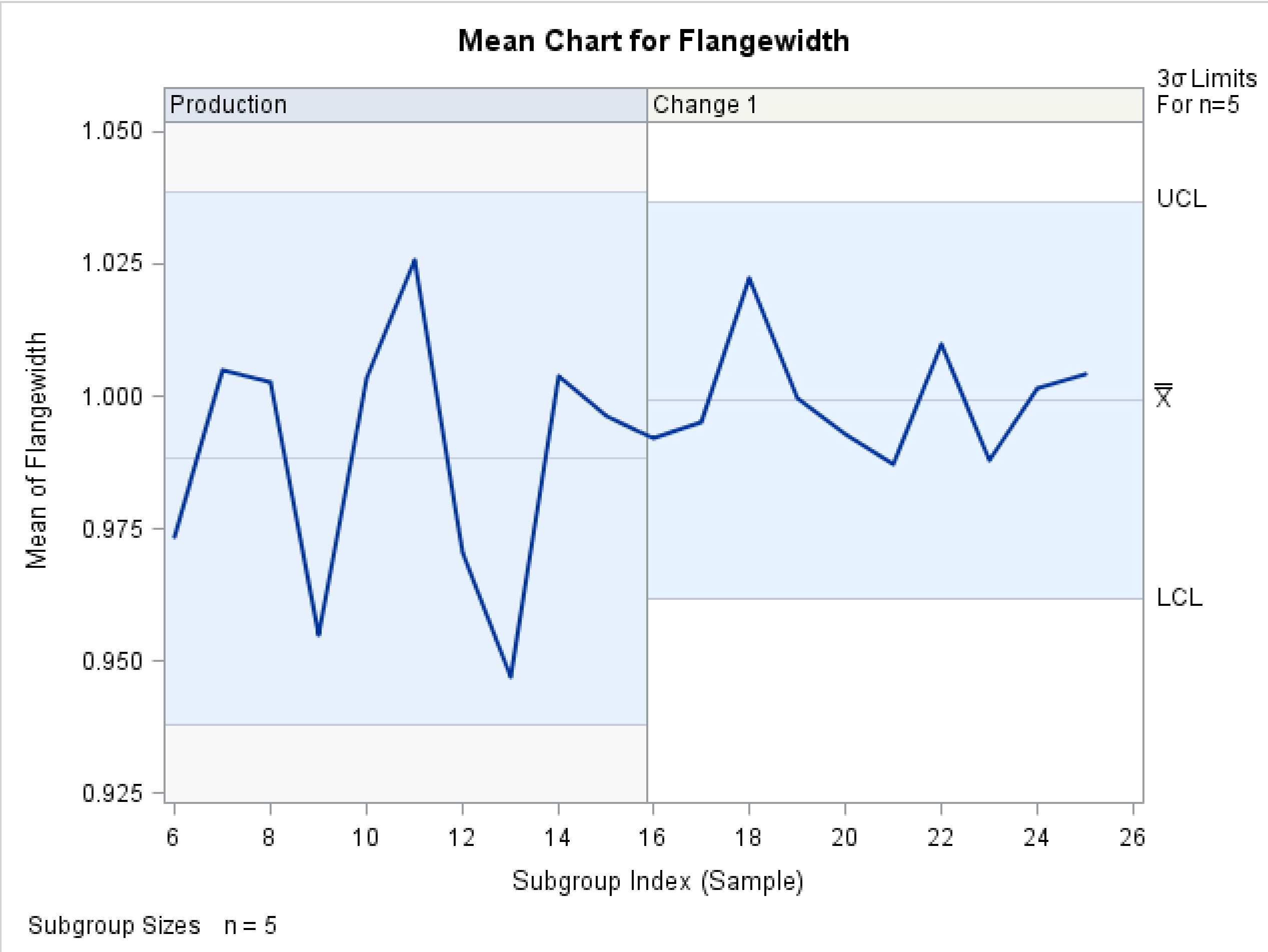
Obs	_uclx_	_lclr_	_r_	_uclr_	_stddev_
1	1.03680	0	0.06513	0.13771	0.028000
2	1.03862	0	0.08729	0.18458	0.037530
3	1.06517	0	0.16842	0.35612	0.072409

The following statements use the READINDEX= and READPHASES= options to create a historical control chart for the *Production* and *Change 1* phases:

```
ods graphics on;
proc shewhart history=Flange limits=Flangelim;
   xchart Flangewidth*Sample /
      readphases  = ('Production' 'Change 1')
      readindexes = ('Production' 'Change 1')
      phaseref
      phaselegend;
run;
```

The chart is displayed in Figure 17.138.

Figure 17.138 Multiple Control Limits for Multiple Phases



You can also use the keyword ALL with the READPHASES= option to match control limits to phases. For more information and examples about specifying multiple control limits, including the use of the keyword ALL, see "Displaying Multiple Sets of Control Limits" on page 2033.

REPEAT

REP

specifies that the horizontal axis of a chart that spans multiple pages be arranged so that the last subgroup position on a page is repeated as the first subgroup position on the next page. The REPEAT option facilitates cutting and pasting panels together. If a SAS DATETIME format is associated with the subgroup variable, REPEAT is used by default.

RSYMBOL='*label*'

RSYMBOL=R | RBAR | RPM | R0

specifies a label for the central line in an *R* chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol=value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Used in Graphics	Line Printer Charts
R	R	R
RBAR	$\overline{R}$	$\overline{R}$
RPM	R'	R'
R0	R_0	R0

The default keyword is RBAR. The RSYMBOL= option is available only in the IRCHART, MRCHART, RCHART, and XRCHART statements.

RTMARGIN=*value*

RTM=*value*

specifies the width of the right marginal area for the plot requested with the RTMPLOT= option. For traditional graphics, the width is specified in horizontal percent screen units. For ODS Graphics output, the width is specified in pixels. The RTMARGIN= option is available only in the IRCHART statement.

RTMPLOT=*keyword*

requests a univariate plot of the control chart statistics that is positioned in the right margin of the control chart. The *keywords* that you can specify and the associated plots are listed in the following table:

Keyword	Marginal Plot
DIGIDOT	digidot plot
HISTOGRAM	histogram
SKELETAL	skeletal box-and-whisker plot
SCHEMATIC	schematic box-and-whisker plot
SCHEMATICID	schematic box-and-whisker plot with outliers labeled
SCHEMATICIDFAR	schematic box-and-whisker plot with far outliers labeled

NOTE: Digidot plots are not available in ODS Graphics output.

The RTMPLOT= option is available only in the IRCHART statement; see Example 17.13 for an example. Refer to Hunter (1988) for a description of digidot plots, and see the entry for the BOXSTYLE= option for a description of the various box-and-whisker plots. Related options are LTMARGIN=, LTMPLOT=, and RTMARGIN=.

SEPARATE

displays primary and secondary charts on separate screens or pages. This option is useful if you are displaying line printer charts on a terminal and the number of lines on the screen limits the resolution of the chart. The SEPARATE option is available only in the IRCHART, MRCHART, XRCHART, and XSCHART statements.

SERIFS

adds serifs to the whiskers of *skeletal box-and-whisker charts*. The SERIFS option is available only in the BOXCHART statement.

SIGMA0=*value*

specifies a known (standard) value σ_0 for the process standard deviation σ. By default, σ_0 is estimated from the data.

The SIGMA0= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements.

NOTE: As an alternative to specifying SIGMA0=σ_0, you can read a predetermined value for σ_0 from the variable _STDDEV_ in a LIMITS= data set. For details, see "Input Data Sets" in the section for the chart statement in which you are interested.

SIGMAS=*k*

specifies the width of the control limits in terms of the multiple k of the standard error of the subgroup summary statistic plotted on the chart. The value of k must be positive. By default, $k = 3$ and the control limits are "3σ limits."

The particular subgroup summary statistic whose standard error is multiplied by k depends on the chart statement, as indicated by the following table:

Statement	Subgroup Summary Statistic
BOXCHART	mean or median
CCHART	number nonconforming
IRCHART	individual measurements and moving ranges
MCHART	median
MRCHART	median and range
NPCHART	number nonconforming
PCHART	proportion nonconforming
RCHART	range
SCHART	standard deviation
UCHART	number of nonconformities per unit
XCHART	mean
XRCHART	mean and range
XSCHART	mean and standard deviation

For details, see the Options for Specifying Control Limits table and the "Details" subsection in the section for the particular chart statement that you are using.

Note that

- as an alternative to specifying SIGMAS=k, you can read k from the variable _SIGMAS_ in a LIMITS= data set. For details, see "Input Data Sets" in the section for the chart statement in which you are interested.
- as an alternative to specifying SIGMAS=k (or reading _SIGMAS_ from a LIMITS= data set), you can request probability limits by specifying ALPHA=α (or reading the variable _ALPHA_ from a LIMITS= data set by specifying the READALPHA option).

SKIPHLABELS=*n*

SKIPHLABEL=*n*

specifies the number *n* of consecutive tick mark labels, beginning with the second tick mark label, that are thinned (not displayed) on the horizontal (subgroup) axis. For example, specifying SKIPHLABEL=1 causes every other label to be skipped (not displayed). Specifying SKIPHLABEL=2 causes the second and third labels to be skipped, the fifth and sixth labels to be skipped, and so forth.

The default value of the SKIPHLABELS= option is the smallest value *n* for which tick mark labels do not collide. A specified *n* will be overridden to avoid collision, unless you specify SKIPHLABELS=0, which forces all tick mark labels to be displayed. To avoid both collisions and thinning, you can use the TURNHLABELS option.

SMETHOD=NOWEIGHT | MVLUE | RMSDF | MAD | MMR | MVGRANGE

specifies a method for estimating the process standard deviation, σ, as summarized by the following table:

Keyword	Method for Estimating Standard Deviation
NOWEIGHT	estimates σ as an unweighted average of unbiased subgroup estimates of σ
MVLUE	calculates a minimum variance linear unbiased estimate for σ
RMSDF	calculates a root-mean square estimate for σ
MAD	calculates a median absolute deviation estimate for σ (IRCHART only)
MMR	calculates a median moving range estimate for σ (IRCHART only)
MVGRANGE	estimates σ based on a moving range of subgroup means (XRCHART and XSCHART only)

For formulas, see "Methods for Estimating the Process Standard Deviation" in the section for the particular chart statement you are using.

The default keyword is NOWEIGHT. The SMETHOD= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements. You can specify SMETHOD=RMSDF only in the BOXCHART, MCHART, XCHART, SCHART, and XSCHART statements and only when used with the STDDEVIATIONS option (or only in the absence of the RANGES option with a BOXCHART statement). You can specify SMETHOD=MAD and SMETHOD=MMR only in the IRCHART statement. You can specify SMETHOD=MVGRANGE only in the XRCHART and XSCHART statements.

SPLIT='*character***'**

specifies a special *character* that is inserted into the label of a process variable or summary statistic variable and whose purpose is to split the label into two parts. The first part is used to label the vertical axis of the primary chart, and the second part is used to label the vertical axis of the secondary chart. The *character* is not displayed in either label. See Figure 17.173 for an example.

The SPLIT= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

SSYMBOL=‘*label*’

SSYMBOL=S | SBAR | SPM | S0

specifies a label for the central line in an s chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol=value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

	Symbol Used in	
Keyword	**Graphics**	**Line Printer Charts**
S	S	S
SBAR	$\overline{\text{S}}$	$\overline{\text{S}}$
SPM	S′	S′
S0	S_0	S0

The default keyword is SBAR. The SSYMBOL= option is available only in the SCHART and XSCHART statements.

STARBDRADIUS=*value*

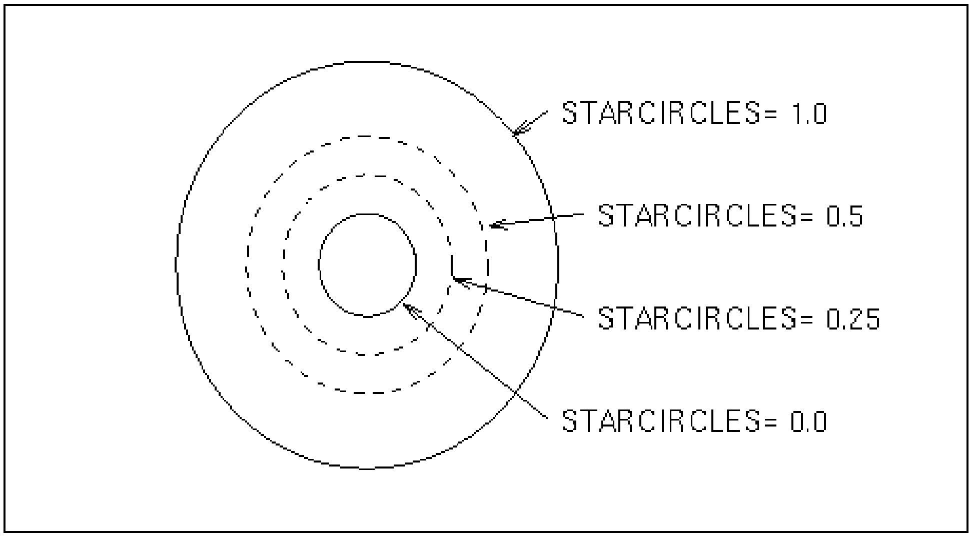
specifies the radius of an imaginary circle that is the outer bound for vertices of stars requested with the STARVERTICES= option. For traditional graphics, the radius is specified in horizontal percent screen units. For ODS Graphics output, the radius is specified in pixels. Vertices that exceed the outer bound are truncated to this value in order to prevent gross distortion of stars due to extreme values in the data. The *value* must be greater than or equal to the value specified with the STAROUTRADIUS= option. See Figure 17.140 or “Displaying Auxiliary Data with Stars” on page 2042.

STARCIRCLES=*values*

specifies reference circles that are superimposed on the stars requested with the STARVERTICES= option. All of the circles are displayed and centered at each point plotted on the primary chart. The *value* determines the diameter of the circle as follows: a *value* of zero specifies a circle with the *inner radius*, and a value of one specifies a circle with the *outer radius*. In general, a value of h specifies a circle with a radius equal to $inradius + h \times (outradius - inradius)$.

Figure 17.139 shows four circles specified with the STARCIRCLES= option. The values 0.0 and 1.0 correspond to the *inner circle* and *outer circle* (see the entries for the STARINRADIUS= and STAROUTRADIUS= options). The value 0.5 specifies a circle with a radius of $inradius + 0.5 \times (outradius - inradius)$ or a circle halfway between the inner circle and the outer circle. Likewise, the value 0.25 specifies a circle one-fourth of the way from the inner circle to the outer circle. Note also that the line types for the circles are specified with the LSTARCIRCLES= option. For more information, see “Displaying Auxiliary Data with Stars” on page 2042.

Figure 17.139 Circles Specified by STARCIRCLES=0.0 1.0 0.25 0.5

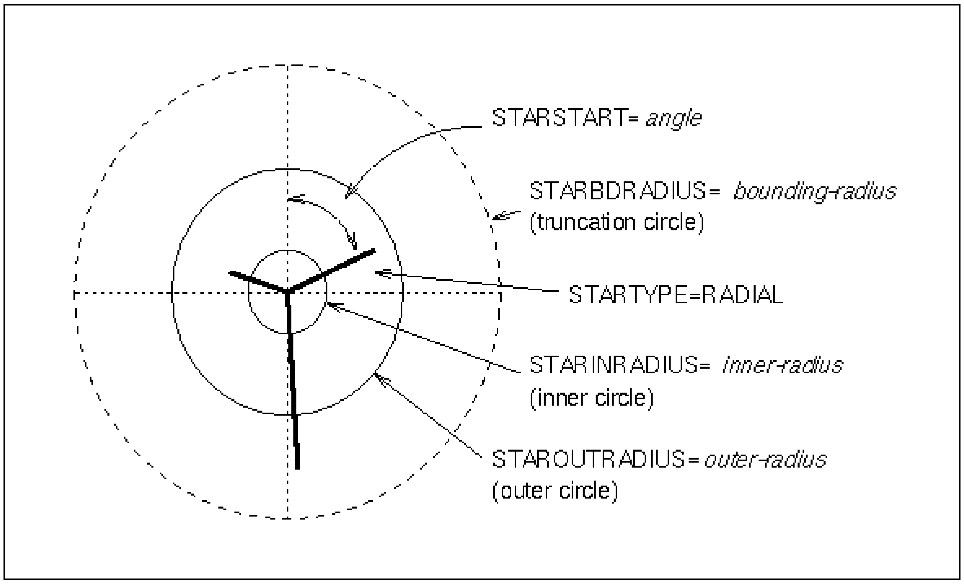


STARINRADIUS=*value*

specifies the inner radius of stars requested with the STARVERTICES= option. For traditional graphics, the radius is specified in horizontal percent screen units. For ODS Graphics output, the radius is specified in pixels. The *value* must be less than the value that is specified with the STAROUTRADIUS= option. The inner radius of a star is the distance from the center of the star to the circle that represents the lower limit of the standardized vertex variables. The lower limit can correspond to the minimum value, a multiple of standard deviations below the mean, or a lower specification limit. The default *value* is one-third of the outer radius.

Figure 17.140 illustrates five of the star options. The STARSTART= option determines the angle between the vertical axis and the first vertex. The STARINRADIUS= and STAROUTRADIUS= options specify the radii (in horizontal percent screen units) of the inner and outer circles that are associated with each star. Extremely large vertex values are truncated at the imaginary circle whose radius is specified by the STARBDRADIUS= option. The STARTYPE=RADIAL option specifies that the vertices are to be displayed as endpoints of line segments connecting each vertex to the center point. For more information, see the entries for these options or "Displaying Auxiliary Data with Stars" on page 2042.

Figure 17.140 Illustration of Star Options



STARFILL=*variable*

specifies colors for filling the interior of stars requested with the STARVERTICES= option. The STARFILL= option is analogous to the CSTARFILL= option, but the values of the STARFILL= variable are used only to group the stars. Stars in the same group are filled with the same color from the ODS style.

STARLABEL=ALL | FIRST | HIGH | LOW | OUT

specifies a method for labeling the vertices of stars requested with the STARVERTICES= option. The following table describes the method corresponding to each keyword:

Keyword	Method for Labeling Star Vertices
ALL	labels all vertices of all stars
FIRST	labels all vertices of the leftmost star
HIGH	labels only vertices that lie outside the outer circle
LOW	labels only vertices that lie inside the inner circle
OUT	labels only vertices that lie inside the inner circle or outside the outer circle

The label used for a particular vertex is the value of the variable _LABEL_ in the STARSPECS= data set. If this data set is not specified, or if the _LABEL_ variable is not provided, then the name of the vertex variable is used as the label. See "Displaying Auxiliary Data with Stars" on page 2042. By default, vertices are not labeled.

STARLEGEND=CLOCK | CLOCK0 | DEGREES | NONE

specifies the style of the legend used to identify the vertices of stars requested with the STARVERTICES= option. The following table describes the method corresponding to each keyword:

Keyword	Star Vertices Legend Style
CLOCK	identifies the vertex variables by their positions on the clock (starting with 12:00)
CLOCK0	identifies the vertex variables by their positions on the clock (starting with 0:00)
DEGREES	identifies the vertex variables by angles in degrees, with 0 degrees corresponding to 12 o'clock
NONE	suppresses the legend

See "Displaying Auxiliary Data with Stars" on page 2042. The default keyword is CLOCK.

STARLEGENDLAB='*label*'

specifies the label displayed to the left of the legend for stars requested with the STARLEGEND= option. The label can be up to 16 characters and must be enclosed in quotes. See "Displaying Auxiliary Data with Stars" on page 2042. The default label is *Vertices:*.

STAROUTRADIUS=*value*

specifies the outer radius of stars requested with the STARVERTICES= option. For traditional graphics, the radius is specified in horizontal percent screen units. For ODS Graphics output, the radius is specified in pixels. The outer radius of a star is the distance from the center of the star to the circle that represents the upper limit of the standardized vertex variables. The upper limit can correspond to the maximum value, a multiple of standard deviations above the mean, or an upper specification limit.

See Figure 17.140 "Displaying Auxiliary Data with Stars" on page 2042. For an example, see Figure 17.223. The default *value* depends on the number of subgroup positions per panel, and it is as large as possible without causing overlap of adjacent stars.

STARS=*variable*

specifies colors for the outlines of stars requested with the STARVERTICES= option. The STARS= option is analogous to the CSTARS= option, but the values of the STARS= variable are used only to group the stars. The outlines of stars in the same group are drawn with the same color from the ODS style.

STARSPECS=*value*|*SAS-data-set*

STARSPEC=*value*|*SAS-data-set*

specifies the method used to standardize the star vertex variables listed with the STARVERTICES= option. The method determines how the value of a vertex variable is transformed to determine the distance between the center of the star and the vertex. The STARSPECS= option also determines how the inner and outer radii of the star are to be interpreted.

A *value* of zero specifies standardization by the range of the variable. In this case, the distance between the center and the vertex is proportional to the difference between the variable value and the minimum variable value (taken across all subgroups). The inner radius of the star corresponds to the minimum variable value, and the outer radius of the star corresponds to the maximum variable value.

A positive STARSPECS= value requests standardization by a multiple of standard deviations above and below the mean. For example, STARSPECS=3 specifies that the inner radius of the star corresponds to three standard deviations below the mean, and the outer radius corresponds to three standard deviations above the mean. Thus, a vertex variable value exactly equal to the mean is represented by a vertex whose distance to the center of the star is halfway between the inner and outer radii.

You can request a distinct method of standardization for each vertex variable by specifying a STARSPECS= *data set*. Each observation provides standardization and related information for a distinct vertex variable. The variables read from a STARSPECS= *data set* are described in the following table:

Variable	**Description**
CSPOKE	color of spokes used with STARTYPE=RADIAL and STARTYPE=SPOKE; this must be a character variable of length 8 or less
LABEL	label for identifying the vertex when you specify STARLEGEND=FIRST or STARLEGEND=ALL; this must be a character variable of up to 16 characters
LSL	lower specification limit
LSPOKE	line style for spokes used with STARTYPE=RADIAL, STARTYPE=SPOKE, and STARTYPE=WEDGE
NOMVAL	nominal value substituted for missing values
SIGMAS	multiple of standard deviations above and below the average
UBOUND	upper bound for truncating extremely high values
USL	upper specification limit
VAR	name of vertex variable; this must be a character variable of length 32 or less

Only the variable _VAR_ is mandatory. If you provide the variables _LSL_ and _USL_, standardization is based on the specification limits; in this case, the variable _LSL_ corresponds to the inner radius of the star, and the variable _USL_ corresponds to the outer radius of the star. If you do not provide the variables _LSL_ and _USL_, standardization is based on the value of the variable _SIGMAS_, and if you do not provide the variable _SIGMAS_, standardization is based on the range.

See "Displaying Auxiliary Data with Stars" on page 2042. If you do not specify the STARSPECS= option, each vertex variable is standardized by its range across subgroups. In other words, the minimum corresponds to the inner radius, and the maximum corresponds to the outer radius.

STARSTART=*value*

specifies the vertex angle for the first variable in the STARVERTICES= list. Vertex angles for the remaining variables are uniformly spaced clockwise and assigned in the order listed. You can specify the *value* in the following ways:

- *Clock position:* If you specify the value as a time literal (between '0:00'T and '12:00'T), the corresponding clock position is used for the first vertex variable.
- *Degrees:* If you specify the value as a nonpositive number, the absolute value in degrees is used for the first vertex angle. Here, 0 degrees corresponds to 12:00.

The default *value* is zero, so the first vertex variable is positioned at 12:00. See Figure 17.140 or "Displaying Auxiliary Data with Stars" on page 2042.

STARTYPE=CORONA | POLYGON | RADIAL | SPOKE | WEDGE

specifies the style of the stars requested with the STARVERTICES= option. The following table describes the method corresponding to each keyword.

Keyword	Star Style
CORONA	polygon with star-vertices emanating from the inner circle
POLYGON	closed polygon
RADIAL	rays emanating from the center
SPOKE	rays emanating from the inner circle
WEDGE	closed polygon with rays from the center to each vertex

See Figure 17.140 or "Displaying Auxiliary Data with Stars" on page 2042. "Adding Reference Circles to Stars" on page 2045 describes the inner and outer circles, and "Specifying the Style of Stars" on page 2047 provides examples of each value of the STARTYPE= option. The default keyword is POLYGON.

STARVERTICES=*variable* | (*variable-list*)

superimposes a star (polygon) at each point on the primary chart. The star is centered at the point, and the distance between the center and each star vertex represents the standardized value of a *variable* in the STARVERTICES= list. The *variables* must be provided in the input data set.

The star display is suggested as a method for monitoring quantitative variables (such as environmental factors) that are measured simultaneously with the process variable. For examples and details, see "Displaying Auxiliary Data with Stars" on page 2042. By default, stars are not superimposed on the chart.

STDDEVIATIONS

STDDEVS

specifies that the estimate of the process standard deviation σ is to be calculated from subgroup standard deviations. This, in turn, affects the calculation of control limits; for details, see "Methods for Estimating the Process Standard Deviation" in the section for the chart statement in which you are interested. By default, the estimate of σ is calculated from subgroup ranges, except with the BOXCHART statement, where subgroup standard deviations are used by default.

If you specify the STDDEVIATIONS option and read summary data from a HISTORY= data set, the data set must contain a subgroup standard deviation variable for each *process*. Conversely, if you omit the STDDEVIATIONS option, the HISTORY= data set must contain a subgroup range variable for each *process* listed in the chart statement.

You should specify STDDEVIATIONS when your subgroup sample sizes are large (typically, 15 or greater). The STDDEVIATIONS option is available only in the MCHART and XCHART statements.

SUBGROUPN=*value*

SUBGROUPN=*variable*

specifies the subgroup sample sizes as a constant *value* or as the values of a *variable* in the DATA= data set. The SUBGROUPN= option is available only in the CCHART, NPCHART, PCHART, and UCHART statements.

You must specify SUBGROUPN= in the NPCHART, PCHART, and UCHART statements when your input data set is a DATA= data set. If you are using a CCHART statement, the SUBGROUPN= option is available only when your input data set is a DATA= data set. For the CCHART statement, the default value of the SUBGROUPN= option is one.

If you specify multiple *processes* in a chart statement, the SUBGROUPN= option is used with all of the *processes* listed.

SYMBOLLEGEND=LEGEND*n*

SYMBOLLEGEND=NONE

controls the legend for the levels of a *symbol-variable* (see “Displaying Stratification in Levels of a Classification Variable” on page 2025). For traditional graphics, you can specify SYMBOLLEGEND=LEGEND*n*, where *n* is the number of a LEGEND statement defined previously. You can specify SYMBOLLEGEND=NONE to suppress the default legend.

SYMBOLORDER=DATA | INTERNAL | FORMATTED

SYMORD=DATA | INTERNAL | FORMATTED

specifies the order in which symbols are assigned for levels of *symbol-variable*. The DATA keyword assigns symbols to values in the order in which values appear in the input data. This is how symbols were assigned in SAS 6.12 and earlier releases of SAS/QC software. The INTERNAL keyword assigns symbols based on sorted order of internal values of *symbol-variable* and FORMATTED assigns them based on sorted formatted values. The default value is FORMATTED.

TABLE < (EXCEPTIONS) >

TABLES < (EXCEPTIONS) >

creates a basic table of the subgroup values, the subgroup sample sizes, the subgroup summary statistics, and the upper and lower control limits. Rows of the table correspond to subgroups. The keyword **EXCEPTIONS** (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

You can request extended versions of the basic table by specifying one or more of the following options: TABLEBOX, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUTLIM, and TABLETESTS. Specifying the TABLEALL option is equivalent to specifying all of these options, and it provides the most extensive table.

TABLEALL < (EXCEPTIONS) >

tabulates the information about the control chart and is equivalent to specifying all of the following options: TABLES, TABLECENTRAL, TABLEID, TABLELEGEND, TABLEOUT, and TABLETESTS. If you specify the TABLEALL option in a BOXCHART statement, the TABLEBOX option is also implied. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive. You can use the OUTTABLE= option to create a data set that saves the information tabulated with the TABLEALL option.

TABLEBOX < (EXCEPTIONS) >

augments the basic table created by the TABLES option with columns for the minimum, 25th percentile, median, 75th percentile, and maximum of the observations in a subgroup. The TABLEBOX option is available only in the BOXCHART statement. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLECENTRAL < (EXCEPTIONS) >

TABLEC < (EXCEPTIONS) >

augments the basic table created by the TABLES option with columns for the values of the central lines. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLEID < (EXCEPTIONS) >

augments the basic table created by the TABLES option with a column for each of the ID variables. The keyword **EXCEPTIONS** (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLELEGEND < (EXCEPTIONS) >

TABLELEG < (EXCEPTIONS) >

adds a legend to the basic table created by the TABLE option. The legend describes the tests for special causes that were requested with the TESTS= option and for which a positive signal is found for at least one subgroup. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLEOUTLIM < (EXCEPTIONS) >

TABLEOUT < (EXCEPTIONS) >

augments the basic table created by the TABLE option with columns indicating which control limits (if any) are exceeded. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TABLETESTS < (EXCEPTIONS) >

augments the basic table created by the TABLES option with a column that indicates which of the tests for special causes (requested with the TESTS= option) are positive. The column contains the numbers of all the tests that are positive at a particular subgroup. The keyword EXCEPTIONS (enclosed in parentheses) is optional and restricts the tabulation to those subgroups for which the control limits are exceeded or a test for special causes is positive.

TARGET=*value-list*

provides target values used to compute the capability index C_{pm}, which is saved in the OUTLIMITS= data set. If you provide more than one *value*, the number of *values* must match the number of *processes* listed in the chart statement. If you specify only one *value*, it is used for all the *processes*.

CAUTION: You can use the TARGET= options only in conjunction with the LSL= and USL= options. For more information, see “Capability Indices” on page 1874 and “Output Data Sets” in the section for the chart statement in which you are interested. Also see the entries for the LSL= and USL= options. The TARGET= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements.

TEST2RESET=*variable*

TEST2RESET=*value*

enables tests for special causes to be reset in a secondary chart. The specified variable must be a character variable of length 8, or length 16 if customized tests are requested. The variable values have the same format as those of the _TESTS_ variable in a TABLE= data set. A test that is flagged by the

TEST2RESET= value for a given subgroup is reset starting with that subgroup. That means a positive result for the test can include the given subgroup only if it is the first subgroup in the pattern. For example, the value "12345678" for the TEST2RESET= variable will reset all standard tests for special causes.

TEST2RUN=*run-length*
specifies the length of the pattern for Test 2 requested with the TESTS= and TESTS2= options. The values allowed for the *run-length* are 7, 8, 9, 11, 14, and 20. The form of the test for each *run-length* value is given in the following table. The default *run-length* is 9. See "Tests for Special Causes: SHEWHART Procedure" on page 2073 for more information.

Run-length	Number of Points on One Side of the Central Line
7	7 in a row
8	8 in a row
9	9 in a row
11	at least 10 out of 11 in a row
14	at least 12 out of 14 in a row
20	at least 16 out of 20 in a row

TEST3RUN=*run-length*
specifies the length of the pattern for Test 3 requested with the TESTS= and TESTS2= options. Test 3 searches for a pattern of steadily increasing or decreasing values, where the length of the pattern is at least the value given as the *run-length*. The values allowed for the *run-length* are 6, 7, and 8. The default *run-length* is 6. See "Tests for Special Causes: SHEWHART Procedure" on page 2073 for more information.

TESTACROSS
specifies that tests for special causes requested with the TESTS= or TESTS2= options be applied without regard to phases (blocks of consecutive subgroups) determined by the READPHASES= option and the variable _PHASE_ in the input data set. With varying control limits, if you specify the READPHASES= option but do not specify the TESTACROSS option, tests for special causes are applied within (but not across) phases. With constant control limits, tests are applied across phases by default, and you can use the NOTESTACROSS option to specify that they be applied only within phases. See "Tests for Special Causes: SHEWHART Procedure" on page 2073.

TESTLABBOX
requests that labels for subgroups with positive tests for special causes are positioned so they do not overlap. The labels are enclosed in boxes that are connected to the associated subgroup points with line segments.

TESTLABEL='*label***'**
TESTLABEL=(*variable***)**
TESTLABEL=TESTINDEX
TESTLABEL=SPACE
TESTLABEL=NONE
provides labels for points at which one of the tests for special causes (requested with the TESTS= or TESTS2= option) is positive. The values for the TESTLABEL= option are as follows:

- You can specify a *label* of up to 16 characters enclosed in quotes. This label is displayed at all points where a test is signaled.
- You can specify a *variable* (enclosed in parentheses) whose values are used as labels. The *variable* must be provided in the input data set, and it can be numeric or character. If the *variable* is character, its length cannot exceed 16. For each subgroup of observations at which a test is signaled, the formatted value of the *variable* in the observations is used to label the point representing the subgroup. If you are reading a DATA= data set with multiple observations per subgroup, the values of the *variable* should be identical for observations within a subgroup.
- You can specify TESTINDEX to label points with the single-digit *index* that requested the test in a TESTS= or TESTS2= list. If the test was requested with a customized *pattern* in a TESTS= or TESTS2= list, then points are labeled with the letter that you specified using the CODE= option.
- You can specify SPACE to request a label of the form *Test k*. This is slightly more legible than the default label of the form *Testk* (a description of *Testk* follows).
- You can specify NONE to suppress labeling.

If you do not use the TESTLABEL= option, the default label is of the form *Testk*, where *k* is the index of the test as requested with the TESTS= or TESTS2= options, or *k* is the CODE= *character* of the test as requested in a pattern specified with the TESTS= or TESTS2= options.

See "Tests for Special Causes: SHEWHART Procedure" on page 2073. Related options include OUTLABEL=, OUTLABEL2=, TESTFONT=, TESTHEIGHT=, and TESTLABEL*n*=.

TESTLABEL*n*='*label*'

specifies a *label* for points at which the test for special causes requested with the *index n* in a TESTS= or TESTS2= list is positive. The *index n* can be a number from 1 to 8. The TESTLABEL*n*= option overrides a TESTLABEL= option and the default label *Test n*. The *label* that you specify with the TESTLABEL*n*= option can be up to 16 characters and must be enclosed in quotes.

See "Tests for Special Causes: SHEWHART Procedure" on page 2073. Related options are TESTFONT=, TESTHEIGHT=, and TESTLABEL=.

TESTNMETHOD=STANDARDIZE

applies the tests for special causes requested with the TESTS= and TESTS2= options to standardized test statistics when the subgroup sample sizes are not constant. This method was suggested by Nelson (1994). See "Tests for Special Causes: SHEWHART Procedure" on page 2073. By default, the tests are not applied to data with varying subgroup sample sizes.

TESTOVERLAP

applies tests for special causes (requested with the TESTS= or TESTS2= option) to overlapping patterns of points.

The TESTOVERLAP option modifies the way in which the search for a subsequent pattern is done when a pattern is encountered. If you omit the TESTOVERLAP option, the search begins with the first subgroup after the current pattern ends. If you specify the TESTOVERLAP option, the search begins with the second subgroup in the current pattern.

The following statements illustrate the use of the TESTOVERLAP option:

```
proc shewhart;
   xrchart Width*Hour / test=3;
   xrchart Width*Hour / test=3 testoverlap;
run;
```

Test 3 looks for six subgroup means in a row steadily increasing or decreasing. Suppose that the subgroup means of WIDTH are steadily increasing for HOUR=5, 6, 7, 8, 9, 10, and 11. The first XRCHART statement will signal that Test 3 is positive at HOUR=10 but not at HOUR=11, since the search for the next pattern begins with HOUR=11. The second XRCHART statement will signal that Test 3 is positive at HOUR=10 and HOUR=11, since the search for the next pattern begins with HOUR=6 and thus finds a second pattern ending with HOUR=11. See "Tests for Special Causes: SHEWHART Procedure" on page 2073 for more information.

CAUTION: Specifying TESTOVERLAP affects the interpretation of the standard tests for special causes, because a particular point can contribute to more than one positive test. Typically, this option should not be used.

TESTRESET=*variable*

TESTRESET=*value*

enables tests for special causes to be reset in a primary chart. The specified variable must be a character variable of length 8, or length 16 if customized tests are requested. The variable values have the same format as those of the _TESTS_ variable in a TABLE= data set. A test that is flagged by the TESTRESET= value for a given subgroup is reset starting with that subgroup. That means that a positive result for the test can include the given subgroup only if it is the first subgroup in the pattern. For example, the value "12345678" for the TESTRESET= variable will reset all standard tests for special causes.

TESTS=*index-list*

TESTS=*customized-pattern-list*

requests one or more tests for special causes, which are also known as *runs tests*, *pattern tests*, and *Western Electric rules*. These tests detect particular nonrandom patterns in the points plotted on the primary control chart. The occurrence of a pattern, referred to as a *signal*, suggests the presence of a special cause of variation.

Each pattern is defined in terms of zones A, B, and C, which are constructed by dividing the interval between the control limits into six equally spaced subintervals. Zone A is the union of the subintervals immediately below the upper control limit and immediately above the lower control limit. Zone C is the union of the subintervals immediately above and below the central line. Zone B is the union of the subintervals between zones A and C. See Figure 17.178 for an illustration of test zones.

You can use the TESTS= option in three ways. First, you can specify an *index-list* to request a combination of standard tests (this is the approach most commonly used). Second, you can specify a *customized-pattern-list* to request a combination of tests based on customized patterns. Third, you can specify a list consisting of both *indexes* and *customized-patterns*. The first two approaches are described as follows.

Standard tests. The following table lists the standard tests that you can request by specifying TEST=*index-list*. The tests are indexed according to the sequence used by Nelson (1984, 1985).

Index	Pattern Description
1	one point beyond Zone A (outside the control limits)
2	nine points in a row in Zone C or beyond on one side of the central line (see the entry for the TEST2RUN option)
3	six points in a row steadily increasing (see the entry for the TEST3RUN option)
4	fourteen points in a row alternating up and down
5	two out of three points in a row in Zone A or beyond
6	four out of five points in a row in Zone B or beyond
7	fifteen points in a row in Zone C on either or both sides of the central line
8	eight points in a row on either or both sides of the central line with no points in Zone C

You can specify any combination of the eight *indexes* with an explicit list or with an implicit list, as in the following example:

```
proc shewhart;
   xrchart Width*Hour / tests=1 2 3 4;
   xrchart Width*Hour / tests=1 to 4;
run;
```

The TESTS= option is available in all but the RCHART and SCHART statements. Use only tests 1, 2, 3, and 4 in the CCHART, NPCHART, PCHART, and UCHART statements. By default, the TESTS= option is not applied in any chart statement unless the control limits are 3σ limits. You can use the NO3SIGMACHECK option to request tests for special causes when you use the SIGMAS= option to specify control limits other than 3σ limits.

Customized tests. Although the standard tests supported by the TESTS= option are appropriate for the vast majority of control chart applications, there may be situations in which you want to work with customized tests. You can define your own tests by specifying TESTS=*customized-pattern-list*. There are two types of patterns that you can include in this list: *T-patterns* and *M-patterns*.

Use a T-pattern to request a search for *k* out of *m* points in a row in the interval (a, b). The required syntax for a T-pattern is

T(K=*k* M=*m* LOWER=*a* UPPER=*b* CODE=*character* LABEL=*'label'*)

The default value for SCHEME= is ONESIDED. The options for a T-pattern are summarized in the following table:

Option	**Description**
K=*k*	number of points
M=*m*	number of consecutive points
LOWER=*value*	lower limit of interval (a, b)
UPPER=*value*	upper limit of interval (a, b)
SCHEME=ONESIDED	one-sided scheme using (a, b)
SCHEME=TWOSIDED	two-sided scheme using $(a, b) \cup (-b, -a)$
CODE=*character*	identifier for test (A-H)
LABEL=*'label'*	label for points if signal
LEGEND=*'legend'*	legend used with the TABLELEGEND option

Use an M-pattern to request a search for *k* points in a row increasing or decreasing. The required syntax for an M-pattern is

M(K=*k* DIR=*direction* CODE=*character* LABEL=*'label'*)

The options for an M-pattern are summarized in the following table:

Option	**Description**
K=*k*	number of points
DIR=INC	increasing pattern
DIR=DEC	decreasing pattern
CODE=*character*	identifier for test (A-H)
LABEL=*'label'*	label for points if signal
LEGEND=*'legend'*	legend used with the TABLELEGEND option

For details on the TESTS= option, see "Tests for Special Causes: SHEWHART Procedure" on page 2073. Related options include CTEST=, CZONES=, LTEST=, TABLETESTS, TABLELEGEND, TEST2RUN=, TEST3RUN=, TESTACROSS, TESTCHAR=, TESTLABEL=, TESTLABELn=, TESTNMETHOD=, TESTOVERLAP, TESTS2=, ZONES, ZONECHAR=, and ZONELABELS.

TESTS2=*index-list*

TESTS2=*customized-pattern-list*

requests one or more tests for special causes for an *R* chart or *s* chart. The syntax for the TESTS2= option is identical to the syntax for the TESTS= option. The TESTS2= option is available in the MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements. For details on the TESTS2= option, see "Tests for Special Causes: SHEWHART Procedure" on page 2073. Related options include CTEST=, CZONES=, LTEST=, TABLETESTS, TABLELEGEND, TEST2RUN=, TEST3RUN=, TESTACROSS, TESTCHAR=, TESTLABEL=, TESTLABELn=, TESTNMETHOD=, TESTOVERLAP, TESTS=, ZONES2, ZONECHAR=, and ZONE2LABELS.

TOTPANELS=*n*

specifies the total number of panels to be used to display the chart. This option overrides the NPANEL= option.

TRENDVAR=*variable* **| (***variable-list***)**

specifies a list of trend variables, one for each *process* listed in the chart statement. The TRENDVAR= option is available only in the BOXCHART, MCHART, and XCHART statements and only when your input data set is a DATA= or HISTORY= data set.

The values of the trend variables are subtracted from the values of the corresponding process variables (if you read a DATA= data set) or subgroup mean variables (if you read a HISTORY= data set). The chart is then created for the residuals (differences), and the trend values are plotted in a secondary chart. If you specify a single trend variable and two or more *processes*, the trend variable is used with each *process*.

The TRENDVAR= option does not apply if you are reading a TABLE= data set. In this case, the procedure produces a trend chart only if the variable _TREND_ is included in the TABLE= data set.

For more details, see "Displaying Trends in Process Data" on page 2054. Related options include NOTRENDCONNECT, SEPARATE, SPLIT=, WTREND=, and YPCT1=.

TYPE=*value*

specifies the *value* of the _TYPE_ variable in the OUTLIMITS= data set, which in turn indicates whether certain parameter variables in this data set represent estimates or standard (known) values.

If you are using a chart statement that creates a variables chart, _TYPE_ is a bookkeeping variable that indicates whether the values of the variables _MEAN_ and _STDDEV_ in the OUTLIMITS= data set are estimates or standard values of the process mean μ and standard deviation σ. The following table summarizes the *values* that you can specify:

Value	_MEAN_	_STDDEV_
ESTIMATE	estimate	estimate
STDMU	standard	estimate
STDSIGMA	estimate	standard
STANDARD	standard	standard

The default *value* is ESTIMATE, unless you specify standard values for μ or σ with the MU0= or SIGMA0= options.

For PCHART and NPCHART statements, the *value* you specify for the TYPE= option can be either ESTIMATE or STANDARD, indicating that the value of the variable _P_ in the OUTLIMITS= data set is an estimate or standard value of the proportion p of nonconforming items. The default *value* is ESTIMATE, unless you specify a standard value for p with the P0= option.

For UCHART and CCHART statements, the *value* you specify for the TYPE= option can be either ESTIMATE or STANDARD, indicating that the value of the variable _U_ in the OUTLIMITS= data set is an estimate or standard value of the average number u of nonconformities per unit. The default *value* is ESTIMATE, unless you specify a standard value for u with the U0= option.

U0=*value*

specifies a known (standard) value u_0 for the average number u of nonconformities per unit produced by the process. By default, u_0 is estimated from the data. The U0= option is available only in the CCHART and UCHART statements.

NOTE: As an alternative to specifying the U0= option, you can read a predetermined value for u_0 from the variable _U_ in a LIMITS= data set. For details, see "Input Data Sets" in the section for the chart statement in which you are interested.

UCLLABEL='*label*'

specifies a label for the upper control limit in the primary chart. The label can be up to 16 characters. Enclose the label in quotes. The default label is of the form *UCL=value* if the control limit has a fixed value; otherwise, the default label is *UCL*. Related options are UCLLABEL2=, LCLLABEL=, and LCLLABEL2=.

UCLLABEL2='*label*'

specifies a label for the upper control limit in the secondary chart. The label can be up to 16 characters. Enclose the label in quotes. The default label is of the form *UCL=value* if the control limit has a fixed value; otherwise, the default label is *UCL*. This option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements. Related options are LCLLABEL2=, LCLLABEL=, and UCLLABEL=.

USL=*value-list*

provides upper specification limits used to compute capability indices. If you provide more than one *value*, the number of *values* must match the number of *processes* listed in the chart statement. If you specify only one *value*, it is used for all the *processes*.

The SHEWHART procedure uses the specification limits to compute capability indices, and it saves the limits and indices in the OUTLIMITS= data set. For more information, see "Capability Indices" on page 1874 and "Output Data Sets" in the section for the chart statement in which you are interested. A related option is LSL=. The USL= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, RCHART, SCHART, XCHART, XRCHART, and XSCHART statements.

USYMBOL='*label*'

USYMBOL=U | UBAR | UPM | UPM2 | U0

specifies a label for the central line in a *u* chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters.
- You can specify one of the keywords listed in the following table. Each keyword requests a label of the form *symbol=value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

	Symbol Used in	
Keyword	**Graphics**	**Line Printer Charts**
U	U	U
UBAR	$\overline{\mathrm{U}}$	$\overline{\mathrm{U}}$
UPM	U′	U′
UPM2	U″	U″
U0	U_0	U0

The default keyword is UBAR. The USYMBOL= option is available only in the UCHART statement.

VAXIS=*value-list*

VAXIS=AXIS*n*

specifies major tick mark values for the vertical axis of a primary chart. The *values* must be listed in increasing order, must be evenly spaced, and must span the range of summary statistics and control limits displayed in the chart. You can specify the *values* with an explicit list or with an implicit list, as shown in the following example:

```
proc shewhart;
   xrchart Width*Hour / vaxis=0 2 4 6 8;
   xrchart Width*Hour / vaxis=0 to 8 by 2;
run;
```

If you are producing traditional graphics, you can also specify a previously defined AXIS statement with the VAXIS= option. Related options are HAXIS= and VAXIS2=.

VAXIS2=*value-list*

VAXIS2=AXIS*n*

specifies major tick mark values for the vertical axis of a secondary chart. The specifications and restrictions are the same as for the VAXIS= option. The VAXIS2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option. Related options are HAXIS= and VAXIS=.

VFORMAT=*format*

specifies a format to be used for displaying tick mark labels on the vertical axis of a primary chart.

VFORMAT2=*format*

specifies a format to be used for displaying tick mark labels on the vertical axis of a secondary chart.

VOFFSET=*value*

specifies the length of the offset at each end of the vertical axis. For traditional graphics, the offset is specified in percent screen units. For ODS Graphics output, the offset is specified in pixels.

VREF=*value-list*

VREF=*SAS-data-set*

draws reference lines perpendicular to the vertical axis on the primary chart. Reference line values can be expressed as simple values or as multiples of the standard error of the subgroup summary statistic. You can use this option in the following ways:

- Specify the *values* for the lines with a VREF= list. Examples of the VREF= option follow:

  ```
  vref=20
  vref=20 40 80
  vref=(2.5 sigma)
  vref=20 (1.5 2.0 2.5 sigma) 80
  ```

  Values expressed as multiples of σ must be enclosed in parentheses with the SIGMA keyword.

- Specify the values for the lines as the values of a numeric variable named _REF_ in a VREF= data set. Optionally, you can provide labels for the lines as values of a variable named _REFLAB_, which must be a character variable of length 16 or less. If you want distinct reference lines to be displayed in charts for different *processes* specified in the chart statement, you must include a character variable of length 32 or less named _VAR_, whose values are the *processes*. If you do not include the variable _VAR_, all of the lines are displayed in all of the charts. If you want to display reference lines whose values are multiples of σ, you must include a character variable named _TYPE_, whose values are “VALUES” or “SIGMAS.” The value of _TYPE_ indicates whether the reference line value is expressed as a simple value or as a multiple of σ.

Each observation in the VREF= data set corresponds to a reference line. If BY variables are used in the input data set (DATA=, HISTORY=, or TABLE=), the same BY variable structure must be used in the VREF= data set unless you specify the NOBYREF option.

This option can be used to add warning limits to be displayed on a chart.

Related options are CVREF=, LVREF=, NOBYREF, VREFCHAR=, VREFLABELS=, and VREFLABPOS=.

VREF2=*value-list*

VREF2=*SAS-data-set*

draws reference lines perpendicular to the vertical axis on the secondary chart. The conventions for specifying the VREF2= option are identical to those for specifying the VREF= option. Related options are CVREF=, LVREF=, NOBYREF, VREFCHAR=, VREF2LABELS=, and VREFLABPOS=. The VREF2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

VREF2LABELS='*label1*' **. . .** '*labeln*'

VREF2LAB='*label1*' **. . .** '*labeln*'

specifies labels for the reference lines requested by the VREF2= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters. The VREF2LABELS= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

VREFLABELS='*label1*' **. . .** '*labeln*'

specifies labels for the reference lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=*n*

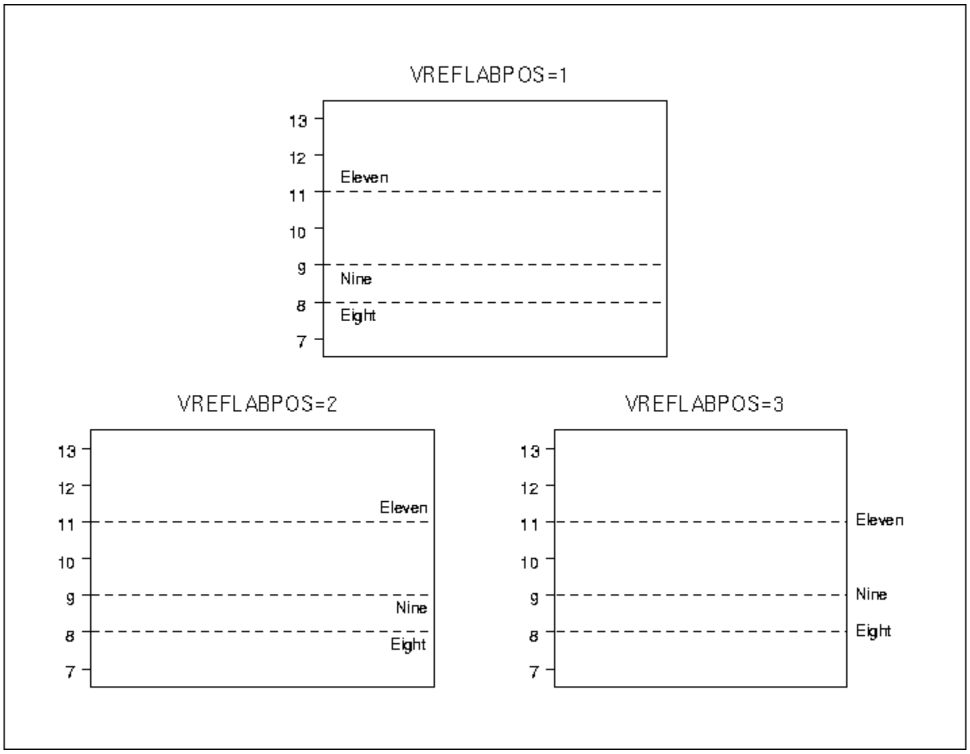
specifies the horizontal position of the VREFLABEL= and VREF2LABEL= labels, as described in the following table. By default, $n = 1$.

n	**Label Position**
1	left-justified in subplot area
2	right-justified in subplot area
3	left-justified in right margin

Figure 17.141 illustrates label positions for values of the VREFLABPOS= option when the VREF= and VREFLABELS= options are as follows:

```
vref       = 8 9 11
vreflabels = 'Eight' 'Nine' 'Eleven'
```

Figure 17.141 Positions for Reference Line Labels



VZERO

forces the origin to be included in the vertical axis for a primary chart.

VZERO2

forces the origin to be included in the vertical axis for a secondary chart.

WESTGARD=*index-list*

requests that one or more of the Westgard rules be applied. The Westgard rules are tests for special causes that were developed specifically for use in healthcare laboratories. Westgard (2002) describes the rules and their proper use in detail.

The Westgard rules are similar to the Western Electric rules that are implemented by the TESTS= option. They detect unusual patterns of points plotted on the primary control chart. The patterns are defined in terms of the zones A, B, and C that are illustrated in Figure 17.178. The occurrence of one or more of these patterns suggests the presence of a special cause of variation.

Table 17.88 lists the Westgard tests that you can request.

Table 17.88 Westgard Rules

Index	Notation	Pattern Description
1	1:2s	One point in Zone A or beyond
2	1:3s	One point beyond Zone A (outside the control limits)
3	2:2s	Two points in a row in Zone A or beyond on the same side of the central line
4	R:4s	At least one point in Zone A or beyond on each side of the central line
5	4:1s	Four points in a row in Zone B or beyond on the same side of the central line
6	10x	Ten points in a row on the same side of the central line

WHISKERPERCENTILE=*pctl*

specifies that the whiskers of the box-and-whisker plots be drawn to the *pctl* and $100 - \mathit{pctl}$ percentiles. For example, if you specify WHISKERPERCENTILE=10 the whiskers are drawn to the 10th and 90th percentiles. Observations that lie beyond the whiskers are outliers, and there are no far outliers. This option in available only in the BOXCHART statement.

XSYMBOL=‘*label*’

XSYMBOL=*keyword*

specifies a label for the central line in an $\bar{X}$ chart or a median chart. You can use the option in the following ways:

- You can specify a quoted *label* up to 16 characters.
- You can specify one of the *keywords* listed in the following table. Each *keyword* requests a label of the form *symbol=value*, where *symbol* is the symbol given in the table, and *value* is the value of the central line. If the central line is not constant, only the symbol is displayed.

Keyword	Symbol Used in Graphics	Symbol Used in Line Printer Charts
MBAR	$\overline{\mathrm{M}}$	$\overline{\mathrm{M}}$
MTIL	$\tilde{\mathrm{M}}$	$\tilde{\mathrm{M}}$
MU	μ	MU
MU0	μ_0	MU0
XBAR	$\overline{\mathrm{X}}$	$\overline{\mathrm{X}}$
XBAR2	$\overline{\overline{\mathrm{X}}}$	$\overline{\overline{\mathrm{X}}}$
XBARPM	$\overline{\mathrm{X}}'$	$\overline{\mathrm{X}}'$
XBAR0	$\overline{\mathrm{X}}_0$	$\overline{\mathrm{X}}0$
XBAR0PM	$\overline{\mathrm{X}}'_0$	$\overline{\mathrm{X}}0'$

For the IRCHART statement, the default *keyword* is XBAR. For the MCHART and MRCHART statements, the default *keyword* is MBAR. For all other chart statements, the default *keyword* is XBAR2. The XSYMBOL= option is available in the BOXCHART, IRCHART, MCHART, MRCHART, XCHART, XRCHART, and XSCHART statements.

YPCT1=*value*

specifies a percent (ranging from 0 to 100) that determines the length of the vertical axis for the primary chart in proportion to the sum of the lengths of the vertical axes for the primary and secondary charts. For example, you can specify YPCT1=50 in an XRCHART statement to request that the vertical axes for the $\bar{X}$ and R charts have the same length. The default *value* is 60. The YPCT1= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

YSCALE=PERCENT

scales the vertical axis on a p chart in percent units. The YSCALE= option is available only in the PCHART statement.

ZEROSTD

ZEROSTD=NOLIMITS

specifies that a control chart is to be constructed and displayed regardless of whether the estimated process standard deviation $\hat{\sigma}$ is zero. When $\hat{\sigma}$ is zero, the control limits are degenerate (collapsed around the central line), and the chart simply serves as a placeholder, particularly when a series of charts is to be created. Specify ZEROSTD=NOLIMITS to suppress the display of the degenerate limits. By default, a chart is not displayed when $\hat{\sigma}$ is zero.

ZONE2LABELS

adds the labels A, B, and C to zone lines requested with the ZONES2 or ZONE2VALUES options. The ZONE2LABELS option is available in the MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements.

ZONE2VALUES

labels R or s chart zones lines with their values. If the ZONE2VALUES option is specified the ZONES2 option is not required.

ZONELABELS

adds the labels A, B, and C to zone lines requested with the ZONES or ZONEVALUES options. The ZONELABELS option is not available in the RCHART or SCHART statements.

ZONES

adds lines to a primary chart that delineate zones A, B, and C for standard tests requested with the TESTS= option. Related options are CZONES= and ZONELABELS. The ZONES option is not available in the RCHART or SCHART statements.

ZONES2

adds lines to an R or s chart that delineate zones A, B, and C for tests requested with the TESTS2= option. Related options are CZONES= and ZONE2LABELS. The ZONES2 option is available in the MRCHART, RCHART, SCHART, XRCHART, and XSCHART statements.

ZONEVALPOS=*n*

specifies the horizontal position of the ZONEVALUES= and ZONE2VALUES= labels, as described in the following table. By default, $n = 1$.

n	**Label Position**
1	left-justified in subplot area
2	right-justified in subplot area
3	left-justified in right margin

ZONEVALUES
labels the primary chart zones lines with their values. If the ZONEVALUES option is specified the ZONES option is not required.

Options for ODS Graphics

BLOCKREFTRANSPARENCY=*value*

PHASEREFTRANSPARENCY=*value*

REFFILLTRANSPARENCY=*value*
specifies the wall fill transparency for blocks and phases when transparency is used in ODS Graphics output. The *value* must be between 0 and 1, where 0 is completely opaque and 1 is completely transparent. The default wall fill transparency is 0.85.

BOXTRANSPARENCY=*value*
specifies the box fill transparency for box-and-whisker charts when transparency is used in ODS Graphics output. The *value* must be between 0 and 1, where 0 is completely opaque and 1 is completely transparent. The default box fill transparency is 0.25.

INFILLTRANSPARENCY=*value*
specifies the control limit infill transparency when transparency is used in ODS Graphics output. The *value* be between 0 and 1, where 0 is completely opaque and 1 is completely transparent. The default control limit infill transparency is 0.75.

MARKERS
plots subgroup points with markers. By default, subgroup points are plotted with markers only by the BOXCHART statement. On other types of charts, subgroup points are connected by line segments and are not plotted with markers by default.

NOBLOCKREF

NOPHASEREF

NOREF
suppresses block and phase reference lines from ODS Graphics output. By default, block and phase reference lines are drawn when ODS Graphics is in effect.

NOBLOCKREFFILL

NOPHASEREFFILL

NOREFFILL
suppresses the block and phase wall fills from ODS Graphics output. By default, block and phase walls are filled when ODS Graphics is in effect.

NOBOXFILLLEGEND

NOFILLLEGEND

NOSTARFILLLEGEND
suppresses the legend for the levels of a BOXFILL= or STARFILL= variable in ODS Graphics output.

NOTRANSPARENCY

disables transparency in ODS Graphics output, so that all graph features are opaque. By default, transparency is enabled when ODS Graphics is in effect.

ODSFOOTNOTE=FOOTNOTE | FOOTNOTE1 | '*string*'

adds a footnote to ODS Graphics output. If you specify the FOOTNOTE (or FOOTNOTE1) keyword, the value of SAS FOOTNOTE statement is used the as the graph footnote. If you specify a quoted string, that is used as the footnote. The quoted string can contain any of the following escaped characters, which are replaced with the appropriate values from the analysis:

\n	process variable name
\l	process variable label (or name if the process variable has no label)
\x	subgroup variable name
\s	subgroup variable label (or name if the subgroup variable has no label)

ODSFOOTNOTE2=FOOTNOTE2 | '*string*'

adds a secondary footnote to ODS Graphics output. If you specify the FOOTNOTE2 keyword, the value of SAS FOOTNOTE2 statement is used as the secondary graph footnote. If you specify a quoted string, that is used as the secondary footnote. The quoted string can contain any of the following escaped characters, which are replaced with the appropriate values from the analysis:

\n	process variable name
\l	process variable label (or name if the process variable has no label)
\x	subgroup variable name
\s	subgroup variable label (or name if the subgroup variable has no label)

ODSLEGENDEXPAND

specifies that legend entries contain all levels observed in the data. By default, a legend shows only the levels used on the current page.

ODSTITLE=TITLE | TITLE1 | NONE | DEFAULT | LABELFMT | '*string*'

specifies a title for ODS Graphics output.

TITLE (or TITLE1)	uses the value of SAS TITLE statement as the graph title.
NONE	suppresses all titles from the graph.
DEFAULT	uses the default ODS Graphics title (a descriptive title consisting of the plot type and the process variable name.)
LABELFMT	uses the default ODS Graphics title with the variable label instead of the variable name.

If you specify a quoted string, that is used as the graph title. The quoted string can contain any of the following escaped characters, which are replaced with the appropriate values from the analysis:

\n	process variable name
\l	process variable label (or name if the process variable has no label)

\x	subgroup variable name
\s	subgroup variable label (or name if the subgroup variable has no label)

ODSTITLE2=TITLE2 | '*string*'

specifies a secondary title for ODS Graphics output. If you specify the TITLE2 keyword, the value of SAS TITLE2 statement is used as the secondary graph title. If you specify a quoted string, that is used as the secondary title. The quoted string can contain any of the following escaped characters, which are replaced with the appropriate values from the analysis:

\n	process variable name
\l	process variable label (or name if the process variable has no label)
\x	subgroup variable name
\s	subgroup variable label (or name if the subgroup variable has no label)

OUTFILLTRANSPARENCY=*value*

specifies the control limit outfill transparency when transparency is used in ODS Graphics output. The *value* must be between 0 and 1, where 0 is completely opaque and 1 is completely transparent. The default control limit outfill transparency is 0.75.

OUTHIGHURL=*variable*

specifies a variable whose values are URLs to be associated with outlier points above the upper fence on a schematic box chart when ODS Graphics output is directed into HTML.

OUTLOWURL=*variable*

specifies a variable whose values are URLs to be associated with outlier points below the lower fence on a schematic box chart when ODS Graphics output is directed into HTML.

OVERLAY2URL=(*variable-list*)

specifies variables whose values are URLs to be associated with points on secondary chart overlays. These URLs are associated with points on an overlay plot when ODS Graphics output is directed into HTML. Variables in the OVERLAY2URL= list are matched with variables in the corresponding positions in the OVERLAY2= list. The value of the OVERLAY2URL= variable should be the same for each observation with a given value of the subgroup variable.

OVERLAYURL=(*variable-list*)

specifies variables whose values are URLs to be associated with points on primary chart overlays. These URLs are associated with points on an overlay plot when ODS Graphics output is directed into HTML. Variables in the OVERLAYURL= list are matched with variables in the corresponding positions in the OVERLAY= list. The value of the OVERLAYURL= variable should be the same for each observation with a given value of the subgroup variable.

PHASEBOXLABELS

draws phase labels as titles along the top of phase boxes.

PHASEPOS=*n*

specifies the vertical position of the phase legend. Values of *n* and the corresponding positions are as follows. By default, PHASEPOS=1.

n	**Legend Position**
1	Top of chart, offset from axis frame
2	Top of chart, immediately above axis frame
3	Bottom of chart, immediately above horizontal axis
4	Bottom of chart, below horizontal axis label

PHASEREFLEVEL=INNER | OUTER | NONE
enables you to associate phase reference lines (block reference lines) with either the innermost or the outermost level. The default value is INNER.

POINTSURL=*variable*
specifies a variable whose values are URLs to be associated with points on a box chart when the BOXSTYLE= value is POINTS, POINTSJOIN, POINTSBOX, POINTSID, or POINTSJOINID. These URLs are associated with points on a box chart when ODS Graphics output is directed into HTML.

SIMULATEQCFONT
draws the central line labels using a simulated software font rather than a hardware font.

STARTRANSPARENCY=*value*
specifies the star fill transparency when transparency is used in ODS Graphics output. The *value* must be between 0 and 1, where 0 is completely opaque and 1 is completely transparent. The default star fill transparency is 0.25.

URL=*variable*
specifies URLs as values of the specified character variable (or formatted values of a numeric variable). These URLs are associated with subgroup points on a primary control chart when ODS Graphics output is directed into HTML. The value of the URL= variable should be the same for each observation with a given value of the subgroup variable.

URL2=*variable*
specifies URLs as values of the specified character variable (or formatted values of a numeric variable). These URLs are associated with subgroup points on a secondary control chart when ODS Graphics output is directed into HTML. The value of the URL2= variable should be the same for each observation with a given value of the subgroup variable.

WBOXES=*n*
specifies the width in pixels for the outlines of the box-and-whisker plots created with the BOXCHART statement in ODS Graphics output.

Options for Traditional Graphics

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*
specifies an ANNOTATE= type data set, as described in *SAS/GRAPH: Reference*, that enhances a primary chart. The ANNOTATE= data set specified in a chart statement enhances all charts created by that particular statement. You can also specify an ANNOTATE= data set in the PROC SHEWHART statement to enhance all primary charts created by the procedure.

ANNOTATE2=*SAS-data-set*

ANNO2=*SAS-data-set*

specifies an ANNOTATE= type data set, as described in *SAS/GRAPH: Reference*, that enhances a secondary chart. The ANNOTATE2= data set specified in a chart statement enhances all charts created by that particular statement. You can also specify an ANNOTATE2= data set in the PROC SHEWHART statement to enhance all secondary charts created by the procedure.

This option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements and in the BOXCHART, MCHART, and XCHART statements with the TRENDVAR= option.

BILEVEL

arranges the Shewhart chart in two levels (rather than the default of one level) so that twice as much data can be displayed on a page or screen. The second level is a continuation of the first level, and this arrangement is continued on subsequent pages until all the subgroups are displayed. You use the NPANELPOS= option to control the number of subgroup positions in each level. If you specify the BILEVEL option in a chart statement that produces primary and secondary charts, you must also specify the SEPARATE option.

CAXIS=*color*

CAXES=*color*

CA=*color*

specifies the color for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement.

CBLOCKLAB=*color* **| (***color-list***)**

specifies fill colors for the frames that enclose the *block-variable* labels in a block legend. By default, these areas are not filled. Colors in the CBLOCKLAB= list are matched with *block-variables* in the order in which they appear in the chart statement. Related options are BLOCKLABELPOS=, BLOCKLABTYPE=, BLOCKREP, BLOCKPOS=, and CBLOCKVAR=.

CBLOCKVAR=*variable* **| (***variable-list***)**

specifies variables whose values are colors for filling the background of the legend associated with *block-variables*. Each CBLOCKVAR= variable must be a character variable of no more than eight characters in the input data set (a DATA=, HISTORY=, or TABLE= data set). A list of CBLOCKVAR= variables must be enclosed in parentheses. You can use the BLOCKVAR= option to specify that the block variable legend be filled with different colors from the ODS style.

The procedure matches the CBLOCKVAR= variables with *block-variables* in the order specified. That is, each block legend will be filled with the color value of the CBLOCKVAR= variable of the first observation in each block. In general, values of the *i*th CBLOCKVAR= variable are used to fill the block of the legend corresponding to the *i*th *block-variable*. For examples of the CBLOCKVAR= option, see Figure 17.146 and Figure 17.147.

By default, fill colors are not used for the *block-variable* legend. The CBLOCKVAR= option is available only when *block-variables* are used in the chart statement.

CBOXES=*color*

CBOXES=(*variable***)**

specifies the colors for the outlines of the box-and-whisker plots created with the BOXCHART statement. You can use one of the following approaches:

- You can specify CBOXES=*color* to provide a single outline color for all the box-and-whisker plots.
- You can specify CBOXES=(*variable*) to provide a distinct outline color for *each* box-and-whisker plot as the value of the *variable*. The *variable* must be a character variable of length 8 less in the input data set, and its values must be valid SAS/GRAPH color names. The outline color of the plot displayed for a particular subgroup is the value of the *variable* in the observations corresponding to this subgroup. Note that if there are multiple observations per subgroup in the input data set, the values of the *variable* should be identical for all the observations in a given subgroup.

You can use the BOXES= option to group boxes to be drawn with different colors from the ODS style.

The CBOXES= option is available only in the BOXCHART statement.

CBOXFILL=*color*

CBOXFILL=(*variable*)

specifies the interior fill colors for the box-and-whisker plots created with the BOXCHART statement. You can use one of the following approaches:

- You can specify CBOXFILL=*color* to provide a single color for all of the box-and-whisker plots.
- You can specify CBOXFILL=(*variable*) to provide a distinct color for *each* box-and-whisker plot as the value of the *variable*. The *variable* must be a character variable of length 8 or less in the input data set, and its values must be valid SAS/GRAPH color names (or the value *EMPTY*, which you can use to suppress color filling). The interior color of the plot displayed for a particular subgroup is the value of the *variable* in the observations corresponding to this subgroup. Note that if there are multiple observations per subgroup in the input data set, the values of the *variable* should be identical for all the observations in a given subgroup.

You can use the BOXFILL= option to group boxes to be filled with different colors from the ODS style. By default, all boxes are filled with a single color from the ODS style. The CBOXFILL= option is available only in the BOXCHART statement.

CCLIP=*color*

specifies a color for the plotting symbol that is specified with the CLIPSYMBOL= option to mark clipped points. The default color is the color specified in the COLOR= option in the SYMBOL1 statement.

CCONNECT=*color*

specifies the color for the line segments connecting points on the chart. The default color is the color specified in the COLOR= option in the SYMBOL1 statement. This option is not applicable in the BOXCHART statement unless you also specify the BOXCONNECT option.

CCOVERLAY=(*color-list*)

specifies the colors for the line segments connecting points on primary chart overlays. Colors in the CCOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list. By default, points are connected by line segments of the same color as the plotted points. You can specify the value NONE to suppress the line segments connecting points on an overlay.

CCOVERLAY2=(*color-list***)**
specifies the colors for the line segments connecting points on secondary chart overlays. Colors in the CCOVERLAY2= list are matched with variables in the corresponding positions in the OVERLAY2= list. By default, points are connected by line segments of the same color as the plotted points. You can specify the value NONE to suppress the line segments connecting points on an overlay.

CFRAME=*color*

CFRAME=(*color-list***)**
specifies the colors for filling the rectangle enclosed by the axes and the frame. By default, this area is not filled. The CFRAME= option cannot be used in conjunction with the NOFRAME option.

You can specify a single *color* to fill the entire area. Alternatively, if you are displaying phases (blocks) of data read with the READPHASES= option, you can specify a *color-list* with the CFRAME= option to fill the sub-rectangles of the framed area corresponding to the phases. The colors, in order of specification, are applied to the sub-rectangles starting from left to right. You can use the value *EMPTY* in the *color-list* to avoid filling a particular sub-rectangle. If the number of colors is less than the number of phases, the colors are applied cyclically. The colors are also used for phase legends requested with the PHASELEGEND option.

CGRID=*color*
specifies the color for the grid requested by the ENDGRID or GRID option. By default, the grid is the same color as the axes.

CHREF=*color*
specifies the color for the lines requested by the HREF= and HREF2= options.

CLABEL=*color*
specifies the color for labels produced by the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and OUTLABEL2= options.

CLIMITS=*color*
specifies the color for the control limits, the central line, and the labels for these lines.

CLIPLEGPOS=TOP | BOTTOM
specifies the position for the legend that indicates the number of clipped points when the CLIPFACTOR= option is used. The keywords TOP and BOTTOM position the legend at the top or bottom of the chart, respectively. Do not specify CLIPLEGPOS=TOP together with the PHASELEGEND option or the BLOCKPOS=1 or BLOCKPOS=2 options. By default, CLIPLEGPOS=BOTTOM.

CLIPSYMBOL=*symbol*
specifies a plot symbol used to identify clipped points on the chart and in the legend when the CLIPFACTOR= option is used. You should use this option in conjunction with the CLIPFACTOR= option. The default *symbol* is CLIPSYMBOL=SQUARE.

CLIPSYMBOLHT=*value*
specifies the height for the symbol marker used to identify clipped points on the chart when the CLIPFACTOR= option is used. The default is the height specified with the H= option in the SYMBOL statement.

For general information about clipping options, refer to "Clipping Extreme Points" on page 2059.

CNEEDLES=*color*

requests that points are to be connected to the central line with vertical line segments (needles) and specifies the color of the needles. You can use needles to visually represent the process as a series of shocks or vertical displacements away from a constant mean. See Figure 17.168 for an example. The CNEEDLES= option is available in all chart statements except the BOXCHART statement.

COUTFILL=*color*

specifies the fill color for the areas outside the control limits that lie between the connected points and the control limits and are bounded by connecting lines. This option is useful for highlighting out-of-control points. See Figure 17.202 for an example. By default, these areas are not filled. You can use the OUTFILL option to fill this area with an appropriate color from the ODS style. Note that you can use the CINFILL= option to fill the area inside the control limits.

COVERLAY=(*color-list*)

specifies the colors used to plot primary chart overlay variables. Colors in the COVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list.

COVERLAY2=(*color-list*)

specifies the colors used to plot secondary chart overlay variables. Colors in the COVERLAY2= list are matched with variables in the corresponding positions in the OVERLAY2= list.

COVERLAYCLIP=*color*

specifies the color used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

CPHASELEG=*color*

specifies a text color for the phase labels requested with the PHASELEGEND option. By default, if you specify a list of fill colors with the CFRAME= option, these colors are used for the corresponding phase labels, otherwise, the CTEXT= color is used for the phase labels.

CSTARCIRCLES=*color*

specifies a color for the circles requested with the STARCIRCLES= option. See "Displaying Auxiliary Data with Stars" on page 2042. By default, the color specified with the CSTARS= option is used.

CSTARFILL=*color*

CSTARFILL=(*variable*)

specifies a color or colors for filling the interior of stars requested with the STARVERTICES= option. You can use one of the following approaches:

- Specify a single color to be used for all stars with CSTARFILL=*color*.
- Specify a distinct color for *each* star (or subsets of stars) by providing the colors as values of a variable specified with CSTARFILL=(*variable*). The variable must be a character variable of length 8 or less in the input data set, and its values must be valid SAS/GRAPH colors or the value *EMPTY*. The color for the star positioned at the *i*th subgroup on the chart is the value of the CSTARFILL= *variable* in the observations corresponding to the *i*th subgroup. Note that if there are multiple observations per subgroup in the input data set (for instance, if you are using the XRCHART statement in the SHEWHART procedure to analyze observations from a DATA= input data set), the values of the CSTARFILL= *variable* should be identical for all the observations in a given subgroup.

See "Displaying Auxiliary Data with Stars" on page 2042.

You can use the STARFILL= option to group stars to be filled with different colors from the ODS style. By default, all stars are filled with a single color from the ODS style.

CSTARS=*color*
CSTARS=(*variable*)
specifies a color or colors for the outlines of stars requested with the STARVERTICES= option.

You can use one of the following approaches:

- You can specify a single color to be used for all the stars on the chart with CSTARS=*color*.
- You can specify a distinct outline color for *each* star (or subsets of stars) by providing the colors as values of a variable specified with CSTARS=(*variable*). The variable must be a character variable of length 8 or less in the input data set. The outline color for the star positioned at the *i*th subgroup on the chart is the value of the CSTARS=*variable* in the observations corresponding to the *i*th subgroup. Note that if there are multiple observations per subgroup in the input data set (for instance, if you are using the XRCHART statement in the SHEWHART procedure to analyze observations from a DATA= input data set), the values of the CSTARS= *variable* should be identical for all the observations in a given subgroup.

See "Displaying Auxiliary Data with Stars" on page 2042.

You can use the STARS= option to group stars to be drawn with different colors from the ODS style. By default, all stars are drawn with a single color from the ODS style.

CTESTLABBOX=*color*
specifies the color for boxes enclosing labels for positive tests for special causes requested with the TESTLABBOX option. If you use the CTESTLABBOX= option, you do not need to specify the TESTLABBOX option.

CTESTS=*color* | *test-color-list*
CTEST=*color* | *test-color-list*
specifies colors for labels indicating points where a test is positive.

- You can specify the *color* for the labels used to identify points at which tests for special causes specified in the TESTS= option are positive. For Tests 2 through 8, this color is also used for the line segments that connect patterns of points for which a test is positive.
- You can specify the *test-color-list* to enable different colors to be used for the labels and highlighted line segments associated with different tests for special causes. Any positive tests for which no specific CTESTS= value is specified are displayed using the general CTESTS= color. A non-default general CTESTS= color can be specified using the CTESTS=*color* syntax.

The following options request the standard tests for special causes 1 through 4 and one user-defined test designated B.

```
TESTS  = 1 to 4 M(K=4 DIR=DEC Code=B);
CTESTS = green;
CTESTS = (1 purple 3 yellow B blue);
```

Test 1 will be displayed in purple, Test 3 in yellow, and Test B in blue. Tests 2 and 4 will be displayed in green, the general CTESTS= color.

CTESTSYMBOL=*color*

CTESTSYM=*color*

specifies the color of the symbol used to plot subgroups with positive tests for special causes.

CTEXT=*color*

specifies the color for tick mark values and axis labels. This color is also used for the sample size legend and for the control limit legend. The default color is the color specified in the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*

CV=*color*

specifies the color for reference lines requested by the VREF= and VREF2= options.

CZONES=*color*

requests lines marking zones A, B, and C for the tests for special causes (see the TESTS= option) and specifies the *color* for these lines. This color is also used for labels requested with the ZONELABELS option.

DESCRIPTION=‘*string***’**

DES=‘*string***’**

specifies a description, up to 256 characters long, for the GRSEG catalog entry for the primary chart. The default *string* is the variable name. A related option is NAME=.

DESCRIPTION2=‘*string***’**

DES2=‘*string***’**

specifies a description, up to 256 characters long, for the GRSEG catalog entry for the secondary chart. The default *string* is the variable name. The DESCRIPTION2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements, and it is used in conjunction with the SEPARATE option. A related option is NAME2=.

ENDGRID

adds a grid to the rightmost portion of the chart, beginning with the first labeled major tick mark position that follows the last plotted point. This grid is useful in situations where you want to add points by hand after the chart is created. You can use the HAXIS= option to force space to be added to the horizontal axis.

FONT=*font*

specifies a software font for labels and legends. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the GOPTIONS statement. Hardware characters are used by default.

HEIGHT=*value*

specifies the height (in vertical screen percent units) of the text for axis labels and legends. This *value* takes precedence over the HTEXT= value specified in the GOPTIONS statement. This option is recommended for use with software fonts specified with the FONT= option or with the FTEXT= option in the GOPTIONS statement. Related options are LABELHEIGHT= and TESTHEIGHT=.

HMINOR=*n*

HM=*n*

specifies the number of minor tick marks between each major tick mark on the horizontal axis. Minor tick marks are not labeled. The default is 0.

HTML=*variable*

specifies a variable whose values create links associated with subgroup points on a primary control chart when traditional graphics output is directed into HTML. You can specify a character variable or formatted numeric variable. The value of the HTML= variable should be the same for each observation with a given value of the subgroup variable. See the section "Interactive Control Charts: SHEWHART Procedure" on page 2136 for more information.

HTML2=*variable*

specifies a variable whose values create links associated with subgroup points on a secondary control chart when traditional graphics output is directed into HTML. You can specify a character variable or formatted numeric variable. The value of the HTML2= variable should be the same for each observation with a given value of the subgroup variable. See the section "Interactive Control Charts: SHEWHART Procedure" on page 2136 for more information.

HTML_LEGEND=*variable*

specifies HTML links as values of the specified character variable (or formatted values of a numeric variable). These links are associated with symbols in the legend for the levels of a *symbol-variable*. The value of the HTML_LEGEND= variable should be the same for each observation with a given value of *symbol-variable*.

IDCOLOR=*color*

specifies the color of the symbol marker used to identify outliers in schematic box-and-whisker plots produced with the BOXCHART statement when you use one of the following options: BOXSTYLE=SCHEMATIC, BOXSTYLE=SCHEMATICID, and BOXSTYLE=SCHEMATICIDFAR. The default *color* is the color specified with the CBOXES= option. The IDCOLOR option is available only in the BOXCHART statement.

IDCTEXT=*color*

specifies the color for the text used to label outliers or indicate process variable values when you specify one of the keywords SCHEMATICID, SCHEMATICIDFAR, POINTSID, or POINTSJOINID with the BOXSTYLE= option. The default is the color specified with the CTEXT= option.

IDFONT=*font*

specifies the font for the text used to label outliers or indicate process variable values when you specify one of the keywords SCHEMATICID, SCHEMATICIDFAR, POINTSID, or POINTSJOINID with the BOXSTYLE= option. The default *font* is SIMPLEX.

IDHEIGHT=*value*

specifies the height for the text used to label outliers or indicate process variable values when you specify one of the keywords SCHEMATICID, SCHEMATICIDFAR, POINTSID, or POINTSJOINID with the BOXSTYLE= option. The default is the height specified with the HTEXT= option in the GOPTIONS statement.

IDSYMBOL=*symbol*

specifies the symbol marker used to identify outliers in schematic box-and-whisker plots produced with the BOXCHART statement when you use one of the following options: BOXSTYLE=SCHEMATIC, BOXSTYLE=SCHEMATICID, and BOXSTYLE=SCHEMATICIDFAR. The default *symbol* is SQUARE. The IDSYMBOL= option is available only in the BOXCHART statement.

IDSYMBOLHEIGHT=*value*

specifies the height of the symbol marker used to identify outliers in schematic box-and-whisker plots produced with the BOXCHART statement. This option is available only in the BOXCHART statement.

LABELANGLE=*angle*

specifies the angle at which labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and OUTLABEL2= options are drawn. A positive angle rotates the labels counterclockwise; a negative angle rotates them clockwise. By default, labels are oriented horizontally.

LABELFONT=*font*

TESTFONT=*font*

specifies a software font for labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, OUTLABEL2=, STARLABEL=, TESTLABEL=, and TESTLABEL*n*= options. Hardware characters are used by default.

LABELHEIGHT=*value*

TESTHEIGHT=*value*

specifies the height (in vertical percent screen units) for labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, OUTLABEL2=, STARLABEL=, TESTLABEL=, and TESTLABEL*n*= options. The default height is the height specified with the HEIGHT= option or the HTEXT= option in the GOPTIONS statement.

LBOXES=*linetype*

LBOXES=(*variable***)**

specifies the line types for the outlines of the box-and-whisker plots created with the BOXCHART statement. You can use one of the following approaches:

- You can specify LBOXES=*linetype* to provide a single *linetype* for all of the box-and-whisker plots.
- You can specify LBOXES=(*variable*) to provide a distinct line type for *each* box-and-whisker plot. The *variable* must be a numeric variable in the input data set, and its values must be valid SAS/GRAPH *linetype* values (numbers ranging from 1 to 46). The line type for the plot displayed for a particular subgroup is the value of the *variable* in the observations corresponding to this subgroup. Note that if there are multiple observations per subgroup in the input data set, the values of the *variable* should be identical for all of the observations in a given subgroup.

The default value is 1, which produces solid lines. The LBOXES= option is available only in the BOXCHART statement.

LENDGRID=*n*

specifies the line type for the grid requested with the ENDGRID option. The default is $n = 1$, which produces a solid line. If you use the LENDGRID= option, you do not need to specify the ENDGRID option.

LGRID=*n*
specifies the line type for the grid requested with the GRID option. The default is $n = 1$, which produces a solid line. If you use the LGRID= option, you do not need to specify the GRID option.

LHREF=*linetype*
LH=*linetype*
specifies the line type for reference lines requested with the HREF= and HREF2= options. The default is 2, which produces a dashed line.

LLIMITS=*linetype*
specifies the line type for control limits. The default is 4, which produces a dashed line.

LOVERLAY=(*linetypes*)
specifies line types for the line segments connecting points on primary chart overlays. Line types in the LOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list.

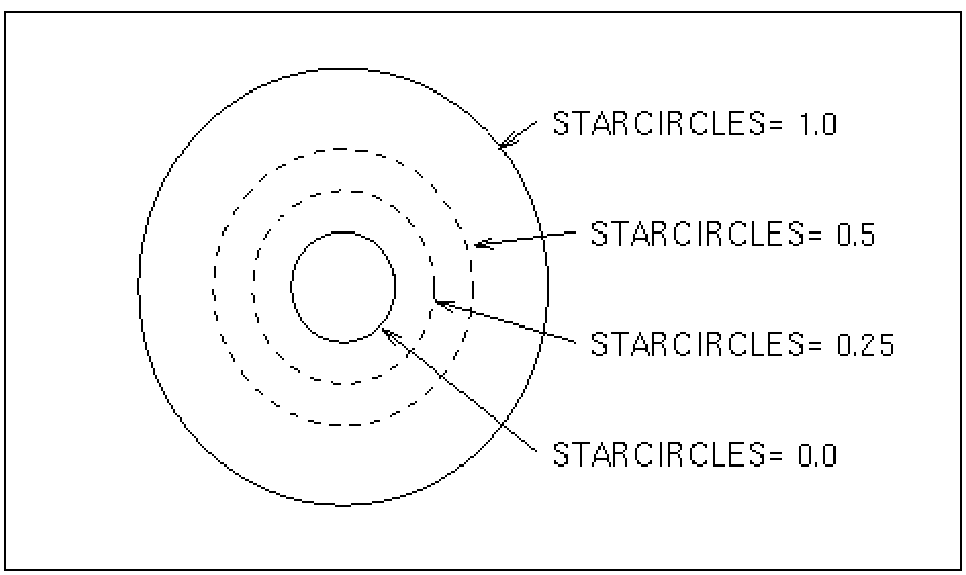
LOVERLAY2=(*linetypes*)
specifies line types for the line segments connecting points on secondary chart overlays. Line types in the LOVERLAY2= list are matched with variables in the corresponding positions in the OVERLAY2= list.

LSTARCIRCLES=*linetypes*
specifies one or more line types for the circles requested with the STARCIRCLES= option. The number of line types should match the number of circles requested, and the line types are paired with the circles in the order specified. The default *linetype* is 1, which produces a solid line.

Figure 17.142 illustrates circles displayed by the following LSTARCIRCLES= and STARCIRCLES= options:

```
starcircles  = 0.0 1.0 0.25 0.5
lstarcircles = 1 1 2 2
```

Figure 17.142 Line Types for Reference Circles



LSTARS=*linetype*

LSTARS=(*variable***)**

specifies the line types for the outlines of stars requested with the STARVERTICES= option. You can use one of the following approaches:

- You can specify LSTARS=*linetype* to provide a single line type for all of the stars.
- You can specify LSTARS=(*variable*) to provide a distinct line type for *each* star. The variable must be a numeric variable in the input data set, and its values must be valid SAS/GRAPH line types. The line type for the star positioned at a particular subgroup is the value of the *variable* in the observations corresponding to this subgroup. Note that if there are multiple observations per subgroup in the input data set, the *variable* values should be identical for all of the observations in a given subgroup.

See "Displaying Auxiliary Data with Stars" on page 2042. The default *linetype* is 1, which produces a solid line.

LTESTS=*linetype*

LTEST=*linetype*

specifies the line type for the line segments that connect patterns of points for which a test for special causes (requested with the TESTS= option) is positive. The default is 1, which produces a solid line.

LVREF=*linetype*

LV=*linetype*

specifies the line type for reference lines requested by the VREF= and VREF2= options. The default is 2, which produces a dashed line.

LZONES=*n*

specifies the line type for lines that delineate zones A, B, and C for standard tests requested with the TESTS= and/or TESTS2= options. The default is $n = 2$, which produces a dashed line.

NAME='*string*'

specifies the name of the GRSEG catalog entry for the primary chart, and the name of the graphics output file if one is created. The name can be up to 256 characters long, but the GRSEG name is truncated to eight characters. The default name is 'SHEWHART'. A related option is DESCRIPTION=.

NAME2='*string*'

specifies the name of the GRSEG catalog entry for the secondary chart, and the name of the graphics output file if one is created. The name can be up to 256 characters long, but the GRSEG name is truncated to eight characters. The default name is 'SHEWHART'. The NAME2= option is available in the IRCHART, MRCHART, XRCHART, and XSCHART statements, and it is used in conjunction with the SEPARATE option. A related option is DESCRIPTION2=.

NOFRAME

suppresses the default frame drawn around the chart.

NOLIMITSFRAME

suppresses the default frame for the control limit information that is displayed across the top of the chart when multiple sets of control limits with distinct multiples of σ and nominal control limit sample sizes are read from a LIMITS= data set.

NOPHASEFRAME
suppresses the default frame for the legend requested by the PHASELEGEND option.

NOVANGLE
requests vertical axis labels that are strung out vertically. By default, the labels are drawn at an angle of 90 degrees if a software font is used.

OUTHIGHHTML=*variable*
specifies a variable whose values create links to be associated with outlier points above the upper fence on a schematic box chart when traditional graphics output is directed into HTML.

OUTLOWHTML=*variable*
specifies a variable whose values create links to be associated with outlier points below the lower fence on a schematic box chart when traditional graphics output is directed into HTML.

OVERLAY2HTML=(*variable-list***)**
specifies variables whose values create links to be associated with points on secondary chart overlays. These links are associated with points on an overlay plot when traditional graphics output is directed into HTML. Variables in the OVERLAY2HTML= list are matched with variables in the corresponding positions in the OVERLAY2= list. The value of the OVERLAY2HTML= variable should be the same for each observation with a given value of the subgroup variable.

OVERLAY2SYM=(*symbol-list***)**
specifies symbols used to plot overlays on a secondary control chart. Symbols in the OVERLAY2SYM= list are matched with variables in the corresponding positions in the OVERLAY2= list.

OVERLAY2SYMHT=(*value-list***)**
specifies the heights of symbols used to plot overlays on a secondary control chart. Heights in the OVERLAY2SYMHT= list are matched with variables in the corresponding positions in the OVERLAY2= list.

OVERLAYCLIPSYM=*symbol*
specifies the symbol used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

OVERLAYCLIPSYMHT=*value*
specifies the height for the symbol used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

OVERLAYHTML=(*variable-list***)**
specifies variables whose values create links to be associated with points on primary chart overlays. These links are associated with points on an overlay plot when traditional graphics output is directed into HTML. Variables in the OVERLAYHTML= list are matched with variables in the corresponding positions in the OVERLAY= list. The value of the OVERLAYHTML= variable should be the same for each observation with a given value of the subgroup variable.

OVERLAYSYM=(*symbol-list***)**
specifies symbols used to plot overlays on the primary control chart. Symbols in the OVERLAYSYM= list are matched with variables in the corresponding positions in the OVERLAY= list.

OVERLAYSYMHT=(*value-list*)

specifies the heights of symbols used to plot overlays on the primary control chart. Heights in the OVERLAYSYMHT= list are matched with variables in the corresponding positions in the OVERLAY= list.

POINTSHTML=*variable*

specifies a variable whose values create links to be associated with points on a box chart when the BOXSTYLE= value is POINTS, POINTSJOIN, POINTSBOX, POINTSID, or POINTSJOINID. These URLs are associated with points on a box chart when traditional graphics output is directed into HTML.

TESTFONT=*font*

LABELFONT=*font*

specifies a software font for labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, OUTLABEL2=, STARLABEL=, TESTLABEL=, and TESTLABEL*n*= options. Hardware characters are used by default.

TESTHEIGHT=*value*

LABELHEIGHT=*value*

specifies the height (in vertical percent screen units) for labels requested with the ALLLABEL=, ALLLABEL2=, OUTLABEL=, OUTLABEL2=, STARLABEL=, TESTLABEL=, and TESTLABEL*n*= options. The default height is the height specified with the HEIGHT= option or the HTEXT= option in the GOPTIONS statement.

TESTSYMBOL=*symbol*

TESTSYM=*symbol*

specifies the symbol for plotting subgroups with positive tests for special causes.

TESTSYMBOLHT=*value*

TESTSYMHT=*value*

specifies the height of the symbol used to plot subgroups with positive tests for special causes.

TURNALL

TURNOUT

turns the labels produced by the ALLLABEL=, ALLLABEL2=, OUTLABEL=, and OUTLABEL2= options so that they are strung out vertically. By default, labels are arranged horizontally.

TURNHLABELS

TURNHLABEL

turns the major tick mark labels for the horizontal (subgroup) axis so that they are strung out vertically. By default, labels are arranged horizontally.

If you are producing traditional graphics with the NOGSTYLE option in effect, you should specify a font (with the FONT= option) in conjunction with the TURNHLABELS option. Otherwise, the labels may be displayed with a mixture of hardware and software fonts.

NOTE: Turning the labels vertically may leave insufficient room on the screen or page for a chart.

VMINOR=*n*

VM=*n*

specifies the number of minor tick marks between each major tick mark on the vertical axis. No values are printed on the minor tick marks. By default, VMINOR=0.

WAXIS=*n*

specifies the width in pixels for the axis and frame lines. By default, *n* = 1.

WEBOUT=*SAS-data-set*

produces an output data set containing all the data in an OUTTABLE= data set plus graphics coordinates for points (subgroup summary statistics) that are displayed on a control chart. You can use an WEBOUT= data set to facilitate the development of web-based applications. See "Interactive Control Charts: SHEWHART Procedure" on page 2136 for details.

WGRID=*n*

specifies the width in pixels for grid lines requested with the ENDGRID and GRID options. By default, *n* = 1.

WLIMITS=*n*

specifies the width in pixels for the control limits and central line. By default, *n* = 1.

WNEEDLES=*n*

specifies the width in pixels of needles connecting plotted points to the central line, as requested with the NEEDLES option. If you use the WNEEDLES= option, you do not need to specify the NEEDLES option. By default, *n* = 1.

WOVERLAY=(*value-list*)

specifies the widths in pixels for the line segments connecting points on primary chart overlay plots. Widths in the WOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list.

WOVERLAY2=(*value-list*)

specifies the widths in pixels for the line segments connecting points on secondary chart overlay plots. Widths in the WOVERLAY2= list are matched with variables in the corresponding positions in the OVERLAY2= list.

WSTARCIRCLES=*n*

specifies the width in pixels of the outline of circles requested by the STARCIRCLES= option. See "Displaying Auxiliary Data with Stars" on page 2042. By default, *n* = 1.

WSTARS=*n*

specifies the width in pixels of the outline of stars requested by the STARVERTICES= option. See "Displaying Auxiliary Data with Stars" on page 2042. By default, *n* = 1.

WTESTS=*n*

WTEST=*n*

specifies the width in pixels of the line segments that connect patterns of points for which a test for special causes (requested with the TESTS= or TESTS2= option) is positive. By default, *n* = 1.

WTREND=*n*

specifies the width in pixels of the line segments that connect points on trend charts requested with the TRENDVAR= option. By default, $n = 1$. The WTREND= option is available in the BOXCHART, MCHART, and XCHART statements.

Options for Legacy Line Printer Charts

CLIPCHAR='*character*'

specifies a plot character that identifies clipped points, as requested with the CLIPFACTOR= option. Specifying the CLIPCHAR= option is recommended when the CLIPFACTOR= option is used. The default character is an asterisk (*).

CONNECTCHAR='*character*'

CCHAR='*character*'

specifies the character used to form line segments that connect points on a chart. The default character is a plus (+) sign.

HREFCHAR='*character*'

specifies the character used for reference lines requested by the HREF= and HREF2= options on line printer charts. The default is the vertical bar (|).

SYMBOLCHARS='*character-list*'

specifies a list of characters used to mark the points plotted on line printer charts when a *symbol-variable* is used. See "Displaying Stratification in Levels of a Classification Variable" on page 2025.

Each character is associated with a level (unique value) of the *symbol-variable* and is used to mark points associated with that value. For example, consider the following statements:

```
proc shewhart;
   xrchart Gap*Shift=Machine / symbolchars='12345';
run;
```

Here the *symbol-variable* is Machine. The $\bar{X}$ and R charts use a '1' to mark points associated with the first unique value of Machine, a '2' to mark points associated with the second unique value of Machine, and so on.

If the number of levels of the *symbol-variable* exceeds the number of *characters*, the last character listed is used for points associated with the additional values. Thus, in the preceding example, if there are six levels of Machine, points with the fifth and six values are indicated by '5'.

The default *character-list* is ABCDEFGHIJKLMNOPQRSTUVWXYZ*. Thus, the procedure uses 'A' for the first unique value of the *symbol-variable*, 'B' for the second unique value, and so on. An asterisk is used for points associated with the 27th and subsequent levels when the *symbol-variable* has more than 26 levels.

TESTCHAR='*character*'

specifies the character for the line segments that connect any sequence of points for which a test for special causes (requested with the TESTS= or TESTS2= option) is positive. The default *character* is the number of the test (with values 1 to 8).

VREFCHAR='*character*'

specifies the character used for reference lines requested by the VREF= and VREF2= options on line printer charts. The default is the hyphen (-).

ZONECHAR='*character*'

specifies the character used to form the zone lines requested by the ZONES option. See the entry for the TESTS= option for a description of the zones. You do not need to specify the ZONES option if you specify the ZONECHAR= option. By default, the line between Zone A and Zone B uses the character 'B', and the line between Zone B and Zone C uses the character 'C'. Related options are TESTS=, TESTS2=, ZONES, and ZONES2.

Graphical Enhancements: SHEWHART Procedure

Overview: Graphical Enhancements

This section provides details on the following topics:

- displaying process data stratified into levels using a *symbol-variable*
- displaying process data stratified into blocks using *block-variables*
- displaying process data stratified into time phases using the READPHASES= option
- displaying multiple sets of control limits using the READPHASES= and READINDEXES= options
- displaying multivariate process data using star charts
- displaying trends in process data
- clipping extreme points to create more readable charts
- labeling axes
- selecting subgroups for computation and display

The options described in this section can be specified in all the chart statements available in the SHEWHART procedure.

Displaying Stratified Process Data

If the data for a Shewhart chart can be classified by factors relevant to the process (for instance, machines or operators), displaying the classification on the chart can facilitate the identification of special or common causes of variation that are related to the factors. Kume (1985) refers to this type of classification as "stratification" and describes various ways to create stratified control charts.

There are important differences between stratification and subgrouping. The data must always be classified into subgroups before a control chart can be produced. Subgrouping affects how control limits are computed from the data as well as the outcome of tests for special causes (see “Tests for Special Causes: SHEWHART Procedure” on page 2073). The values of the *subgroup-variable* specified in the chart statement classify the data into subgroups. In contrast, stratification is optional and involves classification variables other than the *subgroup-variable*. Displaying stratification influences how the chart is interpreted, but it does not affect control limits or tests for special causes.

This section describes three types of variables that you can specify to create stratified control charts.

- A *symbol-variable* stratifies data into levels of a classification variable.
- The *block-variables* stratify data into blocks of consecutive observations.
- A _PHASE_ variable stratifies data into *time phases*.

You can specify any combination of these three variables. You should be careful, however, since it is possible to generate confusing charts by overusing these methods.

The data for the examples in this section consist of diameter measurements for a part produced on one of three different machines. Three subgroups, each consisting of six parts, are sampled each day, corresponding to three shifts worked each day. The data are provided in the data set Parts, which is created by the following statements:

```
data Parts;
   length Machine $ 4;
   input Sample Machine $ Day Shift DiamX DiamS;
   DiamN=6;
   datalines;
 1  A386  01  1  4.32  0.39
 2  A386  01  2  4.49  0.35
 3  A386  01  3  4.44  0.44
 4  A386  02  1  4.45  0.17
 5  A386  02  2  4.21  0.53
 6  A386  02  3  4.56  0.26
 7  A386  03  1  4.63  0.39
 8  A386  03  2  4.38  0.47
 9  A386  03  3  4.47  0.40
10  A455  04  1  4.42  0.37
11  A455  04  2  4.45  0.32
12  A455  04  3  4.62  0.36
13  A455  05  1  4.33  0.31
14  A455  05  2  4.29  0.33
15  A455  05  3  4.17  0.25
16  C334  08  1  4.15  0.28
17  C334  08  2  4.21  0.33
18  C334  08  3  4.16  0.19
19  C334  09  1  4.14  0.13
20  C334  09  2  4.11  0.19
21  C334  09  3  4.10  0.27
22  C334  10  1  3.99  0.14
23  C334  10  2  4.24  0.16
24  C334  10  3  4.23  0.14
```

```
25  A386  11  1  4.27  0.28
26  A386  11  2  4.70  0.45
27  A386  11  3  4.51  0.45
28  A386  12  1  4.34  0.16
29  A386  12  2  4.38  0.29
30  A386  12  3  4.28  0.24
31  A386  15  1  4.47  0.26
32  A386  15  2  4.31  0.46
33  A386  15  3  4.52  0.33
;
```

Displaying Stratification in Levels of a Classification Variable

NOTE: See *Stratifying Data with a Classification Variable* in the SAS/QC Sample Library.

To display process data stratified into levels of a classification variable, specify the name of this variable after an equal sign (=) immediately following the *subgroup-variable* in the chart statement. The classification variable, referred to as the *symbol-variable*, must be a variable in the input data set (a DATA=, HISTORY=, or TABLE= data set). The subgroup summary statistics are classified into groups according to the levels of the *symbol-variable* and are identified on the chart with unique plotting symbols.

When you produce traditional graphics output, you can specify the symbols with SYMBOL statements. It is recommended that you place the SYMBOL statements before the PROC SHEWHART statement. If you omit the SYMBOL statements, the procedure uses the default symbol (+) for all levels of the *symbol-variable* but plots the points for each level in a distinct color. The following example illustrates the use of a *symbol-variable* to stratify the points on an $\bar{X}$ chart according to the machine that produced the parts in each subgroup:

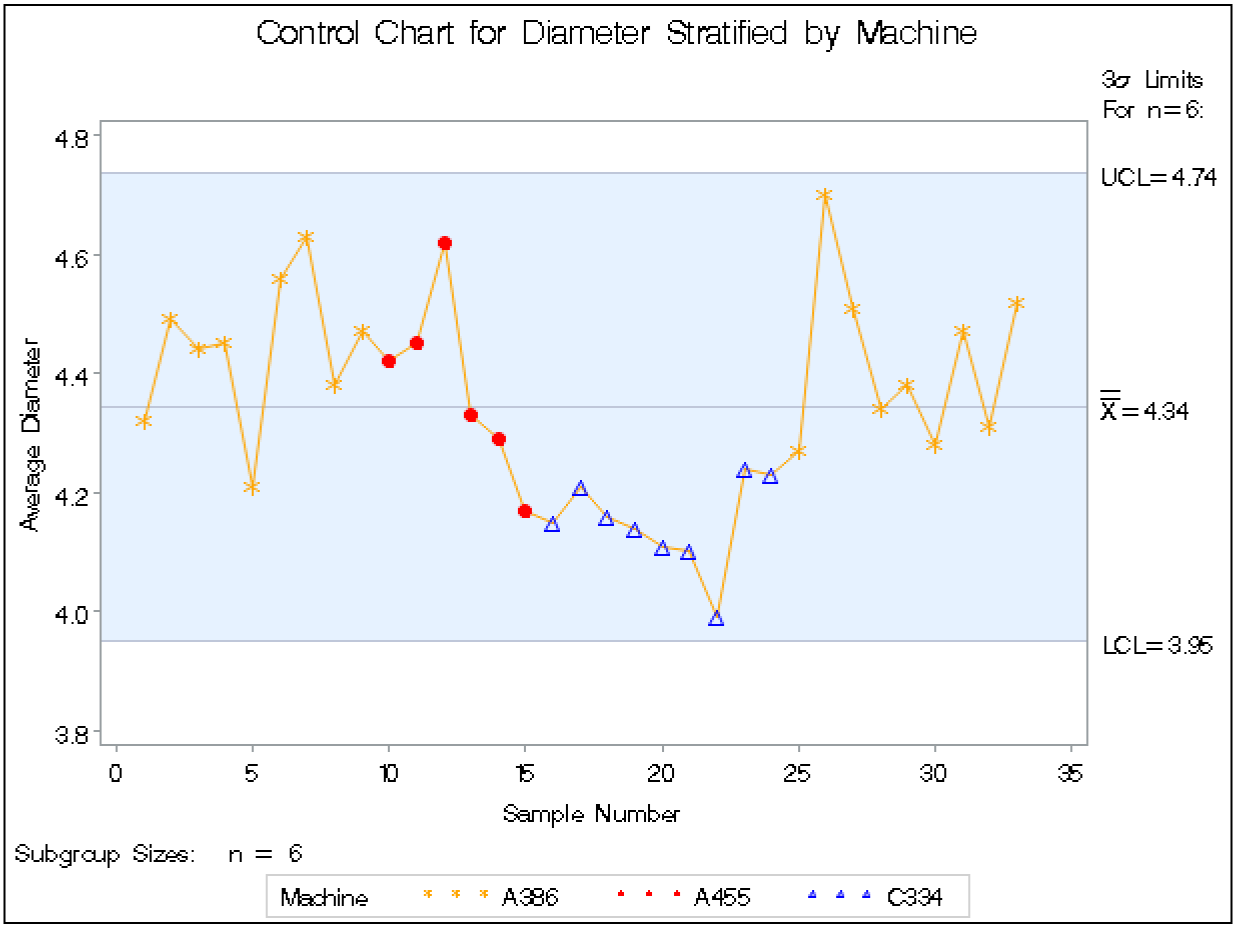
```
ods graphics off;
symbol1 c=orange value=star      h=3.0 pct;
symbol2 c=red    value=dot       h=3.0 pct;
symbol3 c=blue   value=triangle  h=3.0 pct;
title 'Control Chart for Diameter Stratified by Machine';
proc shewhart history=Parts;
   xchart Diam*Sample=Machine / stddeviations
                                symbollegend = legend1;
   label Sample = 'Sample Number'
         DiamX  = 'Average Diameter' ;
   legend1 frame label=('Machine');
run;
```

The symbols are specified with the SYMBOL1, SYMBOL2, and SYMBOL3 statements. The SYMBOLLEGEND= option requests a customized legend for the symbols. For more information about the LEGEND and SYMBOL statements, refer to *SAS/GRAPH: Reference*. The $\bar{X}$ chart, shown in Figure 17.143, reveals an effect due to Machine. In particular, Machine C334 is associated with a run of parts whose diameters are systematically below average, suggesting that this machine may require adjustment.

For line printer charts, you can use the SYMBOLCHARS= option to specify the characters that identify the stratification of the points. For details, see the entry for the SYMBOLCHARS= option in "Dictionary of Options: SHEWHART Procedure" on page 1946.

In this example, Machine A386 is associated with two different blocks of observations that are identified with a common symbol. However, a *symbol-variable* is particularly useful for situations where the stratification is not necessarily chronological or associated with blocks of consecutive groups of observations.

Figure 17.143 Control Chart Stratified into Levels Using Symbols



Displaying Stratification in Blocks of Observations

NOTE: See *Using Block Variables to Stratify Data* in the SAS/QC Sample Library.

To display process data stratified into blocks of consecutive observations, specify one or more *block-variables* in parentheses after the *subgroup-variable* in the chart statement. The procedure displays a legend identifying blocks of consecutive observations with identical values of the *block-variables*. The legend displays one track of values for each *block-variable*. The values are the formatted values of the *block-variable*. For example, Figure 17.144 displays a legend with a single track for Machine, while Figure 17.145 displays a legend with two tracks corresponding to Machine and Day. You can label the tracks themselves by using the LABEL statement to associate labels with the corresponding *block-variables*; see Figure 17.146 for an illustration.

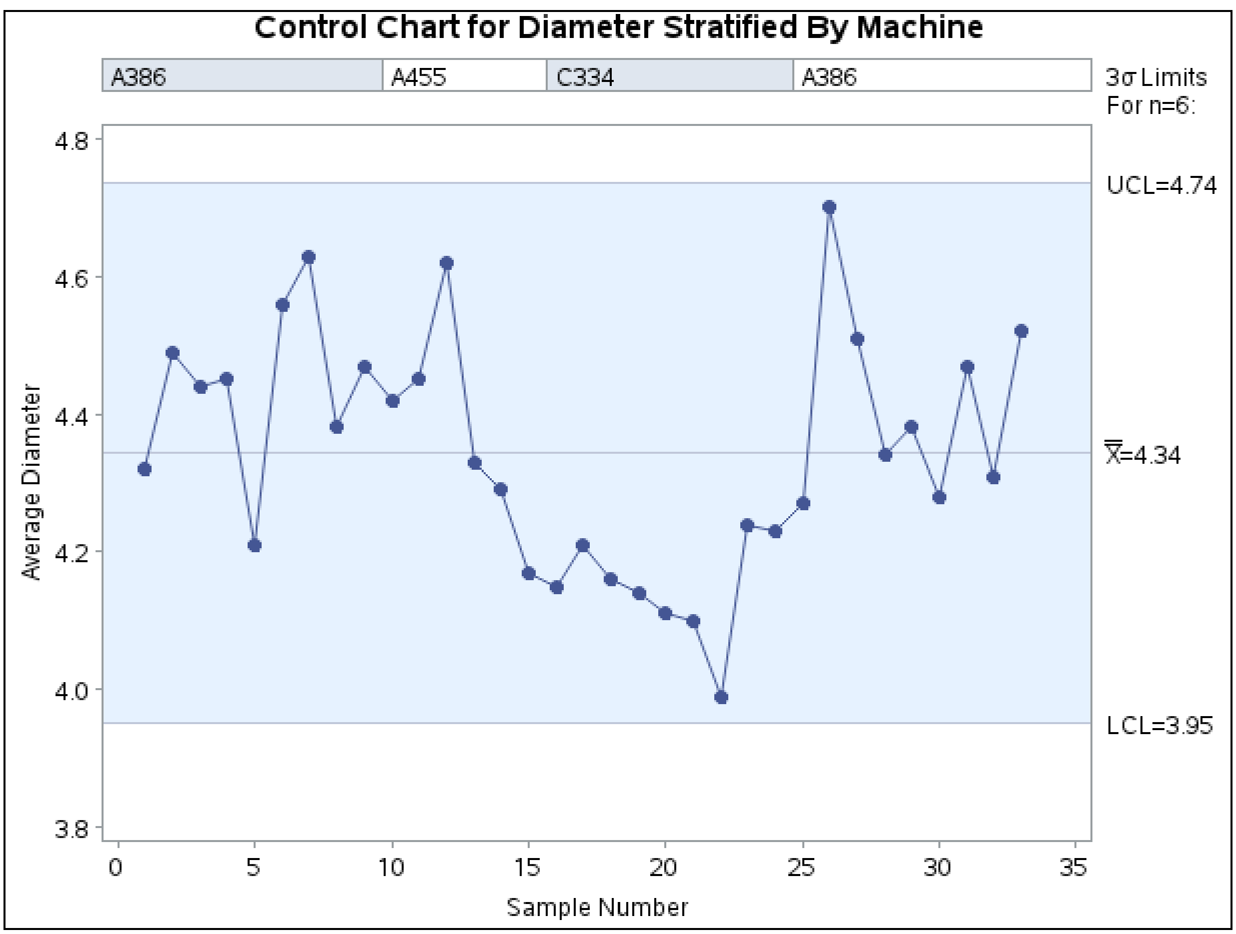
By default, the legend is placed above the chart as in Figure 17.144. You can control the position of the legend with the BLOCKPOS= option and the position of the legend labels with the BLOCKLABELPOS= option. See the entries in "Dictionary of Options: SHEWHART Procedure" on page 1946 as well as the following examples.

The *block-variables* must be variables in the input data set (a DATA=, HISTORY=, or TABLE= data set). If the input data set is a DATA= data set that contains multiple observations with the same value of the *subgroup-variable*, the values of a *block-variable* must be the same for all observations with the same value of the *subgroup-variable*. In other words, subgroups must be nested within groups determined by *block-variables*. The following statements create an $\bar{X}$ chart for the data in Parts stratified by the *block-variable* Machine. The chart is shown in Figure 17.144.

```
symbol v=dot h=3.0 pct;
title 'Control Chart for Diameter Stratified By Machine';
proc shewhart history=Parts;
   xchart Diam*Sample (Machine) / stddeviations
                                  nolegend ;
   label Sample = 'Sample Number'
         DiamX  = 'Average Diameter' ;
run;
```

The unique consecutive values of Machine ('A386', 'A455', 'C334', and 'A386') are displayed in a track above the chart, and they indicate the same relationship between part diameter and machine as the previous example. Note that the track is not labeled (as in Figure 17.146), since no label is associated with Machine. A LABEL statement is used to provide labels for the axes.

Figure 17.144 Stratified Control Chart Using a Single Block Variable

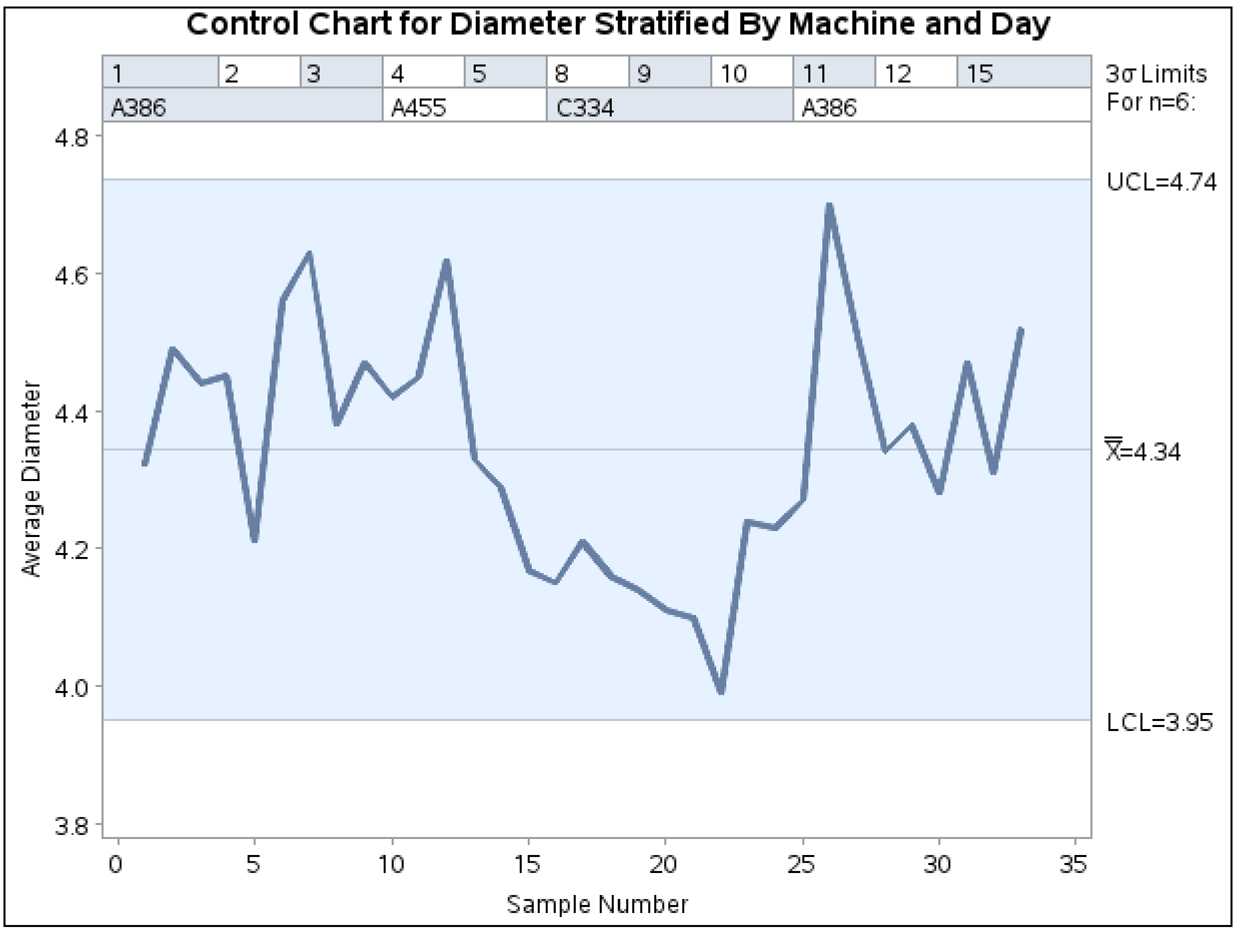


Multiple block variables. You can use multiple *block-variables* to study more than one classification factor with the same chart. The following statements create an $\bar{X}$ chart for the data in Parts, with Machine and Day as *block-variables*:

```
title 'Control Chart for Diameter Stratified By Machine and Day';
proc shewhart history=Parts;
   xchart Diam*Sample (Machine Day) / stddeviations
                                      nolegend
                                      blockpos = 2;
   label Sample = 'Sample Number'
         DiamX  = 'Average Diameter' ;
run;
```

The chart is displayed in Figure 17.145. Specifying BLOCKPOS=2 displays the *block-variable* legend immediately above the chart, without the gap shown in Figure 17.144. The NOLEGEND option suppresses the sample size legend that appears in the lower left of Figure 17.144.

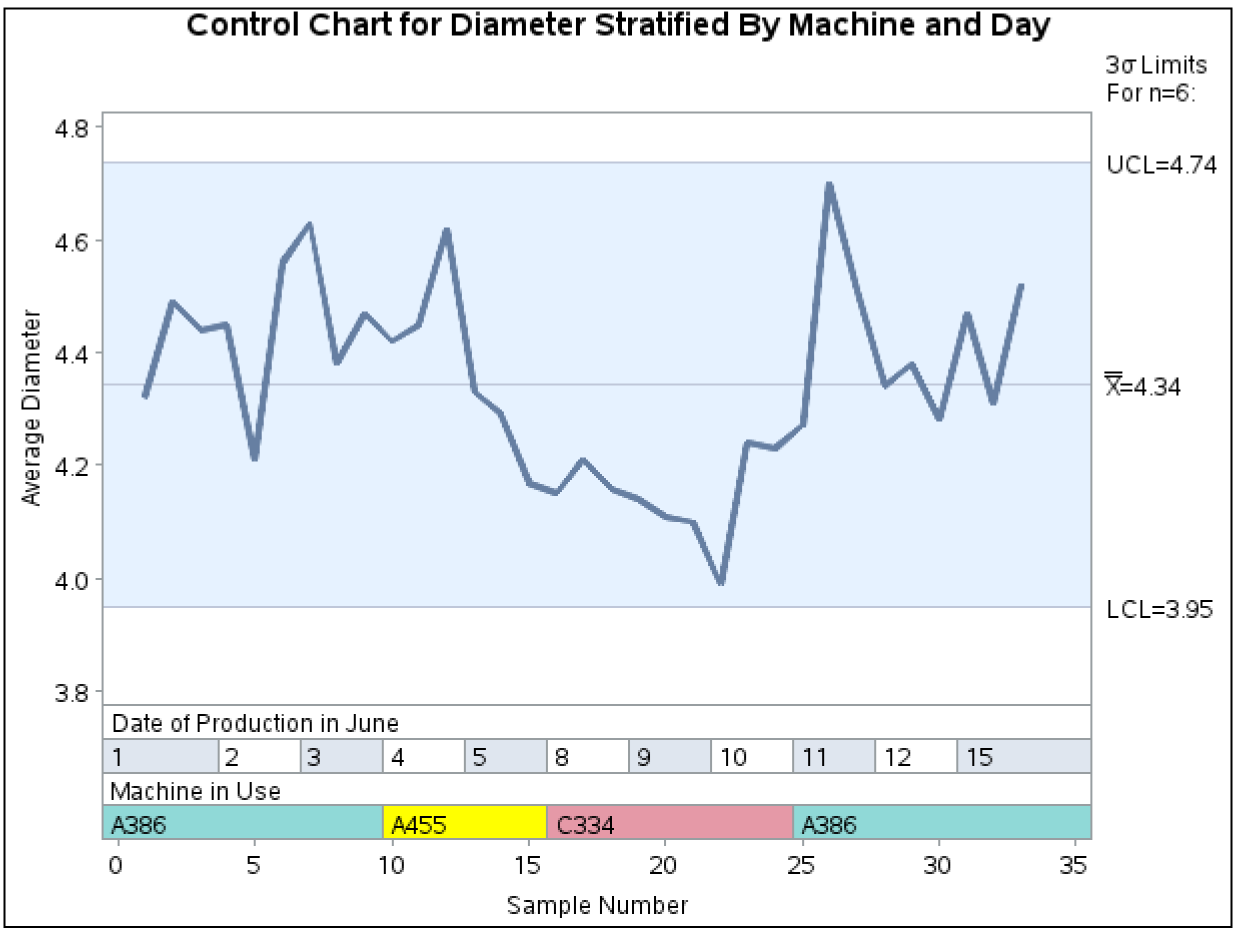
Figure 17.145 Stratified Control Chart Using Multiple Block Variables



Color fills for legend. You can use the CBLOCKVAR= option to fill the legend track sections with colors corresponding to the values of the *block-variables*. Provide the colors as values of variables specified with the CBLOCKVAR= option. The procedure matches the color variables with the *block-variables* in the order specified. Each section is filled with the color for the first observation in the block. For example, the following statements produce an $\bar{X}$ chart using a color variable named CMachine to fill the legend for the *block-variable* Machine:

```
title 'Control Chart for Diameter Stratified By Machine and Day';
proc shewhart history=Parts2;
   xchart Diam*Sample (Machine Day) / stddeviations
                                      nolegend
                                      blockpos  = 3
                                      cblockvar = CMachine;
   label Sample  = 'Sample Number'
         DiamX   = 'Average Diameter'
         Day     = 'Date of Production in June'
         Machine = 'Machine in Use';
run;
```

Figure 17.146 Color Fill for *Block-Variable* Legend



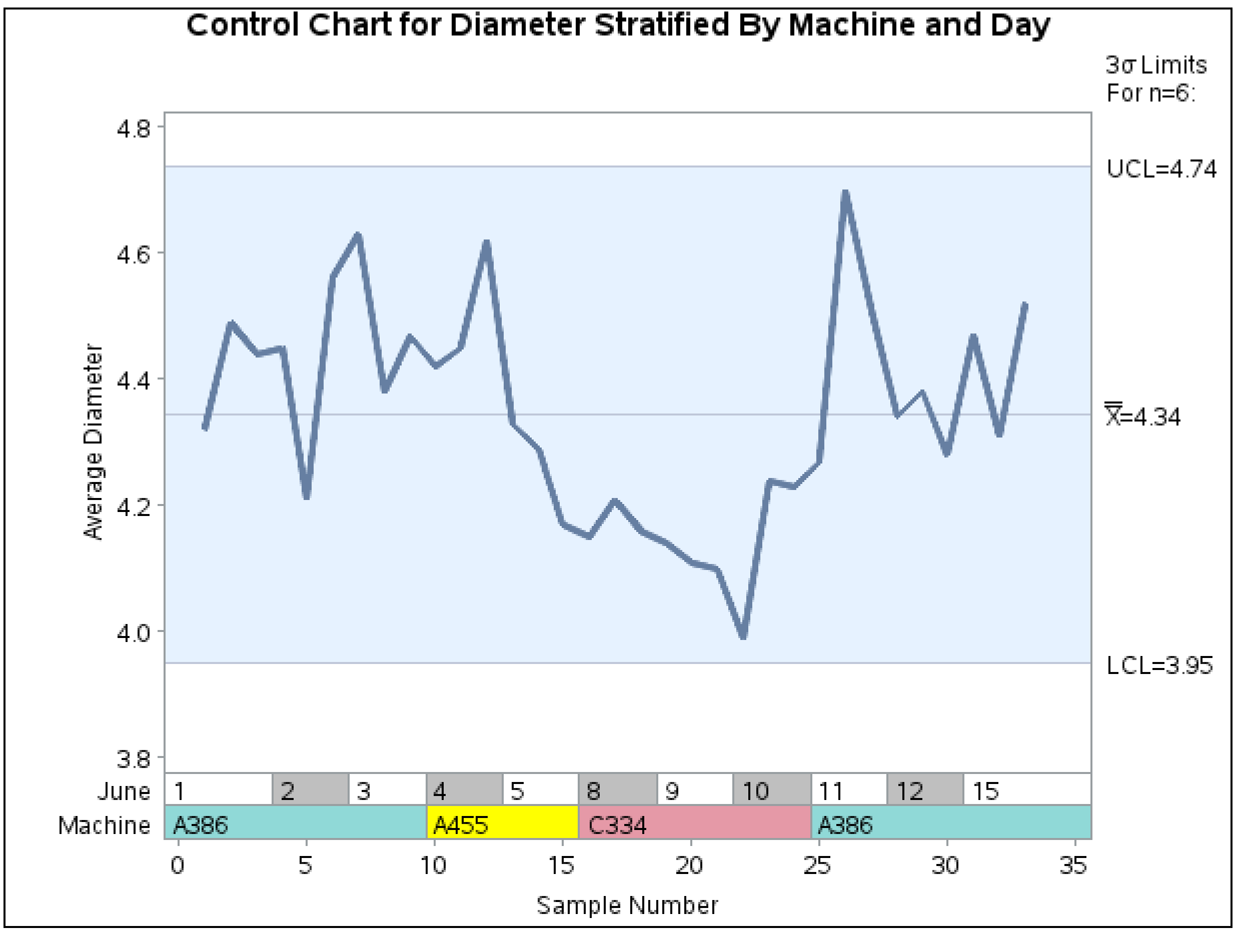
The sections for Machine A386, Machine A455, and Machine C334 are filled with the colors specified as values of CMachine. The legend track for Day is filled with the default alternating colors from the ODS style, because a second color variable was not specified with the CBLOCKVAR= option. Specifying BLOCKPOS=3 positions the legend at the bottom of the chart and facilitates comparison with the subgroup axis. The LABEL statement is used to label the tracks with the labels associated with the *block-variables*.

The following statements produce an $\bar{X}$ chart in which both legend tracks are filled:

```
title 'Control Chart for Diameter Stratified By Machine and Day';
proc shewhart history=Parts3;
   xchart Diam*Sample (Machine Day) /
      stddeviations
      nolegend
      ltmargin       = 5
      blockpos       = 3
      blocklabelpos = left
      cblockvar      = (CMachine CDay);
   label Sample      = 'Sample Number'
         DiamX       = 'Average Diameter'
         Day         = 'June'
         Machine     = 'Machine';
run;
```

The chart is displayed in Figure 17.147. The color values of CMachine are used to fill the track for Machine, and the color values of CDay are used to fill the track for Day. Specifying BLOCKLABELPOS=LEFT displays the block variable labels to the left of the block legend. The LTMARGIN= option provides extra space in the left margin to accommodate the label *Machine*.

Figure 17.147 Stratified Control Chart Using Multiple Block Variables



Displaying Stratification in Phases

NOTE: See *Displaying Stratification in Phases* in the SAS/QC Sample Library.

The preceding section describes the use of *block-variables* to display blocks of consecutive observations that correspond to changes in factors such as machines, shifts, and raw materials. This section describes the use of a *_PHASE_ variable* to display phases of consecutive observations (as in Figure 17.148). Although the terms *block* and *phase* have similar meanings, there are differences in the two methods:

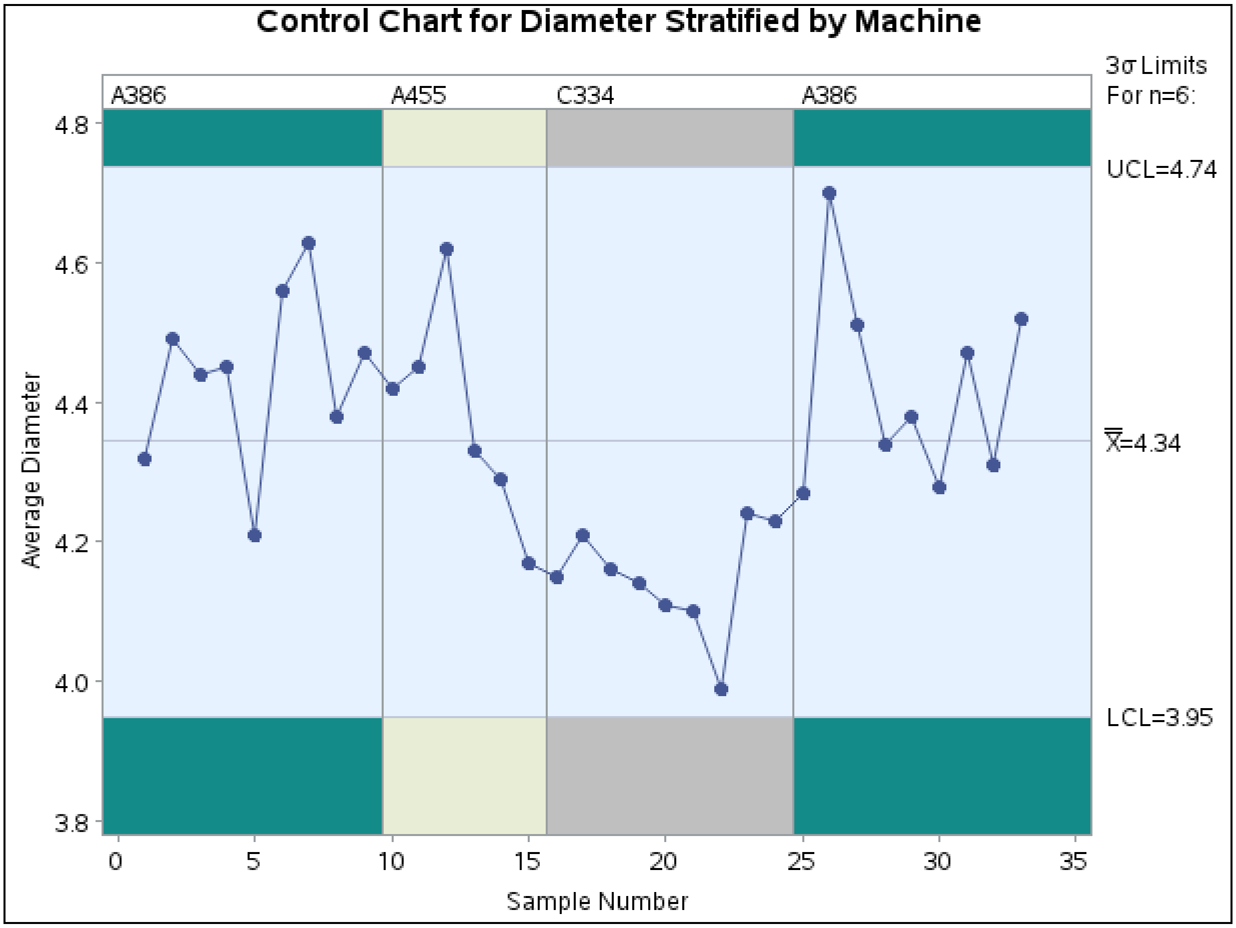
- You can provide only one *_PHASE_ variable*, whereas you can specify multiple *block-variables*.
- You can display distinct control limits for each phase (see "Displaying Multiple Sets of Control Limits" on page 2033) but not for each block.
- Different sets of graphical options are available for identifying blocks and phases.

To display phases, your input data set must include a character variable named _PHASE_ of length 48 or less, and you must specify the READPHASES= option in the chart statement. (If your data set does not include a variable named _PHASE_, you can temporarily rename another character variable to _PHASE_, as illustrated by the following statements.) The procedure classifies the data into phases (groups of consecutive observations with the same value of _PHASE_) and reads only those observations whose _PHASE_ value matches one of the values specified with the READPHASES= option.

You can identify and highlight the phases with various options, as illustrated by the following statements, which produce the chart shown in Figure 17.148. The PHASELEGEND option displays a legend with the _PHASE_ values, and the CPHASELEG= option specifies the color of the legend text. The PHASEREF option delineates the phases with vertical reference lines. The CFRAME= option fills the framed areas for the phases with different colors.

```
ods graphics off;
symbol v=dot h=3.0 pct;
title 'Control Chart for Diameter Stratified by Machine';
proc shewhart history=Parts(rename=(Machine=_phase_));
   xchart Diam*Sample /
      stddeviations
      readphases = ('A386' 'A455' 'C334' 'A386')
      cframe     = ( vibg   ywh    ligr   vibg )
      phaselegend
      cphaseleg  = black
      phaseref
      nolegend;
   label  Sample = 'Sample Number'
          DiamX  = 'Average Diameter';
run;
```

Figure 17.148 Control Chart Stratified by Phases



Note that the data set Parts does not contain a variable named _PHASE_, so the variable Machine is renamed as _PHASE_ for the duration of the procedure step.

The observations read from Parts are those whose value of Machine matches one of the values listed with the READPHASES= option in that order. Here, the value ‘A386’ is listed twice; consequently, both groups of observations for which Machine equals ‘A386’ are read.

In this example, the input data set contains a single observation for each subgroup. If your input data set is a DATA= data set that contains multiple observations with the same value of the *subgroup-variable*, the value of _PHASE_ must be the same for all observations with the same value of the *subgroup-variable*. Thus, in general, subgroups must be nested within phases.

Recall that the horizontal axis scale is determined by the *subgroup-variable* (see “Subgroup Variables” on page 1871). If your *subgroup-variable* is numeric, this scale is continuous; consequently, you should select phases that are reasonably contiguous in order to avoid large empty gaps in your chart. For instance, if you were to specify

```
readphases = ('A386' 'A455' 'A386')
```

in the preceding XCHART statement, there would be a gap between the 15th and 25th points (these points would be connected unless you specified the PHASEBREAK option). You can avoid gaps by specifying a character *subgroup-variable*[11] for which a discrete horizontal axis scale will be displayed.

[11] You can use the PUT function in a DATA step to create a character *subgroup-variable* from a numeric *subgroup-variable*.

Note that the values listed in the READPHASES= option must be listed in the same order as they occur in the input data set. Thus, in order to display all the observations in the data set Parts, 'A386' must be listed as both the first and last value. An alternative method for selecting all the phases from your input data is to specify READPHASES=ALL, as described in the next section.

The control limits shown in Figure 17.148 are computed from the data and are, therefore, the same across all phases. More generally, you can display a distinct set of control limits for each phase. To do so, you must provide the control limits in a LIMITS= data set and specify the READINDEXES= option in addition to the READPHASES= option, as described in the next section.

Displaying Multiple Sets of Control Limits

NOTE: See *Displaying Multiple Sets of Control Limits* in the SAS/QC Sample Library.

This section describes the use of the READPHASES= and READINDEXES= options for creating Shewhart charts that display distinct sets of control limits for multiple phases of observations. The term *phase* refers to a group of consecutive observations in the input data set. For example, the phases might correspond to the time periods during which a new process was brought into production and then put through successive changes.

To display phases, your input data must include a character variable named _PHASE_, whose length cannot exceed 48. (If your data set does not include a variable named _PHASE_, you can temporarily rename another character variable to _PHASE_, as illustrated in the statements in the section "Displaying Stratification in Phases" on page 2031.) Each phase consists of a group of consecutive observations with the same value of _PHASE_.

To display distinct sets of predetermined control limits for the phases, you must provide the limits in a LIMITS= data set. This data set must include a character variable named _INDEX_, whose length cannot exceed 48. This variable identifies the sets of control limits (observations) in the LIMITS= data set that are to be associated with the phases. This data set must also include a number of other variables with reserved names that begin and end with an underscore. The particular structure of a LIMITS= data set depends on the chart statement that you are using; for details, see the sections titled "LIMITS= Data Set" in the sections for the various chart statements. In addition to specifying a LIMITS= data set, you must also specify the READINDEXES= and READPHASES= options in the chart statement.

NOTE: To display a *single* set of predetermined control limits with multiple phases, simply specify a LIMITS= data set in the procedure statement. If you are using SAS 6.09 or an earlier release, you must also specify the READLIMITS option. The control limits are read from the first observation in the LIMITS= data for which the variable _VAR_ is equal to the name of the *process* and the variable _SUBGRP_ is equal to the name of the *subgroup-variable*. For an example, see "Reading Preestablished Control Limits" on page 1848.

This section describes the combinations of the READINDEXES= and READPHASES= options that you can specify. The examples that follow use the HISTORY= data set Flange listed in Figure 17.149 and the LIMITS= data set Flangelim listed in Figure 17.150. The data in Flange consist of means and ranges of flange width measurements for subgroups of size five. The observations are grouped into three phases determined by the _PHASE_ values 'Production', 'Change 1', and 'Change 2'. Three sets of control limits are provided in Flangelim, corresponding to the _INDEX_ values 'Start', 'Production', and 'Change 1'.

Figure 17.149 Listing of the HISTORY= Data Set Flange

Obs	_phase_	Day	Sample	FlwidthX	FlwidthR	FlwidthN
1	Production	08FEB90	6	0.97360	0.06247	5
2	Production	09FEB90	7	1.00486	0.11478	5
3	Production	10FEB90	8	1.00251	0.13537	5
4	Production	11FEB90	9	0.95509	0.08378	5
5	Production	12FEB90	10	1.00348	0.09993	5
6	Production	15FEB90	11	1.02566	0.06766	5
7	Production	16FEB90	12	0.97053	0.07608	5
8	Production	17FEB90	13	0.94713	0.10170	5
9	Production	18FEB90	14	1.00377	0.04875	5
10	Production	19FEB90	15	0.99604	0.08242	5
11	Change 1	22FEB90	16	0.99218	0.09787	5
12	Change 1	23FEB90	17	0.99526	0.02017	5
13	Change 1	24FEB90	18	1.02235	0.10541	5
14	Change 1	25FEB90	19	0.99950	0.11476	5
15	Change 1	26FEB90	20	0.99271	0.05395	5
16	Change 1	01MAR90	21	0.98695	0.03833	5
17	Change 1	02MAR90	22	1.00969	0.06183	5
18	Change 1	03MAR90	23	0.98791	0.05836	5
19	Change 1	04MAR90	24	1.00170	0.05243	5
20	Change 1	05MAR90	25	1.00412	0.04815	5
21	Change 2	08MAR90	26	1.00261	0.05604	5
22	Change 2	09MAR90	27	0.99553	0.02818	5
23	Change 2	10MAR90	28	1.01463	0.05558	5
24	Change 2	11MAR90	29	0.99812	0.03648	5
25	Change 2	12MAR90	30	1.00047	0.04309	5
26	Change 2	15MAR90	31	0.99714	0.03689	5
27	Change 2	16MAR90	32	0.98642	0.04809	5
28	Change 2	17MAR90	33	0.98891	0.07777	5
29	Change 2	18MAR90	34	1.00087	0.06409	5
30	Change 2	19MAR90	35	1.00863	0.02649	5

Figure 17.150 Listing of the LIMITS= Data Set Flangelim

Obs	_index_	_var_	_subgrp_	_type_	_limitn_	_alpha_	_sigmas_	_lclx_	_mean_
1	Change 1	FLWIDTH	SAMPLE	ESTIMATE	5	.0026998	3	0.96167	0.99924
2	Production	FLWIDTH	SAMPLE	ESTIMATE	5	.0026998	3	0.93792	0.98827
3	Start	FLWIDTH	SAMPLE	ESTIMATE	5	.0026998	3	0.87088	0.96803

Obs	_uclx_	_lclr_	_r_	_uclr_	_stddev_
1	1.03680	0	0.06513	0.13771	0.028000
2	1.03862	0	0.08729	0.18458	0.037530
3	1.06517	0	0.16842	0.35612	0.072409

For each of the READINDEXES= and READPHASES= options, you can specify a single value, a list of values, or the keyword ALL. You can also leave these options unspecified. Thus, there are 16 possible combinations of specifications for the two options, as explained by the following table and notes. The two most commonly encountered combinations are

- reading a single set of limits for one or more phases (see Case 1)
- reading a set of limits matched with a set of phases (see Case 4)

	READPHASES=			
READINDEXES=	**Single *Value***	**Multiple *Values***	***Keyword* ALL**	**Not Specified**
Single *Value*	See Case 1	See Case 1	See Case 2	See Case 3
Multiple *Values*	See Case 9	See Case 4	See Case 2	See Case 2
Keyword ALL	See Case 5	See Case 5	See Case 6	See Case 6
Not Specified	See Case 7	See Case 7	See Case 8	See Case 8

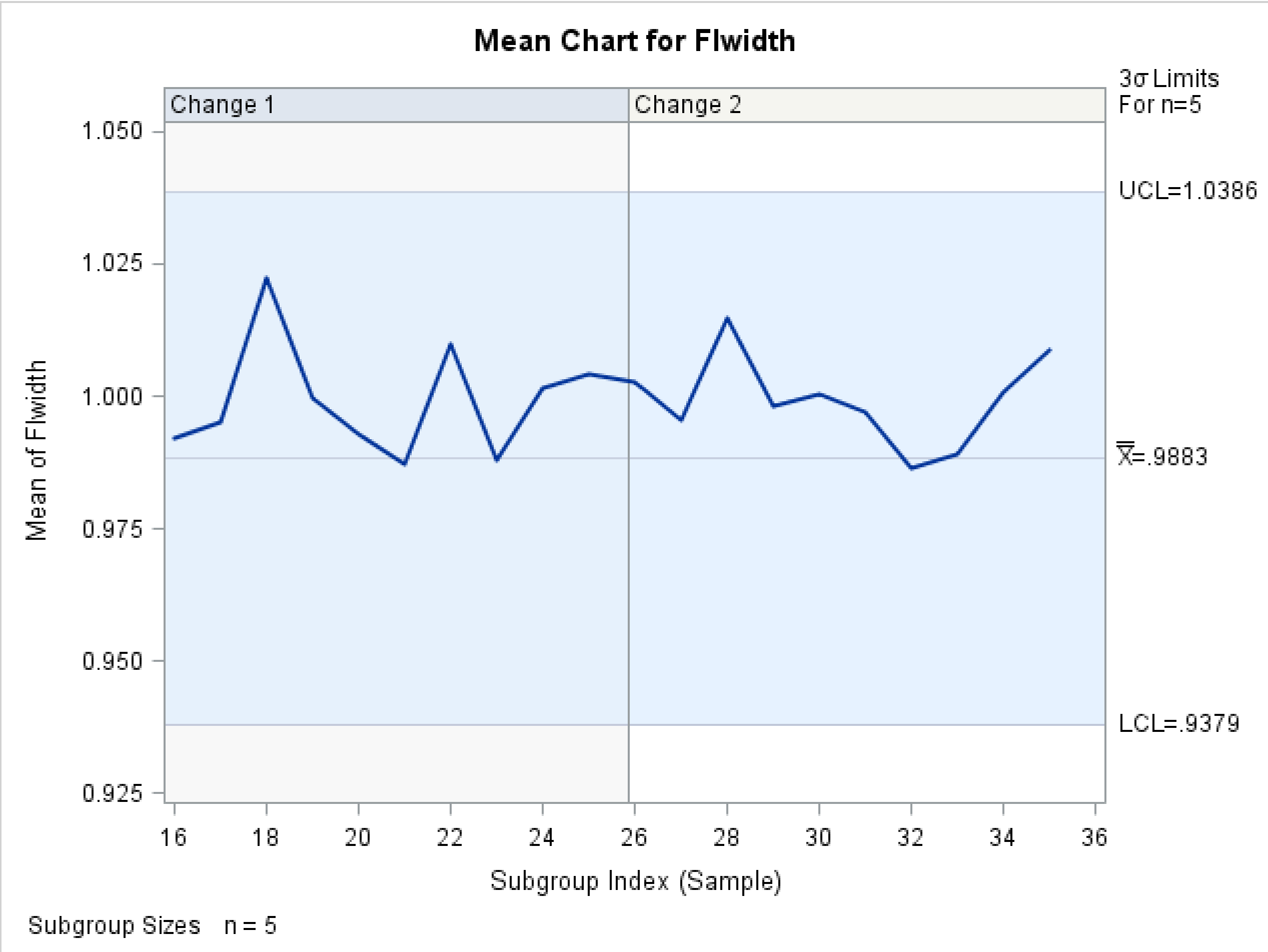
Case 1. READPHASES=*value|value-list* and READINDEXES=*value*

The only phases (groups of observations) read are those for which _PHASE_ equals one of the *values* specified with the READPHASES= option. The chart displays a single set of control limits given by the first observation in the LIMITS= data set for which _INDEX_ is equal to the READINDEXES= *value*.

For example, the following statements create a chart for the phases ‘Change 1’ and ‘Change 2’, with control limits read from the second observation in Flangelim. The chart is displayed in Figure 17.151.

```
ods graphics on;
proc shewhart history=Flange limits=Flangelim;
   xchart Flwidth*Sample /
      readphase = ('Change 1' 'Change 2')
      readindex = ('Production')
      phaseref
      phaselegend;
run;
```

Figure 17.151 A Single Set of Control Limits for Multiple Phases



Case 2. READPHASES=ALL and READINDEXES=*value*|*value-list* or READPHASES= is omitted and READINDEXES=*value-list*

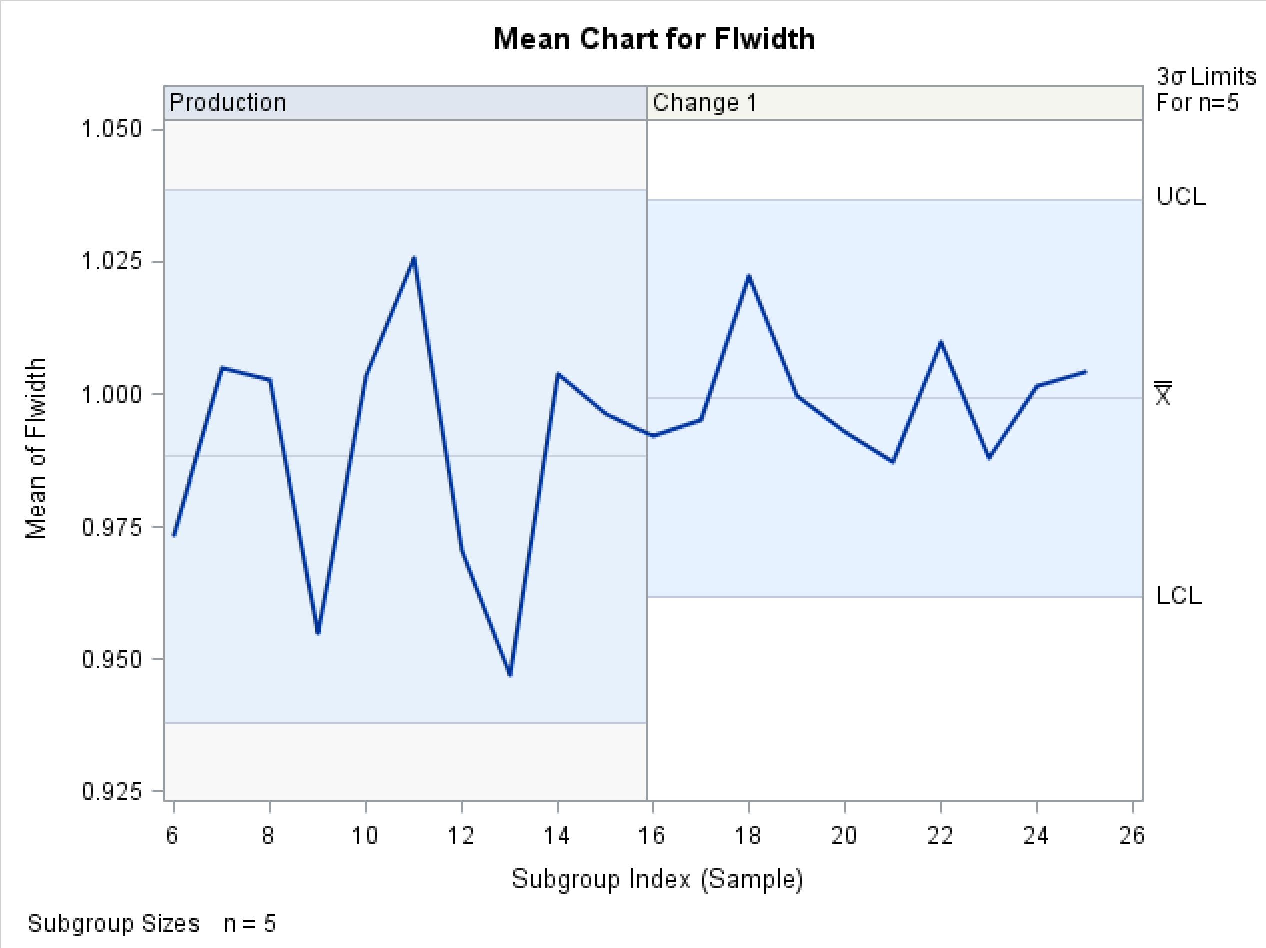
The only phases read are those for which _PHASE_ equals one of the *values* specified with the READINDEXES= option. The chart displays a different set of control limits for each phase, read from the first observation in the LIMITS= data set for which _INDEX_ is equal to the corresponding *value*.

For example, the following statements create a chart for the phases 'Production' and 'Change 1' with control limits read from the second and first observations in Flangelim, respectively. The chart is displayed in Figure 17.152.

```
proc shewhart history=Flange limits=Flangelim;
   xchart Flwidth*Sample /
      readphase = all
      readindex = ('Production' 'Change 1')
      phaseref
      phaselegend;
run;
```

If you wish to specify a single set of control limits to use with all the phases, use the READINDEXES= option *without* the READPHASES= option (see Case 3).

Figure 17.152 READPHASES=ALL with a List of Values for READINDEXES=



Case 3. READPHASES= is omitted and READINDEXES=*value*

All observations are read from the input data set. The chart displays a single set of control limits read from the first observation in the LIMITS= data for which _INDEX_ equals the *value*.

Case 4. READPHASES=*value-list* and READINDEXES=*value-list*

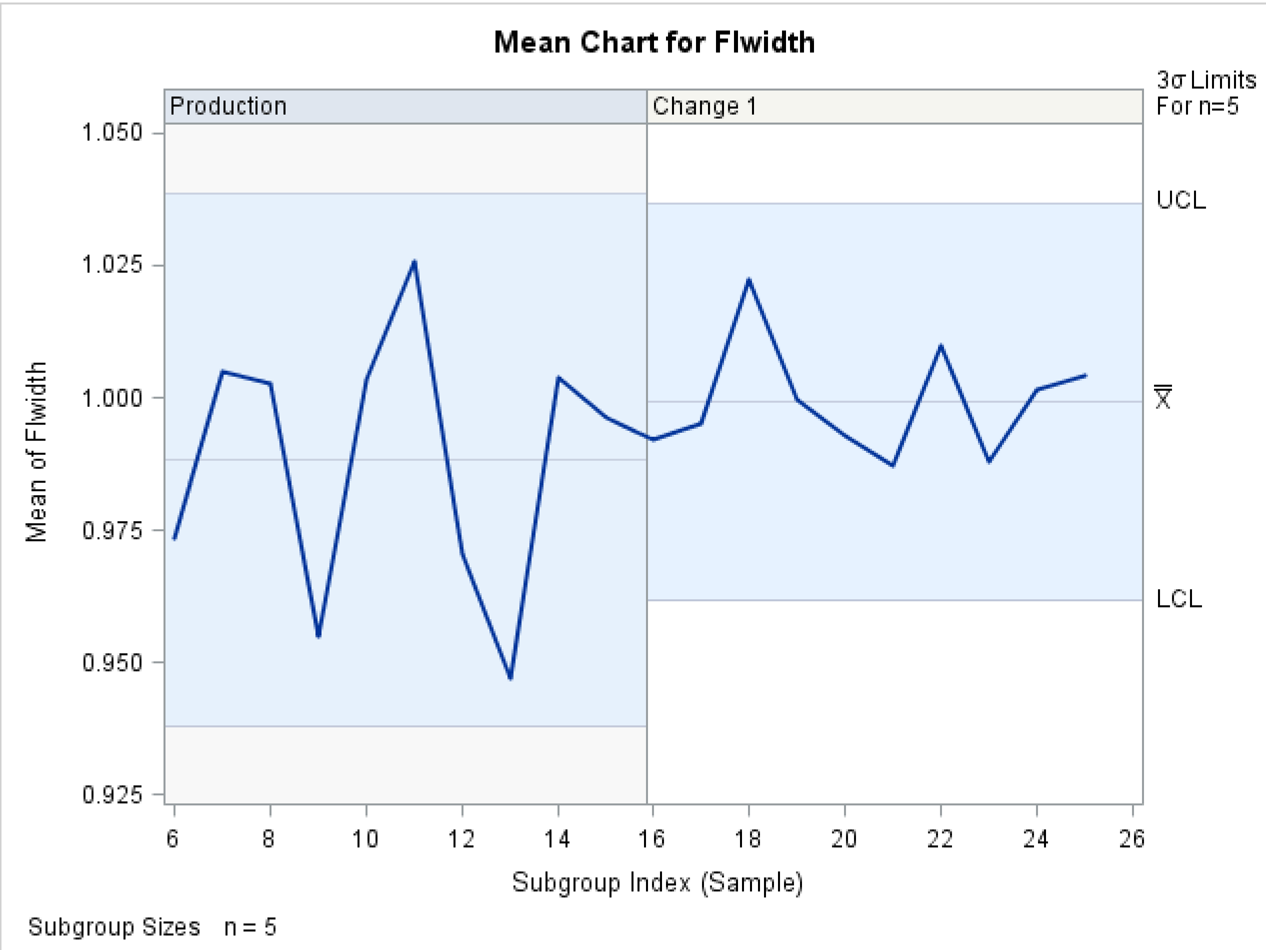
The only phases read are those for which _PHASE_ equals one of the values specified with the READPHASES= option. The chart displays a different set of control limits for each phase, given by the first observation in the LIMITS= data set for which _INDEX_ equals the READINDEXES=*value*. Control limits are matched with phases in the order listed.

For example, the following statements create a chart for the phases 'Production' and 'Change 1' with control limits read from the first and second observations in Flangelim, respectively. The chart produced by these statements is identical to the chart in Figure 17.152.

```
proc shewhart history=Flange limits=Flangelim;
   xchart Flwidth*Sample /
      readphases  = ('Production' 'Change 1')
      readindexes = ('Production' 'Change 1')
      phaseref
      phaselegend;
run;
```

The order of the READINDEX=*value-list* is critical. For instance, the previous statements with READINDEXES=('Change 1' 'Production') create the chart in Figure 17.153, in which the control limits are mismatched with the phases.

Figure 17.153 Multiple Phases with Mismatched Control Limits



Case 5. READPHASES=*value|value-list* and READINDEXES=ALL

The only phases read are those for which _PHASE_ equals one of the *values* specified with the READPHASES= option. The chart displays a different set of control limits for each phase, read from the first observation in the LIMITS= data set for which _INDEX_ equals the *value* corresponding to the phase.

For example, the following statements create a chart for the phases 'Production' and 'Change 1' with the control limits read from the second and first observations in Flangelim, respectively:

```
proc shewhart history=Flange limits=Flangelim;
   xchart Flwidth*Sample /
      readphases  = ('Production' 'Change 1')
      readindexes = all
      phaseref
      phaselegend ;
run;
```

The chart is identical to the chart in Figure 17.152. In general, to read a set of phases with identically labeled control limits, you can specify the phases with either the READPHASES= or READINDEXES= option, and you can specify the keyword ALL with the other option.

Case 6. READPHASES=ALL and READINDEXES=ALL or READPHASES= is omitted and READINDEXES=ALL

All phases are read for which _PHASE_ is a value of _INDEX_ in the LIMITS= data set. The chart displays a different set of control limits for each phase, read from the first observation in the LIMITS= data set for which _INDEX_ equals the value of _PHASE_.

For example, the following statements create a chart for the phases ‘Production’ and ‘Change 1’ with control limits read from the second and first observations in Flangelim, respectively. These two phases are read because they are the only phases in Flange with matching _INDEX_ values in Flangelim. The chart is identical to that in Figure 17.152.

```
proc shewhart history=Flange limits=Flangelim;
   xchart Flwidth*Sample /
      readphase = all
      readindex = all
      phaseref
      phaselegend ;
run;
```

Note that an identical chart would be produced if you were to omit the READPHASES= option.

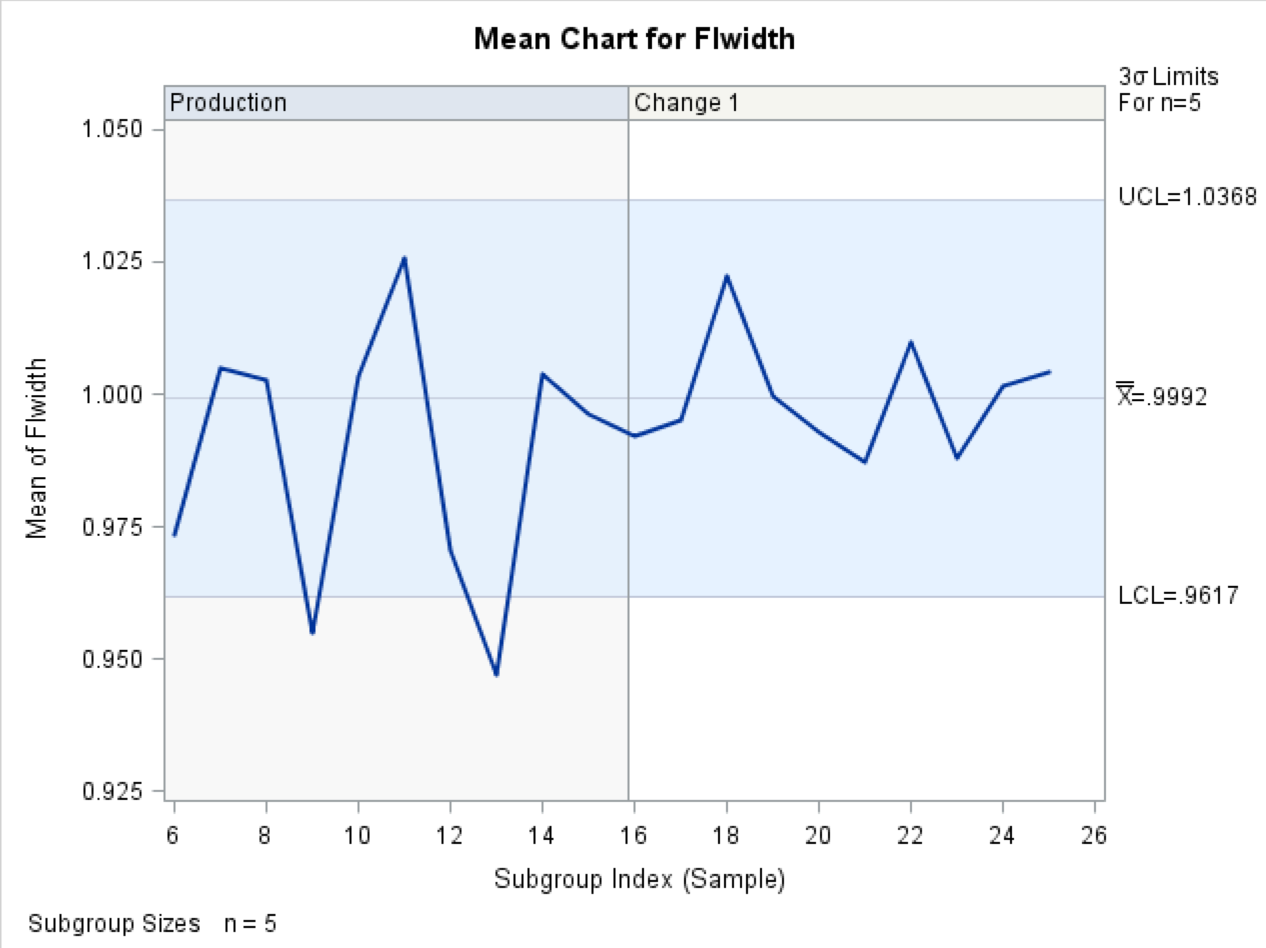
Case 7. READPHASES=*value|value-list* and READINDEXES= is omitted

The only phases read are those for which _PHASE_ equals one of the *values* specified with the READPHASES= option. The chart displays a single set of control limits read from the first observation in the LIMITS= data set for which _VAR_ equals the *process* and _SUBGRP_ equals the name of the *subgroup-variable* specified in the chart statement.

For example, the following statements create a chart for the phases ‘Production’ and ‘Change 1’ with control limits read from the first observation in Flangelim, because this is the first observation for which _VAR_ equals ‘Flwidth’ and _SUBGRP_ equals ‘Sample’.

The chart is displayed in Figure 17.154.

Figure 17.154 *Value-list* for READPHASES= with READINDEXES= Omitted



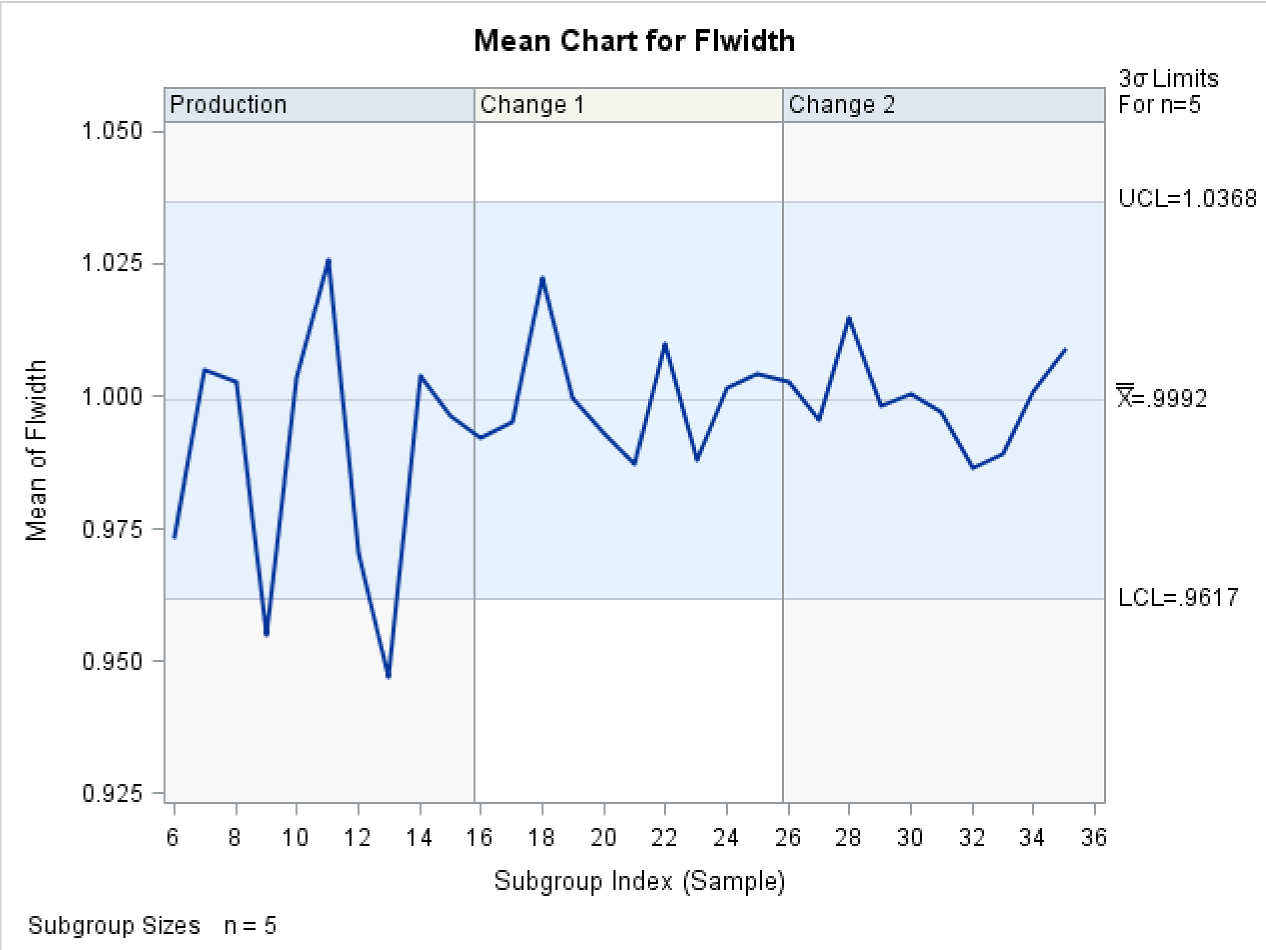
Case 8. READPHASES=ALL and READINDEXES= is omitted or READPHASES= is omitted and READINDEXES= is omitted

All observations are read from the input data set. The chart displays a single set of control limits read from the first observation in the LIMITS= data set for which _VAR_ equals the *process* and _SUBGRP_ equals the name of the *subgroup-variable* specified in the chart statement.

For example, the following statements create a chart for all the phases in Flange with control limits read from the first observation in Flangelim, because this is the first observation for which _VAR_ equals 'FLWIDTH' and _SUBGRP_ equals 'SAMPLE':

The chart is shown in Figure 17.155. Note that an identical chart would be produced if you were to omit the READPHASES= option (except that the phase reference lines and phase legends would be omitted).

Figure 17.155 READPHASES=ALL with READINDEXES= Omitted



Case 9. READPHASES=*value* and READINDEXES=*value-list*

The procedure generates an error message.

The following tables summarize the various combinations of the READPHASES= and READINDEXES= options that you can specify.

Table 17.89 READINDEXES=*index-value*

READPHASES=	Phases Displayed	Control Limits Displayed
phase-value	_PHASE_ = *phase-value*	_INDEX_ = *index-value*
phase-value list	_PHASE_ = *phase-value list*	_INDEX_ = *index-value*
Keyword ALL	_PHASE_ = *index-value*	_INDEX_ = *index-value*
Not Specified	All phases	_INDEX_ = *index-value*

Table 17.90 READINDEXES=*index-value list*

READPHASES=	Phases Displayed	Control Limits Displayed
phase-value	No chart displayed	No chart displayed
phase-value list	_PHASE_ = *phase-value list*	_INDEX_ = *index-value list* with control limits matched to phases in the order listed
Keyword ALL	_PHASE_ = *index-value list*	_INDEX_ = *index-value list*
Not Specified	_PHASE_ = *index-value list*	_INDEX_ = *index-value list*

Table 17.91 READINDEXES=ALL

READPHASES=	Phases Displayed	Control Limits Displayed
phase-value	_PHASE_ = *phase-value*	_INDEX_ = *phase-value*
phase-value list	_PHASE_ = *phase-value list*	_INDEX_ = *phase-value list*
Keyword ALL	_PHASE_ = _INDEX_	_INDEX_ = _PHASE_
Not Specified	_PHASE_ = _INDEX_	_INDEX_ = _PHASE_

Table 17.92 READINDEXES= Not Specified

READPHASES=	Phases Displayed	Control Limits Displayed
phase-value	_PHASE_ = *phase-value*	First LIMITS= observation for which _VAR_=*process* name and _SUBGRP_=*subgroup-variable* name
phase-value list	_PHASE_ = *phase-value list*	same as previous entry
Keyword ALL	All phases	same as previous entry
Not Specified	All phases	same as previous entry

Displaying Auxiliary Data with Stars

NOTE: See *Displaying Auxiliary Data with Stars* in the SAS/QC Sample Library.

In many control chart applications, it is useful to relate the variation of the process to other variables that are being observed simultaneously with the variable that is charted. You can use the features described here to represent auxiliary multivariate data with stars (polygons) that are superimposed on the control chart. See Figure 17.158 for an illustration.

This display, referred to here as a *star chart*, enables you to analyze a process with a control chart while visualizing other quantities such as environmental variables, experimental control variables, or other process variables. The control chart itself can be a standard Shewhart chart, a moving average chart (such as an EWMA chart), or a cumulative sum control chart.

The examples in this section use the HISTORY= input data set Paint (listed in Figure 17.156) and the LIMITS= data set Paintlim (listed in Figure 17.157). The data in Paint consist of the subgroup means, ranges, and sample size (pindexx, pindexr, and pindexn) for an index of paint quality that was monitored on an hourly basis, with six auxiliary variables that were measured simultaneously: thickness, gloss, defects, dust, humidity, and temperature.

Figure 17.156 Listing of the HISTORY= Data Set Paint

hour	pindexx	pindexr	pindexn	thick	gloss	defects	dust	humid	temp
1	5.8	3.0	5	0.2550	0.6800	0.2550	0.2125	0.1700	0.5950
2	6.2	2.0	5	0.2975	0.5950	0.0850	0.1700	0.2125	0.5525
3	3.7	2.5	5	0.3400	0.3400	0.4250	0.2975	0.2550	0.2125
4	3.2	6.5	5	0.3400	0.4675	0.3825	0.3485	0.2125	0.2125
5	4.7	0.5	5	0.5100	0.4250	0.5950	0.4080	0.5100	0.4675
6	5.2	3.0	5	0.5100	0.3400	0.6800	0.5525	0.5525	0.5525
7	2.6	2.0	5	0.4250	0.0425	0.8500	0.5355	0.5525	0.2550
8	2.1	1.0	5	0.3400	0.0170	0.8075	0.5950	0.5950	0.1700

Figure 17.157 Listing of the LIMITS= Data Set Paintlim

Obs	_var_	_subgrp_	_type_	_limitn_	_sigmas_	_lclx_	_mean_	_uclx_	_lclr_	_r_	_uclr_	_stddev_
1	pindex	hour	estimate	5	3	2.395	3.875	5.355	0	2.5625	5.4184	1.10171

The basic variable analyzed with the control chart (in this case, paint index) is referred to as the *process*. The auxiliary variables (in this case, thickness, gloss, defects, dust, humidity, and temperature) are referred to as *vertex variables*, because their values are represented by the vertices of the stars. A star chart can reveal relationships between the process and the vertex variables, and it can reveal relationships among the vertex variables.

You can create star charts for any number of vertex variables. However, the resolution of your graphics device and the number of subgroups per page will limit your ability to distinguish the vertices of the stars. A practical upper limit is twelve vertex variables.

You can specify star options in all chart statements of the SHEWHART procedure except the BOXCHART statement. You can use these options to

- specify the style of the star
- add reference circles to indicate limits of variation for the stars
- add a legend identifying the relationship between vertices and vertex variables
- label the vertices
- specify colors and line types for individual stars
- specify the size of the stars
- specify different methods of standardization for the vertex variables

The star options do not apply if the LINEPRINTER option is specified.

NOTE: A star chart is *not* the same as a multivariate control chart or a T^2 chart. A star chart is simply a univariate control chart enhanced with stars that represent auxiliary multivariate data. A multivariate control chart displays summary statistics (such as T^2) and control limits determined for a number of processes simultaneously. For an example of a multivariate control chart, see Figure 17.222. Figure 17.223 displays a multivariate control chart in which the principal components of the T^2 statistic are displayed with stars.

Creating a Basic Star Chart

NOTE: See *Displaying Auxiliary Data with Stars* in the SAS/QC Sample Library.

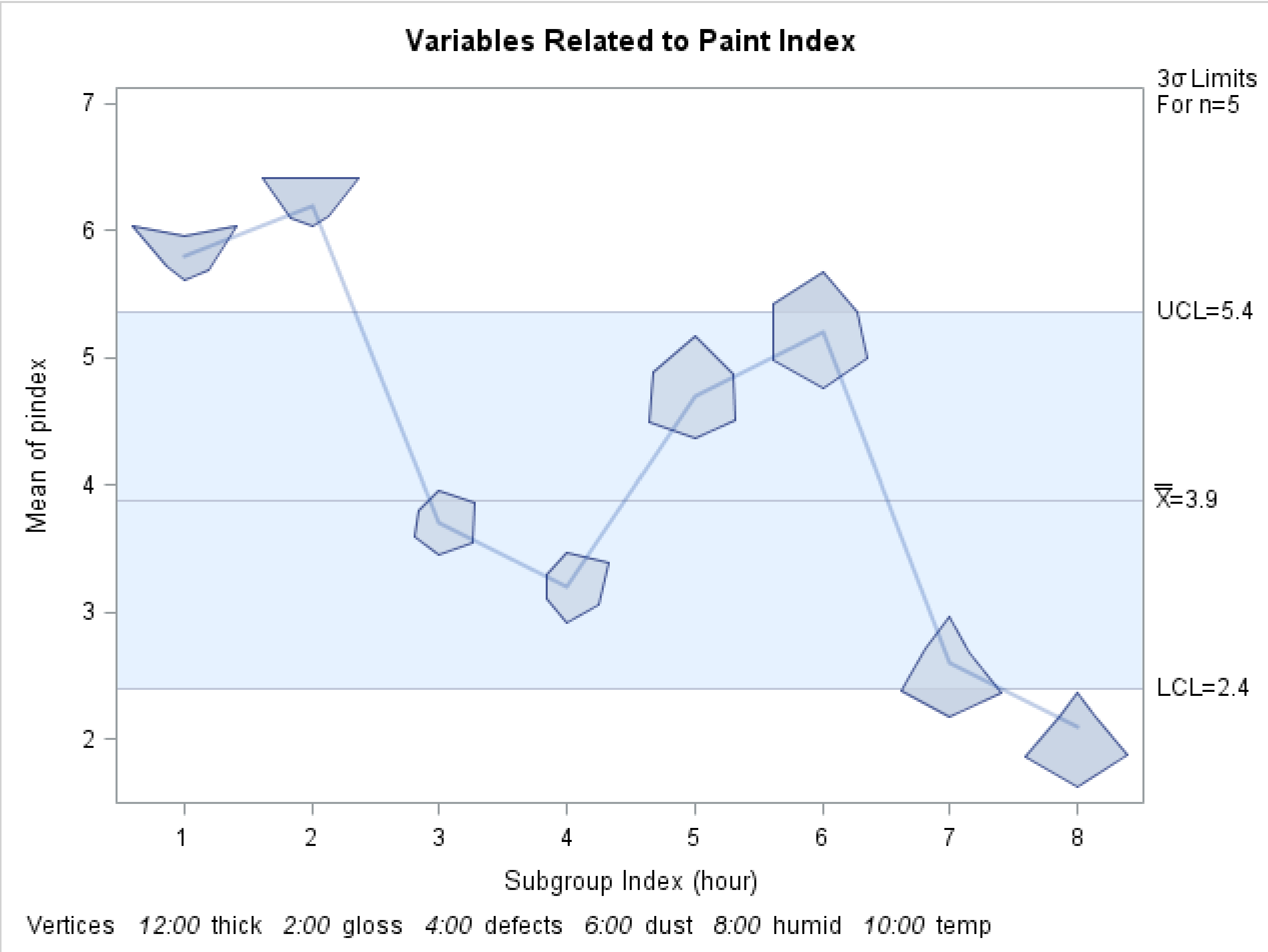
The following statements create the star chart shown in Figure 17.158:

```
ods graphics on;
title 'Variables Related to Paint Index';
proc shewhart history=Paint limits=Paintlim;
   xchart pindex*hour /
      nolegend
      odstitle     = title
      starvertices = (thick gloss defects dust humid temp);
run;
```

This chart is essentially an $\bar{X}$ chart for paint index. However, the chart also provides information about thickness, gloss, defects, dust, humidity, and temperature. These six variables are represented by the vertices of the stars, as indicated by the legend at the bottom of the chart. By default, the legend uses a clock representation for the vertices; for instance, dust corresponds to the vertex at the six o'clock position.

The stars are centered at the points for average paint index, and the distance from the center to a vertex represents the standardized value of the variable corresponding to the vertex. The star chart reveals that relatively high values of gloss (two o'clock) and temperature (ten o'clock) are associated with high out-of-control averages for paint index. Likewise, relatively high values of defects (four o'clock) and humidity (eight o'clock) are associated with low out-of-control averages for paint index. The star shapes reveal similarities in the data for runs 1 and 2, runs 3 and 4, runs 5 and 6, and runs 7 and 8.

Figure 17.158 A Basic Star Chart



Adding Reference Circles to Stars

NOTE: See *Displaying Auxiliary Data with Stars* in the SAS/QC Sample Library.

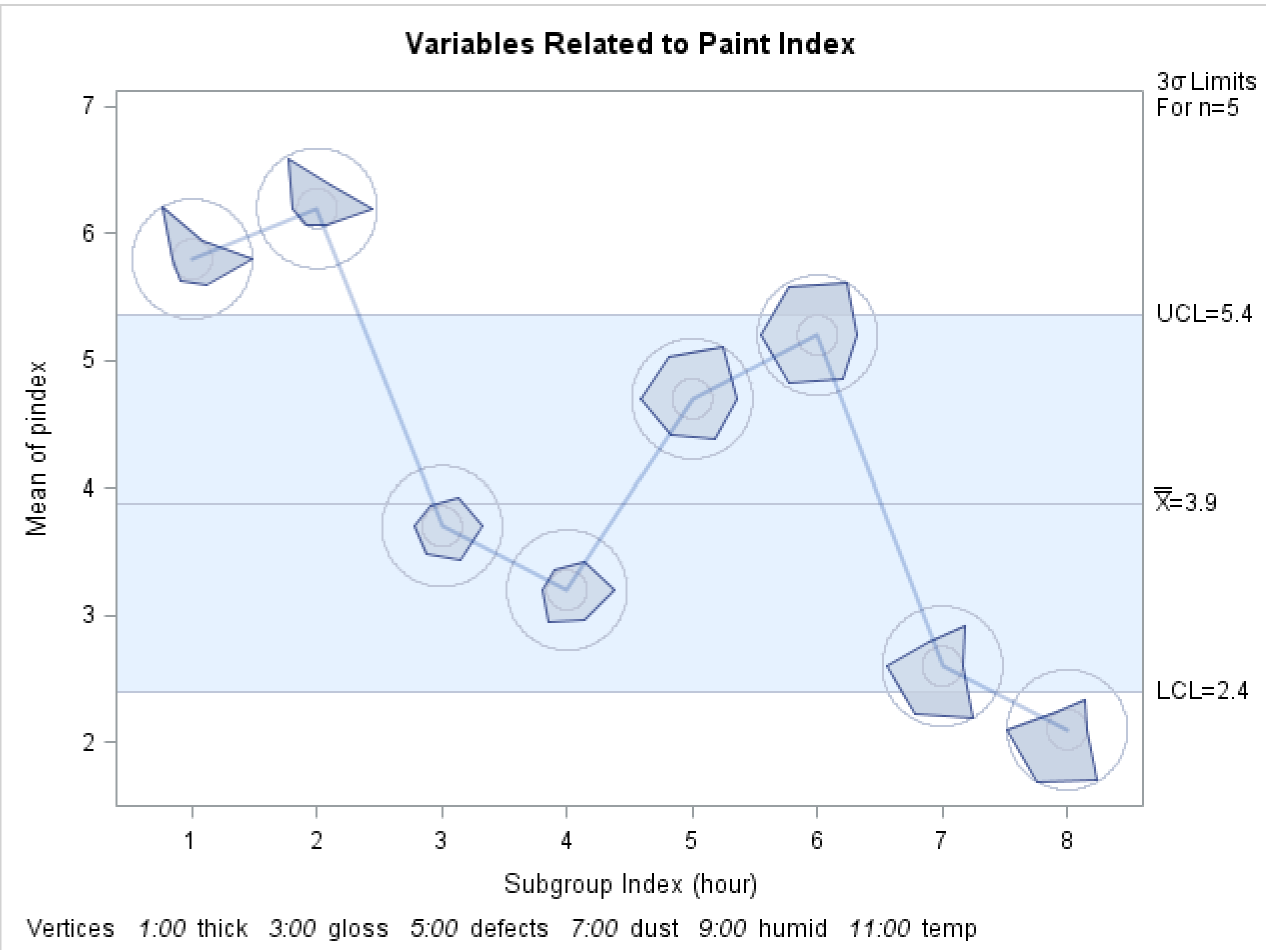
You can add reference circles to a star chart to represent limits of variation for the vertex variables. The following statements add two special reference circles, called the *inner circle* and the *outer circle*, to the star chart in Figure 17.158:

```
title 'Variables Related to Paint Index';
proc shewhart history=Paint limits=Paintlim;
   xchart pindex*hour /
      nolegend
      odstitle     = title
      starvertices = (thick gloss defects dust humid temp)
      starcircles  = 0.0 1.0
      lstarcircles = 1   2
      starstart    = '1:00'T ;
run;
```

The star chart shown in Figure 17.159 displays the two reference circles centered about each point. The STARCIRCLES= value 0.0 requests the *inner circle*, and the value 1.0 requests the *outer circle*. Whether or not they are displayed, these circles are always associated with each star.

The interpretation of the inner and outer circles depends on the method used to standardize the vertex variables. By default (as in this example), the data for each vertex variable are standardized by the range of the variable values taken across subgroups. That is, the inner circle represents the minimum value, and the outer circle represents the maximum value. You can specify other methods of standardization (see "Specifying the Method of Standardization" on page 2051.

Figure 17.159 Star Chart with Inner and Outer Circles Added



Note that the STARCIRCLES= option does not specify the physical radius of a reference circle. Instead, this option specifies the radius relative to the radii of the inner and outer circles. Thus, specifying STARCIRCLES=0.0 always displays the inner circle, and specifying STARCIRCLES=1.0 always displays the outer circle. Specifying STARCIRCLES=0.5 displays a reference circle halfway between the inner and outer circles. You can specify the physical radii (in percent screen units) of the inner and outer circles using the STARINRADIUS= and STAROUTRADIUS= options. In the preceding statements, the LSTARCIRCLES= option specifies line types (1=solid and 2=dashed) for the inner and outer circles. You can also use the WSTARCIRCLES= option to control the thickness of the circles.

The STARSTART= option gives the starting position for the first vertex variable listed. In the preceding example, this option specifies that the vertex corresponding to thick is to be positioned at one o'clock. The remaining vertices are uniformly spaced clockwise and correspond to the vertex variables in the order listed with the STARVERTICES= option.

For more information about the star options, see the appropriate entries in "Dictionary of Options: SHEWHART Procedure" on page 1946.

Specifying the Style of Stars

NOTE: See *Star Charts-Specifying the Style of Stars* in the SAS/QC Sample Library.

The following statements create star charts for paint index using different styles for the stars specified with the STARTYPE= option:

```
ods graphics on;
title 'Variables Related to Paint Index';
proc shewhart history=Paint limits=Paintlim;
   xchart pindex * hour /
      nolegend
      odstitle     = title
      starvertices = ( thick gloss defects dust humid temp )
      starstart    = '1:00'T
      startype     = wedge;
   xchart pindex * hour /
      nolegend
      odstitle     = title
      starvertices = ( thick gloss defects dust humid temp )
      starstart    = '1:00'T
      startype     = radial;
   xchart pindex * hour /
      nolegend
      odstitle     = title
      starvertices = ( thick gloss defects dust humid temp )
      starstart    = '1:00'T
      startype     = spoke;
   xchart pindex * hour /
      nolegend
      odstitle     = title
      starvertices = ( thick gloss defects dust humid temp )
      starstart    = '1:00'T
      startype     = corona;
run;
```

The charts are shown in Figure 17.160, Figure 17.161, Figure 17.162, and Figure 17.163. The default style for the stars is STARTYPE=POLYGON, which is illustrated in Figure 17.158 and Figure 17.159. For more information, see the entry for the STARTYPE= option in "Dictionary of Options: SHEWHART Procedure" on page 1946.

Figure 17.160 Star Chart Using STARTYPE=WEDGE

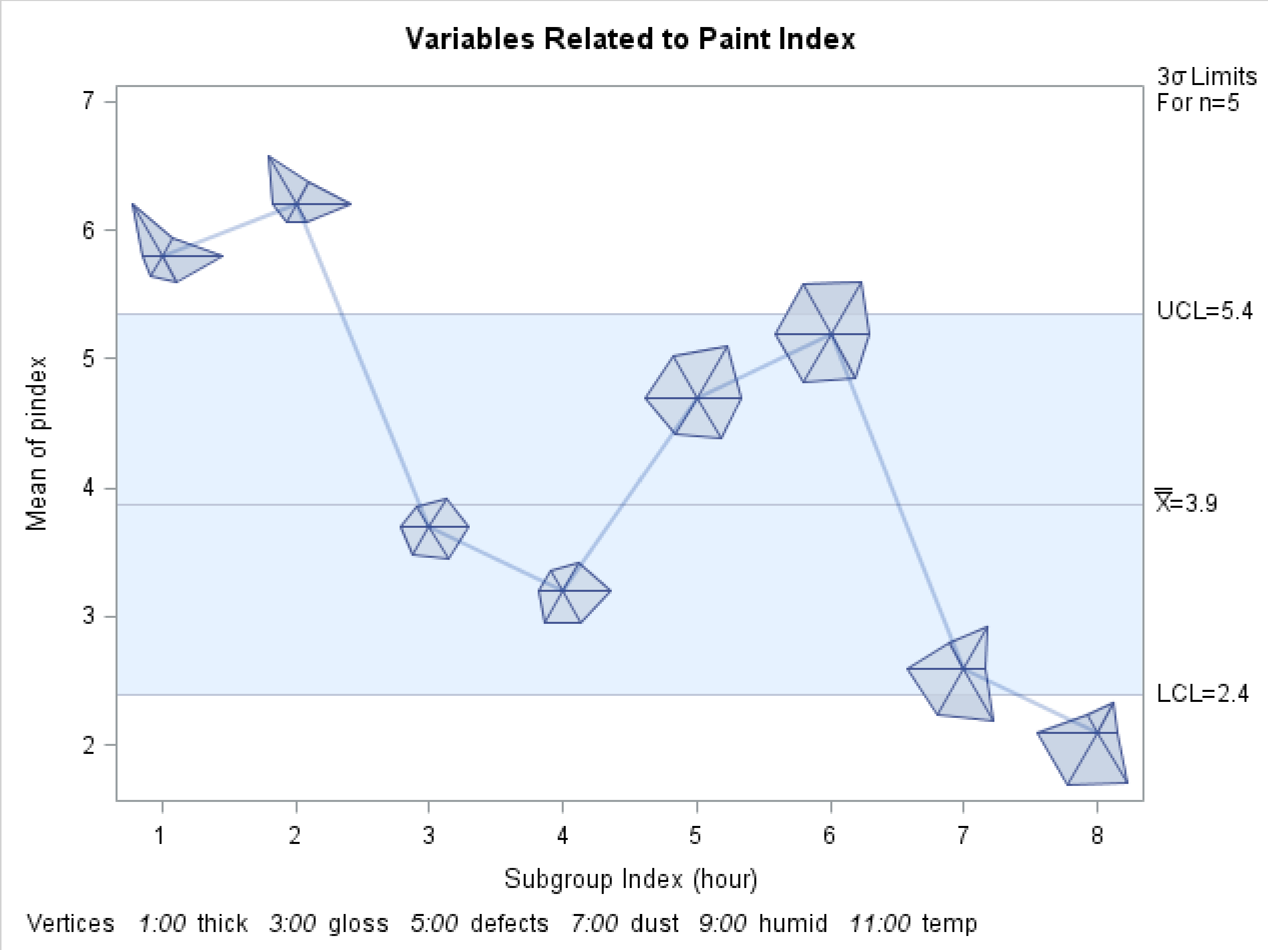


Figure 17.161 Star Chart Using STARTYPE=RADIAL

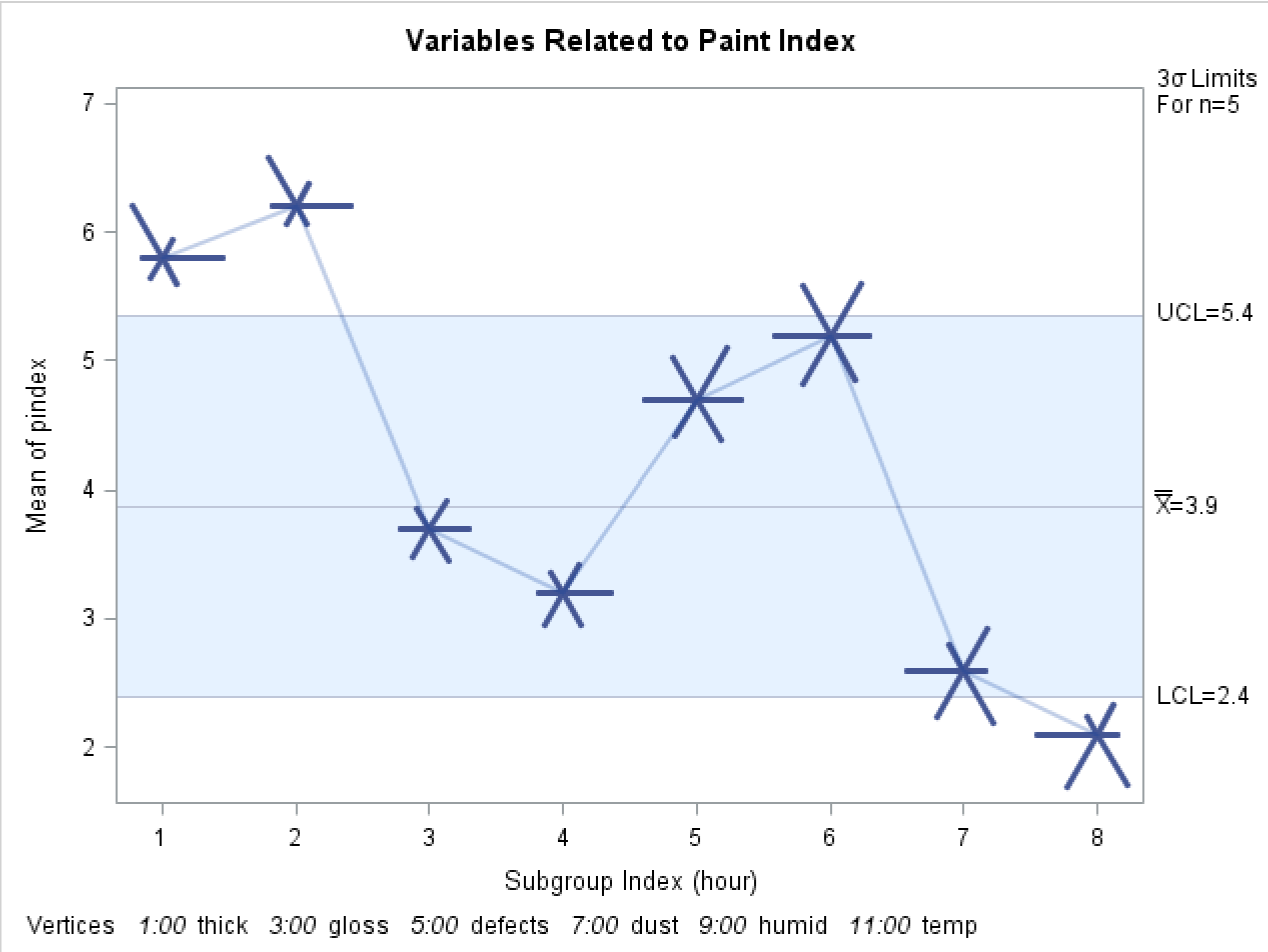


Figure 17.162 Star Chart Using STARTYPE=SPOKE

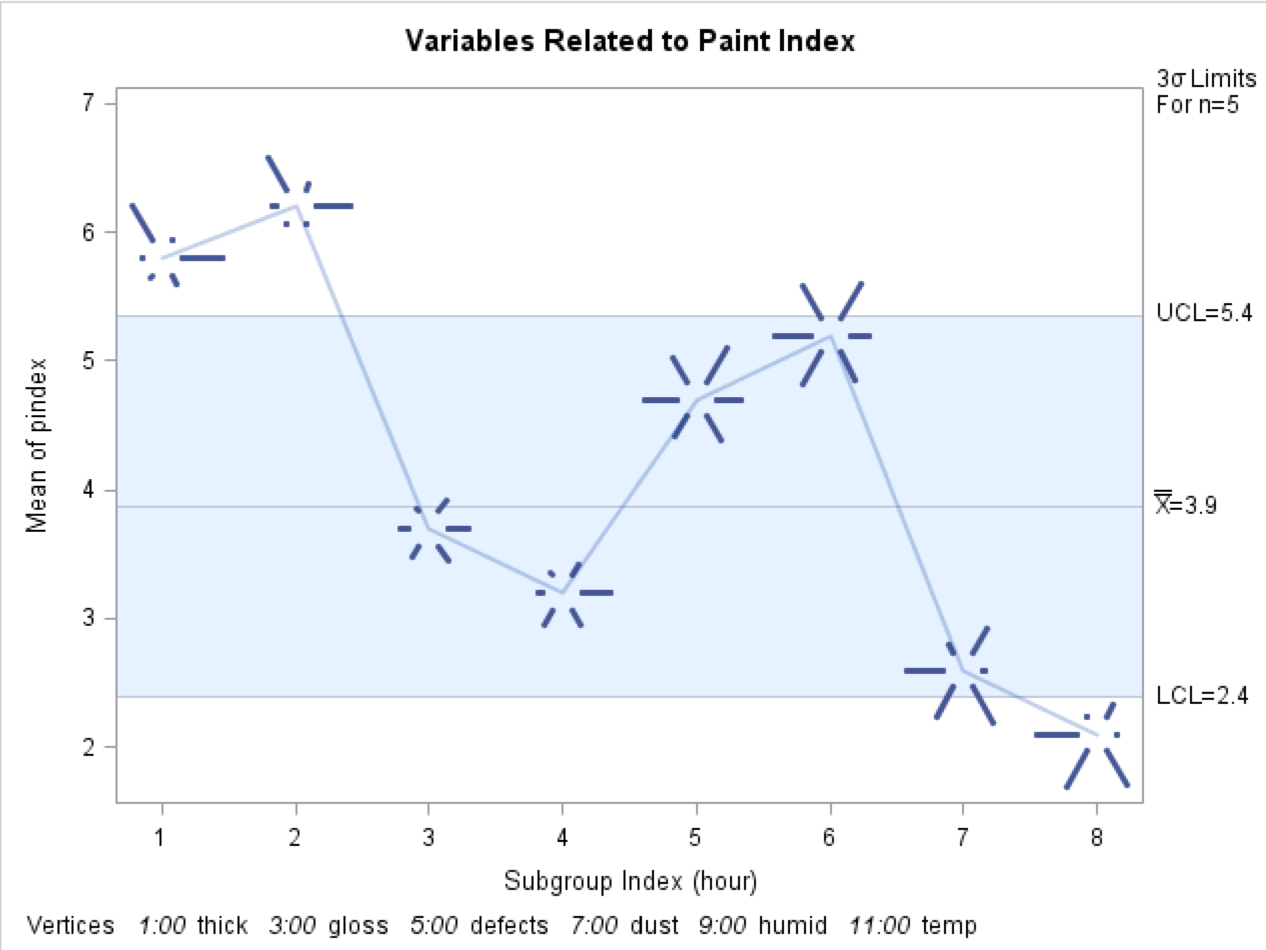
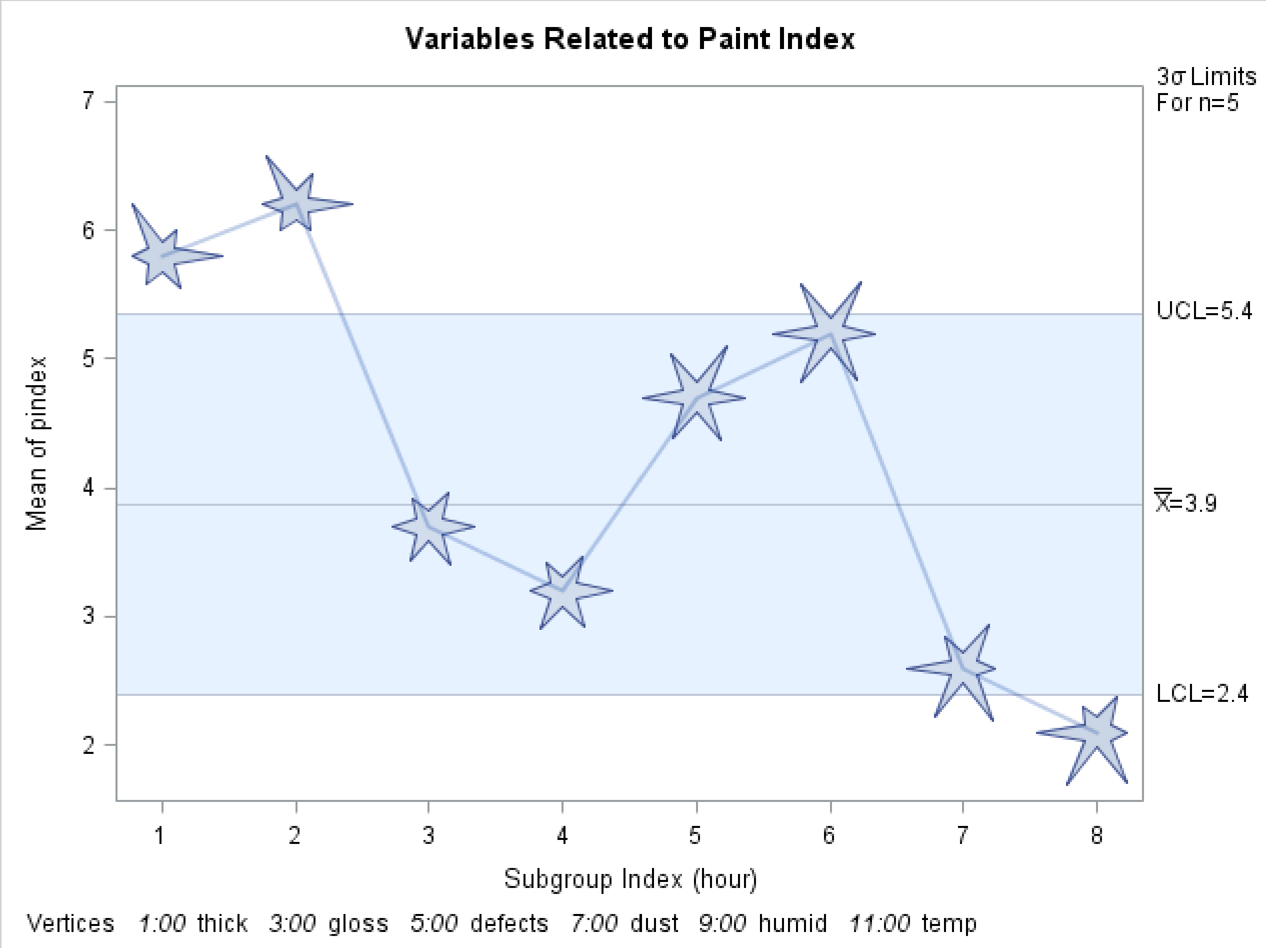


Figure 17.163 Star Chart Using STARTYPE=CORONA



Specifying the Method of Standardization

NOTE: See *Standardization Method on Star Charts* in the SAS/QC Sample Library.

In the previous examples in this section, the default method of standardization (based on ranges) is used for all six vertex variables. You can specify alternative methods with the STARSPECS= option. For example, specifying STARSPECS=3 standardizes each vertex variable so that the inner circle corresponds to three standard deviations below the mean and the outer circle corresponds to three standard deviations above the mean (that is, the circles represent 3σ limits). Specifying STARSPECS=k requests circles corresponding to $k\sigma$ limits, and specifying STARSPECS=0 requests the default method.

In some applications, it may be necessary to use distinct methods of standardization for the vertex variables. You can do this by creating an input SAS data set that provides the method for each vertex variable and specifying this data set with the STARSPECS= option.

The following statements create a data set named myspecs that specifies standardization methods for the vertex variables used in the previous examples:

```
data myspecs;
   length _var_     $8
          _label_   $16 ;
   input  _var_ _label_ _lspoke_ _sigmas_ _lsl_ _usl_ ;
   datalines;
thick    Thickness    1       .      0.25  0.50
gloss    Gloss        1       .      0.10  0.60
defects  Defects      1       .      0.10  0.60
dust     Dust         2      3.0      .     .
humid    Humidity     2      0.0      .     .
temp     Temperature  2      0.0      .     .
;
```

This data set contains a number of special variables whose names begin and end with an underscore.

Variable Name	Description
LABEL	label for identifying the vertex (used in conjunction with the STARLABEL= option). This must be a character variable of length 16 or less.
LSL	lower specification limit
LSPOKE	line style for spokes used with STARTYPE=RADIAL, STARTYPE=SPOKE, and STARTYPE=WEDGE
SIGMAS	multiple of standard deviations above and below the average. A value of zero specifies standardization based on the range.
USL	upper specification limit
VAR	name of vertex variable. This must be a character variable whose length is no greater than 32.

Standardization is specified with the variables _SIGMAS_, _LSL_, and _USL_, as follows:

- Since nonmissing specification limits (_LSL_ and _USL_) are provided for the variables thick, gloss, and defects, the values of these variables are scaled so that the inner circle represents the lower specification limit and the outer circle represents the upper specification limit.
- Since _SIGMAS_ is equal to 3 for dust (and since both _LSL_ and _USL_ are missing), values of dust are scaled so that the inner circle represents three standard deviations below the mean, and the outer circle represents three standard deviations above the mean. The mean and standard deviation are calculated across all subgroups.
- Since _SIGMAS_ is equal to 0 for humid and temp (and since both _LSL_ and _USL_ are missing), values of humid and temp are scaled so that the inner circle represents the minimum and the outer circle represents the maximum. The minimum and maximum are calculated across all subgroups.

The following statements use the data set myspecs to create a star chart for paint index:

```
ods graphics on;
title 'Variables Related to Paint Index';
proc shewhart history=Paint limits=Paintlim;
   xchart pindex * hour /
      nolegend
      odstitle     = title
      starvertices = ( thick gloss defects dust humid temp )
      startype     = wedge
      starcircles  = 0.0 1.0
      lstarcircles = 2   2
      starstart    = -30
      labelfont    = simplex
      starlegend   = degrees
      starspecs    = myspecs
      starlabel    = high ;
run;
```

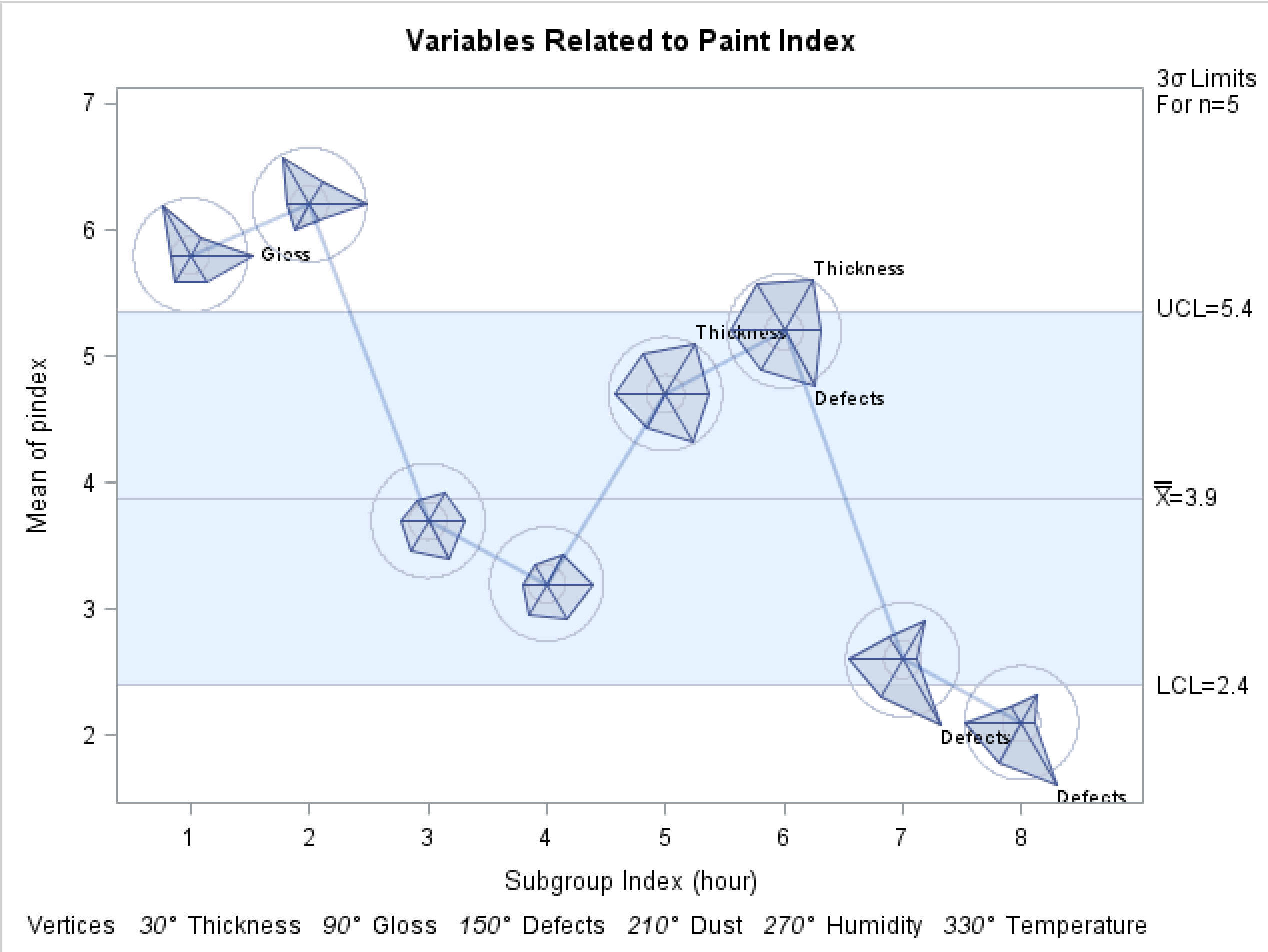
The chart is shown in Figure 17.164. Specifying STARLEGEND=DEGREES requests a legend that identifies the vertex variables by their angles (in degrees) rather than their clock positions. Here, zero degrees corresponds to twelve o'clock, and the degrees are measured clockwise. The first vertex variable is positioned at 30 degrees, as specified with the STARSTART= option. Note that you specify the STARSTART= value as a negative number to indicate that it is in degrees.

In Figure 17.161 the vertices that exceed the outer circle are labeled with the value of the variable _LABEL_ in the STARSPECS= data set. This type of labeling is requested by specifying STARLABEL=HIGH. A font (SIMPLEX) for the labels is specified with the LABELFONT= option.

The vertices for thick at HOUR=5, 6, and 7 are truncated, as indicated in the SAS log. The truncation value is the physical radius of an imaginary circle referred to as the *bounding circle* that lies outside the outer circle. In general, any vertex that exceeds the bounding circle is truncated to the *bounding radius*. This is done so that unusually large vertex variable values will not result in grossly distorted stars. You can specify a different bounding radius with the STARBDRADIUS= option.

The spokes corresponding to the environmental variables dust, humid, and temp are drawn with a dashed line style to distinguish them from the quality variables thick, gloss, and defects, whose spokes are drawn with a solid line. The styles are specified by the variable _LSPOKE_. Refer to *SAS/GRAPH: Reference* for a complete list of line styles. If you are producing charts in color, you can also use the variable _CSPOKE_ in the STARSPECS= data set to assign colors to the spokes.

Figure 17.164 Star Chart Using STARSPECS= Specifications



For more information about the options used in this example, see the appropriate entries in "Dictionary of Options: SHEWHART Procedure" on page 1946.

Displaying Trends in Process Data

NOTE: See *X-Bar Chart for Data with Nonlinear Trend* in the SAS/QC Sample Library.

Time trends due to tool wear, environmental changes, and other gradual process changes are sometimes observed in $\bar{X}$ charts. The presence of a systematic trend makes it difficult to interpret the chart because the control limits are designed to indicate expected variation strictly due to common causes.

You can use the REG procedure (or other modeling procedure) in conjunction with the SHEWHART procedure to determine whether a process with a time trend is in control. With the REG procedure, you can model the trend and save the fitted subgroup means ($\widehat{\bar{X}}_t$) and the residual subgroup means ($\bar{X}_t - \widehat{\bar{X}}_t$) in an output data set. Then, using this data as input to the SHEWHART procedure, you can create a *trend chart*, which displays a trend plot of the fitted subgroup means together with an $\bar{X}$ chart for the residual subgroup means, thus removing the time-dependent component of the data from its random component.

Having accounted for the time trend, you can decide whether the process is in control by examining the $\bar{X}$ chart.

The following example illustrates the steps used to create a trend chart for a SAS data set named TOOLWEAR that contains diameter measurements for 20 subgroup samples each consisting of eight parts:

```
data toolwear;
   input hour @;
   do i=1 to 8;
      input Diameter @;
      output;
   end;
   drop i;
   datalines;
   1    10.0434   9.9427   9.9548   9.8056
        10.0780  10.0302  10.1173  10.0215
   2    10.1976   9.9654  10.0425  10.1183
        10.0963  10.1635  10.1382  10.1265
   3    10.0552  10.0695  10.2495  10.1753
        10.1268  10.1229  10.1351  10.2084
   4    10.1600  10.1378  10.2433  10.2634
        10.1808  10.1601  10.1035  10.0027
   5     9.9611  10.4322  10.1066  10.2653
        10.0310  10.1409  10.2709  10.0585
   6    10.2208  10.2298  10.2427  10.2315
        10.2048  10.2824  10.3347  10.1650
   7    10.2670  10.3793  10.2539  10.4037
        10.3281  10.1327  10.1986  10.1841
   8    10.2537  10.1981  10.2935  10.4308
        10.3195  10.3122  10.2033  10.3220
   9    10.2488  10.1866  10.3678  10.1755
        10.3225  10.2375  10.2466  10.3387
  10    10.3744  10.5221  10.2890  10.3123
        10.5134  10.3212  10.3139  10.1565
  11    10.3525  10.3237  10.4605  10.5139
        10.3650  10.1171  10.3863  10.2061
  12    10.3279  10.3338  10.1885  10.2810
        10.2400  10.3617  10.2938  10.2656
  13    10.1651  10.2404  10.1814  10.2330
        10.3094  10.3373  10.3266  10.3830
  14    10.3554  10.4577  10.5435  10.4805
        10.5358  10.4631  10.3689  10.1750
  15    10.2962  10.4221  10.3578  10.4694
        10.3465  10.4499  10.4645  10.3986
  16    10.6002  10.1924  10.3437  10.3228
        10.3438  10.3503  10.3761  10.3137
  17    10.4015  10.3592  10.3187  10.4108
        10.4834  10.4807  10.2178  10.3897
  18    10.4514  10.4492  10.3373  10.4497
        10.4197  10.3496  10.3949  10.1585
  19    10.3445  10.3310  10.4472  10.4684
        10.3975  10.2714  10.2952  10.6255
  20    10.2612  10.3824  10.4240  10.3120
        10.5744  10.4204  10.4073  10.3783
;
```

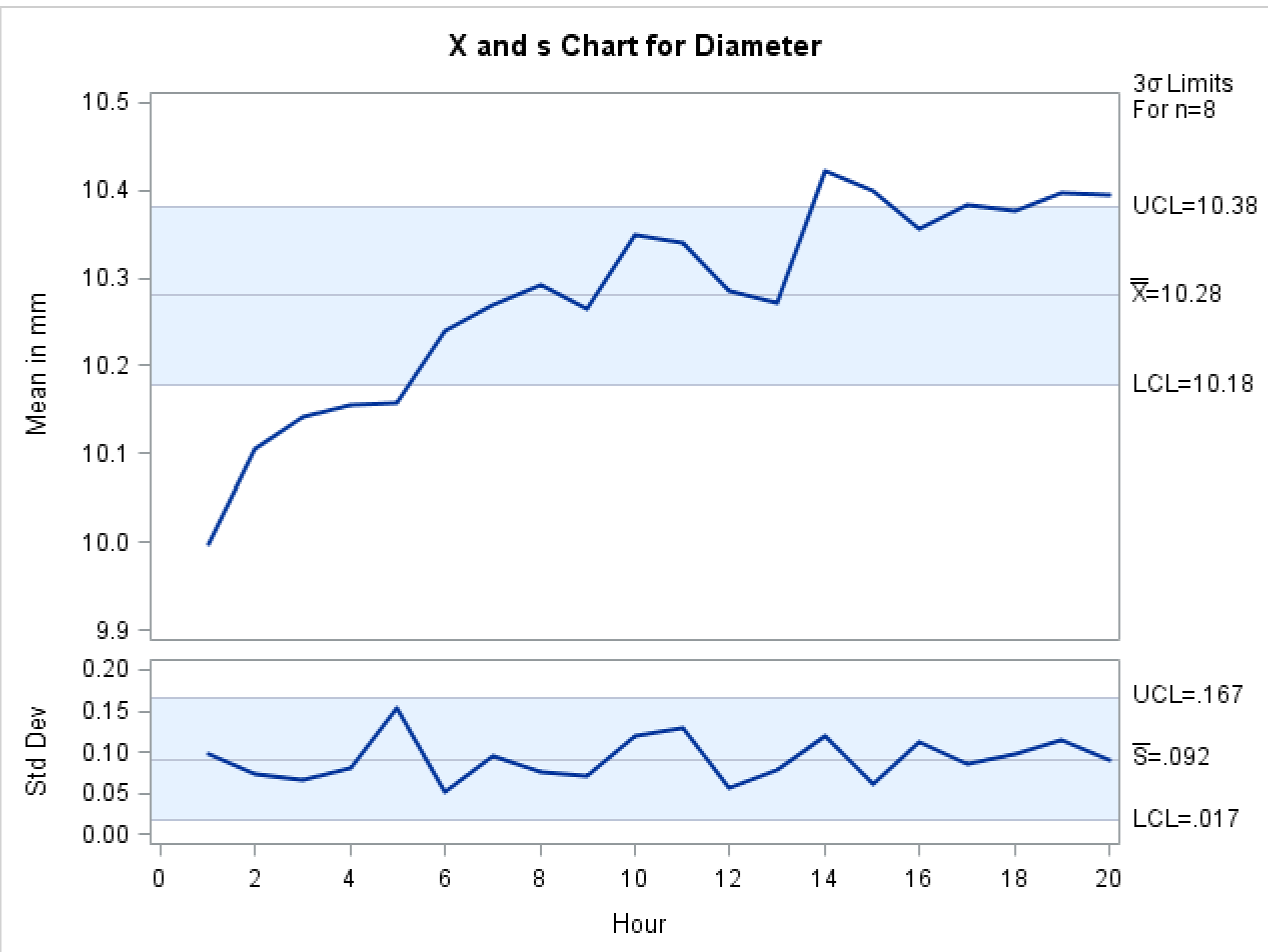
Step 1: Preliminary Mean and Standard Deviation Charts

The following statements create $\bar{X}$ and s charts for the diameter data:

```
ods graphics on;
title f=qcfont1 'X ' f=none 'and s Chart for Diameter';
proc shewhart data=toolwear;
   xschart Diameter*hour /
      odstitle   = title
      outhistory = submeans
      nolegend ;
   label Diameter = 'Mean in mm';
   label hour     = 'Hour';
run;
```

The charts are shown in Figure 17.165. The subgroup standard deviations are all within their control limits, indicating the process variability is stable. However, the $\bar{X}$ chart displays a nonlinear trend that makes it difficult to decide if the process is in control. Subsequent investigation reveals that the trend is due to tool wear.

Figure 17.165 $\bar{X}$ and s Charts for TOOLWEAR Data



Note that the symbol $\bar{X}$ is displayed in the title with the special font QCFONT4, which matches the SWISS font used for the remainder of the title. See Chapter D, "Special Fonts in SAS/QC Software," for a description of the fonts available for displaying $\bar{X}$ and related symbols.

Step 2: Modeling the Trend

The next step is to model the trend as a function of hour. The $\bar{X}$ chart in Figure 17.165 suggests that the mean level of the process (saved as DiameterX in the OUTLIMITS= data set SUBMEANS) grows as the log of HOUR. The following statements fit a simple linear regression model in which DiameterX is the response variable and LOGHOUR (the log transformation of HOUR) is the predictor variable. Part of the printed output produced by PROC REG is shown in Figure 17.166.

```
data submeans;
   set submeans;
   loghour=log(hour);
run;

proc reg data=submeans ;
   model Diameterx=loghour;
   output out=regdata predicted=fitted ;
run;
```

Figure 17.166 Trend Analysis for Diameter from PROC REG

The REG Procedure
Model: MODEL1
Dependent Variable: DiameterX Mean of Diameter

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	9.99056	0.02185	457.29	<.0001
loghour		1	0.13690	0.00967	14.16	<.0001

Figure 17.166 shows that the fitted equation can be expressed as

$$\widehat{\bar{\bar{X}}}_t = 9.99 + 0.14 \times \log(t)$$

where $\widehat{\bar{\bar{X}}}_t$ is the fitted subgroup average.[12] A partial listing of the OUT= data set REGDATA created by the REG procedure is shown in Figure 17.167.

Figure 17.167 Partial Listing of the Output Data Set regdata from the REG Procedure

hour	DiameterX	DiameterS	DiameterN	loghour	fitted
1	9.9992	0.09726	8	0.00000	9.9906
2	10.1060	0.07290	8	0.69315	10.0855
3	10.1428	0.06601	8	1.09861	10.1410
4	10.1565	0.08141	8	1.38629	10.1803
5	10.1583	0.15454	8	1.60944	10.2109

[12] Although this example does not check for the existence of a trend, you should do so by using the hypothesis tests provided by the REG procedure.

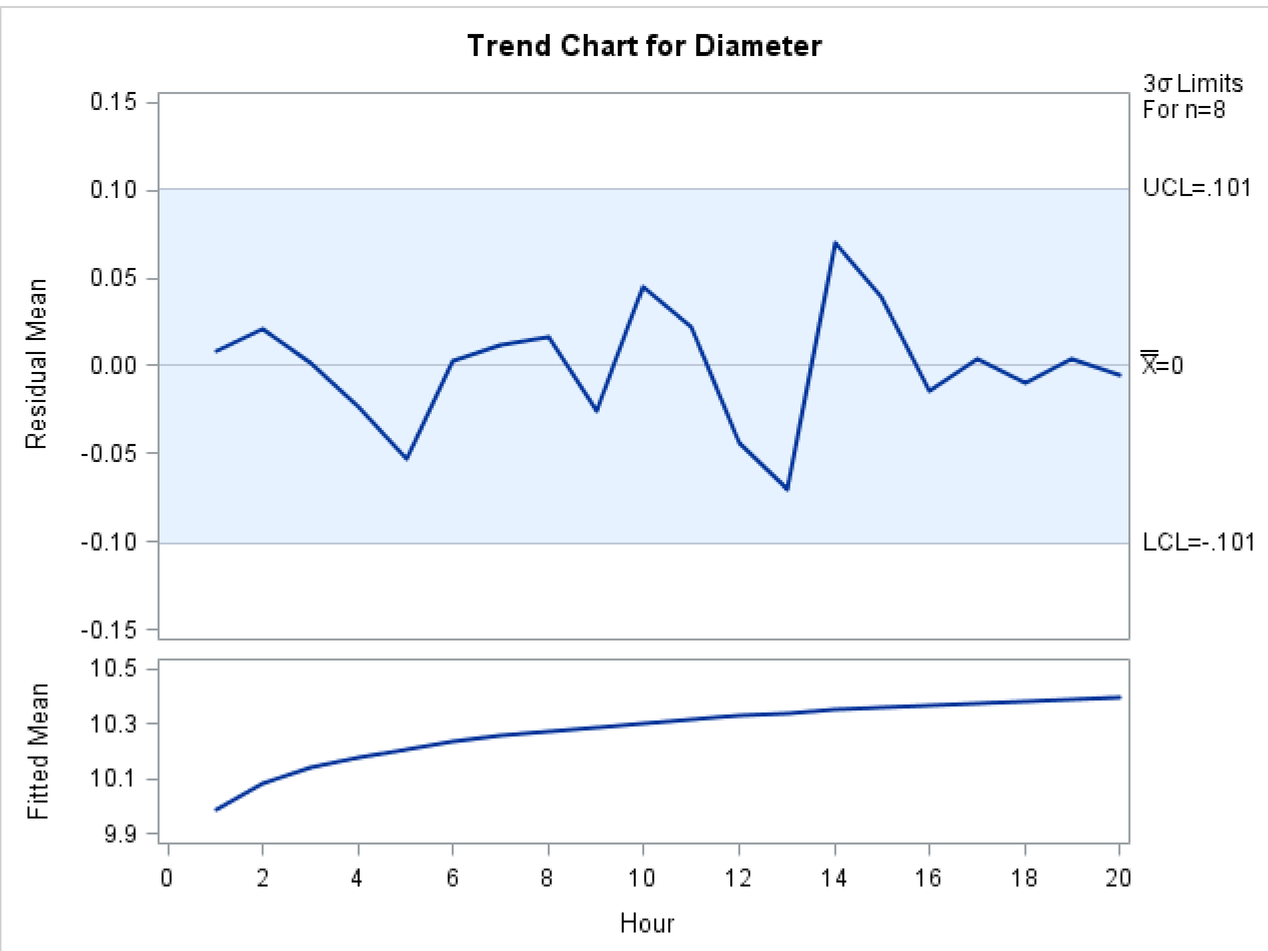
Step 3: Displaying the Trend Chart

The third step is to create a trend chart with the SHEWHART procedure, as follows:

```
title 'Trend Chart for Diameter';
proc shewhart history=regdata;
   xchart Diameter*hour /
      trendvar = fitted
      split    = '/'
      odstitle = title
      stddevs
      nolegend;
   label Diameterx = 'Residual Mean/Fitted Mean';
   label hour      = 'Hour';
run;
```

The chart is shown in Figure 17.168. The values of FITTED are plotted in the lower half of the trend chart. The upper half of the trend chart is an $\bar{X}$ chart for the residual means (DiameterX – FITTED). The CNEEDLES= option specifies that the residuals are to be represented by vertical bars as deviations from the central line. The $\bar{X}$ chart in Figure 17.168 shows that, after accounting for the trend, the mean level of the process is in control.

Figure 17.168 Trend Chart for Diameter Data



If the data are correlated in time, you can use the ARIMA or AUTOREG procedures in place of the REG procedure to remove autocorrelation structure and display a control chart for the residuals; for an example, see "Autocorrelation in Process Data" on page 2096. Another application of the TRENDVAR= option is the display of nominal values in control charts for short runs; see "Short Run Process Control" on page 2113.

Clipping Extreme Points

NOTE: See *Clipping Extreme Points* in the SAS/QC Sample Library.

In some control chart applications, the out-of-control points can be so extreme that the remaining points are compressed to a scale that is difficult to read. In such cases, you can clip the extreme points so that a more readable chart is displayed, as illustrated in the following example.

A company producing copper tubing uses $\bar{X}$ and R charts to monitor the diameter of the tubes. Based on previous production, known values of 70mm and 0.75mm are available for the mean and standard deviation of the diameter. The diameter measurements (in millimeters) for 15 batches of five tubes each are provided in the data set NEWTUBES.

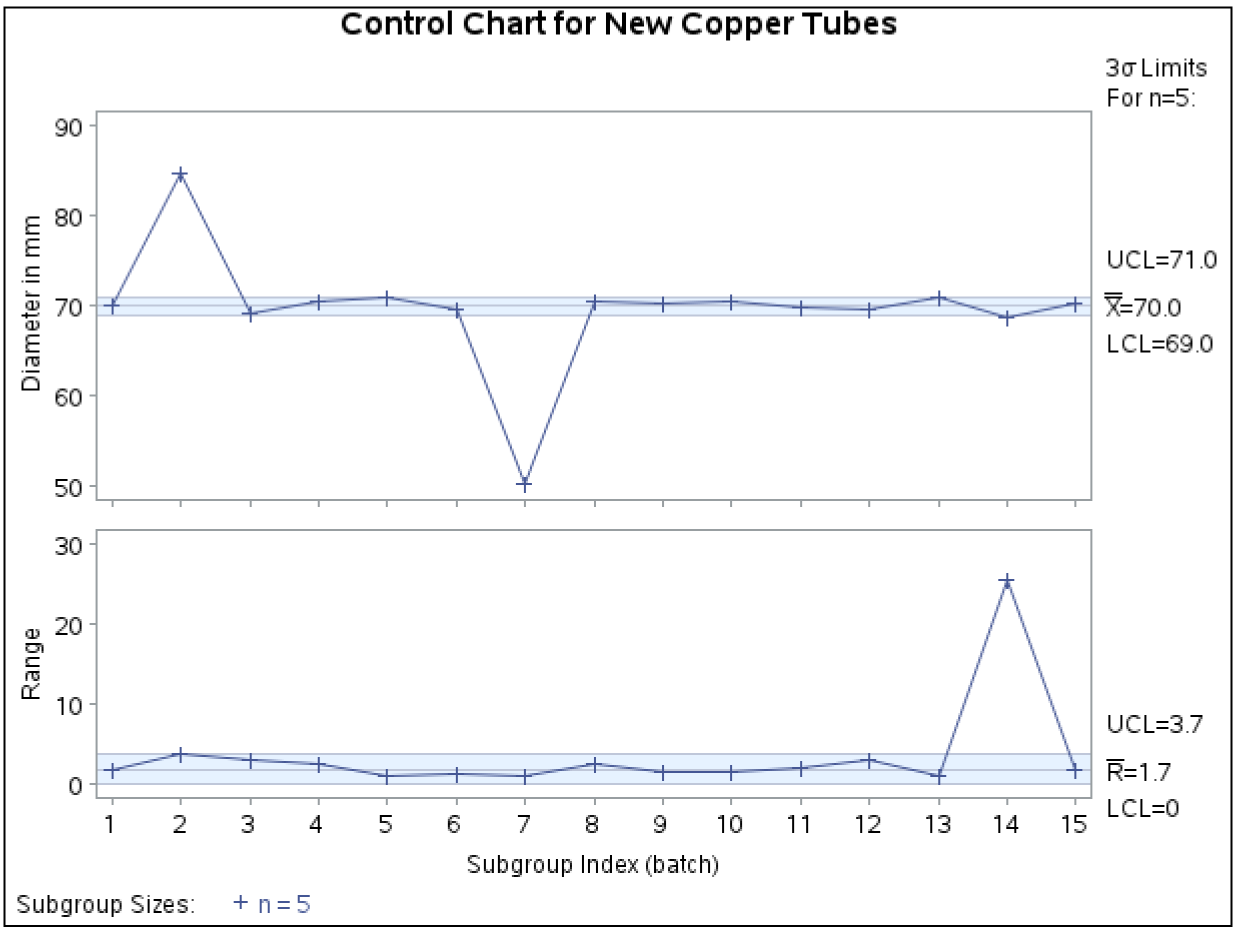
```
data newtubes;
   label Diameter='Diameter in mm';
   do batch = 1 to 15;
      do  i = 1 to 5;
         input Diameter @@;
         output;
      end;
   end;
   datalines;
69.13  69.83  70.76  69.13  70.81
85.06  82.82  84.79  84.89  86.53
67.67  70.37  68.80  70.65  68.20
71.71  70.46  71.43  69.53  69.28
71.04  71.04  70.29  70.51  71.29
69.01  68.87  69.87  70.05  69.85
50.72  50.49  49.78  50.49  49.69
69.28  71.80  69.80  70.99  70.50
70.76  69.19  70.51  70.59  70.40
70.16  70.07  71.52  70.72  70.31
68.67  70.54  69.50  69.79  70.76
68.78  68.55  69.72  69.62  71.53
70.61  70.75  70.90  71.01  71.53
74.62  56.95  72.29  82.41  57.64
70.54  69.82  70.71  71.05  69.24
;
```

The following statements create the $\bar{X}$ and R charts shown in Figure 17.169 for the tube diameter:

```
ods graphics off;
symbol value=plus h=3.0 pct;
title  'Control Chart for New Copper Tubes' ;
proc shewhart data=newtubes;
   xrchart Diameter*batch /
      mu0    = 70
      sigma0 = 0.75;
run;
```

Batches 2 and 7 result in extreme out-of-control points on the mean chart, and batch 14 results in an extreme out-of-control point on the range chart. The vertical axes are scaled to accommodate these extreme out-of-control points, and this in turn forces the control limits to be compressed.

Figure 17.169 $\bar{X}$ and R Charts Without Clipping



You can request clipping by specifying the option CLIPFACTOR=*factor*, where *factor* is a value greater than one (useful values are typically in the range 1.5 to 2). Clipping is applied in two steps, as follows:

1. If a plotted statistic is greater than y_{max}, it is temporarily set to y_{max}, where

   $$y_{\text{max}} = \text{LCL} + (\text{UCL} - \text{LCL}) \times factor$$

   If a plotted statistic is less than y_{min}, it is temporarily set to y_{min}, where

   $$y_{\text{min}} = \text{UCL} - (\text{UCL} - \text{LCL}) \times factor$$

2. Axis scaling is applied to the clipped statistics. Then the y_{max} values are reset to the maximum value on the axis and the y_{min} values are reset to the minimum value on the axis.

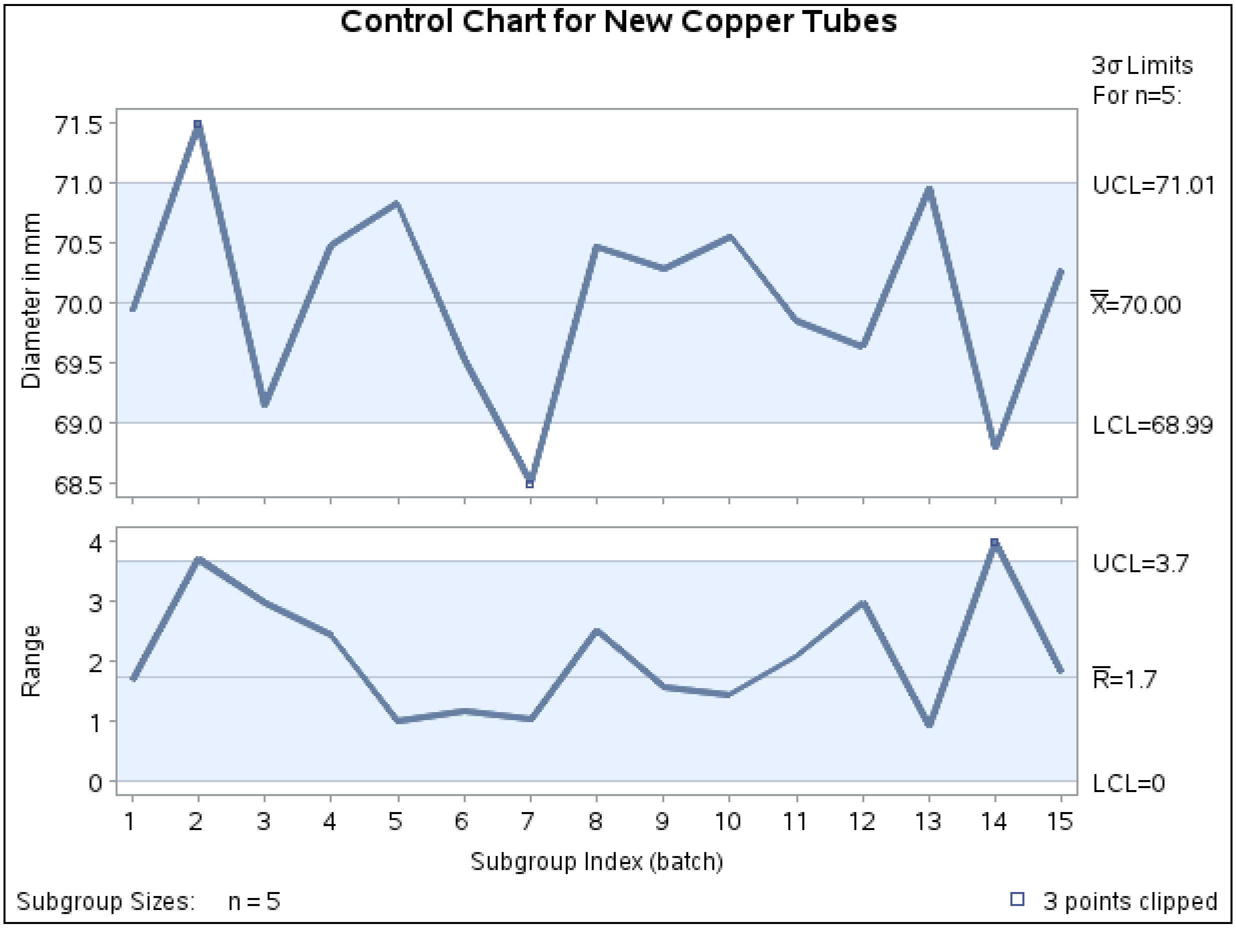
Notes:

- Clipping is applied only to the plotted statistics and not to the statistics tabulated or saved in an output data set.
- Because the *factor* must be greater than one, clipping does not affect whether a plotted statistic is inside or outside the control limits.
- Tests for special causes are applied to the plotted statistics before they are clipped, and clipping does not affect how the tests are flagged on the chart. In some situations, however, clipping can make the patterns associated with the tests less evident on the chart.
- When primary and secondary charts are displayed, the same clipping *factor* is applied to both charts.
- A special symbol is used for clipped points (the default symbol is a square), and a legend is added to the chart indicating the number of points that were clipped.

The following statements create $\bar{X}$ and *R* charts, shown in Figure 17.170, that use a clipping factor of 1.5:

```
title  'Control Chart for New Copper Tubes' ;
proc shewhart data=newtubes;
   xrchart Diameter*batch /
      mu0        = 70
      sigma0     = 0.75
      clipfactor = 1.5;
run;
```

Figure 17.170 $\bar{X}$ and R Charts with Clip Factor of 1.5

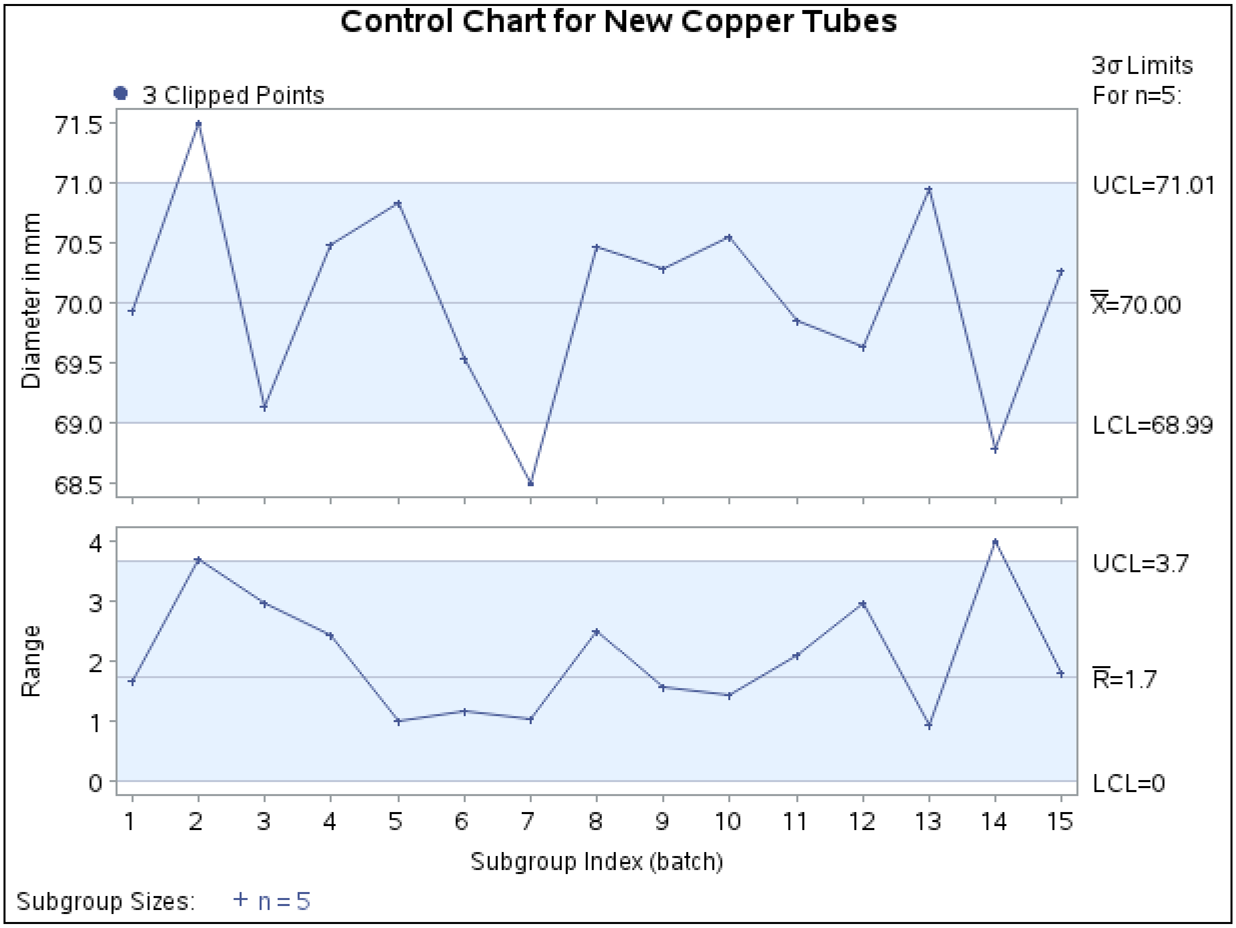


In Figure 17.170, the extreme out-of-control points are clipped making the points plotted within the control limits more readable. The clipped points are marked with a square, and a clipping legend is added at the lower right of the display.

Other clipping options are available, as illustrated by the following statements:

```
symbol value=plus;
title  'Control Chart for New Copper Tubes' ;
proc shewhart data=newtubes;
   xrchart Diameter*batch /
      mu0         = 70
      sigma0      = 0.75
      clipfactor  = 1.5
      clipsymbol  = dot
      cliplegpos  = top
      cliplegend  = '# Clipped Points'
      clipsubchar = '#';
run;
```

Figure 17.171 $\bar{X}$ and R Charts Using Clipping Options



Specifying CLIPSYMBOL=DOT marks the clipped points with a dot instead of the default square. Specifying CLIPLEGPOS=TOP positions the clipping legend at the top of the chart. The options CLIPLEGEND='# Clipped Points' and CLIPSUBCHAR='#' request the clipping legend *3 Clipped Points*. For more information about the clipping options, see the appropriate entries in "Dictionary of Options: SHEWHART Procedure" on page 1946.

Labeling Axes

NOTE: See *Labeling Axes on Shewhart Charts* in the SAS/QC Sample Library.

The SHEWHART procedure provides default labels for the horizontal and vertical axes of control charts. You can specify axis labels by assigning labels to variables, as discussed in the following sections.

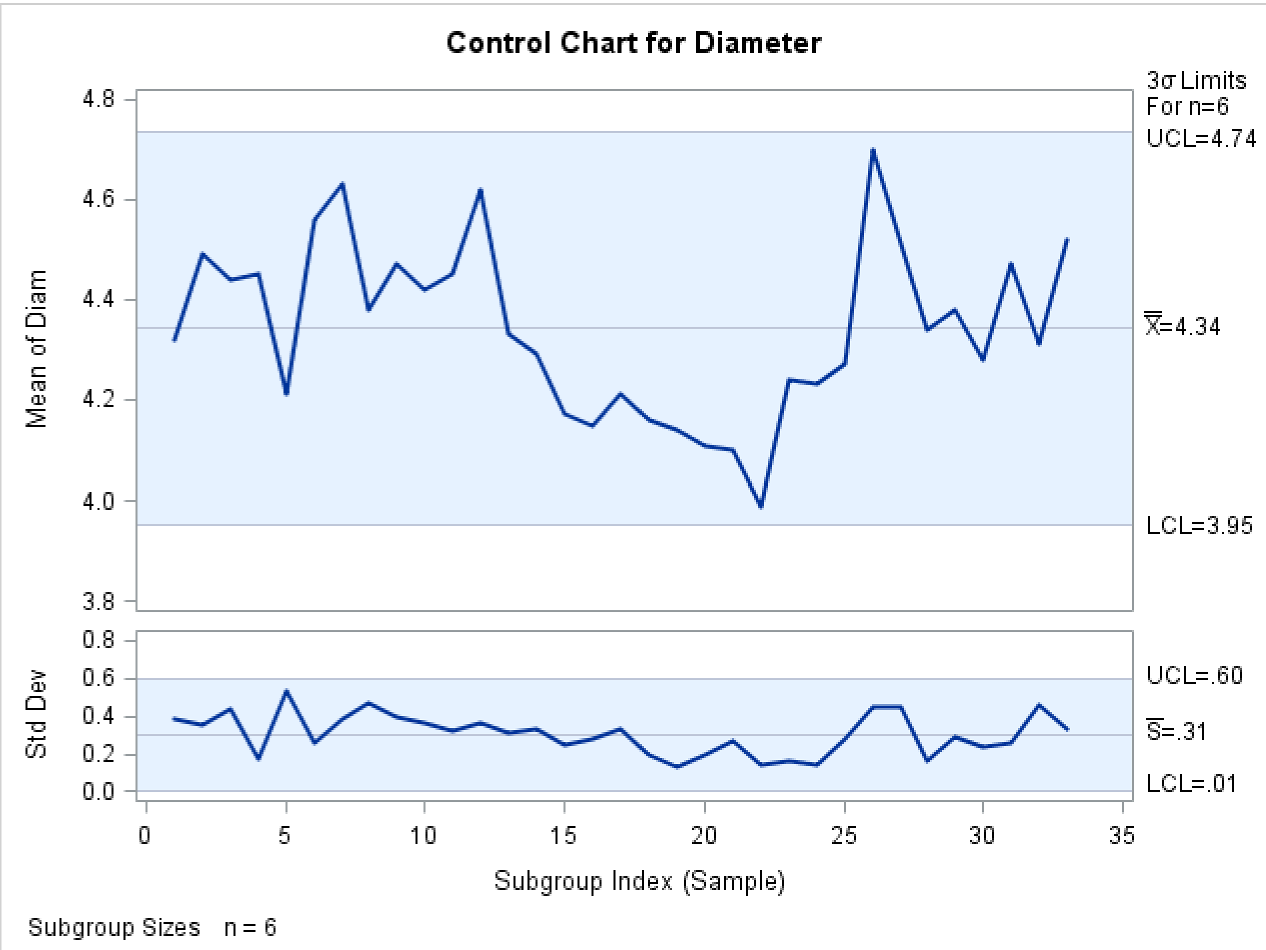
Default Labels

If a label is not associated with the *subgroup-variable*, the default horizontal axis label is "Subgroup Index (*subgroup-variable*)." The default vertical axis label for a primary chart identifies the chart type and the process variable. The default vertical axis label for a secondary chart identifies the chart type only.

For example, the following statements create $\bar{X}$ and s charts with default labels using the data set Parts given in "Displaying Stratified Process Data" on page 2023. The resulting charts are displayed in Figure 17.172.

```
ods graphics on;
title 'Control Chart for Diameter';
proc shewhart history=Parts;
   xschart Diam*Sample / odstitle = title;
run;
```

Figure 17.172 Control Charts with Default Labels



Labeling the Horizontal Axis

You can specify a label of up to 40 characters for the horizontal axis by assigning the label to the *subgroup variable* with a LABEL statement (refer to *SAS Statements: Reference* for a description of LABEL statements). If you use a LABEL statement after the PROC SHEWHART statement and before the RUN statement, the label is associated with the variable only for the duration of the PROC step.

For an example, see "Labeling the Vertical Axis" on page 2065, where Figure 17.173 redisplays the $\bar{X}$ and s charts in Figure 17.172 with specified horizontal and vertical axis labels.

Labeling the Vertical Axis

You can specify a label for the vertical axis of a primary chart by using a LABEL statement to assign the label to a particular variable in the input data set. The type of input data set, the chart statement, and the *process* specified in the chart statement determine which variable to use in the LABEL statement.

- If the input data set is a DATA= data set, assign the label to the process variable (*process*) specified in the chart statement.
- If the input data set is a HISTORY= data set, assign the label to the variable specified in the chart statement whose name begins with the prefix *process* and ends with the appropriate suffix given by the following list:

Chart Statement	**Suffix**
BOXCHART with CONTROLSTAT=MEAN	X
BOXCHART with CONTROLSTAT=MEDIAN	M
CCHART	U
IRCHART	none
MCHART	M
MRCHART	M
NPCHART	P
PCHART	P
RCHART	R
SCHART	S
UCHART	U
XCHART	X
XRCHART	X
XSCHART	X

  If the prefix *process* consists of 32 characters, shorten the prefix to its first 16 characters and last 15 characters before adding the suffix.

- If the input data set is a TABLE= data set, assign the label to the predefined variable given by the following table:

Chart Statement	Variable
BOXCHART with CONTROLSTAT=MEAN	_SUBX_
BOXCHART with CONTROLSTAT=MEDIAN	_SUBMED_
CCHART	_SUBC_
IRCHART	_SUBI_
MCHART	_SUBMED_
MRCHART	_SUBMED_
NPCHART	_SUBNP_
PCHART	_SUBP_
RCHART	_SUBR_
SCHART	_SUBS_
UCHART	_SUBU_
XCHART	_SUBX_
XRCHART	_SUBX_
XSCHART	_SUBX_

If the chart statement produces primary and secondary charts, as in the case of the XSCHART statement, you can break the label into two parts by including a split character in the label. The part before the split character labels the vertical axis of the primary chart, and the part after the split character labels the vertical axis of the secondary chart. To specify the split character, use the SPLIT= option in the chart statement.

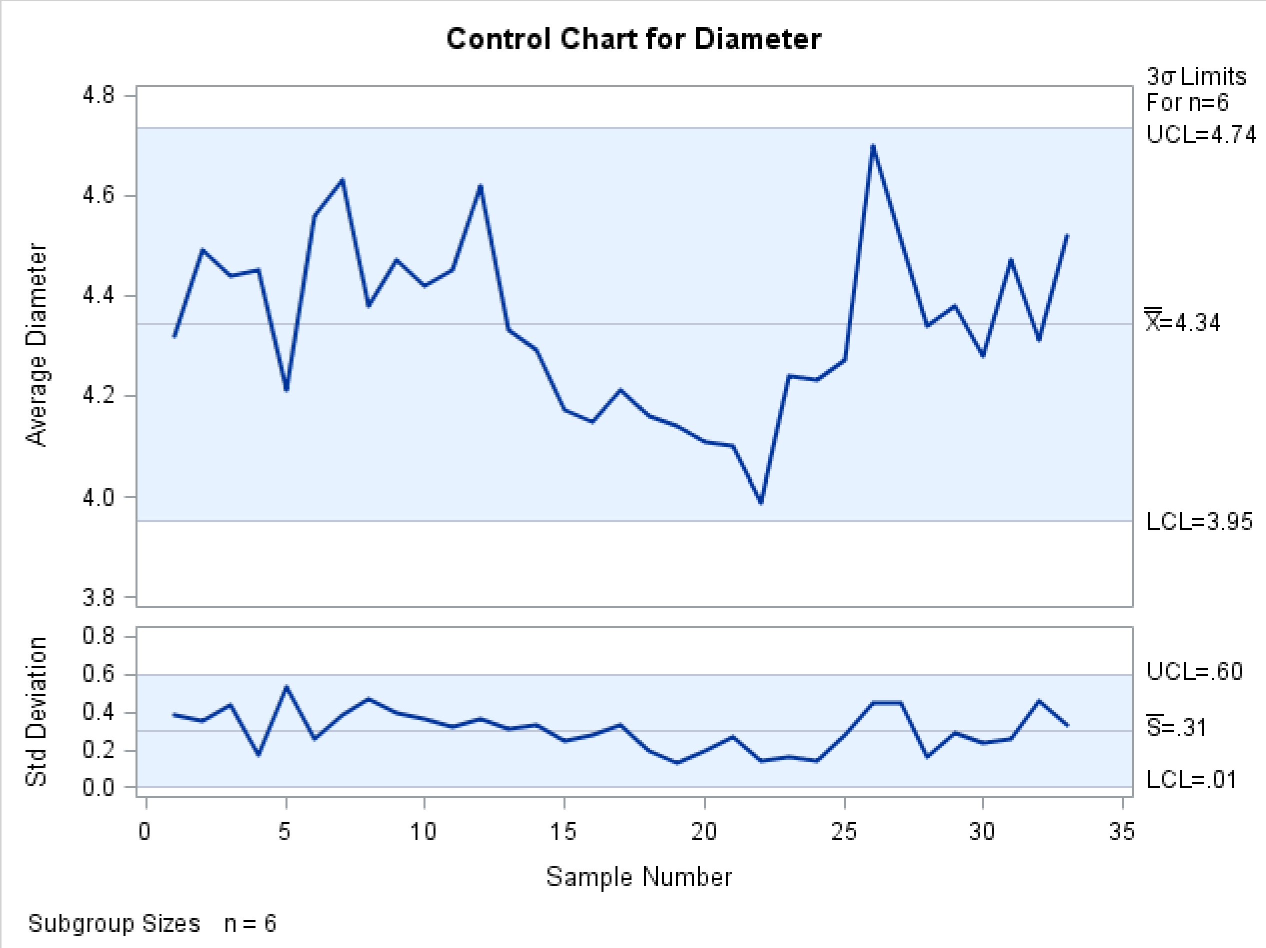
For example, the following statements redisplay the $\bar{X}$ and s charts in Figure 17.172 with specified labels for the horizontal and vertical axes:

```
title 'Control Chart for Diameter';
proc shewhart history=Parts;
   xschart Diam*Sample / split    = '/'
                         odstitle = title;
   label Sample = 'Sample Number'
         DiamX  = 'Average Diameter/Std Deviation';
run;
```

The charts are displayed in Figure 17.173. Because the input data set Parts is a HISTORY= data set, the vertical axes are labeled by assigning a label to the subgroup mean variable DiamX (that is, the *process* Diam with the suffix *X*).[13] Assigning a label to Diam would result in an error message since Diam is interpreted as a prefix rather than a SAS variable.

[13]If the *process* were Diameter rather than Diam, the label would be assigned to the variable DiameterX.

Figure 17.173 Control Charts with Axis Labels Specified



If the input data set were a DATA= data set rather than a HISTORY= data set, you would associate the label with the variable Diam. If the input data set were a TABLE= data set, you would associate the label with the variable _SUBX_.

For another illustration, see Example 17.17.

Selecting Subgroups for Computation and Display

This section describes methods for specifying which subgroups of observations in an input data set (DATA=, HISTORY=, or TABLE=) are to be used to compute control limits and which subgroups are to be displayed as points on the chart.

Using WHERE Statements

NOTE: See *Selecting Subgroups Using WHERE Statements* in the SAS/QC Sample Library.

The following statements create a data set named Bottles that records the number of cracked bottles encountered each day during two months (January and February) of a soft drink bottling operation:

```
data Bottles;
   informat Day date7.;
   format Day date7.;
   nBottles = 3000;
   input Day nCracks @@;
   datalines;
04JAN94  61  05JAN94  56  06JAN94  71  07JAN94  56
10JAN94  51  11JAN94  64  12JAN94  71  13JAN94  91
14JAN94  98  17JAN94  68  18JAN94  63  19JAN94  60
20JAN94  58  21JAN94  55  24JAN94  78  25JAN94  47
26JAN94  54  27JAN94  69  28JAN94  73  31JAN94  66
01FEB94  57  02FEB94  55  03FEB94  63  04FEB94  50
07FEB94  69  08FEB94  54  09FEB94  64  10FEB94  66
11FEB94  70  14FEB94  49  15FEB94  57  16FEB94  56
17FEB94  59  18FEB94  66  21FEB94  60  22FEB94  58
23FEB94  67  24FEB94  60  25FEB94  62  28FEB94  48
;
```

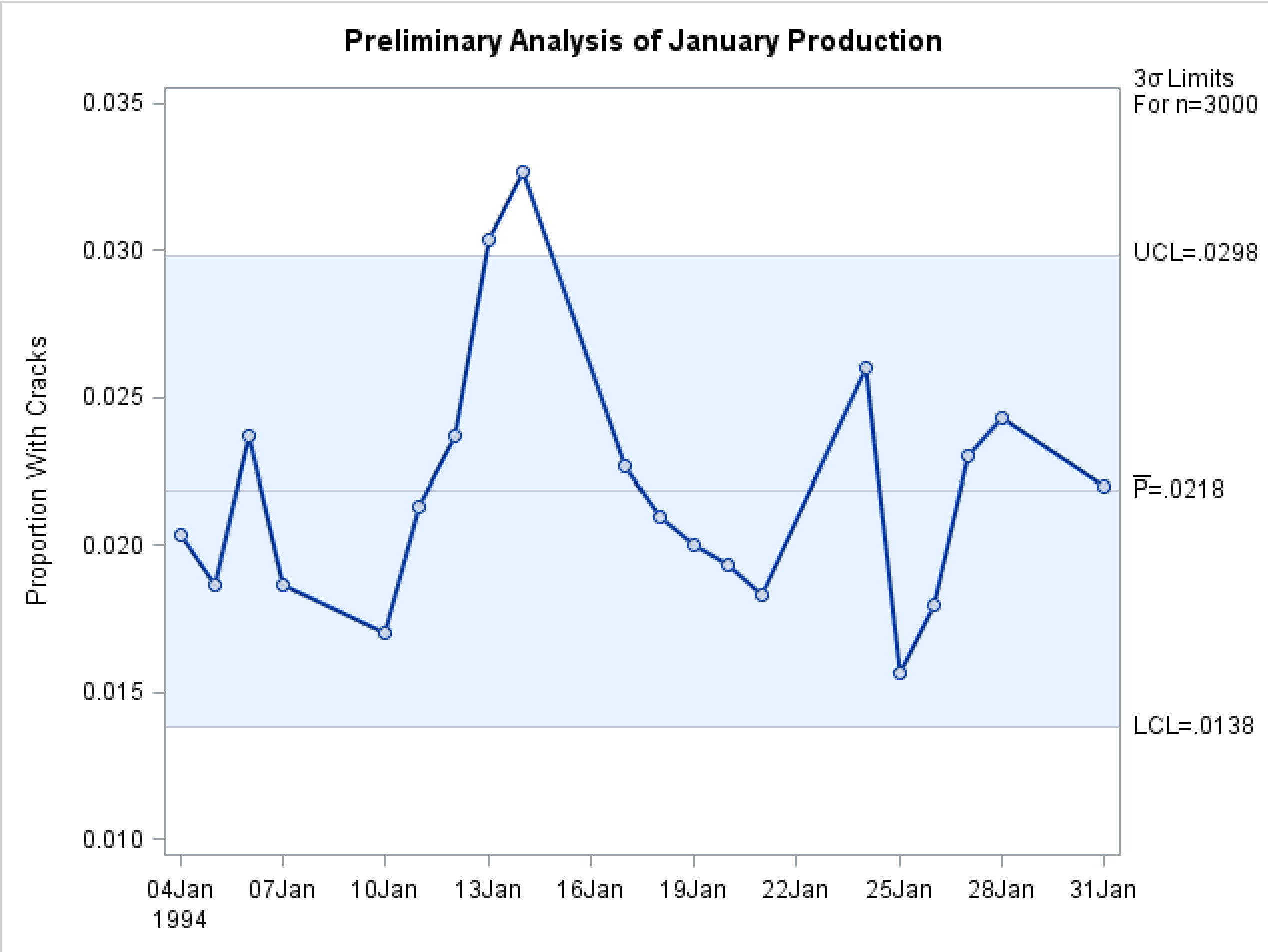
The variable nBottles contains the number of bottles sampled each day, and the variable nCracks contains the number of cracked bottles in each sample.

The following statements create a *p* chart for the number of cracked bottles based on the January production:

```
ods graphics on;
title 'Preliminary Analysis of January Production';
proc shewhart data=Bottles;
   where Day <= '31JAN94'D;
   pchart nCracks * Day / subgroupn = nBottles
                          nohlabel
                          nolegend
                          markers
                          odstitle  = title
                          outlimits = mylim;
   label nCracks = 'Proportion With Cracks';
run;
```

The chart is shown in Figure 17.174. The WHERE statement restricts the observations read from Bottles so that the control limits are estimated from the January data, and only the January data are displayed on the chart. For details concerning the WHERE statement, refer to *SAS Statements: Reference*.

Figure 17.174 Preliminary p Chart for January Data

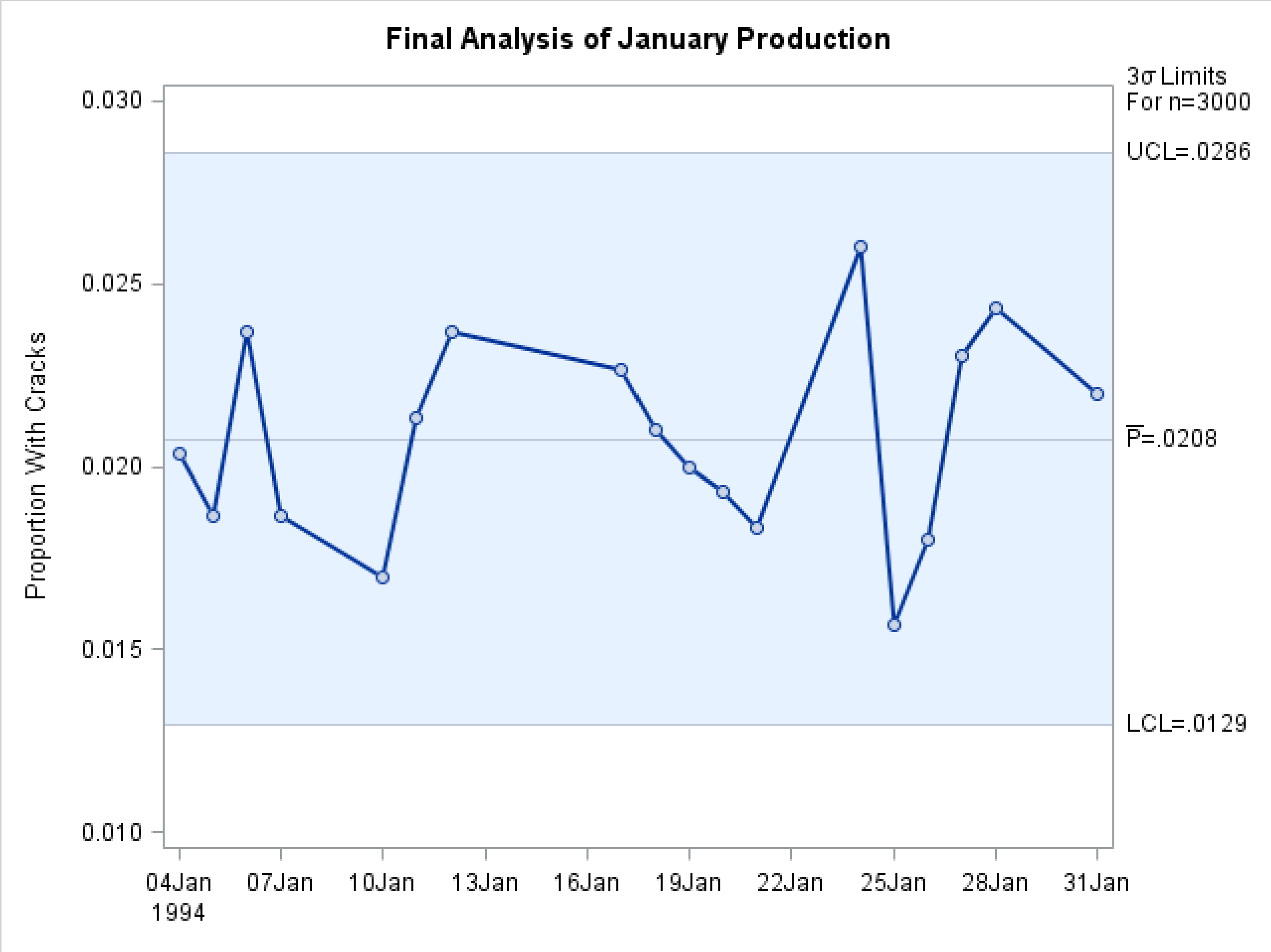


In Figure 17.174, a special cause of variation is signaled by the proportions for January 13 and January 14, which exceed the upper control limit. Since the cause, an improper machine setting, was corrected, it is appropriate to recompute the control limits by excluding the data for these two days. Again, this can be done with a WHERE statement, as follows:

```
title 'Final Analysis of January Production';
proc shewhart data=Bottles;
   where ( Day <= '31JAN94'D ) &
         ( Day ne '13JAN94'D ) &
         ( Day ne '14JAN94'D )   ;
   pchart nCracks * Day / subgroupn = nBottles
                          nohlabel
                          nolegend
                          markers
                          odstitle  = title
                          outlimits = Janlim;
   label nCracks = 'Proportion With Cracks';
run;
```

The chart is shown in Figure 17.175.

Figure 17.175 Final *p* Chart for January Data



The data set Janlim, which saves the control limits, is listed in Figure 17.176.

Figure 17.176 Listing of the LIMITS= Data Set Janlim

Final Analysis of January Production

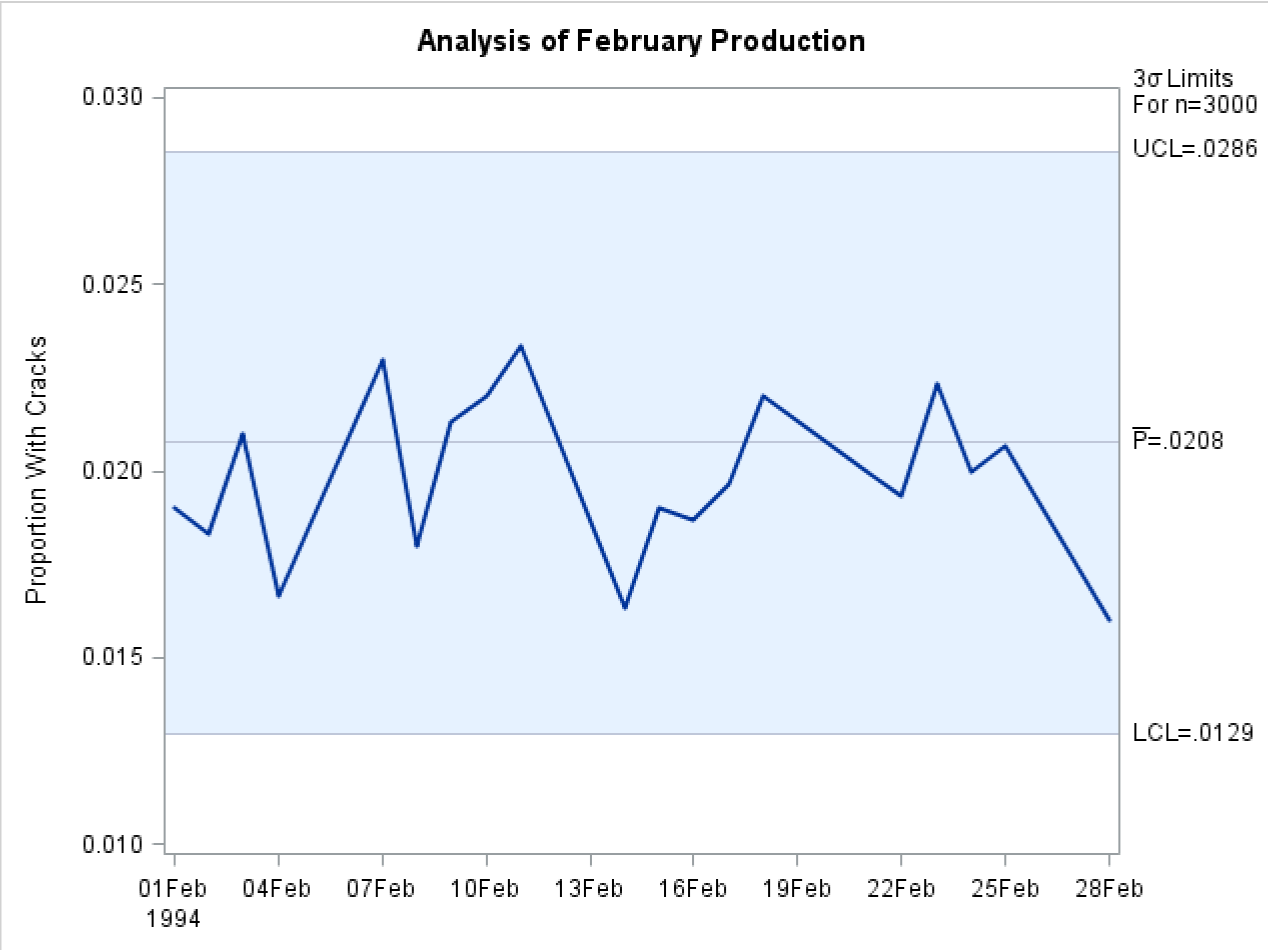
VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLP_	_P_	_UCLP_
nCracks	Day	ESTIMATE	3000	.002298782	3	0.012950	0.020759	0.028569

Now, the control limits based on the January data are to be applied to the February data. Again, this can be done with a WHERE statement, as follows:

```
title 'Analysis of February Production';
proc shewhart data=Bottles limits=Janlim;
   where Day > '31JAN94'D;
   pchart nCracks * Day / subgroupn = nBottles
                          odstitle  = title
                          nolegend
                          nohlabel;
   label nCracks = 'Proportion With Cracks';
run;
```

The chart is shown in Figure 17.177.

Figure 17.177 *p* Chart for February Data



Using Switch Variables

NOTE: See *Selecting Subgroups Using Switch Variables* in the SAS/QC Sample Library.

As an alternative to reading a LIMITS= data set and using a WHERE statement, you can provide two special switch variables named _COMP_ and _DISP_ in the input data set. The rules for using these variables are as follows:

- Switch variables must be character variables of length one. Valid values for these variables are 'Y' (or 'y') and 'N' (or 'n'). A blank value is treated as 'Y'.
- Subgroups for which _COMP_ is equal to 'Y' are included in computations of parameter estimates and control limits, and observations for which _COMP_ is equal to 'N' are excluded.
- Subgroups for which _DISP_ is equal to 'Y' are displayed on the chart, and subgroups for which _DISP_ is equal to 'N' are not displayed.

- If the chart statement creates a chart for variables, you can provide two additional switch variables named _COMP2_ and _DISP2_, which are defined similarly to _COMP_ and _DISP_. In this case, the variable _COMP_ specifies which subgroups are used to estimate the process mean μ, and the variable _COMP2_ specifies which subgroups are used to estimate the process standard deviation σ. The variable _DISP_ specifies which subgroups are displayed on the primary chart ($\bar{X}$ chart, median chart, or individual measurements chart), and the variable _DISP2_ specifies which subgroups are displayed on the secondary chart (R chart or s chart).
- The variables _COMP_ and _COMP2_ are not applicable when control limits or control limit parameters are read from a LIMITS= data set.
- The variables _DISP_ and _DISP2_ take precedence over the display controlled by the LIMITN= and ALLN options.
- If the input data set is a DATA= data set with multiple observations per subgroup, switch variable values must be constant within a subgroup.
- Switch variables are saved in OUTHISTORY= and OUTTABLE= data sets. Subgroups for which _DISP_ is equal to 'N' are not saved in an OUTTABLE= data set, and such subgroups are not displayed in tables created with the TABLE and related options.

The following statements illustrate how the switch variables _COMP_ and _DISP_ can be used with the bottle production data:

```
data Bottles;
   length _comp_ _disp_ $ 1;
   set Bottles;
   if      Day  = '13JAN94'D then _comp_ = 'n';
   else if Day  = '14JAN94'D then _comp_ = 'n';
   else if Day <= '31JAN94'D then _comp_ = 'y';
   else                           _comp_ = 'n';
   if      Day <= '31JAN94'D then _disp_ = 'n';
   else                           _disp_ = 'y';
run;

title 'Analysis of February Production';
proc shewhart data=Bottles;
   pchart nCracks * Day / subgroupn = nBottles
                          odstitle  = title
                          markers
                          nolegend
                          nohlabel;
   label nCracks = 'Proportion With Cracks';
run;
```

The chart is identical to the chart in Figure 17.177.

In general, switch variables are more versatile than WHERE statements in applications where subgroups are simultaneously selected for computation and display. Switch variables also provide a permanent record of which subgroups were selected. The WHERE statement does not alter the input data set; it simply restricts the observations that are read; consequently, the WHERE statement can be more efficient than switch variables for processing large data sets.

Tests for Special Causes: SHEWHART Procedure

This section provides details concerning standard and nonstandard tests for special causes that you can apply with the SHEWHART procedure.

Standard Tests for Special Causes

The SHEWHART procedure provides eight standard *tests for special causes*, also referred to as *rules for lack of control*, *supplementary rules*, *runs tests*, *runs rules*, *pattern tests*, and *Western Electric rules*. These tests improve the sensitivity of the Shewhart chart to small changes in the process. [14] You can also improve the sensitivity of the chart by increasing the rate of sampling, increasing the subgroup sample size, and using control limits that represent less than three standard errors of variation from the central line. However, increasing the sampling rate and sample size is often impractical, and tightening the control limits increases the chances of falsely signaling an out-of-control condition. By detecting particular nonrandom patterns in the points plotted on the chart, the tests can provide greater sensitivity and useful diagnostic information while incurring a reasonable probability of a false signal.

The patterns detected by the eight standard tests are defined in Table 17.93 and Table 17.94, and they are illustrated in Figure 17.178 and Figure 17.179. All eight tests were developed for use with fixed 3σ limits. The tests are indexed according to the numbering sequence used by Nelson (1984, 1985). You can request any combination of the eight tests by specifying the test *indexes* with the TESTS= option in the BOXCHART, CCHART, IRCHART, MCHART, MRCHART, NPCHART, PCHART, UCHART, XCHART, XRCHART, and XSCHART statements.

The following restrictions apply to the tests:

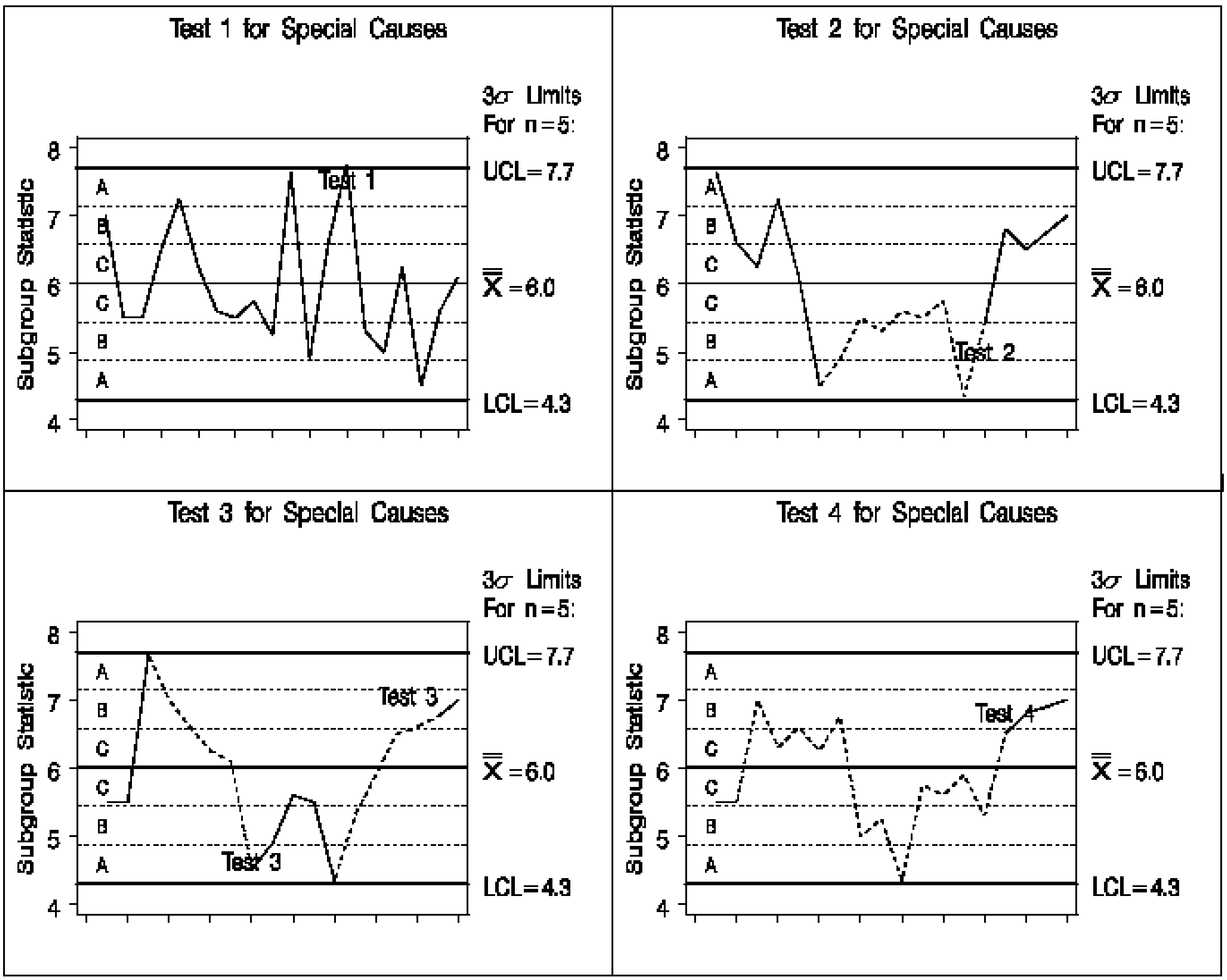
- Only Tests 1, 2, 3, and 4 are recommended for *c* charts, *np* charts, *p* charts, and *u* charts created with the CCHART, NPCHART, PCHART, and UCHART statements, respectively. In these four cases, Test 2 should not be used unless the process distribution is symmetric or nearly symmetric.
- By default, the TESTS= option is not applied with control limits that are not 3σ limits or that vary with subgroup sample size. You can use the NO3SIGMACHECK option to request tests for special causes when the SIGMAS= option specifies control limits other than 3σ limits. This is not recommended for standard control chart applications, since the standard tests for special causes are based on 3σ limits. You can apply tests for special causes when control limits vary with subgroup sample size by using the LIMITN= or TESTNMETHOD= options (see "Requesting Standard Tests" on page 2076 and "Applying Tests with Varying Subgroup Sample Sizes" on page 2079).

[14] Cumulative sum control charts and moving average control charts also detect small shifts more quickly than an ordinary Shewhart chart. See the sections "PROC CUSUM Statement" on page 541 and "PROC MACONTROL Statement" on page 780 for more information.

Table 17.93 Definitions of Tests 1 to 4

Test Index	Pattern Description
1	One point beyond Zone A (outside the control limits)
2	Nine points in a row in Zone C or beyond on one side of the central line (see Note 1 below)
3	Six points in a row steadily increasing or steadily decreasing (see Note 2 below)
4	Fourteen points in a row alternating up and down

Figure 17.178 Examples of Tests 1 to 4



Notes:

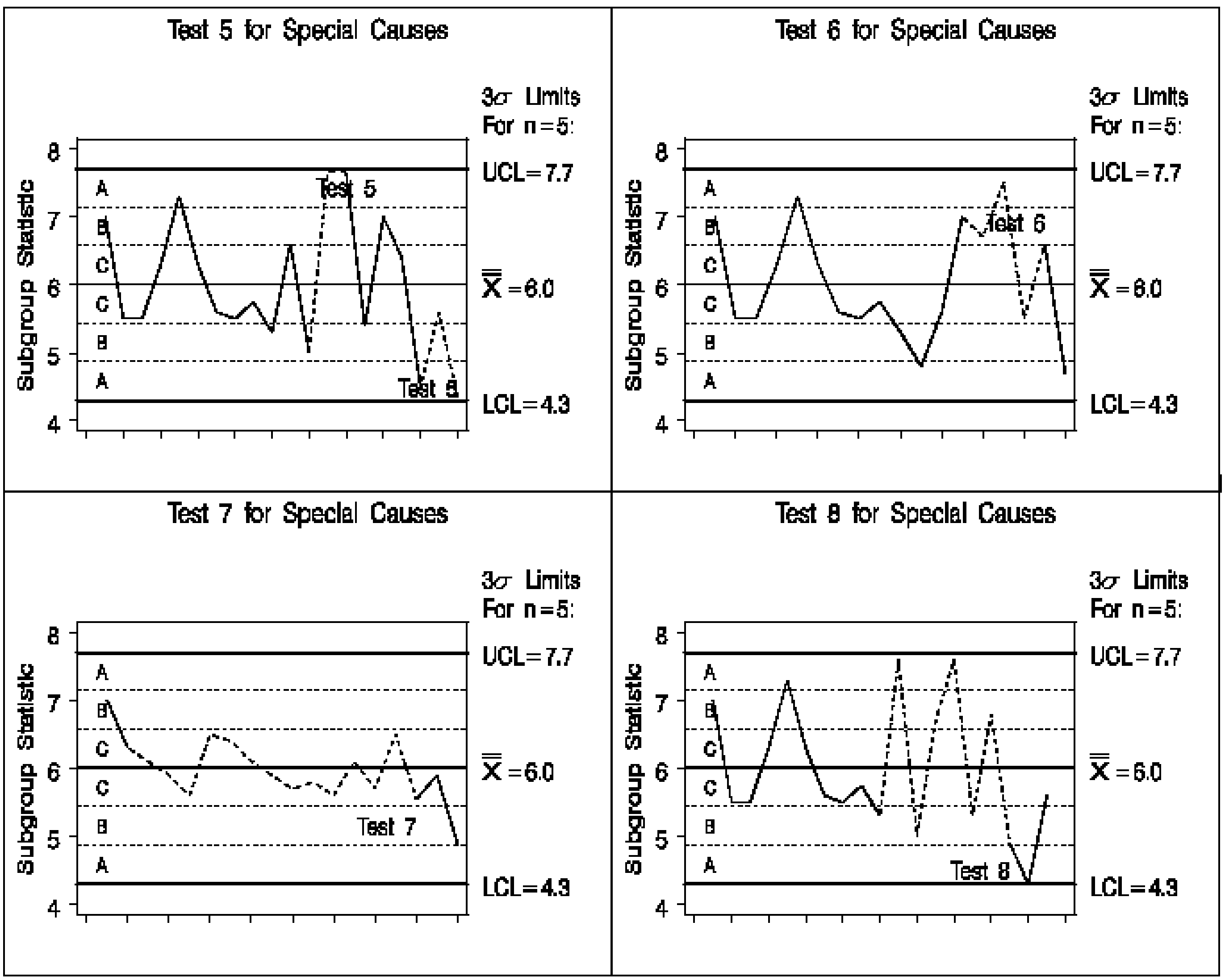
1. The number of points in Test 2 can be specified as 7, 8, 9, 11, 14, or 20 with the TEST2RUN= option.

2. The number of points in Test 3 can be specified as 6, 7, or 8 with the TEST3RUN= option.

Table 17.94 Definitions of Tests 5 to 8

Test Index	Pattern Description
5	Two out of three points in a row in Zone A or beyond
6	Four out of five points in a row in Zone B or beyond
7	Fifteen points in a row in Zone C on either or both sides of the central line
8	Eight points in a row on either or both sides of the central line with no points in Zone C

Figure 17.179 Examples of Tests 5 to 8



Requesting Standard Tests

NOTE: See *Requesting Tests for Special Causes* in the SAS/QC Sample Library.

The following example illustrates how to request the standard tests for special causes. The tests are applied to an $\bar{X}$ chart for assembly offset measurements whose subgroup means, ranges, and sample sizes are provided by the variables OffsetX, OffsetR, and OffsetN, respectively, in a data set named Assembly. [15]

```
data Assembly;
   length System $ 1 comment $ 16;
   label Sample = 'Sample Number';
   input System Sample OffsetX OffsetR OffsetN comment $16. ;
   datalines;
T   1  19.80  3.8  5
T   2  17.16  8.3  5
T   3  20.11  6.7  5
T   4  20.89  5.5  5
T   5  20.83  2.3  5
T   6  18.87  2.6  5
T   7  20.84  2.3  5
T   8  23.33  5.7  5  New Tool
T   9  19.21  3.5  5
T  10  20.48  3.2  5
T  11  22.05  4.7  5
T  12  20.02  6.7  5
T  13  17.58  2.0  5
T  14  19.11  5.7  5
T  15  20.03  4.1  5
R  16  20.56  3.7  5  Changed System
R  17  20.86  3.3  5
R  18  21.10  5.6  5  Reset Tool
R  19  19.05  2.7  5
R  20  21.76  2.8  5
R  21  21.76  6.4  5
R  22  20.54  4.8  5
R  23  20.04  8.2  5
R  24  19.94  8.8  5
R  25  20.70  5.1  5
Q  26  21.40 12.1  7  Bad Reading
Q  27  21.32  3.2  7
Q  28  20.03  5.2  7  New Gauge
Q  29  22.02  5.9  7
Q  30  21.32  4.3  7
;
```

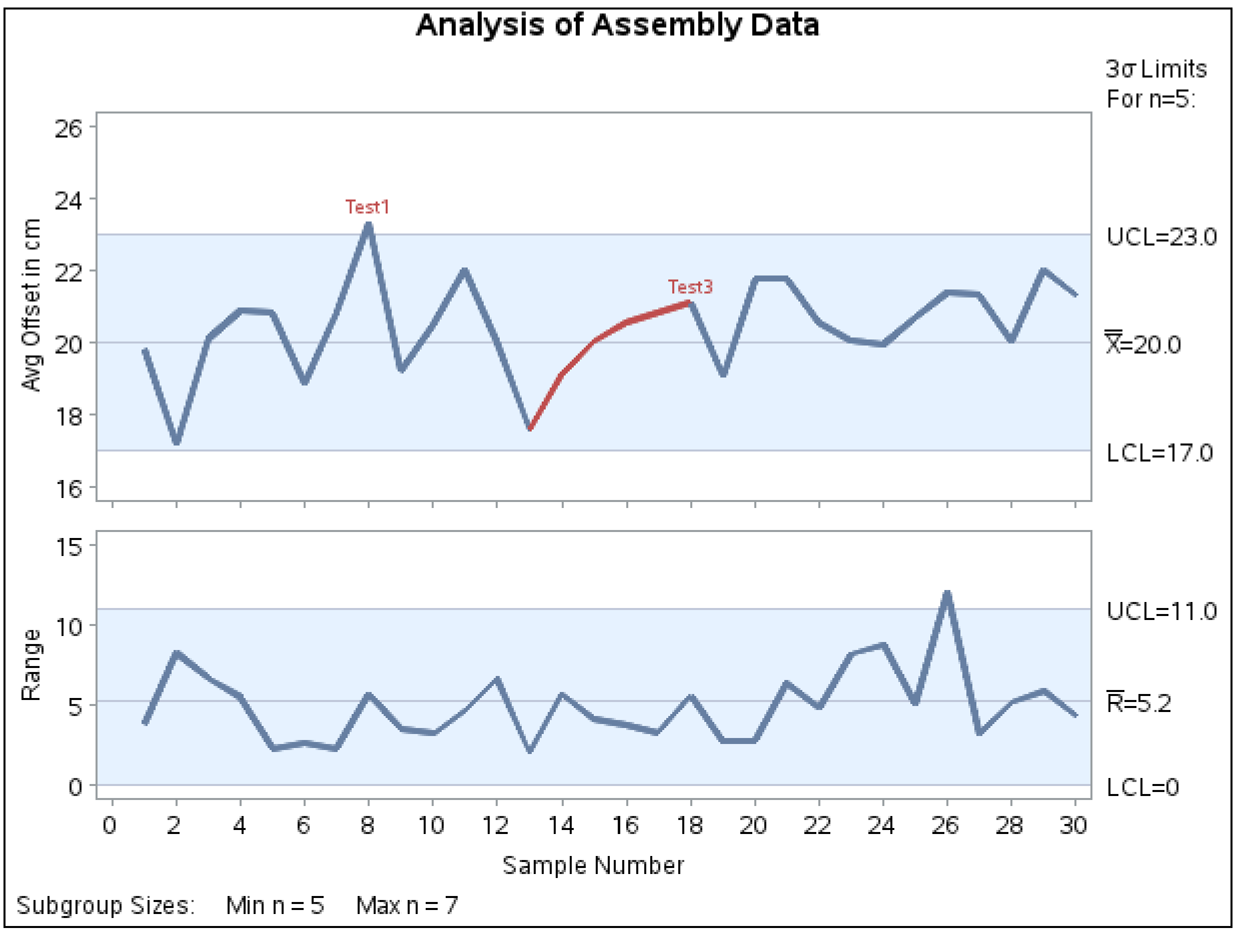
The following statements use the TESTS= option to request Tests 1 to 4. Note that the *process* Offset is specified in the XRCHART statement to indicate that the three summary variables OffsetX, OffsetR, and OffsetN are to be read from the HISTORY= data set Assembly.

[15] The data set Assembly is also used by subsequent examples in this section.

```
ods graphics off;
title 'Analysis of Assembly Data';
proc shewhart history=Assembly;
   xrchart Offset * Sample / mu0      = 20
                             sigma0   = 2.24
                             limitn   = 5
                             alln
                             tests    = 1 to 4
                             vaxis    = 16 to 26 by 2
                             split    = '/';
   label OffsetX = 'Avg Offset in cm/Range';
run;
```

The chart is displayed in Figure 17.180. Test 1 is positive at the 8th subgroup, and Test 3 is positive at the 18th subgroup.

Figure 17.180 Standard Tests Using the TESTS= Option



The control limits in Figure 17.180 are based on standard values for the process mean and standard deviation specified with the MU0= and SIGMA0= options, respectively. Although the subgroup sizes vary, fixed control limits are displayed corresponding to a nominal sample size of five, which is specified with the LIMITN= option. Since ALLN is specified, points are displayed for all subgroups, regardless of sample size.

NOTE: If the LIMITN= option were not specified, the control limits would vary with subgroup sample size, and by default the tests would not be applied. An alternative method for applying the tests with varying subgroup sample sizes is discussed in "Applying Tests with Varying Subgroup Sample Sizes" on page 2079.

Interpreting Standard Tests for Special Causes

Nelson (1984, 1985) makes the following comments concerning the interpretation of the tests:

- When a process is in statistical control, the chance of a false signal for each test is less than five in one thousand.
- Test 1 is positive if there is a shift in the process mean, if there is an increase in the process standard deviation, or if there is a "single aberration in the process such as a mistake in calculation, an error in measurement, bad raw material, a breakdown of equipment, and so on" (Nelson 1985).
- Test 2 signals a shift in the process mean. The use of nine points (rather than seven as in (Grant and Leavenworth 1988) for the pattern that defines Test 2 makes the chance of a false signal comparable to that of Test 1. (To control the number of points for the pattern in test 2, use the TEST2RUN= option in the chart statement.)
- Test 3 signals a drift in the process mean. Nelson (1985) states that causes can include "tool wear, depletion of chemical baths, deteriorating maintenance, improvement in skill, and so on."
- Test 4 signals "a systematic effect such as produced by two machines, spindles, operators or vendors used alternately" (Nelson 1985).
- Tests 1, 2, 3, and 4 should be applied routinely; the combined chance of a false signal from one or more of these tests is less than one in a hundred. Nelson (1985) describes these tests as "a good set that will react to many commonly occurring special causes."
- In the case of charts for variables, the first four tests should be augmented by Tests 5 and 6 when earlier warning is desired. The chance of a false signal increases to two in a hundred.
- Tests 7 and 8 indicate stratification (observations in a subgroup have multiple sources with different means). Test 7 is positive when the observations in the subgroup always have multiple sources. Test 8 is positive when the subgroups are taken from one source at a time.

Nelson (1985) also comments that "the probabilities quoted for getting false signals should not be considered to be very accurate" since the probabilities are based on assumptions of normality and independence that may not be satisfied. Consequently, he recommends that the tests "should be viewed as simply practical rules for action rather than tests having specific probabilities associated with them." Nelson cautions that "it is possible, though unlikely, for a process to be out of control yet not show any signals from these eight tests."

Modifying Standard Tests for Special Causes

Some textbooks and references present slightly different versions of Tests 2 and 3. You can use the following options to request these modifications:

- TEST2RUN=*run-length* specifies the length of the pattern for Test 2. The form of the test for each *run-length* is given in the following table. The default *run-length* is 9.

Run-length	Number of Points on One Side of Central Line
7	7 in a row
8	8 in a row
9	9 in a row
11	at least 10 out of 11 in a row
14	at least 12 out of 14 in a row
20	at least 16 out of 20 in a row

- TEST3RUN=*run-length* specifies the length of the pattern for Test 3. The *run-length* values allowed are 6, 7, and 8. The default *run-length* is 6.

The Western Electric Company (now AT&T) *Statistical Quality Control Handbook* (1956) and Montgomery (1996) discuss a test that is signaled by eight points in a row in Zone C or beyond (on one side of the central line). You can request this test by specifying TESTS=2 and TEST2RUN=8. The *Handbook* also discusses tests corresponding to Tests 1, 5, 6, 7, and 8.

Kume (1985) recommends a number of tests for special causes that can be regarded as modifications of Tests 2 and 3:

- seven points in a row on one side of the central line. Specify TESTS=2 and TEST2RUN=7.
- at least 10 out of 11 points in a row on one side of the central line. Specify TESTS=2 and TEST2RUN=11.
- at least 12 out of 14 points in a row on one side of the central line. Specify TESTS=2 and TEST2RUN=14.
- at least 16 out of 20 points in a row on one side of the central line. Specify TESTS=2 and TEST2RUN=20.
- seven points in a row steadily increasing or decreasing. Specify TESTS=3 and TEST3RUN=7.

Applying Tests with Varying Subgroup Sample Sizes

NOTE: See *Requesting Tests for Special Causes* in the SAS/QC Sample Library.

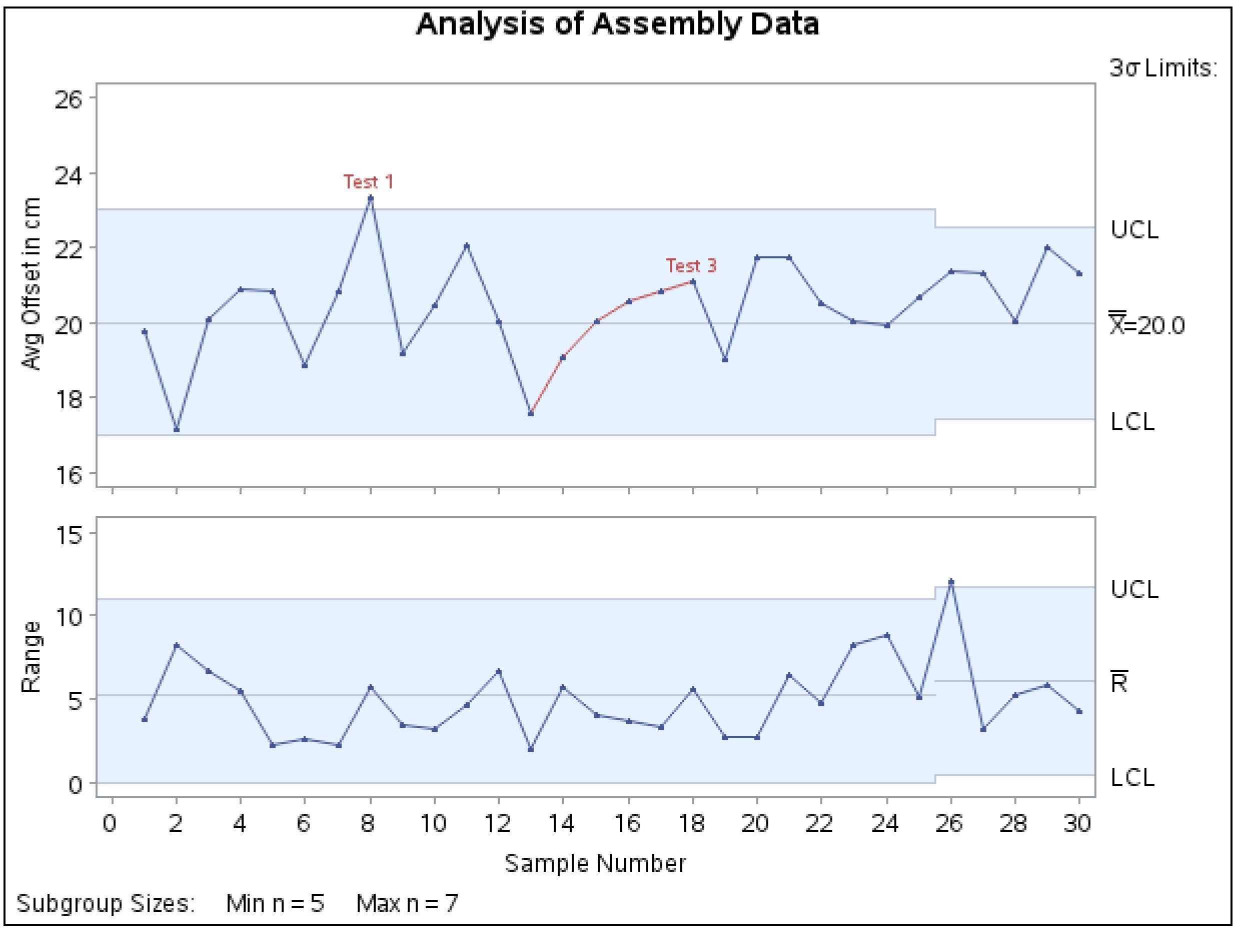
Nelson (1989, 1994) describes the use of standardization to apply the tests for special causes to data involving varying subgroup samples. This approach applies the tests to the standardized subgroup statistics, setting the control limits at ± 3 and the zone boundaries at ± 1 and ± 2. For instance, for an $\bar{X}$ chart with subgroup means $\bar{X}_i$ and varying subgroup sample sizes n_i, the tests are applied to the standardized values $z_i = (\bar{X}_i - \overline{\overline{X}})/(s/\sqrt{n_i})$, where $\overline{\overline{X}}$ estimates the process mean, and s estimates the process standard deviation. You can request this method with the TESTNMETHOD= option,[16] as illustrated by the following statements:

[16]If the TESTNMETHOD= option were omitted in this example, the tests would not be applied, and a warning message would be displayed in the SAS log.

```
ods graphics off;
title 'Analysis of Assembly Data';
symbol v=dot;
proc shewhart history=Assembly;
   xrchart Offset * Sample / mu0          = 20
                             sigma0       = 2.24
                             tests        = 1 to 4
                             testnmethod = standardize
                             testlabel    = space
                             vaxis        = 16 to 26 by 2
                             wtests       = 1
                             split        = '/';
   label OffsetX = 'Avg Offset in cm/Range';
run;
```

Here the tests are applied to $z_i = (\bar{X}_i - 20)/(2.24/\sqrt{n_i})$. The chart, shown in Figure 17.181, displays the results of the tests on a plot of the *unstandardized* means.

Figure 17.181 The TESTNMETHOD= Option for Varying Subgroup Sizes

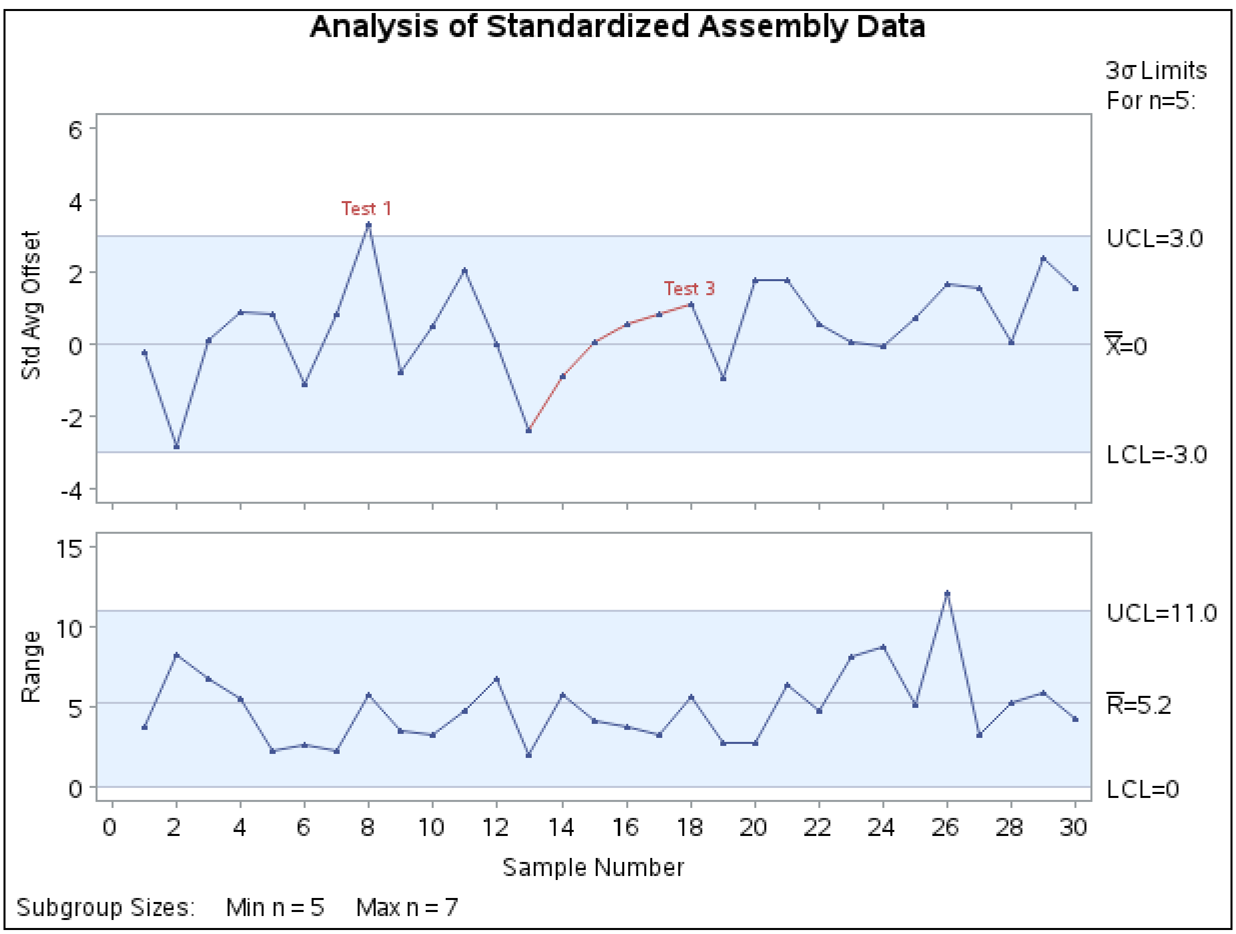


The following statements create an equivalent chart that plots the *standardized* means:

```
data Assembly;
   set Assembly;
   zx = ( OffsetX - 20 ) / ( 2.24 / sqrt( OffsetN ) );
run;

ods graphics off;
title 'Analysis of Standardized Assembly Data';
symbol v=dot;
proc shewhart
   history=Assembly (rename = (OffsetR=zr OffsetN=zn));
   xrchart z * Sample / mu0         = 0
                        sigma0      = 2.2361 /* sqrt 5 */
                        limitn      = 5
                        alln
                        tests       = 1 to 4
                        testlabel   = space
                        vaxis       = -4 to 6 by 2
                        wtests      = 1
                        split = '/';
   label zx = 'Std Avg Offset/Range';
run;
```

Here, the SIGMA0= value is the square root of the LIMITN= value. The chart is shown in Figure 17.182.

Figure 17.182 Tests with Standardized Means



NOTE: In situations where the standard deviation is estimated from the data and the subgroup sample sizes vary, you can use the SMETHOD= option to request various estimation methods, including the method of Burr (1969).

Labeling Signaled Points with a Variable

NOTE: See *Requesting Tests for Special Causes* in the SAS/QC Sample Library.

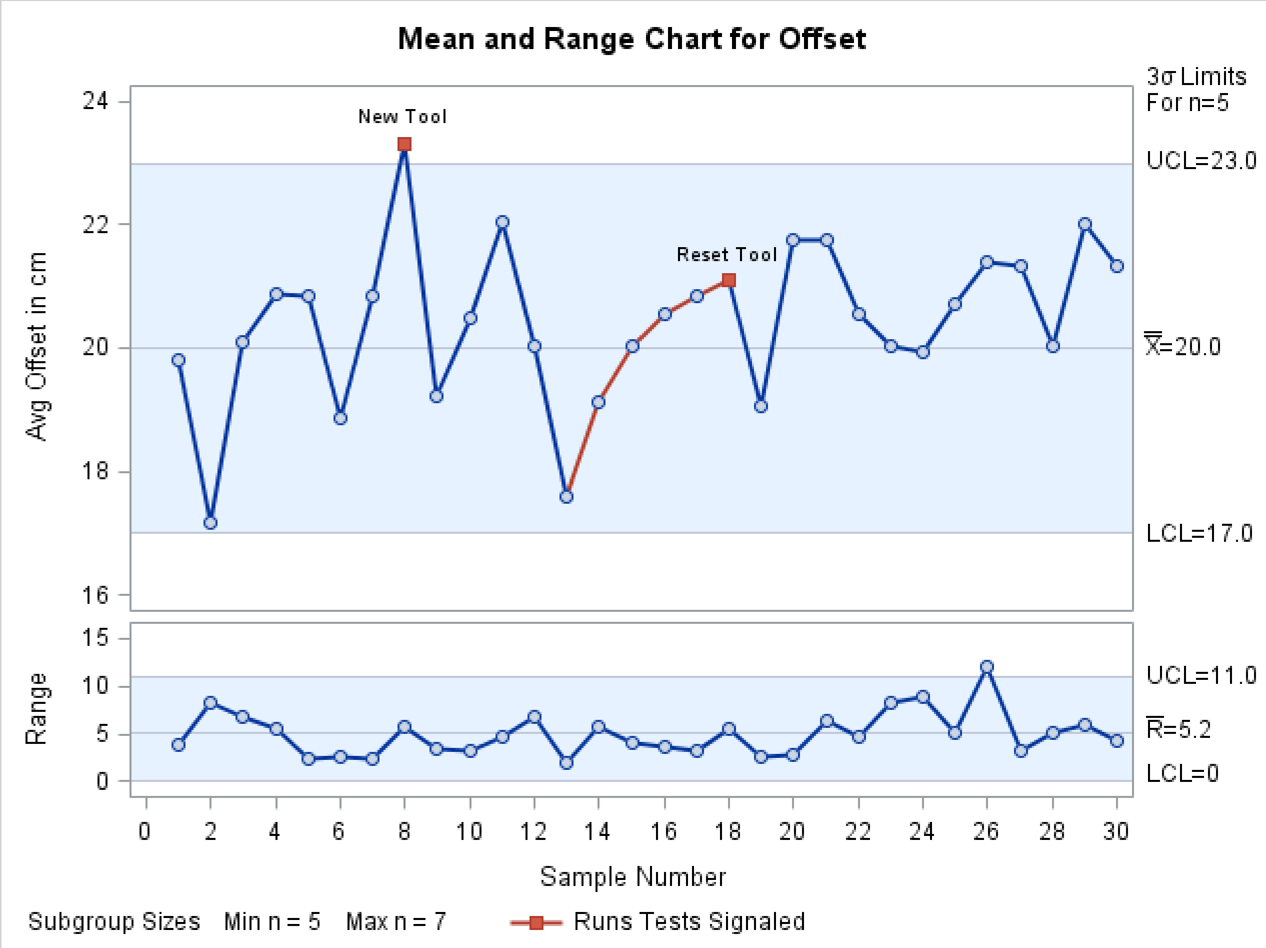
If a test is signaled at a particular point, the point is labeled by default with the *index* of the test, as illustrated in Figure 17.180.[17] You can use the TESTLABEL= option to specify a variable in the input data set whose *values* provide the labels, as illustrated by the following statements:

```
ods graphics on;
title 'Analysis of Assembly Data';
proc shewhart history=Assembly;
   xrchart Offset * Sample / mu0       = 20
                             sigma0    = 2.24
                             limitn    = 5
                             alln
                             tests     = 1 to 4
                             testlabel = ( comment )
                             vaxis     = 16 to 24 by 2
                             split      = '/'
                             markers;
   label OffsetX = 'Avg Offset in cm/Range';
run;
```

The labels are shown in Figure 17.183. It is often helpful to specify a variable with the TESTLABEL= option that provides operator comments or other information that can aid in the identification of special causes.

[17] If two or more tests are positive at a particular point, the default label identifies the *index* of the test that was specified first with the TESTS= option.

Figure 17.183 Labeling Points with a TESTLABEL= Variable



Applying Tests with Multiple Phases

NOTE: See *Requesting Tests for Special Causes* in the SAS/QC Sample Library.

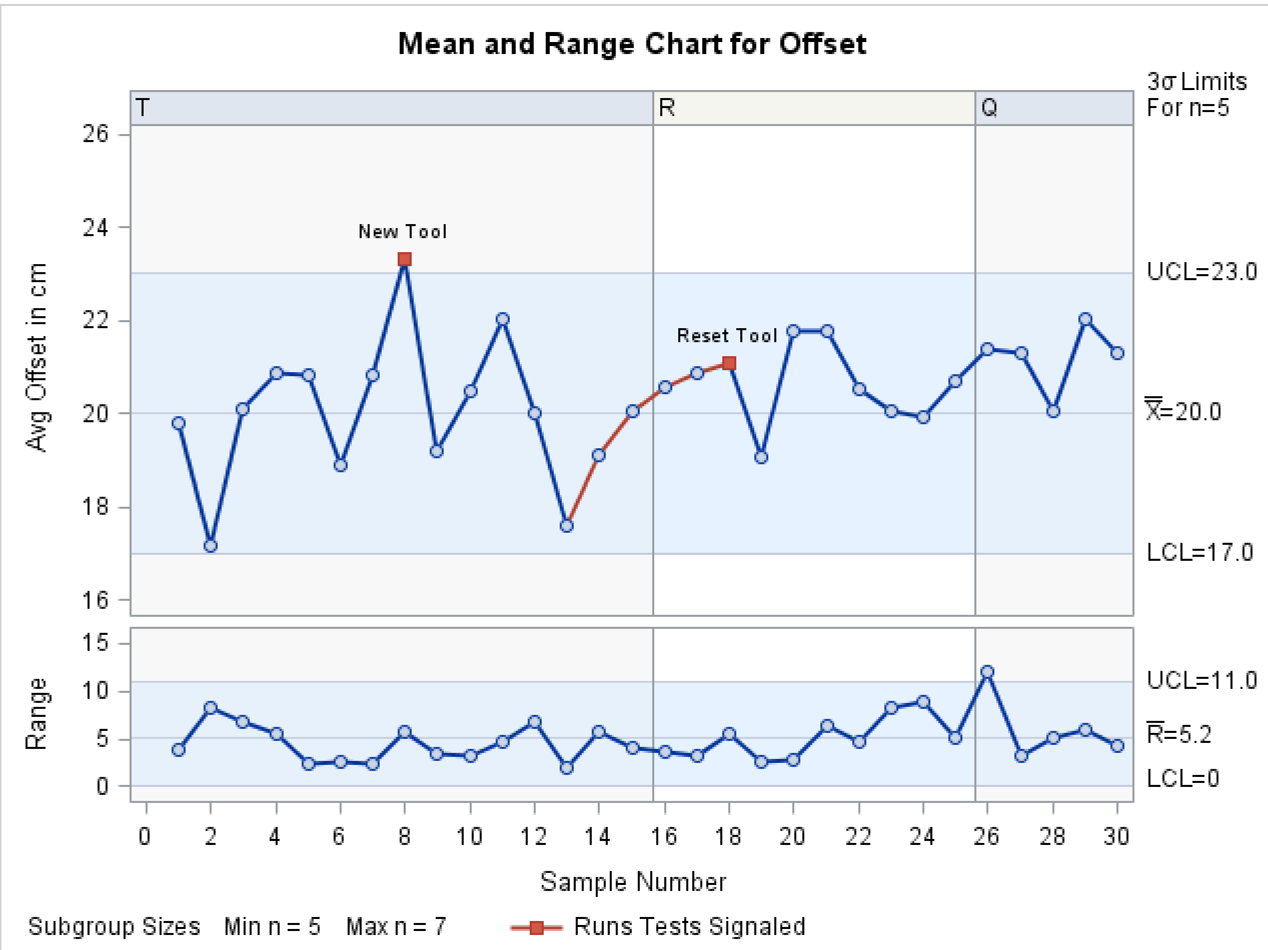
The data set Assembly includes a variable named System, which indicates the manufacturing system used to produce each assembly. As shown by the following statements, this variable can be temporarily renamed and read as the variable _PHASE_ to create a control chart that displays the *phases* (groups of consecutive subgroups) for which System is equal to ‘T’, ‘R’, and ‘Q’:

```
ods graphics on;
title 'Manufacturing Systems Used in Assembly';
proc shewhart
   history=Assembly (rename=(System=_phase_));
   xrchart Offset * Sample /
      mu0         = 20
      sigma0      = 2.24
      limitn      = 5
      alln
      tests       = 1 to 4
      testlabel   = ( comment )
```

```
         readphases  = ('T' 'R' 'Q')
         phaselegend
         phaseref
         vaxis       = 16 to 26 by 2
         split       = '/'
         markers;
      label OffsetX = 'Avg Offset in cm/Range';
   run;
```

The chart is shown in Figure 17.184.

Figure 17.184 Single Set of Limits with Multiple Phases



Note that a single set of fixed 3σ limits is displayed for all three phases because LIMITN=5 and ALLN are specified. Consequently, the tests requested with the TESTS= option are applied independently of the phases. In general, however, it is possible to display distinct sets of control limits for different phases, and in such situations, the tests are not applied independently of phases, as discussed in the next example.

Applying Tests with Multiple Sets of Control Limits

NOTE: See *Applying Tests with Multiple Control Limits* in the SAS/QC Sample Library.

This example is a continuation of the previous example, except that distinct control limits are displayed for each of the phases determined by the variable System. The control limit parameters (mean, standard deviation, and nominal sample size) for each phase (manufacturing system) are provided in the following data set:

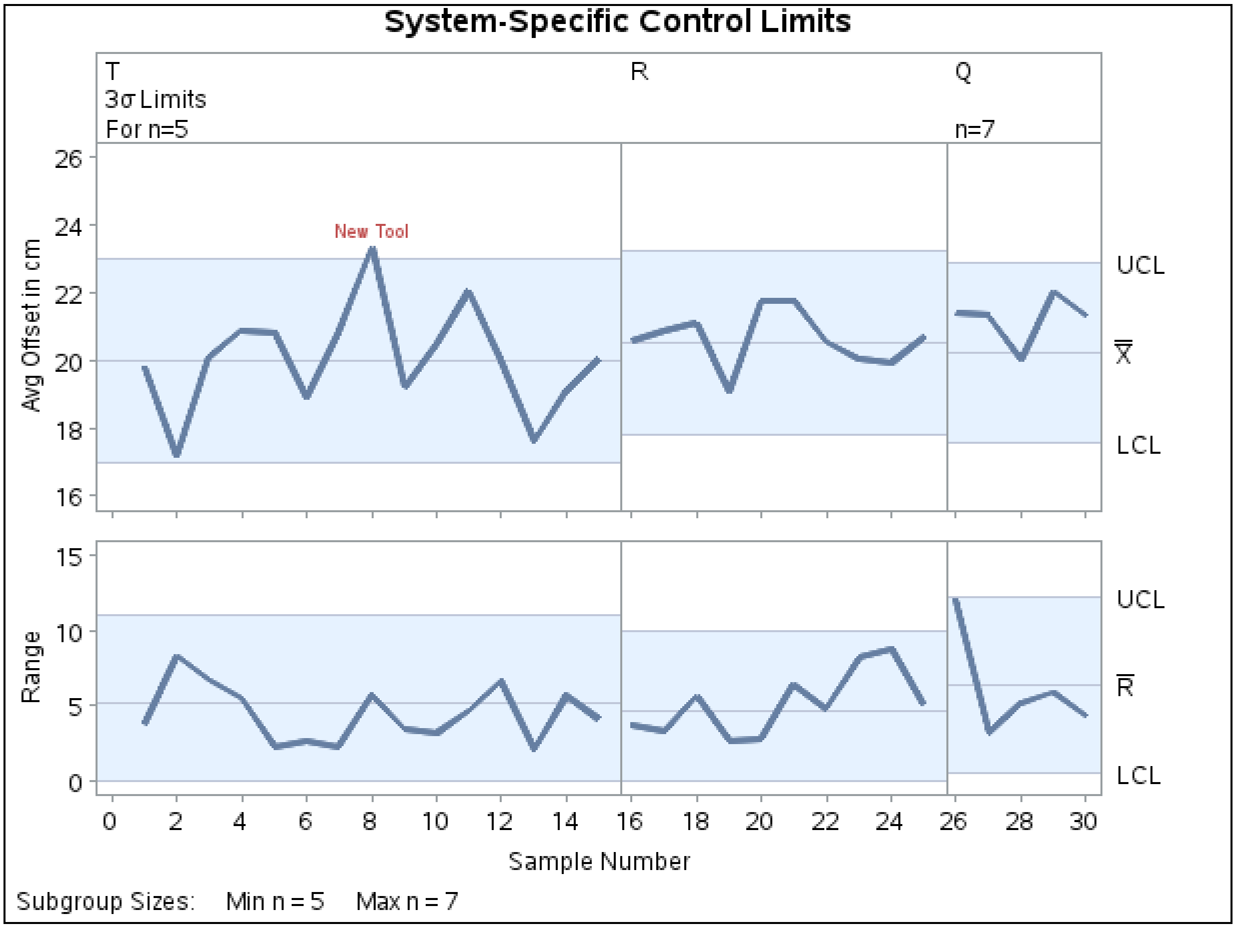
```
data Syslim;
   length _var_ $8 _subgrp_ $8 _type_ $8 _index_ $1;
   input _var_ _subgrp_ _index_ _type_ _mean_ _stddev_
         _limitn_ _sigmas_;
   datalines;
Offset Sample R standard 20.5 2.02 5 3
Offset Sample Q standard 20.2 2.35 7 3
Offset Sample T standard 20.0 2.24 5 3
;
```

The following statements read the control limit parameters from Syslim and use the READPHASES= and READINDEXES= options to display a distinct set of control limits for each phase:

```
ods graphics off;
title 'System-Specific Control Limits';
proc shewhart
   limits=Syslim
   history=Assembly (rename=(System=_phase_));
   xrchart Offset * Sample /
      tests       = 1 to 4
      testlabel   = ( comment )
      readindexes = ('T' 'R' 'Q')
      readphases  = ('T' 'R' 'Q')
      phaselegend
      phaseref
      phasebreak
      vaxis       = 16 to 26 by 2
      split       = '/' ;
   label OffsetX = 'Avg Offset in cm/Range';
run;
```

The chart is shown in Figure 17.185. The tests requested with the TESTS= option are applied strictly within the phases, since the control limits are not constant across the phases (as in Figure 17.184). In particular, note that the pattern labeled *Reset Tool* in Figure 17.184 is not detected in Figure 17.185.

Figure 17.185 Multiple Sets of Control Limits



In most applications involving multiple control limits, a known change or improvement has occurred at the beginning of each phase; consequently, it is appropriate to restart the tests at the beginning of each phase rather than search for patterns that span the boundaries of consecutive phases. In these situations, the PHASEBREAK option is useful for suppressing the connection of points from one phase to the next. Note that it is not necessary to specify the TESTNMETHOD= option here because the subgroup sample sizes are constant within each phase.

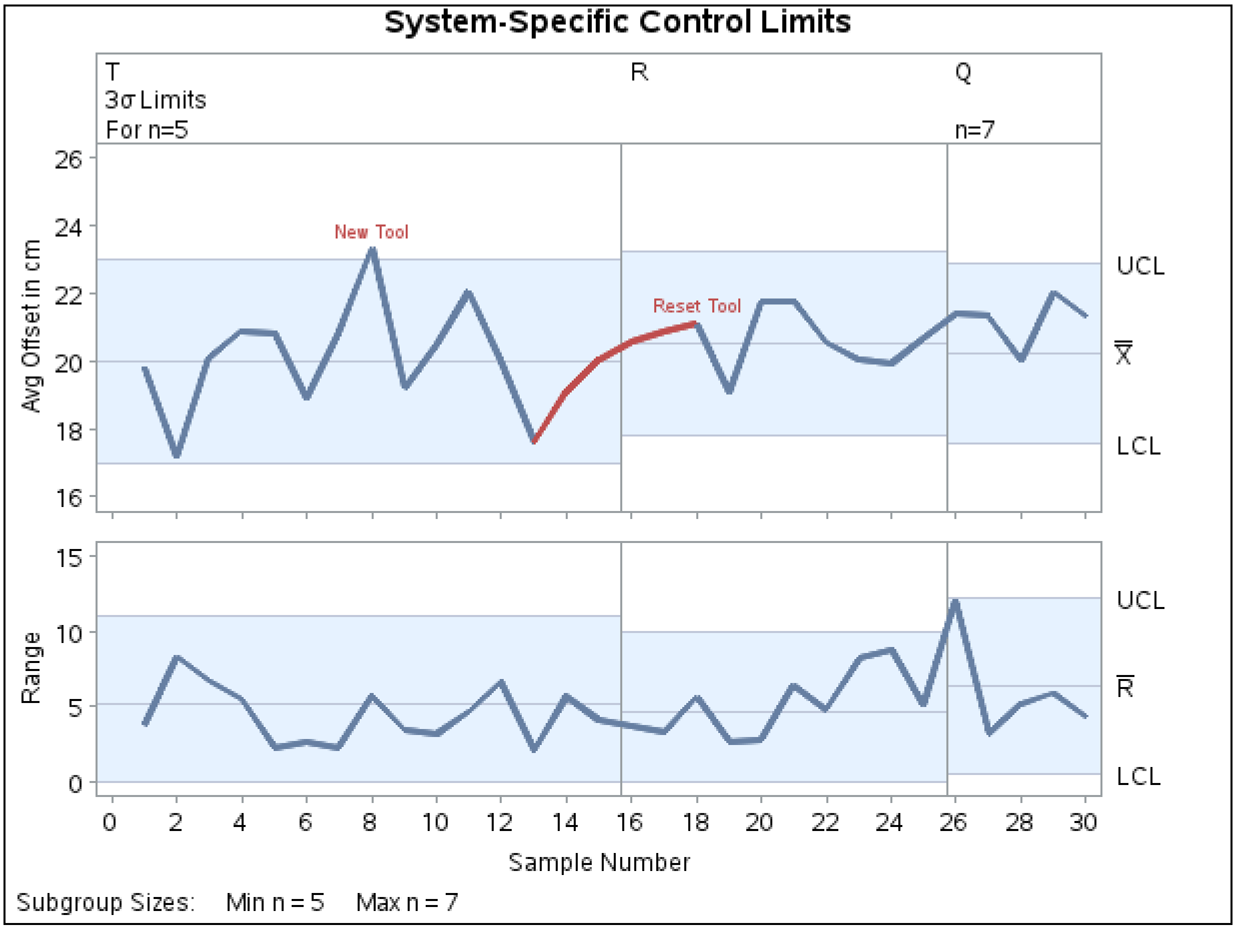
There may be applications in which it is appropriate to apply the tests across phase boundaries. You can use the TESTACROSS option to request this behavior.

```
ods graphics off;
title 'System-Specific Control Limits';
proc shewhart
   limits=Syslim
   history=Assembly (rename=(System=_phase_));
   xrchart Offset * Sample /
      tests       = 1 to 4
      testlabel   = ( comment )
      testnmethod = standardize
      testacross
      readindexes = ('T' 'R' 'Q')
```

```
         readphases  = ('T' 'R' 'Q')
         phaselegend
         phaseref
         vaxis       = 16 to 26 by 2
         split       = '/';
      label OffsetX = 'Avg Offset in cm/Range';
   run;
```

The chart created with the TESTACROSS option is displayed in Figure 17.186.

Figure 17.186 Multiple Sets of Control Limits with the TESTACROSS Option



Here, it is necessary to specify TESTNMETHOD=STANDARDIZE in conjunction with the TESTACROSS option, since the subgroup sample sizes are not constant across phases.

Although Test 3 is now signaled at sample 18, this result should be interpreted with care since the test is applied to standardized average offsets, and the averages for samples 13, 14, and 15 are standardized differently than the averages for samples 16, 17, and 18. If, for instance, the value of _MEAN_ for phase 'R' in Syslim were 21.0 rather than 20.5, the standardized mean for sample 16 would be less than the standardized mean for sample 15, and Test 3 would not be signaled at sample 18.

In summary, when working with multiple control limits, you should

- use the TESTACROSS option only if the process is operating in a continuous manner across phases

- use TESTNMETHOD=STANDARDIZE only if it is clearly understood by users that tests signaled on the chart are based on *standardized* statistics rather than the plotted statistics

Enhancing the Display of Signaled Tests

There are various options for labeling points at which a test is signaled.

- The default label for Test *i* is *Testi*. See Figure 17.180 for an example.
- Specify TESTLABEL=SPACE to request labels of the form *Test i*. See Figure 17.181 for an example.
- Specify TESTLABEL*i*=*'label'* to provide a specific *label* for the *i*th test. See Figure 17.190 for an example.
- Specify TESTLABEL=(*variable*) to request labels provided by a *variable* in the input data set. See Figure 17.183 for an example.

If two or more tests are signaled at a particular point, the label displayed corresponds to the test that was specified first in the TESTS= list.

If you are producing traditional graphics, you can specify the color of the label and the connecting line segments for the pattern with the CTESTS= option. You can specify the line type for the line segments with the LTESTS= option. If you are creating line printer charts, you can specify the plot character for the line segments with the TESTCHAR= option.

You can specify the ZONES option to display the zone lines on the chart, and you can specify ZONELABELS to label the zone lines. If you are creating traditional graphics, you can specify the color of the lines with the CZONES= option, and if you are producing line printer charts, you can specify the plot character for the lines with the ZONECHAR= option.

Nonstandard Tests for Special Causes

This section describes options and programming techniques for requesting various nonstandard tests for special causes.

Applying Tests to Range and Standard Deviation Charts

NOTE: See *Applying Tests for Special Causes-R charts* in the SAS/QC Sample Library.

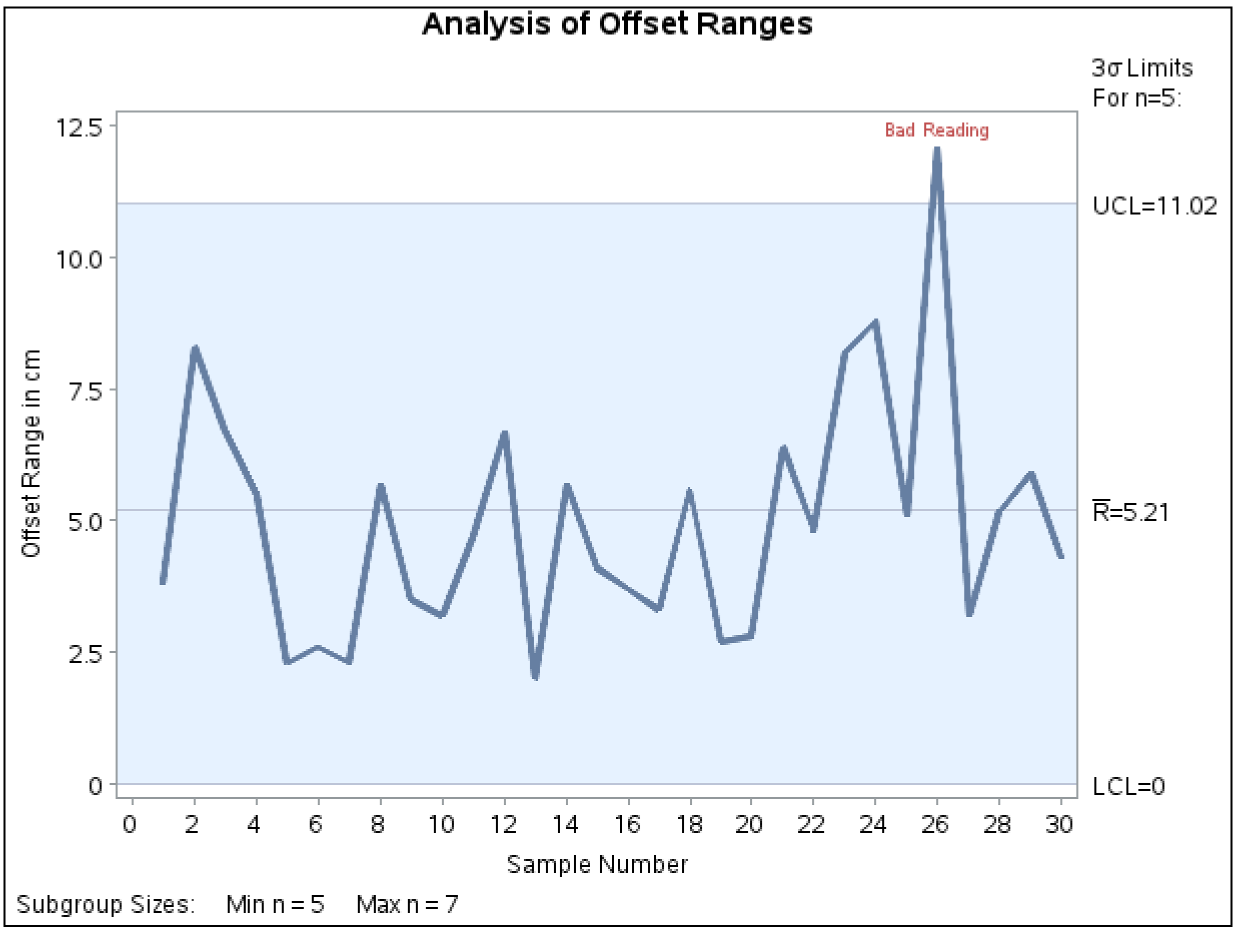
If you are using the MRCHART, RCHART, SCHART, XRCHART, or XSCHART statement, you can use the TESTS2= option to request tests for special causes with an *R* chart or *s* chart. The syntax and test definitions for the TESTS2= option are identical to those for the TESTS= option, and you can use the ZONES2 and ZONE2LABELS options to display the zones on the secondary chart.

The following statements request Test 1 for a range chart of the data in Assembly (see “Requesting Standard Tests” on page 2076):

```
ods graphics off;
title 'Analysis of Offset Ranges';
proc shewhart history=Assembly;
   rchart Offset * Sample / sigma0    = 2.24
                            limitn    = 5
                            alln
                            tests2    = 1
                            testlabel = (comment) ;
   label OffsetR = 'Offset Range in cm';
run;
```

The *R* chart is shown in Figure 17.187.

Figure 17.187 Range Chart with Test 1



Caution: Except for requesting Test 1, use of the TESTS2= option is not recommended for general process control work. At the time of this writing, there is insufficient published research supporting the application of the other tests to *R* charts and *s* charts. There are no established guidelines for interpreting the other tests, nor are there assessments of their false signal probabilities or average run length characteristics. The TESTS2= option is intended primarily as a research tool.

Applying Tests Based on Generalized Patterns

In addition to *indices* for standard tests, you can specify up to eight *T-patterns* or *M-patterns* with the TESTS= option:

- Specifying a T-pattern requests a search for k out of m points in a row in the interval (a, b). Tests based on T-patterns are generalizations of Tests 1, 2, 5, and 6. The average run length properties of tests based on T-patterns have been analyzed by Champ and Woodall (1987). Also refer to Chapter 8 of Wetherill and Brown (1991).
- Specifying an M-pattern requests a search for k points in a row increasing or decreasing. Tests based on M-patterns are generalizations of Test 3.

The general syntax for a T-pattern is of the form

T(K=*k* M=*m* LOWER=*a* UPPER=*b* SCHEME=*scheme* CODE=*character* LABEL='*label*' LEGEND='*legend*')

The options for a T-pattern are summarized in the following table:

Table 17.95 Options for T-Patterns

Option	Description
K=*k*	number of points ($k \leq m$)
M=*m*	number of consecutive points
LOWER=*value*	lower limit of interval (a, b)
UPPER=*value*	upper limit of interval (a, b)
SCHEME=ONESIDED	one-sided scheme using (a, b)
SCHEME=TWOSIDED	two-sided scheme using $(a, b) \cup (-b, -a)$
CODE=*character*	identifier for test (A-H)
LABEL=*'label'*	label for points that are signaled
LEGEND=*'legend'*	legend used with the TABLELEGEND option

The following rules apply to the T-pattern options:

1. You must specify SCHEME=*scheme*. Specifying SCHEME=ONESIDED requests a one-sided test that searches for k out of m points in a row in the interval (a, b). Specifying SCHEME=TWOSIDED with positive values for a and b (where $a < b$) requests a two-sided test that searches for k out of m points in a row in the interval (a, b) or k out of m points in a row in the interval $(-b, -a)$.
2. The values a and b must be specified in standardized units, and they must both have the same sign. For instance, specifying LOWER=2 and UPPER=3 with SCHEME=TWOSIDED corresponds to Zone A in Figure 17.178.

3. Specifying a missing value for the LOWER= option and a negative value for *b* requests a search in the interval $(-\infty, b)$. Specifying a positive value for *a* and a missing value for the UPPER= option requests a search in the interval (a, ∞).

4. You must specify a CODE= *character*, which can be any of the letters A through H. The character identifies the pattern in tables requested with the TABLETESTS and TABLEALL options and in the value of the variable _TESTS_ in the OUTTABLE= data set. The character is analogous to the indices 1 through 8 that are used to identify the standard tests. If you request multiple T-patterns, you must specify a unique character for each pattern.

5. You can specify a *label* with the LABEL= option. The label must be enclosed in quotes and can be up to 16 characters long. The label is used to label points on the chart at which the test defined by the T-pattern is signaled. The LABEL= option is similar to the TESTLABEL*n*= options used with the standard tests.

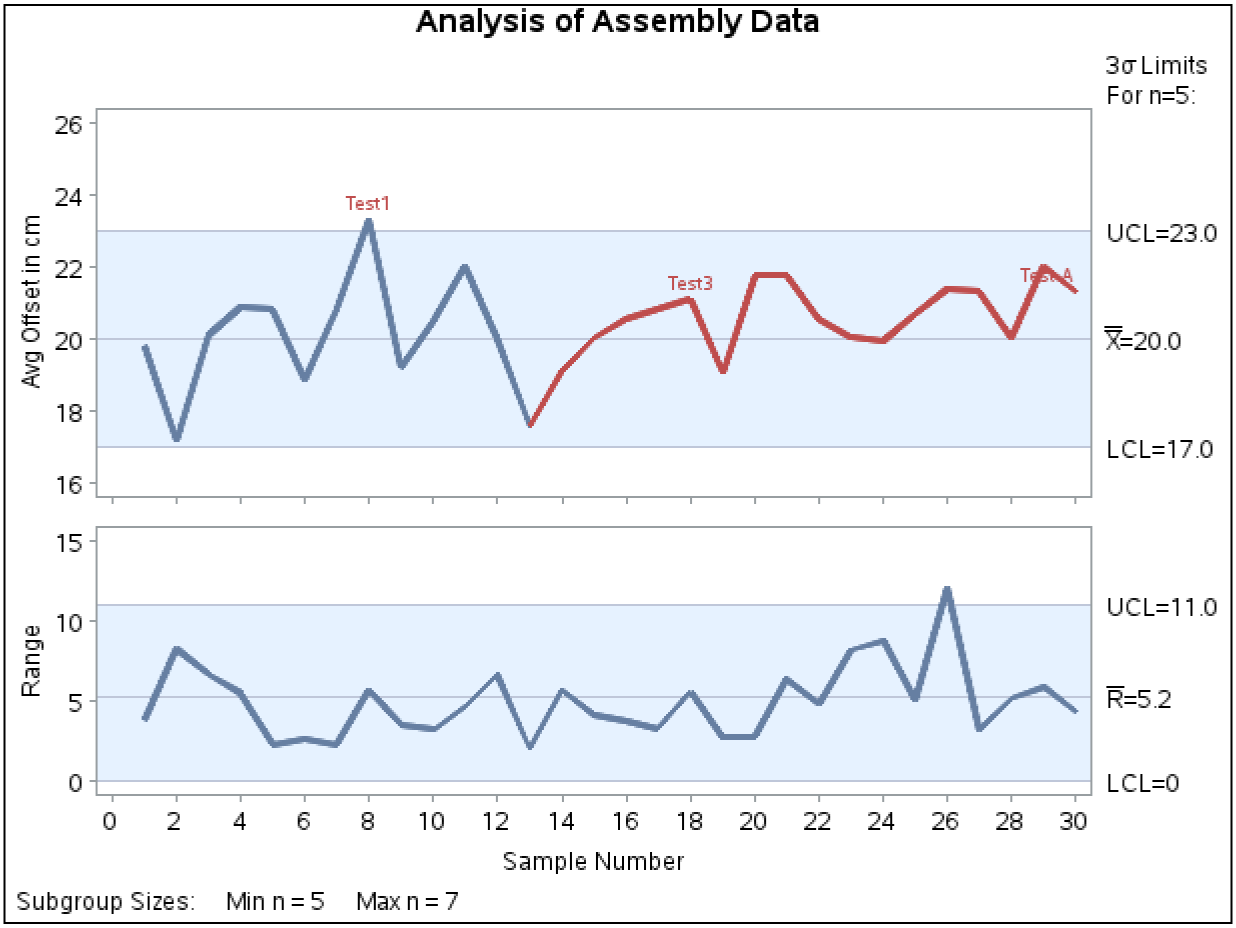
6. You must specify a *legend* with the LEGEND= option if you also specify the TABLELEGEND or TABLEALL option. The legend must be enclosed in quotes and can be up to 40 characters long. The legend is used to describe the test defined by the T-pattern in the table legend requested with the TABLELEGEND and TABLEALL options.

NOTE: See *Applying Tests Based on General Patterns* in the SAS/QC Sample Library.

An example of a nonstandard test using a T-pattern is the run test based on 14 out of 17 points in a row on the same side of the central line that is suggested by Wheeler and Chambers (1986). The following statements apply this test with Tests 1, 3, and 4. The resulting chart is shown in Figure 17.188.

```
ods graphics off;
title 'Analysis of Assembly Data';
proc shewhart history=Assembly;
   xrchart Offset * Sample /
      mu0       = 20
      sigma0    = 2.24
      limitn    = 5
      alln
      tests     = 1
                  t( k=14 m=17
                     lower=0 upper=. scheme=twosided
                     code=A label='Test A' )
                  3 4
      vaxis     = 16 to 26 by 2
      split     = '/' ;
   label OffsetX = 'Avg Offset in cm/Range';
run;
```

Figure 17.188 Generalized T-pattern Applied to Assembly Data



The specified T-pattern is signaled at 30th subgroup. Consequently, this point is labeled *Test A*.

The general syntax for an M-pattern is of the form

M(K=*k* DIR=*direction* CODE=*character* LABEL='*label*' LEGEND='*legend*')

The options for an M-pattern are summarized in the following table:

Table 17.96 Options for M-Patterns

Option	Description
K=*k*	number of points
DIR=INC	increasing pattern
DIR=DEC	decreasing pattern
CODE=*character*	identifier for test (A-H)
LABEL=*'label'*	label for points that are signaled
LEGEND=*'legend'*	legend used with the TABLELEGEND option

You must specify the direction of the pattern with the DIR= option. The CODE=, LABEL=, and LEGEND= options are used as described on page 2091.

CAUTION: You should not substitute tests based on arbitrarily defined T-patterns and M-patterns for standard tests in general process control applications. The pattern options are intended primarily as a research tool.

NOTE: See *ARL With Supplementary Run Rules* in the SAS/QC Sample Library.

Champ and Woodall (1990) provide a FORTRAN program for assessing the run length distribution of tests based on T-patterns. A version of their algorithm is implemented by a SAS/IML program in the SAS/QC Sample Library.

If you specify either a T-pattern or M-pattern with the TESTS= option and save the results in an OUTTABLE= data set, the length of the variable _TESTS_ is 16 rather than 8 (the default). The ninth character of _TESTS_ is assigned the value 'A' if the test with CODE=A is signaled, the tenth character of _TESTS_ is assigned the value 'B' if the test with CODE=B is signaled, and so on. If you also specify one or more standard tests, the *i*th character of _TESTS_ is assigned the value *i* if Test *i* is signaled.

Customizing Tests with DATA Step Programs

NOTE: See *Customizing Tests with DATA Step Programs* in the SAS/QC Sample Library.

Occasionally, you may find it necessary to apply customized tests that cannot be specified with the TESTS= option. You can program your own tests as follows:

1. Run the SHEWHART procedure without the TESTS= option and save the results in an OUTTABLE= data set. Use the NOCHART option to suppress the display of the chart.
2. Use a DATA step program to apply your tests to the subgroup statistics in the OUTTABLE= data set. If tests are signaled at certain subgroups, save these results as values of a flag variable named _TESTS_, which should be a character variable of length 8. Recall that each observation of an OUTTABLE= data set corresponds to a subgroup. Assign the character *i* to the *i*th character of _TESTS_ if the *i*th customized test is signaled at that subgroup (otherwise, assign a blank character).
3. Run the procedure reading the modified data set as a TABLE= data set.

The following example illustrates these steps by creating an $\bar{X}$ chart for the data in Assembly (see "Requesting Standard Tests" on page 2076) that signals a special cause of variation if an average is greater than 2.5 standard errors above the central line. The first step is to compute 2.5σ limits and save both the subgroup statistics and the limits in an OUTTABLE= data set named First.

```
proc shewhart history=Assembly;
   xchart Offset * Sample /
      sigmas   = 2.5
      outtable = First
      nochart ;
run;

title ;
proc print data=First(obs=10) noobs;
run;
```

A partial listing of the data set First is shown in Figure 17.189.

Figure 17.189 Partial Listing of the Data Set First

VAR	Sample	_SIGMAS_	_LIMITN_	_SUBN_	_LCLX_	_SUBX_	_MEAN_	_UCLX_	_STDDEV_	_EXLIM_
Offset	1	2.5	5	5	18.1515	19.80	20.4733	22.7951	2.07665	
Offset	2	2.5	5	5	18.1515	17.16	20.4733	22.7951	2.07665	LOWER
Offset	3	2.5	5	5	18.1515	20.11	20.4733	22.7951	2.07665	
Offset	4	2.5	5	5	18.1515	20.89	20.4733	22.7951	2.07665	
Offset	5	2.5	5	5	18.1515	20.83	20.4733	22.7951	2.07665	
Offset	6	2.5	5	5	18.1515	18.87	20.4733	22.7951	2.07665	
Offset	7	2.5	5	5	18.1515	20.84	20.4733	22.7951	2.07665	
Offset	8	2.5	5	5	18.1515	23.33	20.4733	22.7951	2.07665	UPPER
Offset	9	2.5	5	5	18.1515	19.21	20.4733	22.7951	2.07665	
Offset	10	2.5	5	5	18.1515	20.48	20.4733	22.7951	2.07665	

The second step is to carry out the test and create the flag variable _TESTS_.

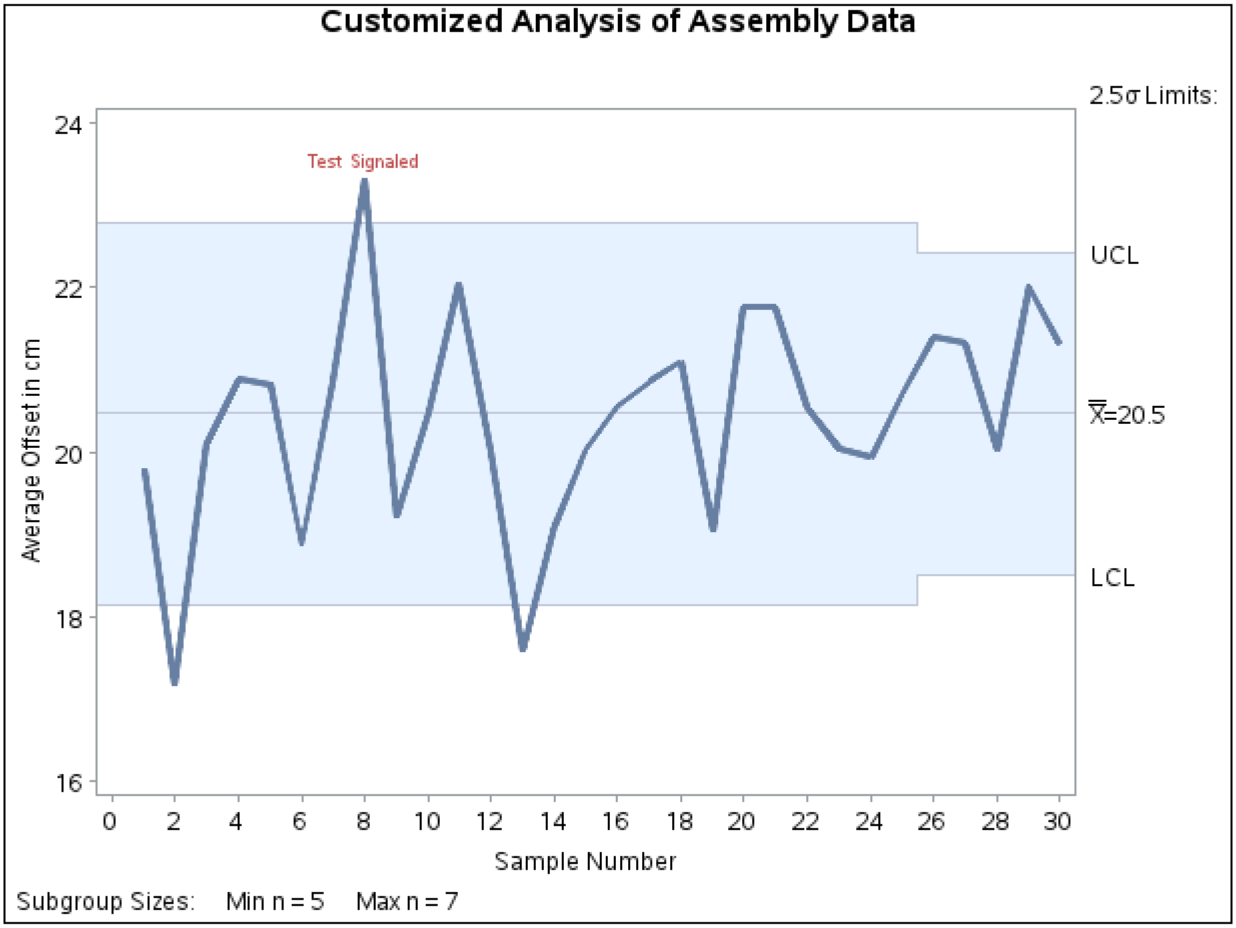
```
data First;
   set First;
   length _tests_ $ 8;
   if _subx_ > _uclx_ then substr( _tests_, 1 ) = '1';
run;
```

Finally, the data set First is read by the SHEWHART procedure as a TABLE= data set.

```
ods graphics off;
title 'Customized Analysis of Assembly Data';
proc shewhart table=First;
   xchart Offset * Sample / tests      = 1
                            testlabel1 = 'Test Signaled';
   label _subx_ = 'Average Offset in cm';
run;
```

The chart is shown in Figure 17.190. Note that the variable _TESTS_ is read “as is” to flag points on the chart, and the standard tests are *not* applied to the data. The option TESTS=1 specifies that a point is to be labeled if the first character of _TESTS_ for the corresponding subgroup is `1`. The label is specified by the TESTLABEL1= option (the default would be *Test1*).

Figure 17.190 Customized Test



In general, you can simultaneously apply up to eight customized tests with the variable _TESTS_, which is of length 8. If two or more tests are signaled at a particular point, the label that is displayed corresponds to the test that appears first in the TESTS= list. In the preceding example, the test involves only the current subgroup. For customized tests involving patterns that span multiple subgroups, you will find it helpful to use the LAG functions described in *SAS Functions and CALL Routines: Reference*.

Notes:

1. If you provide the variable _TESTS_ in a TABLE= data set, you must also use the TESTS= option to specify which characters of _TESTS_ are to be checked.

2. The CTESTS= and LTESTS= options specify colors and line styles for *standard* patterns and may not be applicable with customized tests.

Specialized Control Charts: SHEWHART Procedure

Overview: Specialized Control Charts

Although the Shewhart chart serves well as the fundamental tool for statistical process control (SPC) applications, its assumptions are challenged by many modern manufacturing environments. For example, when standard control limits are used in applications where the process is sampled frequently, autocorrelation in the measurements can result in too many out-of-control signals. This section also considers process control applications involving multiple components of variation, short production runs, nonnormal process data, and multivariate process data.

These questions are subjects of current research and debate. It is not the goal of this section to provide definitive solutions but rather to illustrate some basic approaches that have been proposed and indicate how they can be implemented with short SAS programs. The examples in this section use the SHEWHART procedure in conjunction with various SAS procedures for statistical modeling, as summarized by the following table:

Process Control Application	**Modeling Procedure**
Diagnosing and modeling autocorrelation in process data	ARIMA
Developing control limits for processes involving multiple components of variation	MIXED
Establishing control with short production runs and checking for constant variance	GLM
Developing control limits for nonnormal individual measurements	CAPABILITY
Creating control charts for multivariate process data	PRINCOMP

Autocorrelation in Process Data

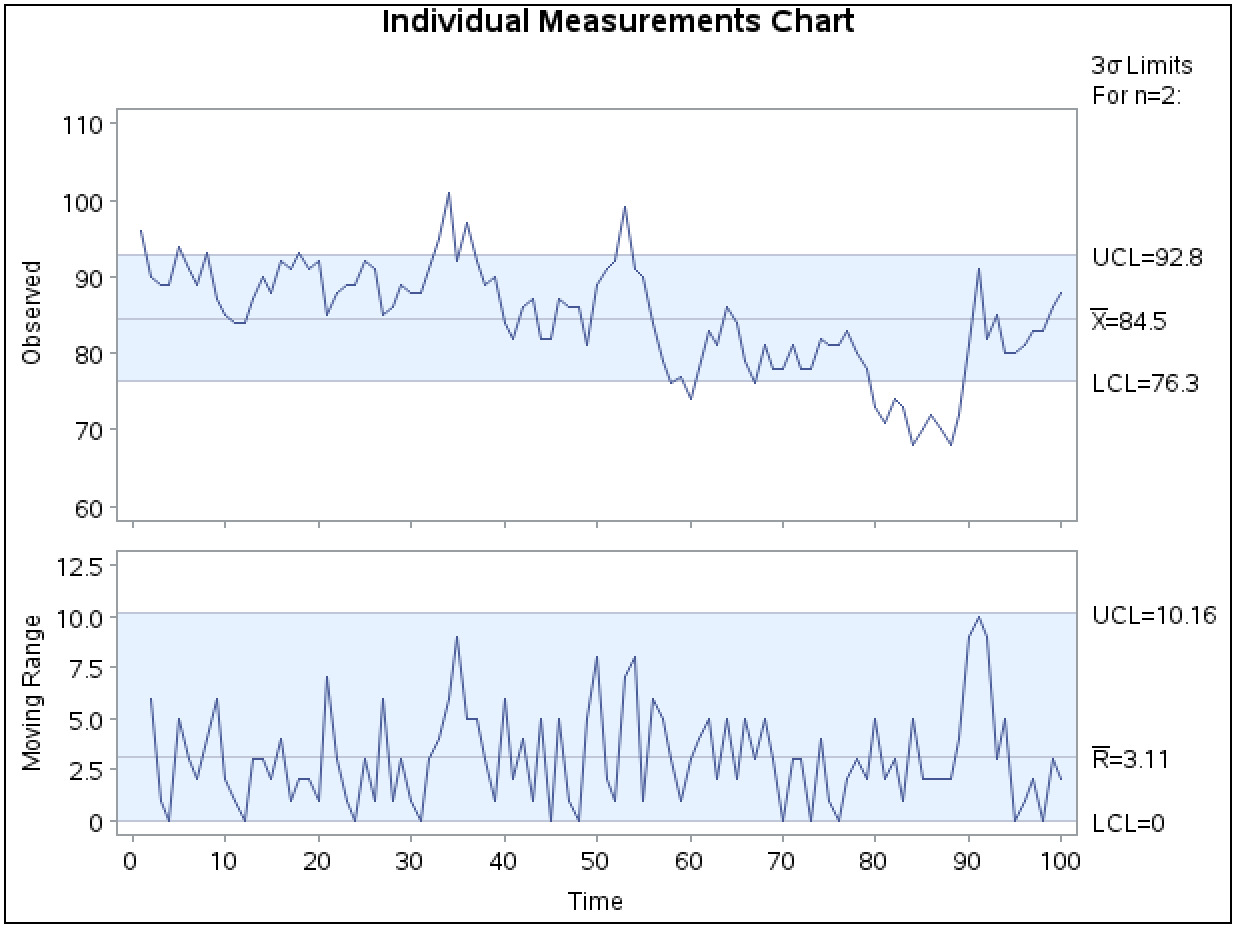
NOTE: See *Autocorrelation in Process Data* in the SAS/QC Sample Library.

Autocorrelation has long been recognized as a natural phenomenon in process industries, where parameters such as temperature and pressure vary slowly relative to the rate at which they are measured. Only in recent years has autocorrelation become an issue in SPC applications, particularly in parts industries, where autocorrelation is viewed as a problem that can undermine the interpretation of Shewhart charts. One reason for this concern is that, as automated data collection becomes prevalent in parts industries, processes are sampled more frequently and it is possible to recognize autocorrelation that was previously undetected. Another reason, noted by Box and Kramer (1992), is that the distinction between parts and process industries is becoming blurred in areas such as computer chip manufacturing. For two other discussions of this issue, refer to Schneider and Pruett (1994) and Woodall (1993).

The standard Shewhart analysis of individual measurements assumes that the process operates with a constant mean μ, and that x_t (the measurement at time t) can be represented as $x_t = \mu + \epsilon_t$, where ϵ_t is a random displacement or error from the process mean μ. Typically, the errors are assumed to be statistically independent in the derivation of the control limits displayed at three standard deviations above and below the central line, which represents an estimate for μ.

When Shewhart charts are constructed from autocorrelated measurements, the result can be too many false signals, making the control limits seem too tight. This situation is illustrated in Figure 17.191, which displays an individual measurement and moving range chart for 100 observations of a chemical process.

Figure 17.191 Conventional Shewhart Chart



The measurements are saved in a SAS data set named Chemical.[18] The chart in Figure 17.191 is created with the following statements:

```
symbol h=2.0 pct;
title 'Individual Measurements Chart';
proc shewhart data=Chemical;
   irchart xt*t / npanelpos = 100
                  split     = '/';
   label xt = 'Observed/Moving Range'
          t = 'Time';
run;
```

[18]The measurements are patterned after the values plotted in Figure 1 of Montgomery and Mastrangelo (1991).

Diagnosing and Modeling Autocorrelation

You can diagnose autocorrelation with an autocorrelation plot created with the ARIMA procedure.

```
ods graphics on;
ods select ChiSqAuto SeriesACFPlot SeriesPACFPlot;
proc arima data=Chemical plots(only)=series(acf pacf);
   identify var = xt;
run;
quit;
```

Refer to *SAS/ETS User's Guide* for details on the ARIMA procedure. The output, shown in Figure 17.192 and Figure 17.193, indicates that the data are highly autocorrelated with a lag 1 autocorrelation of 0.83.

Figure 17.192 Autocorrelation Check for Chemical Data

Individual Measurements Chart

The ARIMA Procedure

Autocorrelation Check for White Noise

To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	228.15	6	<.0001	0.830	0.718	0.619	0.512	0.426	0.381
12	315.34	12	<.0001	0.360	0.364	0.380	0.347	0.348	0.354
18	406.76	18	<.0001	0.349	0.371	0.348	0.353	0.368	0.341
24	442.15	24	<.0001	0.303	0.261	0.230	0.184	0.141	0.098

Figure 17.193 Autocorrelation Plots for Chemical Data

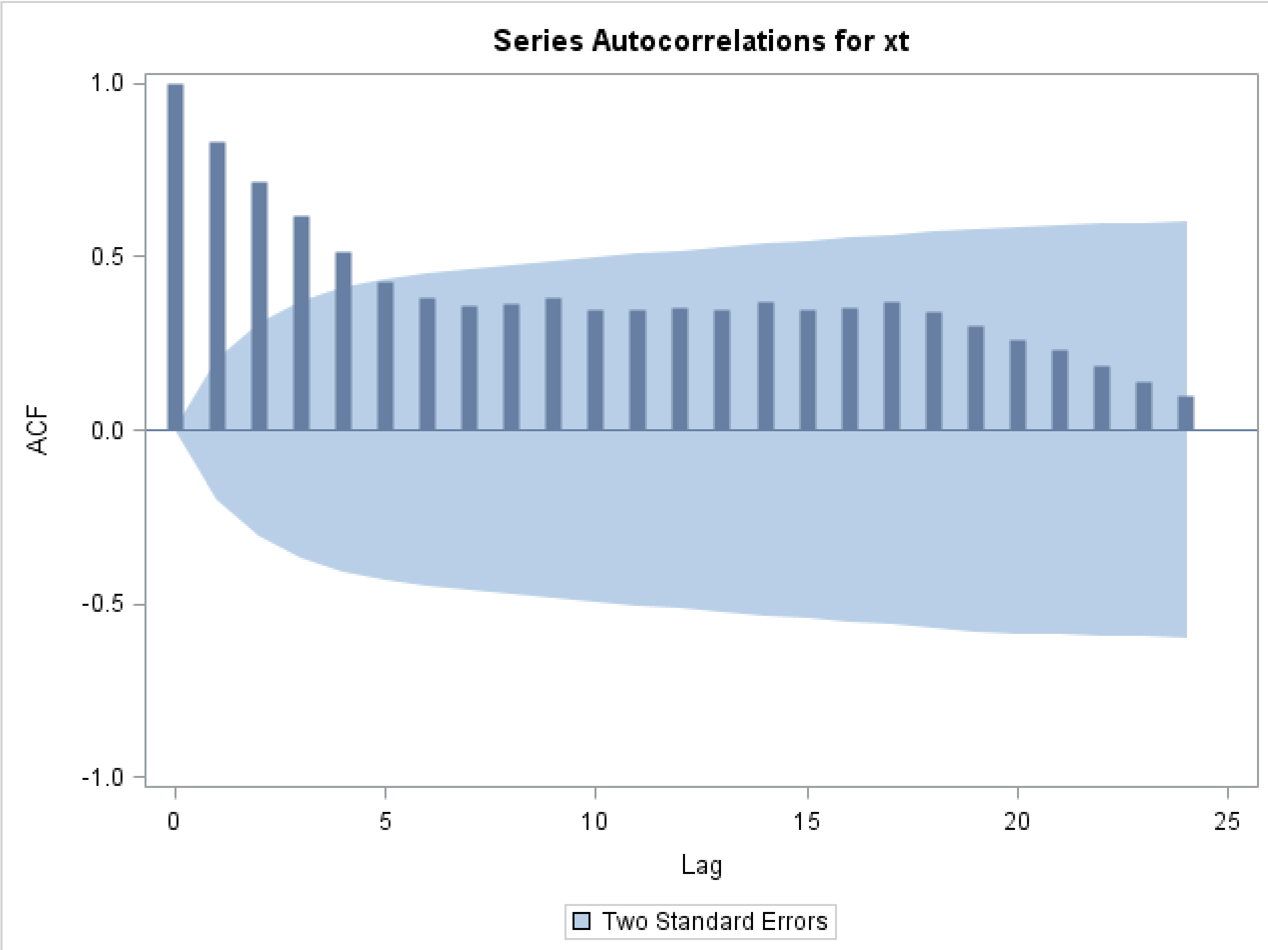
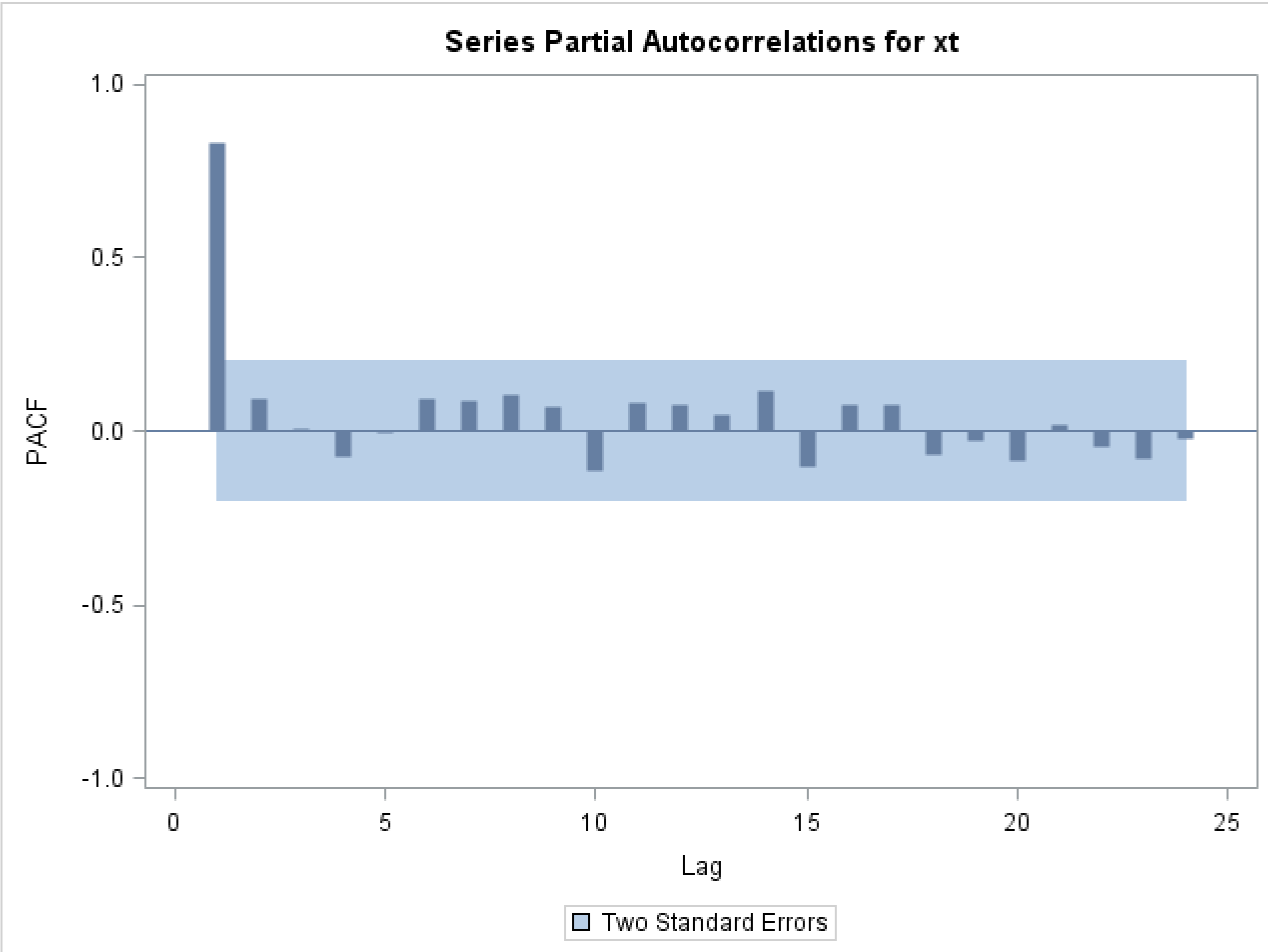


Figure 17.193 *continued*



The partial autocorrelation plot in Figure 17.193 suggests that the data can be modeled with a first-order autoregressive model, commonly referred to as an AR(1) model.

$$\tilde{x}_t \equiv x_t - \mu = \phi_0 + \phi_1 \tilde{x}_{t-1} + \epsilon_t$$

You can fit this model with the ARIMA procedure. The results in Figure 17.194 show that the equation of the fitted model is $\tilde{x}_t = 13.05 + 0.847\tilde{x}_{t-1}$.

```
ods select ParameterEstimates;
proc arima data=Chemical;
   identify var=xt;
   estimate p=1 method=ml;
run;
```

Figure 17.194 Fitted AR(1) Model

Individual Measurements Chart

The ARIMA Procedure

Maximum Likelihood Estimation

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MU	85.28375	2.32973	36.61	<.0001	0
AR1,1	0.84694	0.05221	16.22	<.0001	1

Strategies for Handling Autocorrelation

There is considerable disagreement on how to handle autocorrelation in process data. Consider the following three views:

- At one extreme, Wheeler (1991) argues that the usual control limits are contaminated "only when the autocorrelation becomes excessive (say 0.80 or larger)." He concludes that "one need not be overly concerned about the effects of autocorrelation upon the control chart."
- At the opposite extreme, automatic process control (APC), also referred to as engineering process control, views autocorrelation as a phenomenon to be exploited. In contrast to SPC, which assumes that the process remains on target unless an unexpected but removable cause occurs, APC assumes that the process is changing dynamically due to known causes that cannot be eliminated. Instead of avoiding "overcontrol" and "tampering," which have a negative connotation in the SPC framework, APC advocates continuous tuning of the process to achieve minimum variance control. Descriptions of this approach and discussion of the differences between APC and SPC are provided by a number of authors, including Box and Kramer (1992), MacGregor (1987, 1990), MacGregor, Hunter, and Harris (1988), and Montgomery et al. (1994).
- A third strategy advocates removing autocorrelation from the data and constructing a Shewhart chart (or an EWMA chart or a cusum chart) for the residuals; refer, for example, to Alwan and Roberts (1988).

An example of the last approach is presented in the remainder of this section simply to demonstrate the use of the ARIMA procedure in conjunction with the SHEWHART procedure. The ARIMA procedure models the autocorrelation and saves the residuals in an output data set; the SHEWHART procedure creates a control chart using the residuals as input data.

In the chemical data example, the residuals can be computed as forecast errors and saved in an output SAS data set with the FORECAST statement in the ARIMA procedure.

```
proc arima data=Chemical;
   identify var=xt;
   estimate p=1 method=ml;
   forecast out=Results id=t;
run;
```

The output data set (named Results) saves the one-step-ahead forecasts as a variable named forecast, and it also contains the original variables xt and t. You can create a Shewhart chart for the residuals by using the data set Results as input to the SHEWHART procedure.

```
title 'Residual Analysis Using AR(1) Model';
symbol h=2.0 pct;
proc shewhart data=Results(firstobs=4 obs=100);
   xchart xt*t / npanelpos = 100
                 split     = '/'
                 trendvar  = forecast
                 xsymbol   = xbar
                 ypct1     = 40
                 vref2     = 70 to 100 by 10
                 lvref     = 2
                 nolegend;
   label xt = 'Residual/Forecast'
          t = 'Time';
run;
```

The chart is shown in Figure 17.195. Specifying TRENDVAR=forecast plots the values of forecast in the lower chart and plots the residuals (xt – forecast) together with their 3σ limits in the upper chart.[19]

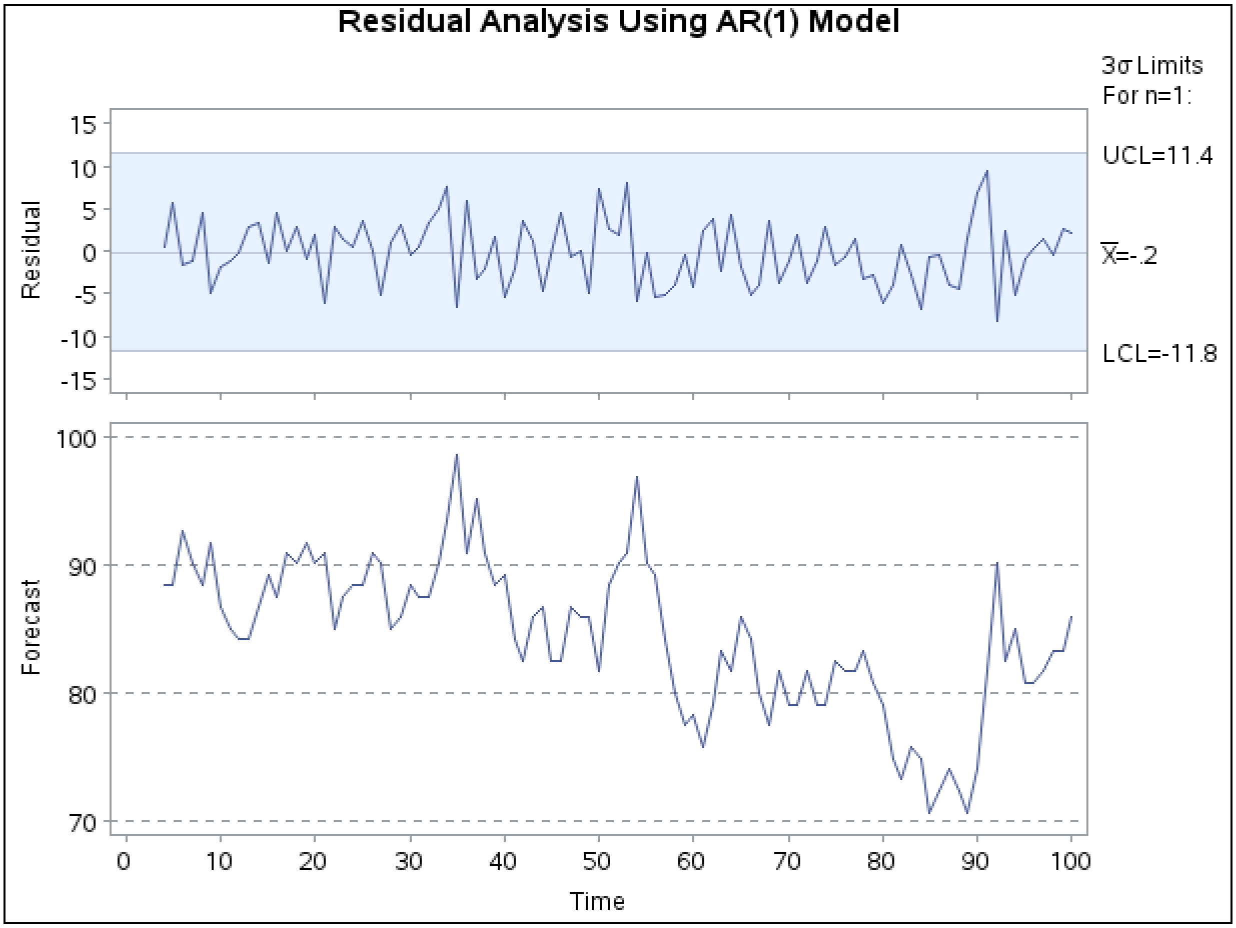
Various other methods can be applied with this data. For example, Montgomery and Mastrangelo (1991) suggest fitting an exponentially weighted moving average (EWMA) model and using this model as the basis for a display that they refer to as an *EWMA central line control chart*.

Before presenting the statements for creating this display, it is helpful to review some terminology. The EWMA *statistic* plotted on a conventional EWMA control chart is defined as

$$z_t = \lambda x_t + (1 - \lambda) z_{t-1}$$

[19] The upper chart in Figure 17.195 resembles Figure 2 of Montgomery and Mastrangelo (1991), who conclude that the process is in control.

Figure 17.195 Residuals from AR(1) Model



The EWMA chart (which you can construct with the MACONTROL procedure) is based on the assumption that the observations x_t are independent. However, in the context of autocorrelated process data (and more generally in time series analysis), the EWMA statistic z_t plays a different role:[20] it is the optimal one-step-ahead forecast for a process that can be modeled by an ARIMA(0,1,1) model

$$x_t = x_{t-1} + \epsilon_t - \theta\epsilon_{t-1}$$

provided that the weight parameter λ is chosen as $\lambda = 1 - \theta$. This statistic is also a good predictor when the process can be described by a subset of ARIMA models for which the process is “positively autocorrelated and the process mean does not drift too quickly.”[21]

[20]For a discussion of these roles, refer to Hunter (1986).

[21]Refer to Montgomery and Mastrangelo (1991) and the discussion that follows their paper.

You can fit an ARIMA(0,1,1) model to the chemical data with the following statements. A summary of the fitted model is shown in Figure 17.196.

```
title ;
proc arima data=Chemical;
   identify var=xt(1);
   estimate q=1 method=ml noint;
   forecast out=EWMA id=t;
run;
```

Figure 17.196 Fitted ARIMA(0, 1, 1) Model

The ARIMA Procedure

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MA1,1	0.15041	0.10021	1.50	0.1334	1

Variance Estimate	14.97024
Std Error Estimate	3.86914
AIC	549.868
SBC	552.4631
Number of Residuals	99

The forecast values and their standard errors (variables forecast and STD), together with the original measurements, are saved in a data set named EWMA. The EWMA central line control chart plots the forecasts from the ARIMA(0,1,1) model as the central "line," and it uses the standard errors of prediction to determine upper and lower control limits. You can construct this chart, shown in Figure 17.197,[22] with the following statements:

```
data EWMA;
   set EWMA(firstobs=2 obs=100);
run;

data EWMAtab;
   length _var_ $ 8 ;
   set EWMA (rename=(forecast=_mean_ xt=_subx_));
   _var_    = 'xt';
   _sigmas_ = 3;
   _limitn_ = 1;
   _lclx_   = _mean_ - 3 * std;
   _uclx_   = _mean_ + 3 * std;
   _subn_   = 1;
run;
```

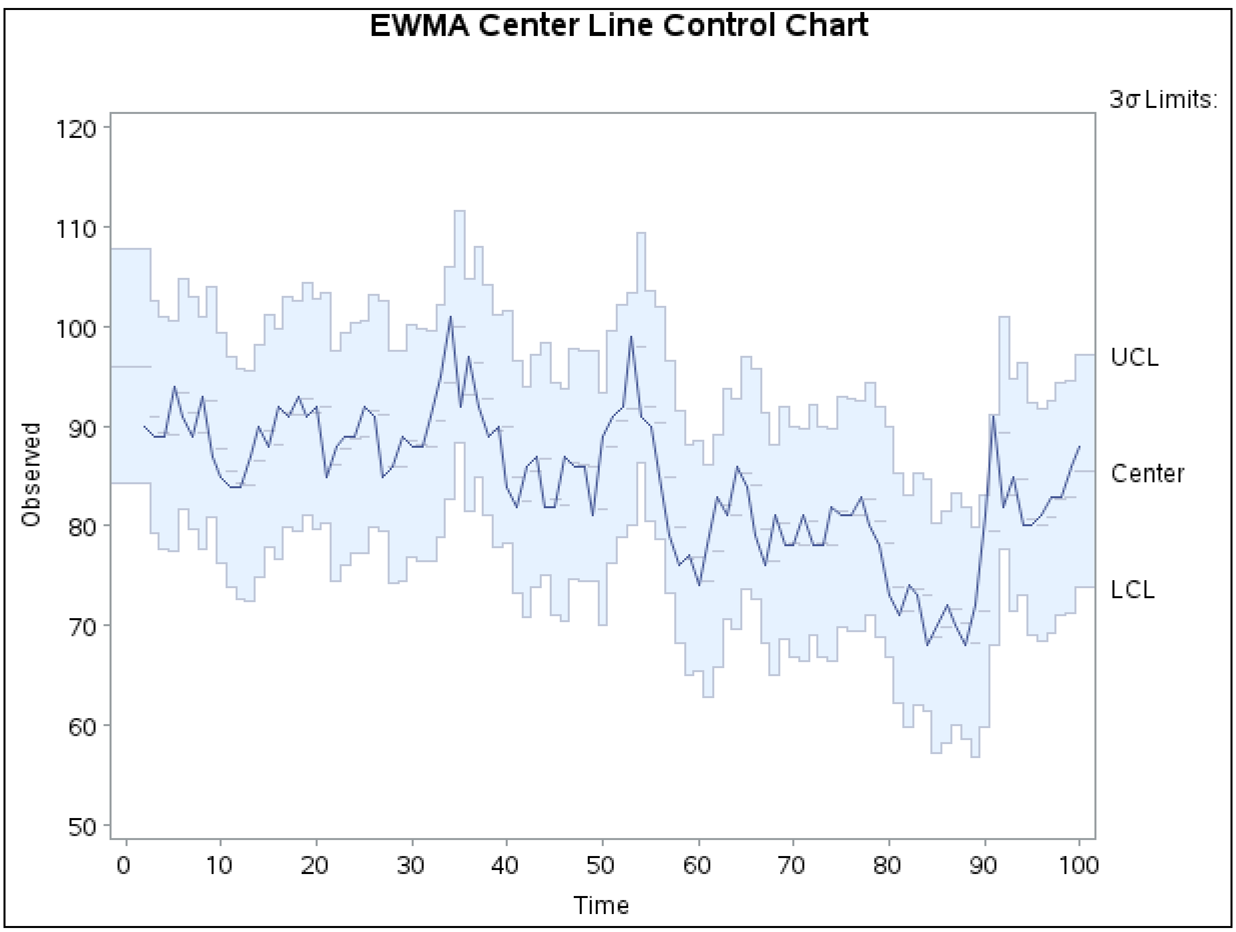
[22]Figure 17.197 is similar to Figure 5 of Montgomery and Mastrangelo (1991).

```
symbol h=2.0 pct;
title 'EWMA Center Line Control Chart';
proc shewhart table=EWMAtab;
   xchart xt*t / npanelpos = 100
                 xsymbol   = 'Center'
                 nolegend;
   label _subx_ = 'Observed'
              t = 'Time' ;
run;
```

Note that EWMA is read by the SHEWHART procedure as a TABLE= input data set, which has a special structure intended for applications in which both the statistics to be plotted and their control limits are pre-computed. The variables in a TABLE= data set have reserved names beginning and ending with the underscore character; for this reason, forecast and xt are temporarily renamed as _MEAN_ and _SUBX_, respectively. For more information about TABLE= data sets, see "Input Data Sets" in the section for the chart statement in which you are interested.

Again, the conclusion is that the process is in control. While Figure 17.195 and Figure 17.197 are not the only displays that can be considered for analyzing the chemical data, their construction illustrates the conjunctive use of the ARIMA and SHEWHART procedures in process control applications involving autocorrelated data.

Figure 17.197 EWMA Center Line Chart



Multiple Components of Variation

NOTE: See *Multiple Components of Variation* in the SAS/QC Sample Library.

In the preceding section, the excessive variation in the conventional Shewhart chart in Figure 17.191 is the result of positive autocorrelation in the data. The variation is "excessive" not because it is due to special causes of variation, but because the Shewhart model is inappropriate. This section considers another form of departure from the Shewhart model; here, measurements are *independent* from one subgroup sample to the next, but there are multiple components of variation for each measurement. This is illustrated with an example involving two components.[23]

A company that manufactures polyethylene film monitors the statistical control of an extrusion process that produces a continuous sheet of film. At periodic intervals of time, samples are taken at four locations (referred to as lanes) along a cross section of the sheet, and a test measurement is made of each sample. The test values are saved in a SAS data set named Film. A partial listing of Film is shown in Figure 17.198.

Figure 17.198 Polyethylene Sheet Measurements in the Data Set Film

Sample	Lane	Testval
1	A	93
1	B	87
1	C	92
1	D	78
2	A	87

Preliminary Examination of Variation

As a preliminary step in the analysis, the data are sorted by lane and visually screened for outliers (test values greater than 130) with box plots created as follows:

```
ods graphics off;
proc sort data=Film;
   by Lane;
run;
symbol v = dot h = 2.0 pct;
title 'Outlier Analysis';
proc shewhart data=Film;
   boxchart Testval*Lane / boxstyle = schematicid
                           idsymbol = dot
                           vref     = 130
                           vreflab  = 'Outlier Cutoff'
                           hoffset  = 5
                           nolegend
                           stddevs
                           nolimits ;
   id Sample;
run;
```

[23] Also refer to Chapter 5 of Wheeler and Chambers (1986) for an explanation of the effects of subgrouping and sources of variation on control charts.

Specifying BOXSTYLE=SCHEMATICID requests schematic box plots with outliers identified by the value of the ID variable Sample. The STDDEVS option specifies that the estimate of the process standard deviation is to be based on subgroup standard deviations. Although this estimate is not needed here because control limits are not displayed, it is recommended that you specify the STDDEVS option whenever you are working with subgroup sample sizes greater than ten. The NOLEGEND and NOLIMITS options suppress the subgroup sample size legend and control limits for lane means that are displayed by default. The display is shown in Figure 17.199.

Figure 17.199 Outlier Analysis for the Data Set Film

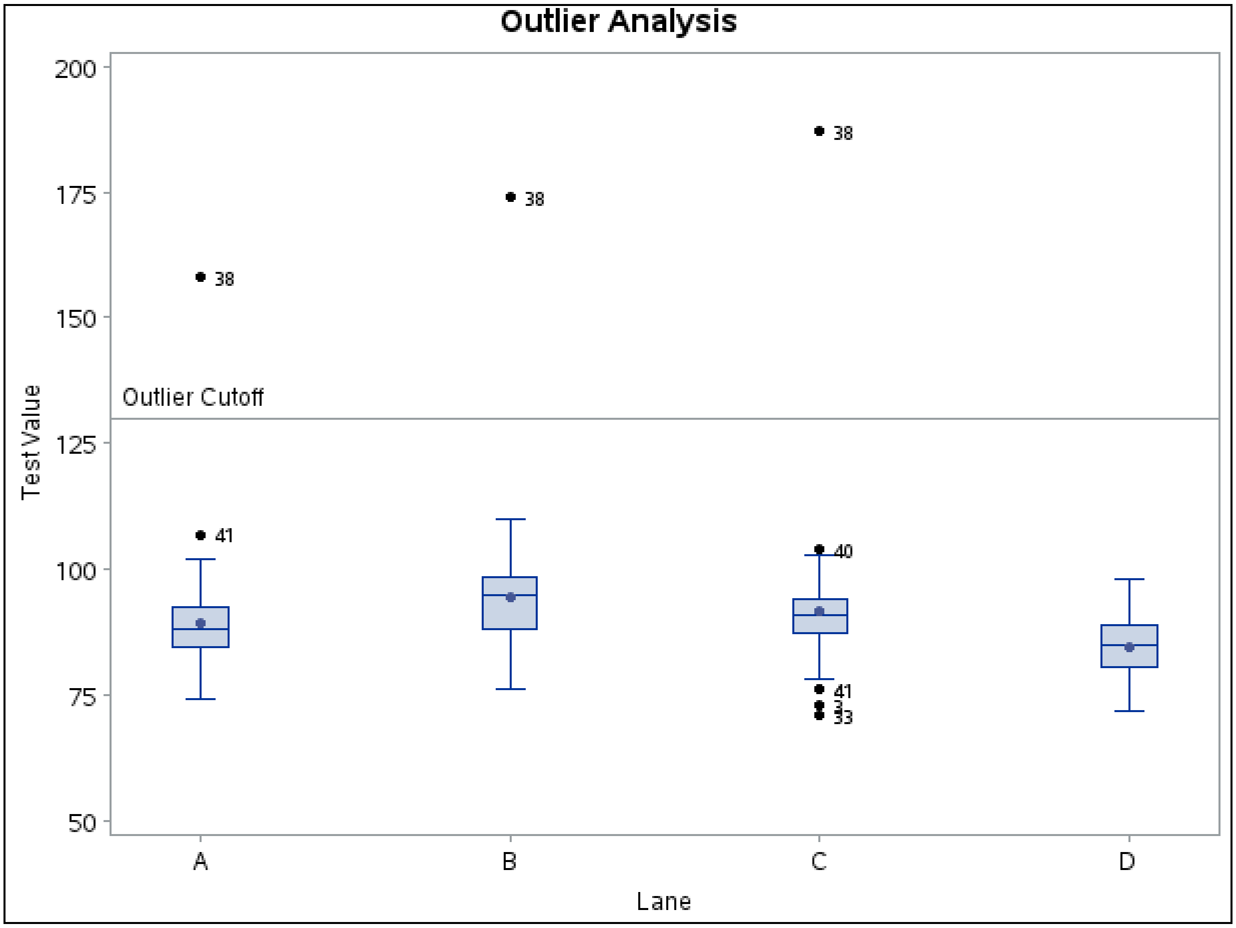
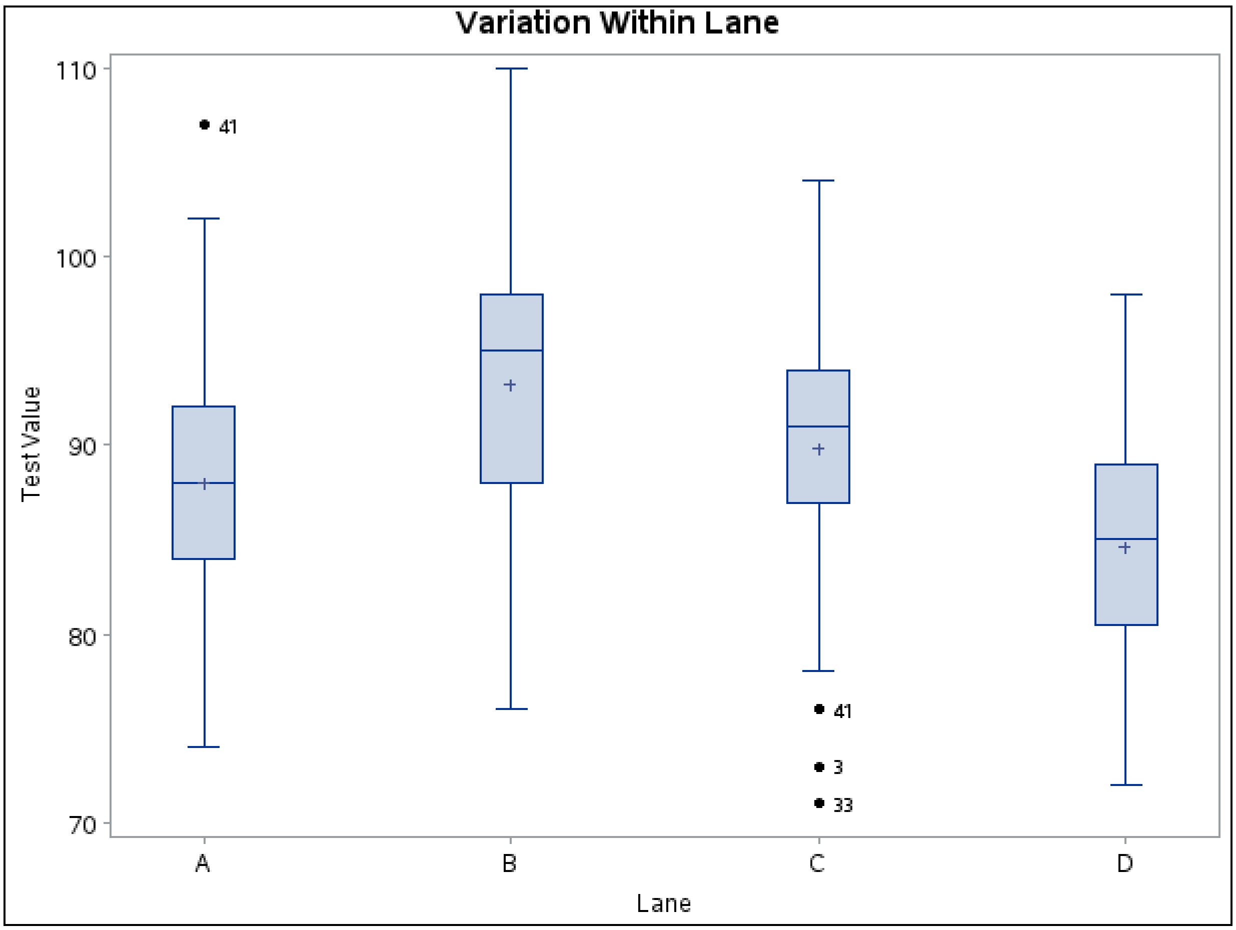


Figure 17.200 shows similarly created box plots for the data in Film2, which is created by removing the outliers from the data set Film.

```
data Film2;
   set Film;
   if Testval < 130;
symbol h = 2.0 pct;
title 'Variation Within Lane';
proc shewhart data=Film2;
   boxchart Testval*Lane / boxstyle = schematicid
                           boxwidth = 5
                           idsymbol = dot
                           hoffset  = 5
                           nolegend
                           stddevs
                           nolimits ;
   id Sample;
run;
```

Figure 17.200 The Data Set Film2 Without Outliers

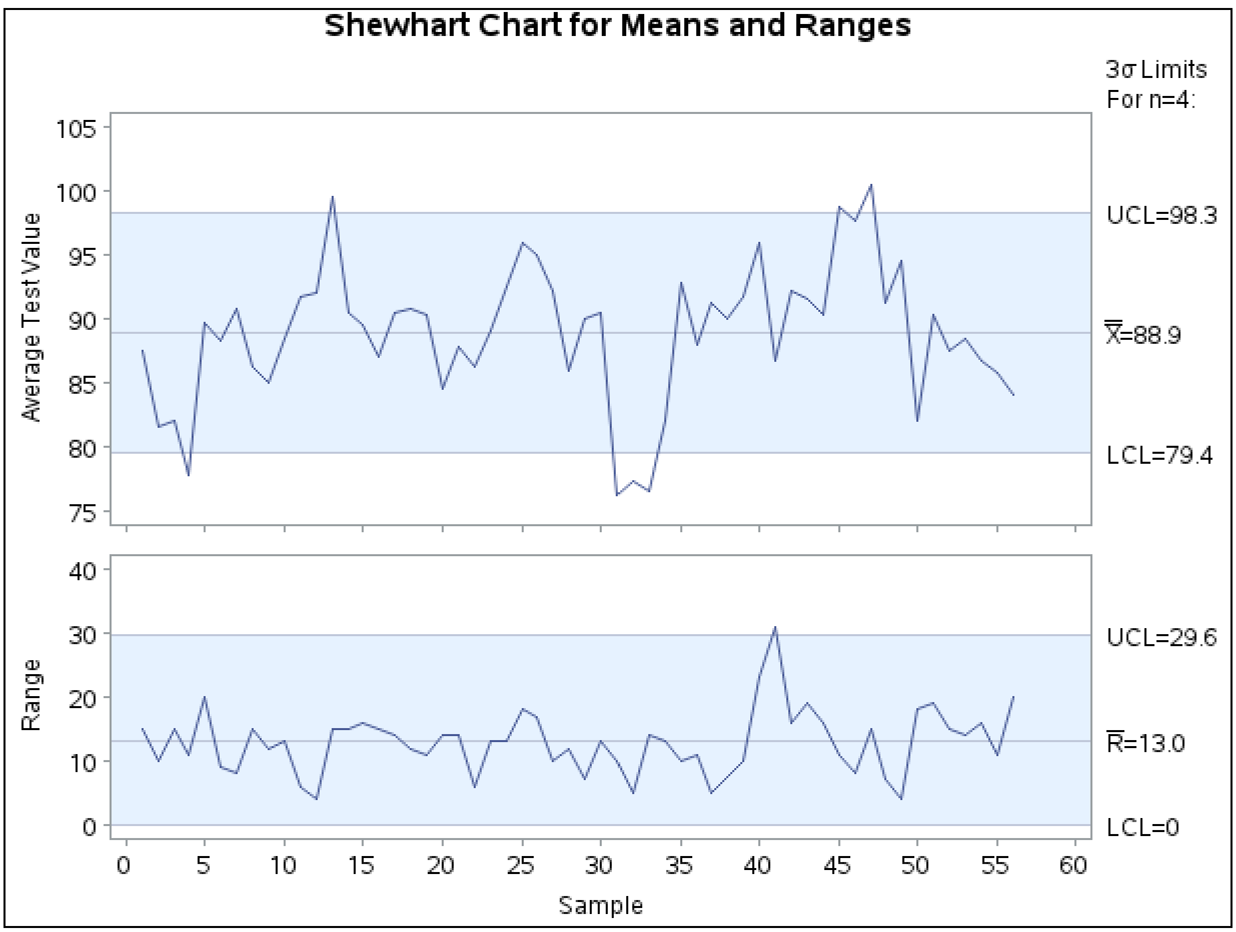


Since you have no additional information about the process, you may want to create a conventional $\bar{X}$ and R chart for the test values grouped by the variable Sample. This is a straightforward application of the XRCHART statement in the SHEWHART procedure.

```
proc sort data=Film2;
   by Sample;
run;
symbol h=2.0 pct;
title 'Shewhart Chart for Means and Ranges';
proc shewhart data=Film2;
   xrchart Testval*Sample /
      split     = '/'
      npanelpos = 60
      limitn    = 4
      outlimits = RLimits
      nolegend
      alln;
   label Testval='Average Test Value/Range';
run;
```

The $\bar{X}$ and R chart is displayed in Figure 17.201. Ordinarily, the out-of-control points in the $\bar{X}$ chart would indicate that the process is not in statistical control. In this situation, however, the process is known to be quite stable, and the data have been screened for outliers. The problem is that the control limits for the average test value were computed from an inappropriate model. This is discussed in the following section.

Figure 17.201 Conventional $\bar{X}$ and R Chart



Determining the Components of Variation

The standard Shewhart analysis assumes that sampling variation, also referred to as *within-group* variation, is the only source of variation. Writing x_{ij} for the *j*th measurement within the *i*th subgroup, you can express the model for the conventional $\bar{X}$ and *R* chart as

$$x_{ij} = \mu + \sigma_W \epsilon_{ij} \qquad (1)$$

for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, n$. The random variables ϵ_{ij} are assumed to be independent with zero mean and unit variance, and σ_W^2 is the within-subgroup variance. The parameter μ denotes the process mean.

In a process such as film manufacturing, this model is not adequate because there is additional variation due to changes in temperature, pressure, raw material, and other factors. A more appropriate model is

$$x_{ij} = \mu + \sigma_B \omega_i + \sigma_W \epsilon_{ij} \qquad (2)$$

where σ_B^2 is the *between-subgroup* variance, the random variables ω_i are independent with zero mean and unit variance, and the random variables ω_i are independent of the random variables ϵ_{ij}.[24]

To plot the subgroup averages $\bar{x}_{i.} \equiv \frac{1}{n} \sum_{j=1}^{n} x_{ij}$ on a control chart, you need expressions for the expectation and variance of $\bar{x}_{i.}$. These are

$$\begin{aligned} E(\bar{x}_{i.}) &= \mu \\ \mathrm{Var}(\bar{x}_{i.}) &= \sigma_B^2 + \frac{\sigma_W^2}{n} \end{aligned}$$

Thus, the central line should be located at $\hat{\mu}$, and 3σ limits should be located at

$$\hat{\mu} \pm 3\sqrt{\widehat{\sigma_B^2} + \frac{\widehat{\sigma_W^2}}{n}} \qquad (3)$$

where $\widehat{\sigma_B^2}$ and $\widehat{\sigma_W^2}$ denote estimates of the variance components. You can use a variety of SAS procedures for fitting linear models to estimate the variance components. The following statements show how this can be done with the MIXED procedure:

```
title;
proc mixed data=Film2;
   class Sample;
   model Testval = / s;
   random Sample;
   ods output solutionf=sf;
   ods output covparms=cp;
run;
```

The results are shown in Figure 17.202. Note that the parameter estimates are $\widehat{\sigma_B^2} = 19.25$, $\widehat{\sigma_W^2} = 39.68$, and $\hat{\mu} = 88.90$.

[24]This notation is used in Chapter 3 of Wetherill and Brown (1991), which discusses this issue.

Figure 17.202 Partial Output from the MIXED Procedure

The Mixed Procedure

Covariance Parameter Estimates	
Cov Parm	**Estimate**
Sample	19.2526
Residual	39.6825

Solution for Fixed Effects					
Effect	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**
Intercept	88.8963	0.7250	55	122.61	<.0001

The following statements merge the output data sets from the MIXED procedure into a SAS data set named Newlim that contains the appropriately derived control limit parameters for average test value:

```
data cp;
   set cp sf;
   keep Estimate;
run;

proc transpose data=cp out=Newlim;
run;

data Newlim (keep=_lclx_ _mean_ _uclx_);
   set Newlim;
   _limitn_ = 4;
   _mean_   = col3;
   _stddev_ = sqrt(4*col1 + col2);
   _lclx_   = _mean_ - 3*_stddev_ / sqrt(_limitn_);
   _uclx_   = _mean_ + 3*_stddev_ / sqrt(_limitn_);
   output;
run;
```

Here, the variable _LIMITN_ is assigned the value of n, the variable _MEAN_ is assigned the value of $\hat{\mu}$, and the variable _STDDEV_ is assigned the value of

$$\hat{\sigma}_{\text{adj}} \equiv \sqrt{4\widehat{\sigma_B^2} + \widehat{\sigma_W^2}}$$

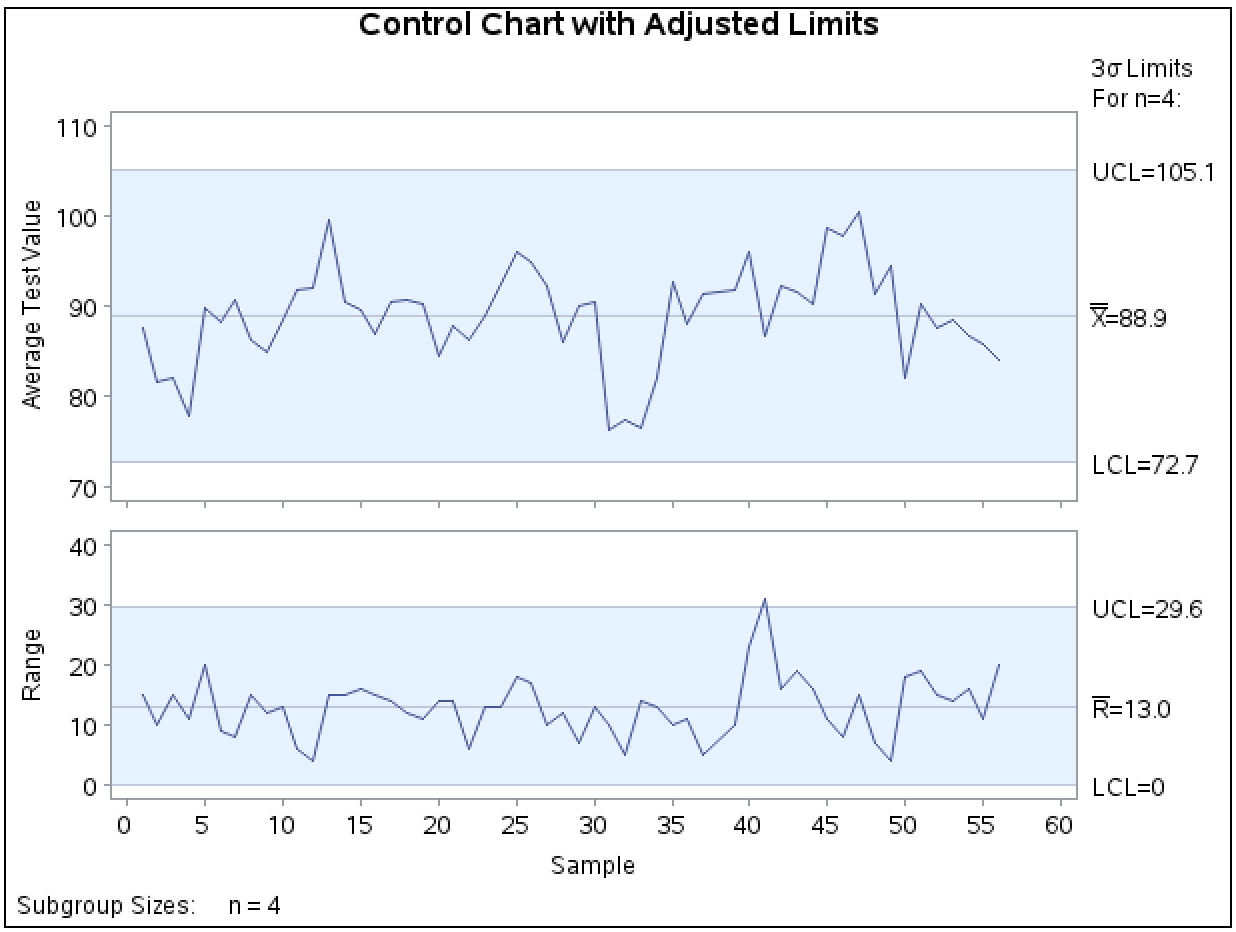
The 3σ limits (_LCLX_ and _UCLX_) are computed according to (3) using $\hat{\sigma}_{\text{adj}}$. The data set Newlim contains the mean and 3σ limits for the average test value.

The following statements compute appropriate control limits for the $\bar{X}$ and R charts, which are shown in Figure 17.203. First, the data set Newlim2 is created by merging the data set RLimits, which contains the original R chart limits computed in "Preliminary Examination of Variation" on page 2106, with Newlim, which saved the appropriate $\bar{X}$ chart limits. The original R chart limits are valid because the range in the ith subgroup is $R_i = \sigma_W(\max_j \epsilon_{ij} - \min_j \epsilon_{ij})$, which is the same for models (1) and (2). The LIMITS= option specifies the data set Newlim2 as the source of the control limits for Figure 17.203.

```
data Newlim2;
   merge Newlim RLimits (drop=_lclx_ _mean_ _uclx_);
run;

title 'Control Chart with Adjusted Limits';
symbol h = 2.0 pct;
proc shewhart data=Film2 limits=Newlim2;
   xrchart Testval*Sample / npanelpos = 60;
   label Testval='Average Test Value';
run;
```

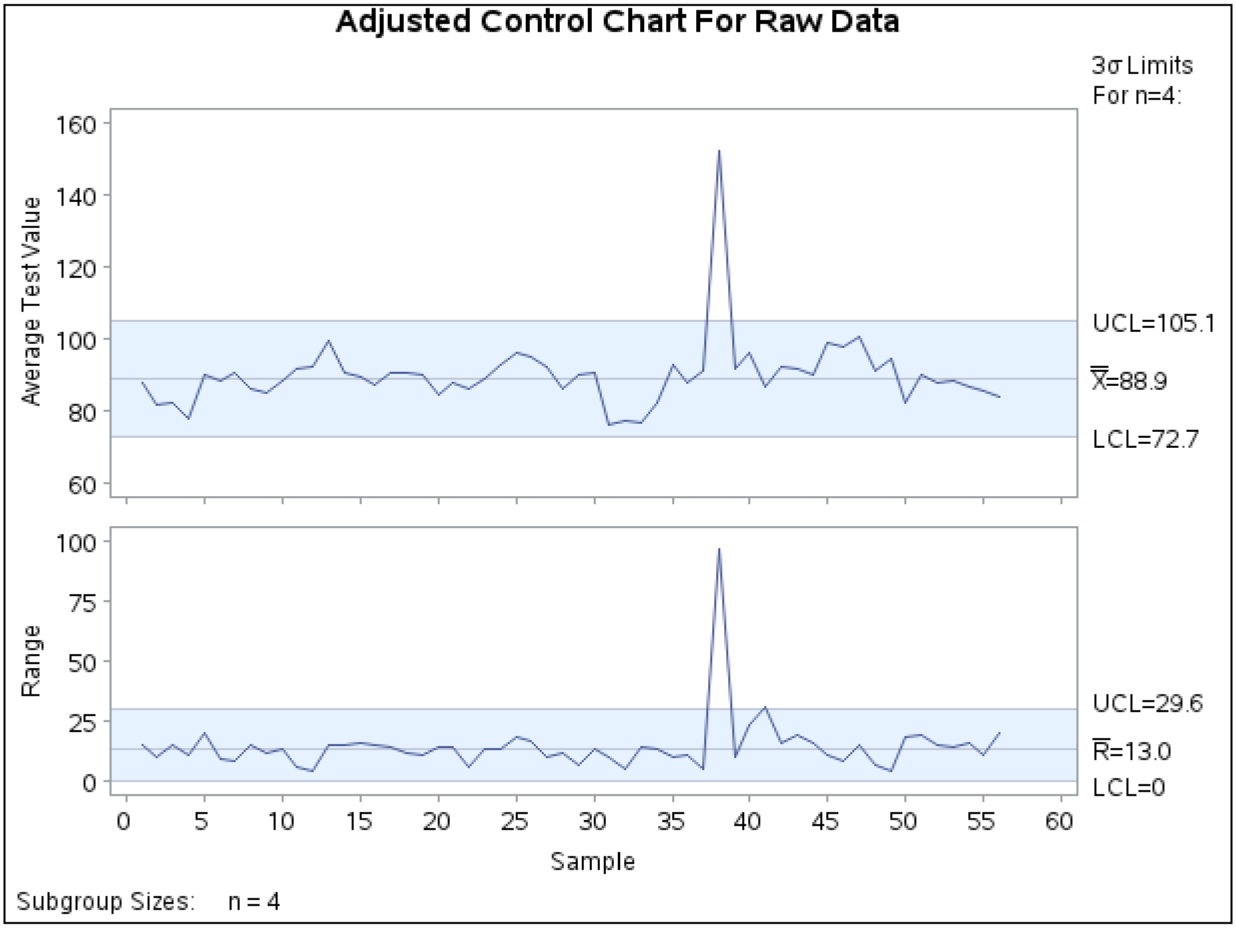
The control limits for the $\bar{X}$ chart in Figure 17.203 are $\hat{\mu} \pm \frac{3}{\sqrt{n}}\hat{\sigma}_{\text{adj}}$. This chart correctly indicates that the variation in the process is due to common causes.

Figure 17.203 $\bar{X}$ and R Chart with Derived Control Limits



You can use a similar set of statements to display the derived control limits in Newlim on an $\bar{X}$ and R chart for the original data (including outliers), as shown in Figure 17.204.

Figure 17.204 $\bar{X}$ and R Chart with Derived Control Limits for Raw Data



A simple alternative to the chart in Figure 17.203 is an "individual measurements" chart for the subgroup means. The advantage of the variance components approach is that it yields separate estimates of the components due to lane and sample, as well as a number of hypothesis tests (these require assumptions of normality). In applying this method, however, you should be careful to use data that represent the process in a state of statistical control.

Short Run Process Control

NOTE: See *Short Run Process Control* in the SAS/QC Sample Library.

When conventional Shewhart charts are used to establish statistical control, the initial control limits are typically based on 25 to 30 subgroup samples. Often, however, this amount of data is not available in manufacturing situations where product changeover occurs frequently or production runs are limited.

A variety of methods have been introduced for analyzing data from a process that is alternating between short runs of multiple products. The methods commonly used in the United States are variations of two basic approaches:[25]

[25] For a review of related methods, refer to Al-Salti and Statham (1994).

- the *difference from nominal* approach. A product-specific nominal value is subtracted from each measured value, and the differences (together with appropriate control limits) are charted. Here it is assumed that the nominal value represents the central location of the process (ideally estimated with historical data) and that the process variability is constant across products.
- the *standardization* approach. Each measured value is standardized with a product-specific nominal and standard deviation values. This approach is followed when the process variability is not constant across products.

These approaches are highlighted in this section because of their popularity, but two alternatives that are technically more sophisticated are worth noting.

- Hillier (1969) provided a method for modifying the usual control limits for $\bar{X}$ and R charts in startup situations where fewer than 25 subgroup samples are available for estimating the process mean μ and standard deviation σ; also refer to Quesenberry (1993).
- Quesenberry (1991b, a) introduced the so-called *Q chart* for short (or long) production runs, which standardizes and normalizes the data using probability integral transformations.

SAS examples illustrating these alternatives are provided in the SAS/QC sample library and are described by Rodriguez and Bynum (1992).

Analyzing the Difference from Nominal

The following example[26] is adapted from an application in aircraft component manufacturing. A metal extrusion process is used to make three slightly different models of the same component. The three product types (labeled M1, M2, and M3) are produced in small quantities because the process is expensive and time-consuming.

Figure 17.205 shows the structure of a SAS data set named Old, which contains the diameter measurements for various short runs. Samples 1 to 30 are to be used to estimate the process standard deviation σ for the differences from nominal.

[26] Refer to Chapter 1 of Wheeler (1991) for a similar example.

Figure 17.205 Diameter Measurements in the Data Set Old

Sample	Prodtype	Diameter
1	M3	13.99
2	M3	14.69
3	M3	13.86
4	M3	14.32
5	M3	13.23
6	M1	17.55
7	M1	14.26
8	M1	14.62
9	M1	12.97
10	M2	16.18
11	M2	15.29
12	M2	16.20
13	M3	13.89
14	M3	12.71
15	M3	14.32
16	M3	15.35
17	M2	15.08
18	M2	14.72
19	M2	14.79
20	M2	15.27
21	M2	15.95
22	M1	14.78
23	M1	15.19
24	M1	15.41
25	M1	16.26
26	M3	16.68
27	M3	15.60
28	M3	14.86
29	M3	16.67
30	M3	14.35

In short run applications involving many product types, it is common practice to maintain a database for the nominal values for the product types. Here, the nominal values are saved in a SAS data set named Nomval, which is listed in Figure 17.206.

Figure 17.206 Nominal Values for Product Types in the Data Set Nomval

Prodtype	Nominal
M1	15.0
M2	15.5
M3	14.8
M4	15.2

To compute the differences from nominal, you must merge the data with the nominal values. You can do this with the following SAS statements. Note that an IN= variable is used in the MERGE statement to allow for the fact that Nomval includes nominal values for product types that are not represented in Old. Figure 17.207 lists the merged data set Old.

```
proc sort data=Old;
   by Prodtype;
run;

data Old;
   format Diff 5.2 ;
   merge Nomval Old(in = a);
      by Prodtype;
   if a;
   Diff = Diameter - Nominal;
run;

proc sort data=Old;
   by Sample;
run;
```

Figure 17.207 Data Merged with Nominal Values

Sample	Prodtype	Diameter	Nominal	Diff
1	M3	13.99	14.8	-0.81
2	M3	14.69	14.8	-0.11
3	M3	13.86	14.8	-0.94
4	M3	14.32	14.8	-0.48
5	M3	13.23	14.8	-1.57
6	M1	17.55	15.0	2.55
7	M1	14.26	15.0	-0.74
8	M1	14.62	15.0	-0.38
9	M1	12.97	15.0	-2.03
10	M2	16.18	15.5	0.68
11	M2	15.29	15.5	-0.21
12	M2	16.20	15.5	0.70
13	M3	13.89	14.8	-0.91
14	M3	12.71	14.8	-2.09
15	M3	14.32	14.8	-0.48
16	M3	15.35	14.8	0.55
17	M2	15.08	15.5	-0.42
18	M2	14.72	15.5	-0.78
19	M2	14.79	15.5	-0.71
20	M2	15.27	15.5	-0.23
21	M2	15.95	15.5	0.45
22	M1	14.78	15.0	-0.22
23	M1	15.19	15.0	0.19
24	M1	15.41	15.0	0.41
25	M1	16.26	15.0	1.26
26	M3	16.68	14.8	1.88
27	M3	15.60	14.8	0.80
28	M3	14.86	14.8	0.06
29	M3	16.67	14.8	1.87
30	M3	14.35	14.8	-0.45

Assume that the variability in the process is constant across product types. To estimate the common process standard deviation σ, you first estimate σ for each product type based on the average of the moving ranges of the differences from nominal. You can do this in several steps, the first of which is to sort the data and compute the average moving range with the SHEWHART procedure.

```
proc sort data=Old;
   by Prodtype;
run;

proc shewhart data=Old;
   irchart Diff*Sample /
      nochart
      outlimits=Baselim;
   by Prodtype;
run;
```

The purpose of this procedure step is simply to save the average moving range for each product type in the OUTLIMITS= data set Baselim, which is listed in Figure 17.208 (note that Prodtype is specified as a BY variable).

Figure 17.208 Values of $\bar{R}$ by Product Type

Control Limits By Product Type

Prodtype	_VAR_	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLI_	_MEAN_
M1	Diff	Sample	ESTIMATE	2	.002699796	3	-3.13258	0.13000
M2	Diff	Sample	ESTIMATE	2	.002699796	3	-1.77795	-0.06500
M3	Diff	Sample	ESTIMATE	2	.002699796	3	-3.22641	-0.19143

UCLI	_LCLR_	_R_	_UCLR_	_STDDEV_
3.39258	0	1.22714	4.00850	1.08753
1.64795	0	0.64429	2.10458	0.57098
2.84356	0	1.14154	3.72887	1.01166

To obtain a combined estimate of σ, you can use the MEANS procedure to average the average ranges in Baselim and then divide by the unbiasing constant d_2.

```
proc means data=Baselim noprint;
   var _r_;
   output out=Difflim (keep=_r_) mean=_r_;
run;

data Difflim;
   set Difflim;
   drop _r_;
   length _var_ _subgrp_ $ 8;
   _var_    = 'Diff';
   _subgrp_ = 'Sample';
   _mean_   = 0.0;
   _stddev_ = _r_ / d2(2);
   _limitn_ = 2;
   _sigmas_ = 3;
run;
```

The data set Difflim is structured for subsequent use by the SHEWHART procedure as an input LIMITS= data set. The variables in a LIMITS= data set provide pre-computed control limits or—as in this case—the parameters from which control limits are to be computed. These variables have reserved names that begin and end with the underscore character. Here, the variable _STDDEV_ saves the estimate of σ, and the variable _MEAN_ saves the mean of the differences from nominal. Recall that this mean is zero, since the nominal values are assumed to represent the process mean for each product type. The identifier variables _VAR_ and _SUBGRP_ record the names of the process and subgroup variables (these variables are critical in applications involving many product types). The variable _LIMITN_ is assigned a value of 2 to specify moving ranges of two consecutive measurements, and the variable _SIGMAS_ is assigned a value of 3 to specify 3σ limits. The data set Difflim is listed in Figure 17.209.

Figure 17.209 Estimates of Mean and Standard Deviation

Control Limit Parameters For Differences

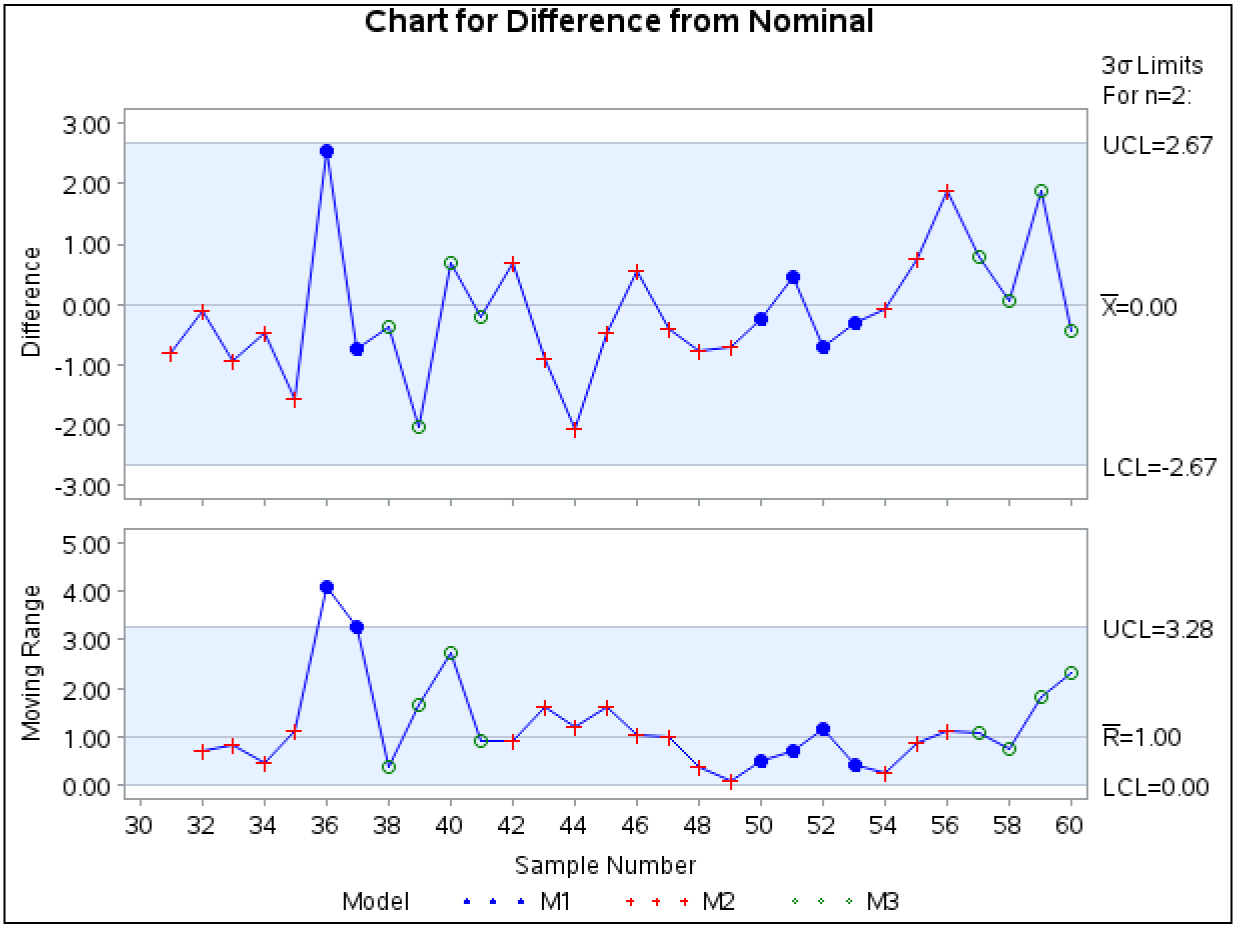
var	_subgrp_	_mean_	_stddev_	_limitn_	_sigmas_
Diff	Sample	0	0.89006	2	3

Now that the control limit parameters are saved in Difflim, diameters for an additional 30 parts (samples 31 to 60) are measured and saved in a SAS data set named New. You can construct short run control charts for this data by merging the measurements in New with the corresponding nominal values in Nomval, computing the differences from nominal, and then constructing the short run individual measurements and moving range charts.

```
proc sort data=new;
   by Prodtype;
run;

data new;
   format Diff 5.2 ;
   merge Nomval new(in = a);
      by Prodtype;
   if a;
   Diff = Diameter - Nominal;
   label Sample   = 'Sample Number'
         Prodtype = 'Model';
run;

proc sort data=new;
   by Sample;
run;

ods graphics off;
symbol1 v=dot    c=blue  h=3.0 pct;
symbol2 v=plus   c=red   h=3.0 pct;
symbol3 v=circle c=green h=3.0 pct;
title 'Chart for Difference from Nominal';
proc shewhart data=new limits=Difflim;
   irchart Diff*Sample=Prodtype / split    = '/';
   label Diff = 'Difference/Moving Range';
run;
```

The chart is displayed in Figure 17.210. Note that the product types are identified with symbol markers as requested by specifying Prodtype as a *symbol-variable*.

Figure 17.210 Short Run Control Chart

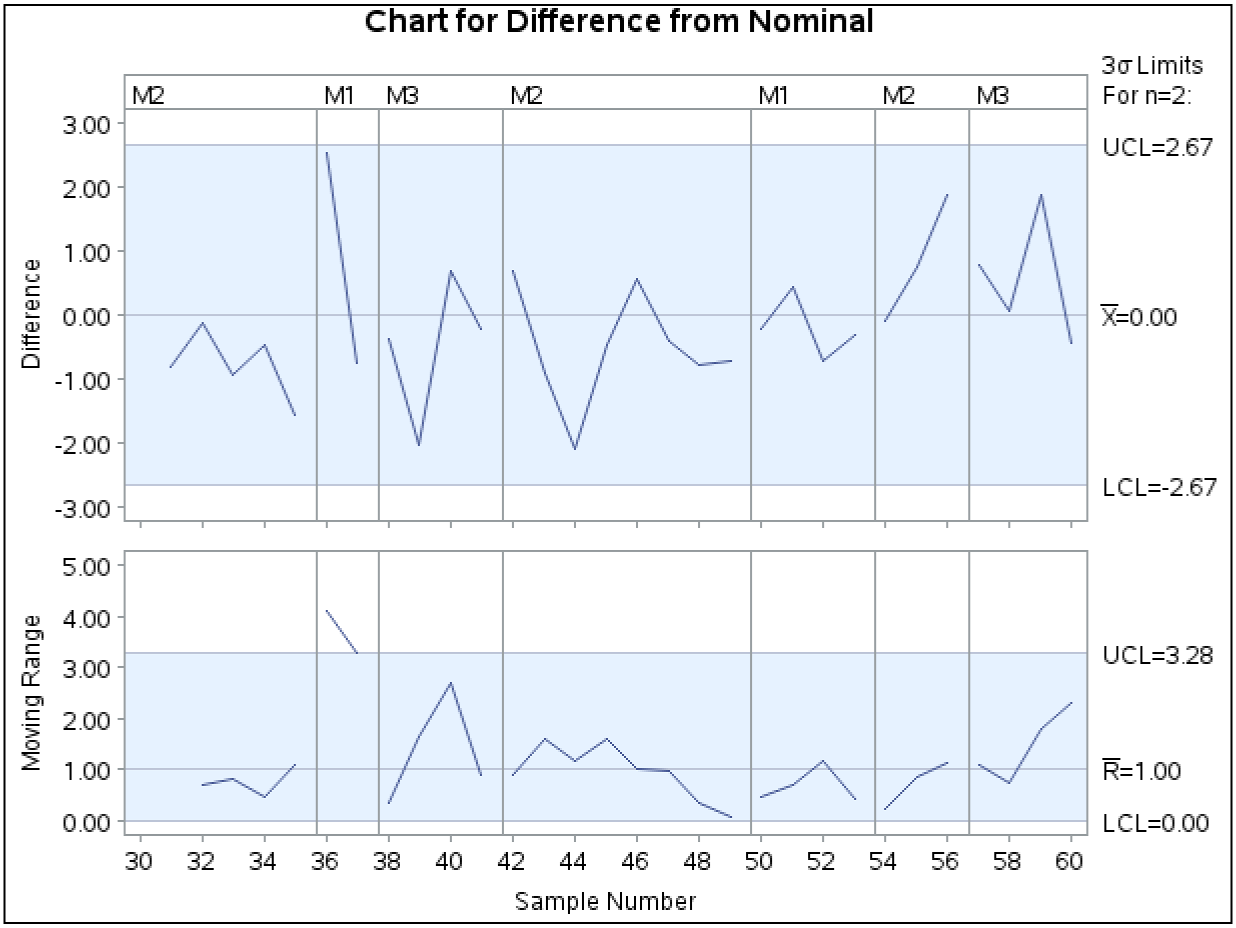


You can also identify the product types with a legend by specifying Prodtype as a _PHASE_ variable.

```
symbol h=3.0 pct;
title 'Chart for Difference from Nominal';
proc shewhart data=new (rename=(Prodtype=_phase_)) limits=Difflim;
   irchart Diff*Sample /
      readphases = all
      phaseref
      phasebreak
      phaselegend
      split      = '/';
   label Diff = 'Difference/Moving Range';
run;
```

The display is shown in Figure 17.211. Note that the PHASEBREAK option is used to suppress the connection of adjacent points in different phases (product types).

Figure 17.211 Identification of Product Types



In some applications, it may be useful to replace the moving range chart with a plot of the nominal values. You can do this with the TRENDVAR= option in the XCHART statement[27] provided that you reset the value of _LIMITN_ to 1 to specify a subgroup sample of size one.

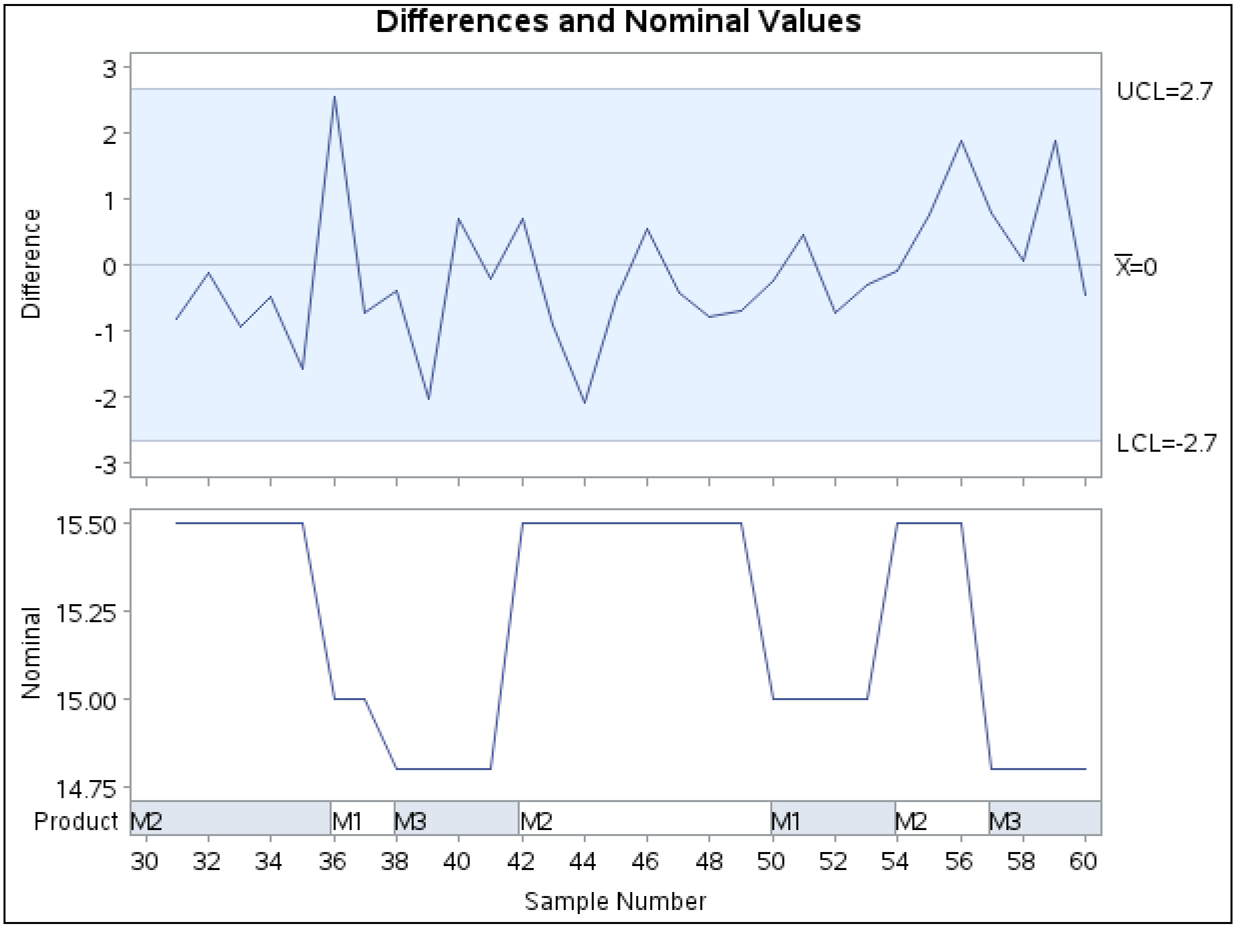
```
data Difflim;
   set Difflim;
   _var_    = 'Diameter';
   _limitn_ = 1;
run;

title 'Differences and Nominal Values';
proc shewhart data=new limits=Difflim;
   xchart Diameter*Sample (Prodtype) /
      nolimitslegend
      nolegend
      split         = '/'
      blockpos      = 3
      blocklabtype  = scaled
      blocklabelpos = left
      xsymbol       = xbar
```

[27]The TRENDVAR= option is not available in the IRCHART statement.

```
        trendvar      = Nominal;
     label Diameter   = 'Difference/Nominal'
           Prodtype   = 'Product';
run;
```

The display is shown in Figure 17.212. Note that you identify the product types by specifying Prodtype as a *block variable* enclosed in parentheses after the subgroup variable Sample. The BLOCKLABTYPE= option specifies that values of the block variable are to be scaled (if necessary) to fit the space available in the block legend. The BLOCKLABELPOS= option specifies that the label of the block variable is to be displayed to the left of the block legend.

Figure 17.212 Short Run Control Chart with Nominal Values



Testing for Constant Variances

The difference-from-nominal chart should be accompanied by a test that checks whether the variances for each product type are identical (homogeneous). Levene's test of homogeneity is particularly appropriate for short run applications because it is robust to departures from normality; refer to Snedecor and Cochran (1980). You can implement Levene's method by using the GLM procedure to construct a one-way analysis of variance for the absolute deviations of the diameters from averages within product types.

```
proc sort data=Old;
   by Prodtype;
run;

proc means data=Old noprint;
   var Diameter;
   by Prodtype;
   output out=Oldmean (keep=Prodtype diammean) mean=diammean;
run;

data Old;
   merge Old Oldmean;
      by Prodtype;
   absdev = abs( Diameter - diammean );
run;

proc means data=Old noprint;
  var absdev;
  by Prodtype;
  output out=stats n=n mean=mean css=css std=std;
run;

title;
proc glm data=Old outstat=glmout;
   class Prodtype;
   model absdev = Prodtype;
run;
```

A partial listing of the results is displayed in Figure 17.213. The large *p*-value (0.3386) indicates that the data do not reject the hypothesis of homogeneity.

Figure 17.213 Levene's Test of Variance Homogeneity

The GLM Procedure

Dependent Variable: absdev

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	1.02901063	0.51450532	1.13	0.3373
Error	27	12.27381243	0.45458565		
Corrected Total	29	13.30282306			

Standardizing Differences from Nominal

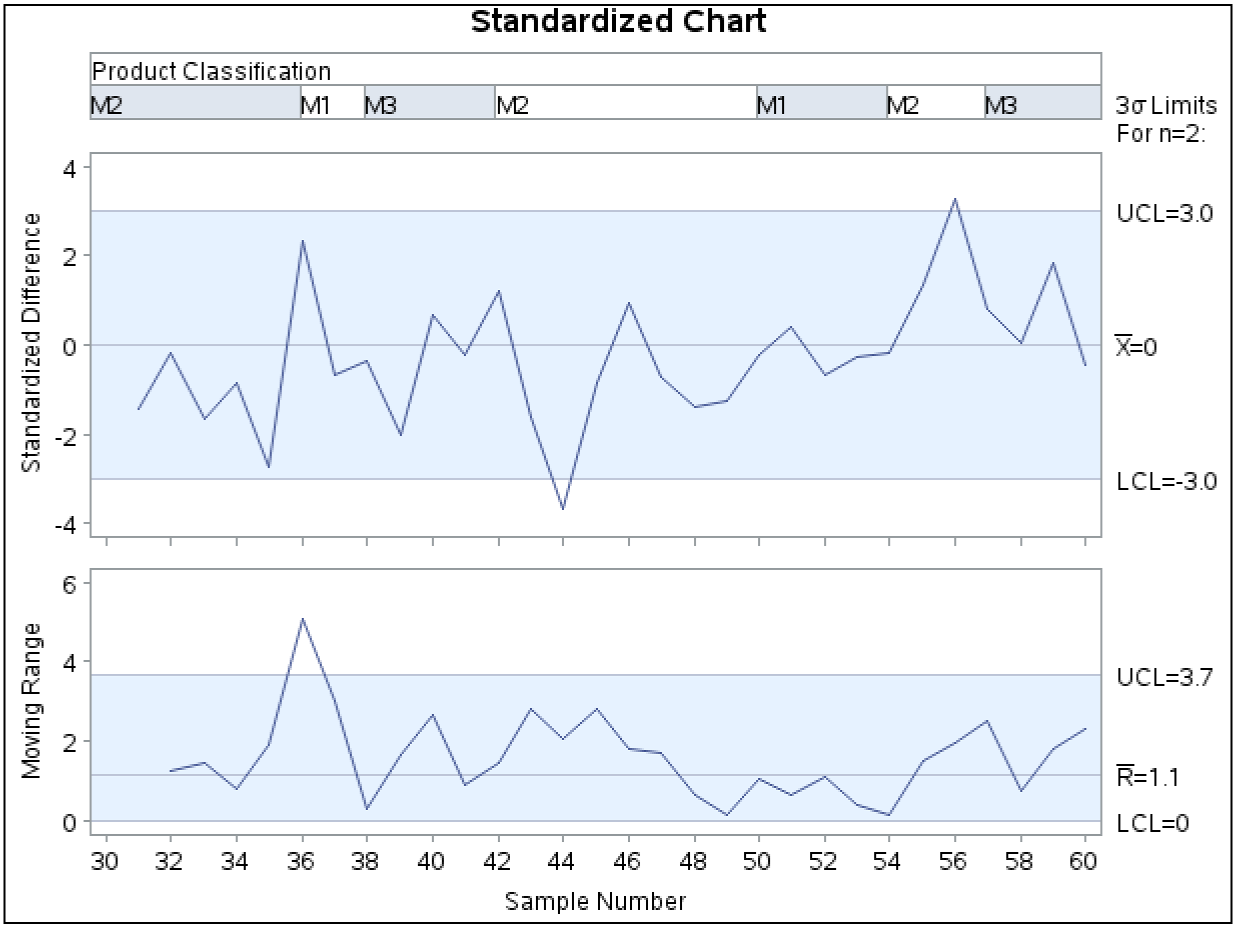
When the variances across product types are *not* constant, various authors recommend standardizing the differences from nominal and displaying them on a common chart with control limits at ± 3.

To illustrate this method, assume that the hypothesis of homogeneity is rejected for the differences in Old. Then you can use the product-specific estimates of σ in Baselim to standardize the differences from nominal in New and create the standardized chart as follows:

```
proc sort data=new;
   by Prodtype;
run;

data new;
   keep Sample Prodtype z Diff Diameter Nominal _stddev_;
   label Sample = 'Sample Number';
   format Diff 5.2 ;
   merge Baselim new(in = a);
      by Prodtype;
   if a;
   z = (Diameter - Nominal) / _stddev_ ;
run;

proc sort data=new;
   by Sample;
run;

title 'Standardized Chart';
proc shewhart data=new;
   irchart z*Sample (Prodtype) /
      blocklabtype  = scaled
      mu0           = 0
      sigma0        = 1
      split         = '/';
   label Prodtype = 'Product Classification'
         z = 'Standardized Difference/Moving Range';
run;
```

Note that the options MU0= and SIGMA= specify that the control limits for the standardized differences from nominal are to be based on the parameters $\mu = 0$ and $\sigma = 1$. The chart is displayed in Figure 17.214.

Figure 17.214 Standardized Difference Chart



Nonnormal Process Data

NOTE: See *Nonnormal Process Data* in the SAS/QC Sample Library.

A number of authors have pointed out that Shewhart charts for subgroup means work well whether the measurements are normally distributed or not.[28] On the other hand, the interpretation of standard control charts for individual measurements (X charts) is affected by departures from normality.

In situations involving a large number of measurements, it may be possible to subgroup the data and construct an $\bar{X}$ chart instead of an X chart. However, the measurements should not be subgrouped arbitrarily for this purpose.[29] If subgrouping is not possible, two alternatives are to transform the data to normality (preferably with a simple transformation such as the log transformation) or modify the usual limits based on a suitable model for the data distribution.

The second of these alternatives is illustrated here with data from a study conducted by a service center. The time taken by staff members to answer the phone was measured, and the delays were saved as values of a variable named Time in a SAS data set named Calls. A partial listing of Calls is shown in Figure 17.215.

[28] Refer to Schilling and Nelson (1976) and Wheeler (1991).
[29] Refer to Wheeler and Chambers (1986) for a discussion of subgrouping.

Figure 17.215 Answering Times from the Data Set Calls

Recnum	Time
1	3.233
2	3.110
3	3.136
4	2.899
5	2.838
6	2.459
7	3.716
8	2.740
9	2.487
10	2.635
11	2.676
12	2.905
13	3.431
14	2.663
15	3.437
16	2.823
17	2.596
18	2.633
19	3.235
20	2.701
21	3.202
22	2.725
23	3.151
24	2.464
25	2.662
26	3.188
27	2.640
28	2.541
29	3.033
30	2.993
31	2.636
32	2.481
33	3.191
34	2.662
35	2.967
36	3.300
37	2.530
38	2.777
39	3.353
40	3.614
41	4.288
42	2.442
43	2.552
44	2.613
45	2.731
46	2.780
47	3.588
48	2.612

Figure 17.215 *continued*

Recnum	Time
49	2.579
50	2.871

Creating a Preliminary Individual Measurements Chart

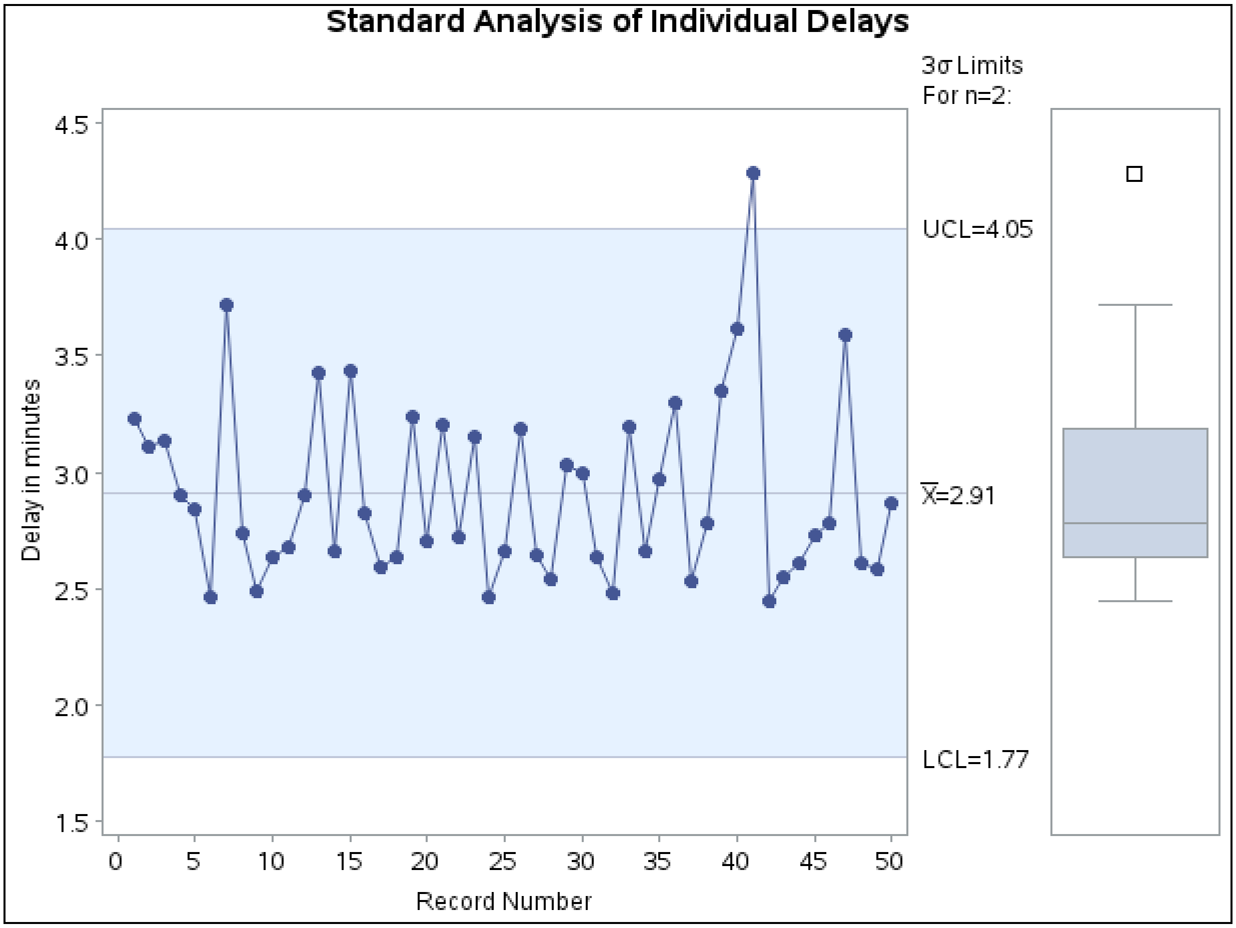
As a first step, the delays were analyzed using an *X* chart created with the following statements. The chart is displayed in Figure 17.216.

```
ods graphics off;
title 'Standard Analysis of Individual Delays';
proc shewhart data=Calls;
   irchart Time * Recnum /
      rtmplot   = schematic
      outlimits = delaylim
      nochart2 ;
   label Recnum = 'Record Number'
         Time   = 'Delay in minutes' ;
run;
```

You may be inclined to conclude that the 41st point signals a special cause of variation. However, the box plot in the right margin (requested with the RTMPLOT= option) indicates that the distribution of delays is skewed. Thus, the reason that the measurements are grouped well within the control limits is that the limits are incorrect and not that the process is too good for the limits.

NOTE: This example assumes the process is in statistical control; otherwise, the box plot could not be interpreted as a representation of the process distribution. You can check the assumption of normality with goodness-of-fit tests by using the CAPABILITY procedure, as shown in the statements that follow.

Figure 17.216 Standard Control Limits for Delays



Calculating Probability Limits

The OUTLIMITS= option saves the control limits from the chart in Figure 17.216 in a SAS data set named DELAYLIM, which is listed in Figure 17.217.

Figure 17.217 Control Limits for Standard Chart from the Data Set Calls

VAR	_SUBGRP_	_TYPE_	_LIMITN_	_ALPHA_	_SIGMAS_	_LCLI_	_MEAN_	_UCLI_	_STDDEV_
Time	Recnum	ESTIMATE	2	.002699796	3	1.77008	2.91038	4.05068	0.38010

The control limits can be replaced with the corresponding percentiles from a fitted lognormal distribution. The equation for the lognormal density function is

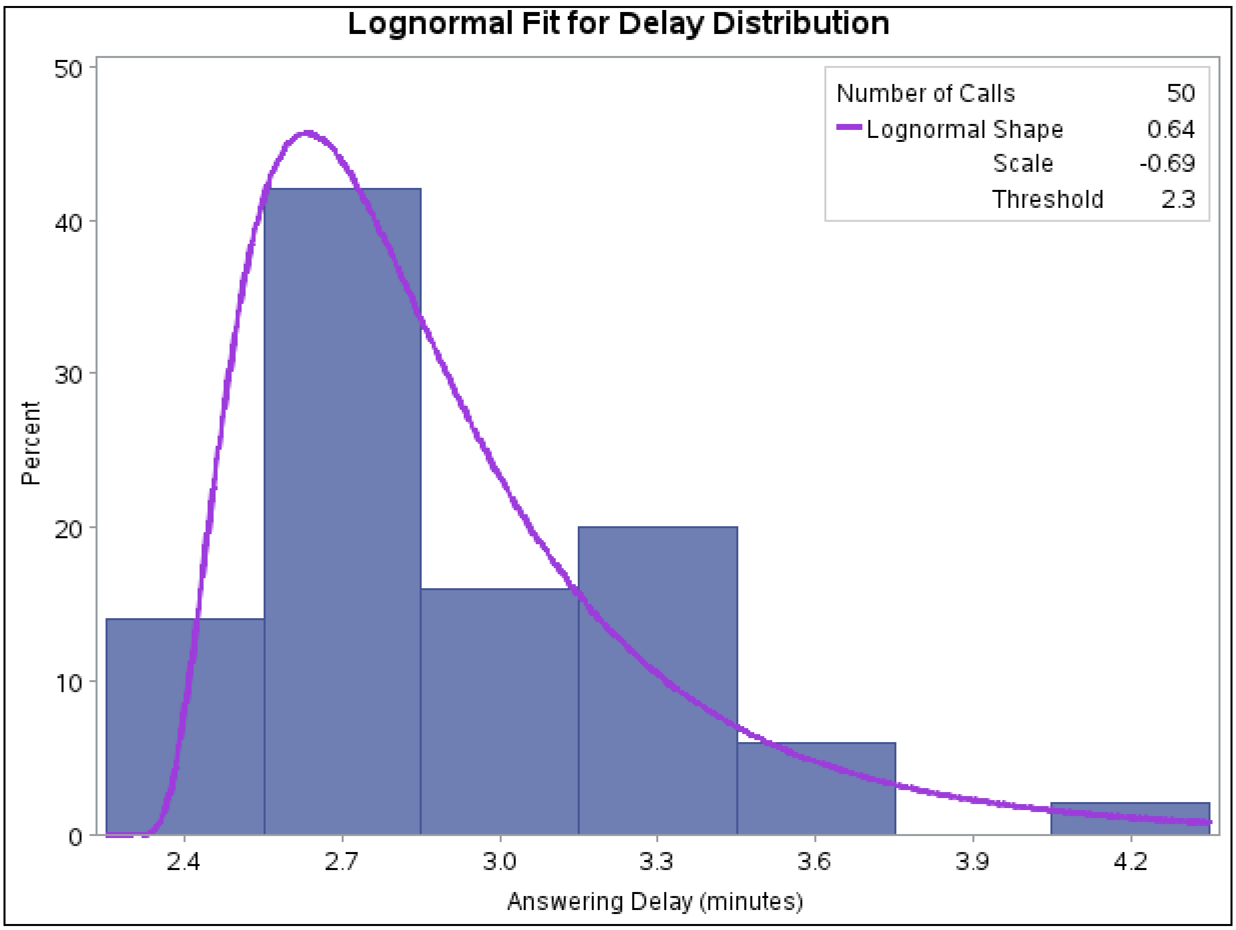
$$f(x) = \frac{1}{x\sqrt{2\pi}\sigma} \exp\left(-\frac{(\log(x)-\zeta)^2}{2\sigma^2}\right) \quad x > 0$$

where σ denotes the shape parameter and ζ denotes the scale parameter.

The following statements use the CAPABILITY procedure to fit a lognormal model and superimpose the fitted density on a histogram of the data, shown in Figure 17.218:

```
title 'Lognormal Fit for Delay Distribution';
proc capability data=Calls noprint;
   histogram Time /
      lognormal(threshold=2.3 w=2)
      outfit = Lnfit
      nolegend ;
   inset n = 'Number of Calls'
      lognormal( sigma = 'Shape' (4.2)
                 zeta  = 'Scale' (5.2)
                 theta ) / pos    = ne;
   label Time  = 'Answering Delay (minutes)';
run;
```

Figure 17.218 Distribution of Delays



Parameters of the fitted distribution and results of goodness-of-fit tests are saved in the data set LNFIT, which is listed in Figure 17.219. The large *p*-values for the goodness-of-fit tests are evidence that the lognormal model provides a good fit.

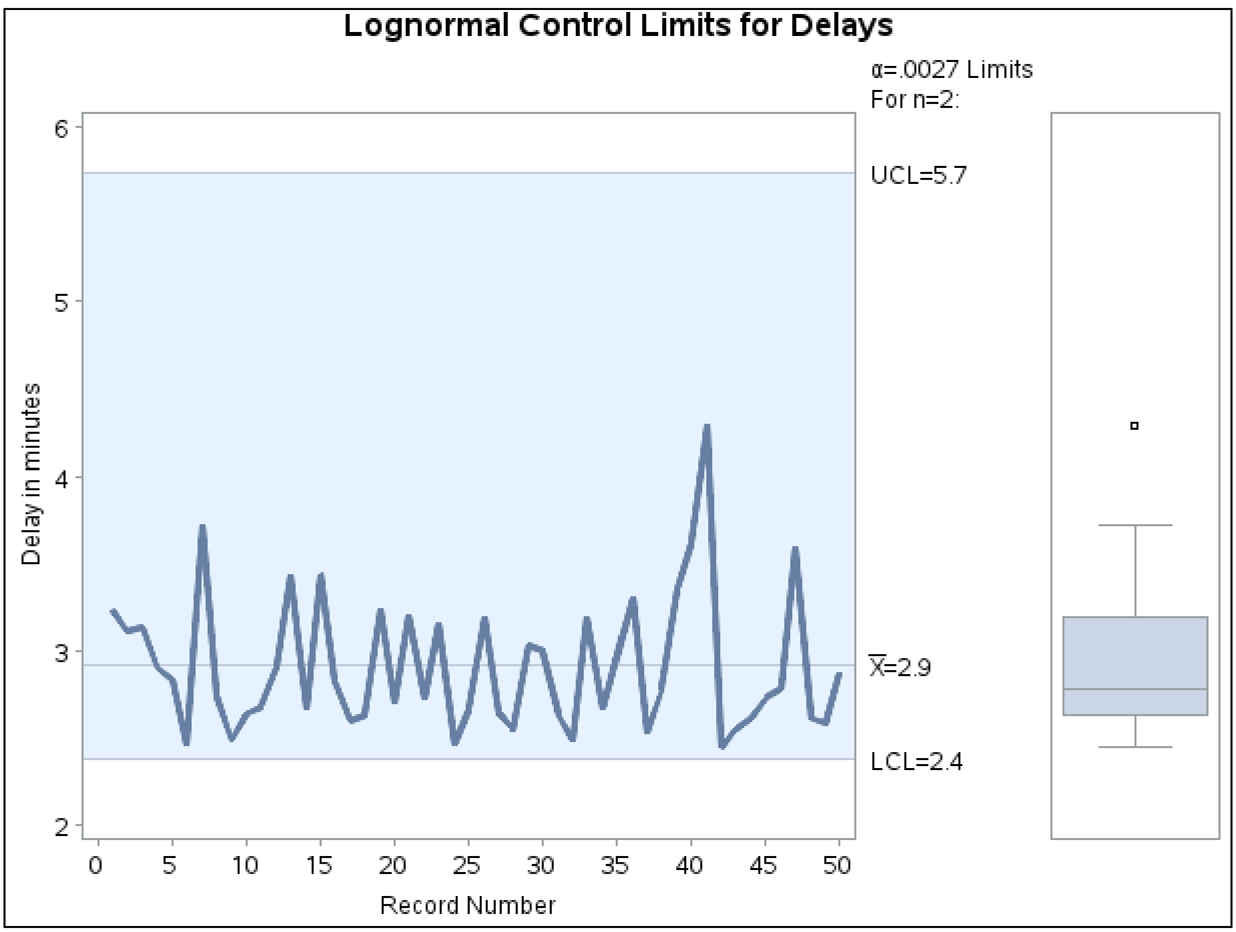
Figure 17.219 Parameters of Fitted Lognormal Model in the Data Set LNFIT

VAR	_CURVE_	_LOCATN_	_SCALE_	_SHAPE1_	_MIDPTN_	_ADASQ_	_ADP_	_CVMWSQ_	_CVMP_	_KSD_	_KSP_
Time	LNORMAL	2.3	-0.68910	0.64110	4.2	0.34854	0.47465	0.058737	0.40952	0.092223	0.15

The following statements replace the control limits in DELAYLIM with limits computed from percentiles of the fitted lognormal model. The 100αth percentile of the lognormal distribution is $P_\alpha = \exp(\sigma\Phi^{-1}(\alpha) + \zeta)$, where Φ^{-1} denotes the inverse standard normal cumulative distribution function. The SHEWHART procedure constructs an X chart with the modified limits, displayed in Figure 17.220.

```
data delaylim;
   merge delaylim Lnfit;
   drop _sigmas_ ;
   _lcli_ = _locatn_ + exp(_scale_+probit(0.5*_alpha_)*_shape1_);
   _ucli_ = _locatn_ + exp(_scale_+probit(1-.5*_alpha_)*_shape1_);
   _mean_ = _locatn_ + exp(_scale_+0.5*_shape1_*_shape1_);
run;

title 'Lognormal Control Limits for Delays';
proc shewhart data=Calls limits=delaylim;
   irchart Time*Recnum /
      rtmplot   = schematic
      nochart2 ;
   label Recnum = 'Record Number'
         Time   = 'Delay in minutes' ;
run;
```

Figure 17.220 Adjusted Control Limits for Delays



Clearly the process is in control, and the control limits (particularly the lower limit) are appropriate for the data. The particular probability level $\alpha = 0.0027$ associated with these limits is somewhat immaterial, and other values of α such as 0.001 or 0.01 could be specified with the ALPHA= option in the original IRCHART statement.

Multivariate Control Charts

NOTE: See *Creating Multivariate Control Charts* in the SAS/QC Sample Library.

In many industrial applications, the output of a process characterized by p variables that are measured simultaneously. Independent variables can be charted individually, but if the variables are correlated, a multivariate chart is needed to determine whether the process is in control.

Many types of multivariate control charts have been proposed; refer to Alt (1985) for an overview. Denote the ith measurement on the jth variable as X_{ij} for $i = 1, 2, \ldots, n$, where n is the number of measurements, and $j = 1, 2, \ldots, p$. Standard practice is to construct a chart for a statistic T_i^2 of the form

$$T_i^2 = (\mathbf{X}_i - \bar{\mathbf{X}}_n)' \mathbf{S}_n^{-1} (\mathbf{X}_i - \bar{\mathbf{X}}_n)$$

where

$$\bar{X}_j = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} X_{ij} \quad , \quad \mathbf{X}_i = \begin{bmatrix} X_{i1} \\ X_{i2} \\ \vdots \\ X_{ip} \end{bmatrix}, \quad \bar{\mathbf{X}}_n = \begin{bmatrix} \bar{X}_1 \\ \bar{X}_2 \\ \vdots \\ \bar{X}_p \end{bmatrix}$$

and

$$\mathbf{S}_n = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{X}_i - \bar{\mathbf{X}}_n)(\mathbf{X}_i - \bar{\mathbf{X}}_n)'$$

It is assumed that $\mathbf{X}_i$ has a p-dimensional multivariate normal distribution with mean vector — $= (\mu_1 \mu_2 \cdots \mu_p)'$ and covariance matrix $\mathbf{\Sigma}$ for $i = 1, 2, \ldots, n$. Depending on the assumptions made about the parameters, a χ^2, Hotelling T^2, or beta distribution is used for T_i^2, and the percentiles of this distribution yield the control limits for the multivariate chart.

In this example, a multivariate control chart is constructed using a beta distribution for T_i^2. The beta distribution is appropriate when the data are individual measurements (rather than subgrouped measurements) and when — and $\mathbf{\Sigma}$ are estimated from the data being charted. In other words, this example illustrates a start-up phase chart where the control limits are determined from the data being charted.

Calculating the Chart Statistic

In this situation, it was shown by Gnanadesikan and Kettenring (1972), using a result of Wilks (1962), that T_i^2 is exactly distributed as a multiple of a variable with a beta distribution. Specifically,

$$T_i^2 \sim \frac{(n-1)^2}{n} B\left(\frac{p}{2}, \frac{n-p-1}{2}\right)$$

Tracy, Young, and Mason (1992) used this result to derive initial control limits for a multivariate chart based on three quality measures from a chemical process in the start-up phase: percent of impurities, temperature, and concentration. The remainder of this section describes the construction of a multivariate control chart using their data, which are given here by the data set Startup.

```
data Startup;
   input Sample Impure Temp Conc;
   label Sample = 'Sample Number'
         Impure = 'Impurities'
         Temp   = 'Temperature'
         Conc   = 'Concentration' ;
   datalines;
 1  14.92  85.77  42.26
 2  16.90  83.77  43.44
 3  17.38  84.46  42.74
 4  16.90  86.27  43.60
 5  16.92  85.23  43.18
 6  16.71  83.81  43.72
 7  17.07  86.08  43.33
 8  16.93  85.85  43.41
 9  16.71  85.73  43.28
10  16.88  86.27  42.59
11  16.73  83.46  44.00
12  17.07  85.81  42.78
13  17.60  85.92  43.11
14  16.90  84.23  43.48
;
```

In preparation for the computation of the control limits, the sample size is calculated and parameter variables are defined.

```
proc means data=Startup noprint ;
   var Impure Temp Conc;
   output out=means n=n;
run;

data Startup;
   if _n_ = 1 then set means;
   set Startup;
   p        = 3;
   _subn_   = 1;
   _limitn_ = 1;
run;
```

Next, the PRINCOMP procedure is used to compute the principal components of the variables and save them in an output data set named Prin.

```
proc princomp data=Startup out=Prin outstat=scores std cov;
   var Impure Temp Conc;
run;
```

The following statements compute T_i^2 and its exact control limits, using the fact that T_i^2 is the sum of squares of the principal components.[30] Note that these statements create several special SAS variables so that the data set Prin can subsequently be read as a TABLE= input data set by the SHEWHART procedure. These special variables begin and end with an underscore character. The data set Prin is listed in Figure 17.221.

```
data Prin (rename=(tsquare=_subx_));
   length _var_ $ 8 ;
   drop prin1 prin2 prin3 _type_ _freq_;
   set Prin;
   comp1   = prin1*prin1;
   comp2   = prin2*prin2;
   comp3   = prin3*prin3;
   tsquare = comp1 + comp2 + comp3;
   _var_   = 'tsquare';
   _alpha_ = 0.05;
   _lclx_  = ((n-1)*(n-1)/n)*betainv(_alpha_/2, p/2, (n-p-1)/2);
   _mean_  = ((n-1)*(n-1)/n)*betainv(0.5, p/2, (n-p-1)/2);
   _uclx_  = ((n-1)*(n-1)/n)*betainv(1-_alpha_/2, p/2, (n-p-1)/2);
   label tsquare = 'T Squared'
         comp1   = 'Comp 1'
         comp2   = 'Comp 2'
         comp3   = 'Comp 3';
run;
```

[30] Refer to Jackson (1980).

Figure 17.221 The Data Set Prin

T2 Chart For Chemical Example

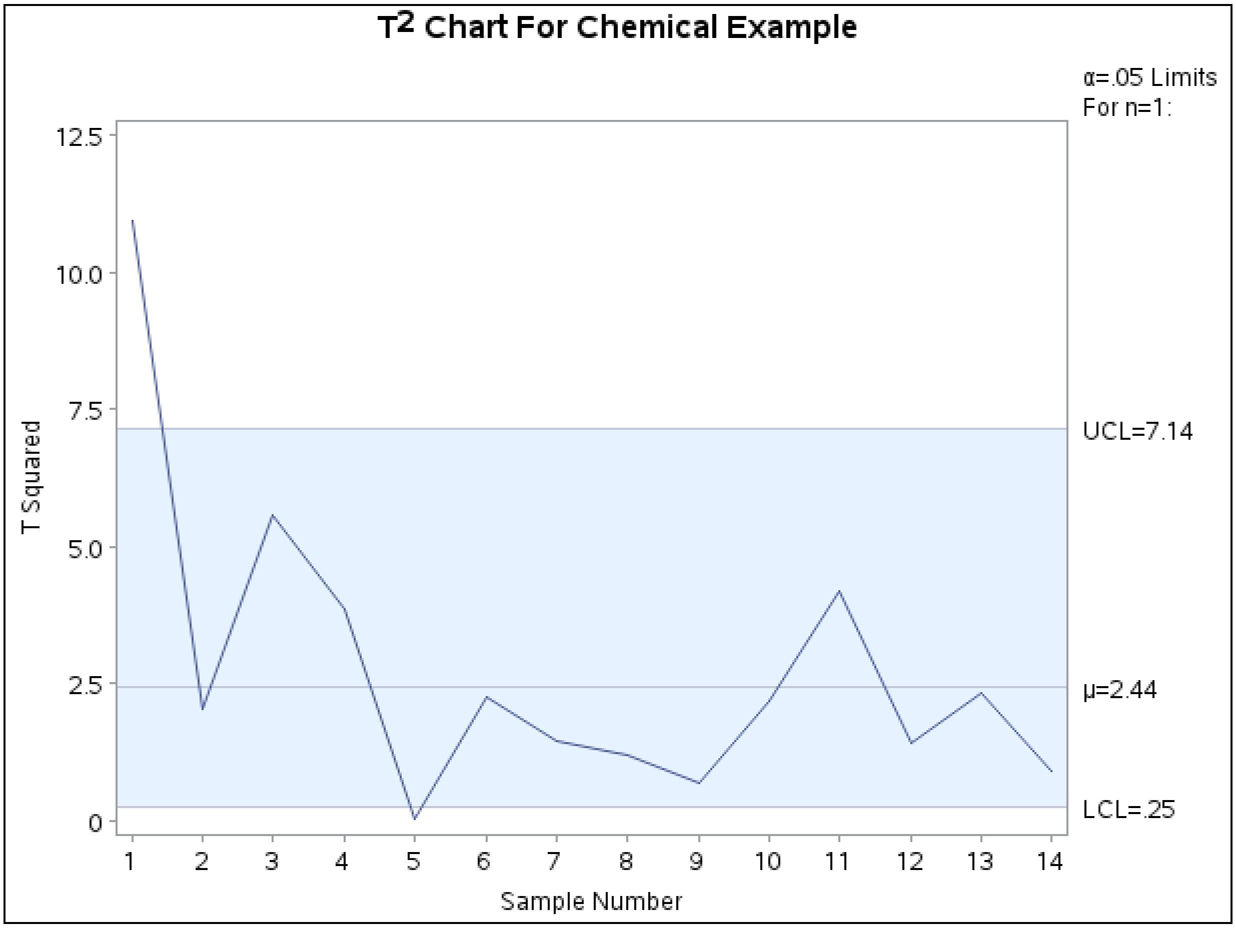
var	n	Sample	Impure	Temp	Conc	p	_subn_	_limitn_	comp1	comp2	comp3
tsquare	14	1	14.92	85.77	42.26	3	1	1	0.79603	10.1137	0.01606
tsquare	14	2	16.90	83.77	43.44	3	1	1	1.84804	0.0162	0.17681
tsquare	14	3	17.38	84.46	42.74	3	1	1	0.33397	0.1538	5.09491
tsquare	14	4	16.90	86.27	43.60	3	1	1	0.77286	0.3289	2.76215
tsquare	14	5	16.92	85.23	43.18	3	1	1	0.00147	0.0165	0.01919
tsquare	14	6	16.71	83.81	43.72	3	1	1	1.91534	0.0645	0.27362
tsquare	14	7	17.07	86.08	43.33	3	1	1	0.58596	0.4079	0.44146
tsquare	14	8	16.93	85.85	43.41	3	1	1	0.29543	0.1729	0.73939
tsquare	14	9	16.71	85.73	43.28	3	1	1	0.23166	0.0001	0.44483
tsquare	14	10	16.88	86.27	42.59	3	1	1	1.30518	0.0004	0.86364
tsquare	14	11	16.73	83.46	44.00	3	1	1	3.15791	0.0274	0.98639
tsquare	14	12	17.07	85.81	42.78	3	1	1	0.43819	0.0823	0.87976
tsquare	14	13	17.60	85.92	43.11	3	1	1	0.41494	1.6153	0.30167
tsquare	14	14	16.90	84.23	43.48	3	1	1	0.90302	0.0001	0.00010

subx	_alpha_	_lclx_	_mean_	_uclx_
10.9257	0.05	0.24604	2.44144	7.13966
2.0410	0.05	0.24604	2.44144	7.13966
5.5827	0.05	0.24604	2.44144	7.13966
3.8640	0.05	0.24604	2.44144	7.13966
0.0372	0.05	0.24604	2.44144	7.13966
2.2534	0.05	0.24604	2.44144	7.13966
1.4354	0.05	0.24604	2.44144	7.13966
1.2077	0.05	0.24604	2.44144	7.13966
0.6766	0.05	0.24604	2.44144	7.13966
2.1692	0.05	0.24604	2.44144	7.13966
4.1717	0.05	0.24604	2.44144	7.13966
1.4003	0.05	0.24604	2.44144	7.13966
2.3320	0.05	0.24604	2.44144	7.13966
0.9032	0.05	0.24604	2.44144	7.13966

You can now use the data set Prin as input to the SHEWHART procedure to create the multivariate control chart displayed in Figure 17.222.

```
ods graphics off;
title 'T' m=(+0,+0.5) '2'
          m=(+0,-0.5) ' Chart For Chemical Example';
proc shewhart table=Prin;
   xchart tsquare*Sample /
      xsymbol  = mu
      nolegend ;
run;
```

Figure 17.222 Multivariate Control Chart for Chemical Process



The methods used in this example easily generalize to other types of multivariate control charts. You can create charts using the χ^2 and F distributions by using the appropriate CINV or FINV function in place of the BETAINV function. For details, refer to Alt (1985), Jackson (1980, 1991), and Ryan (1989).

Examining the Principal Component Contributions

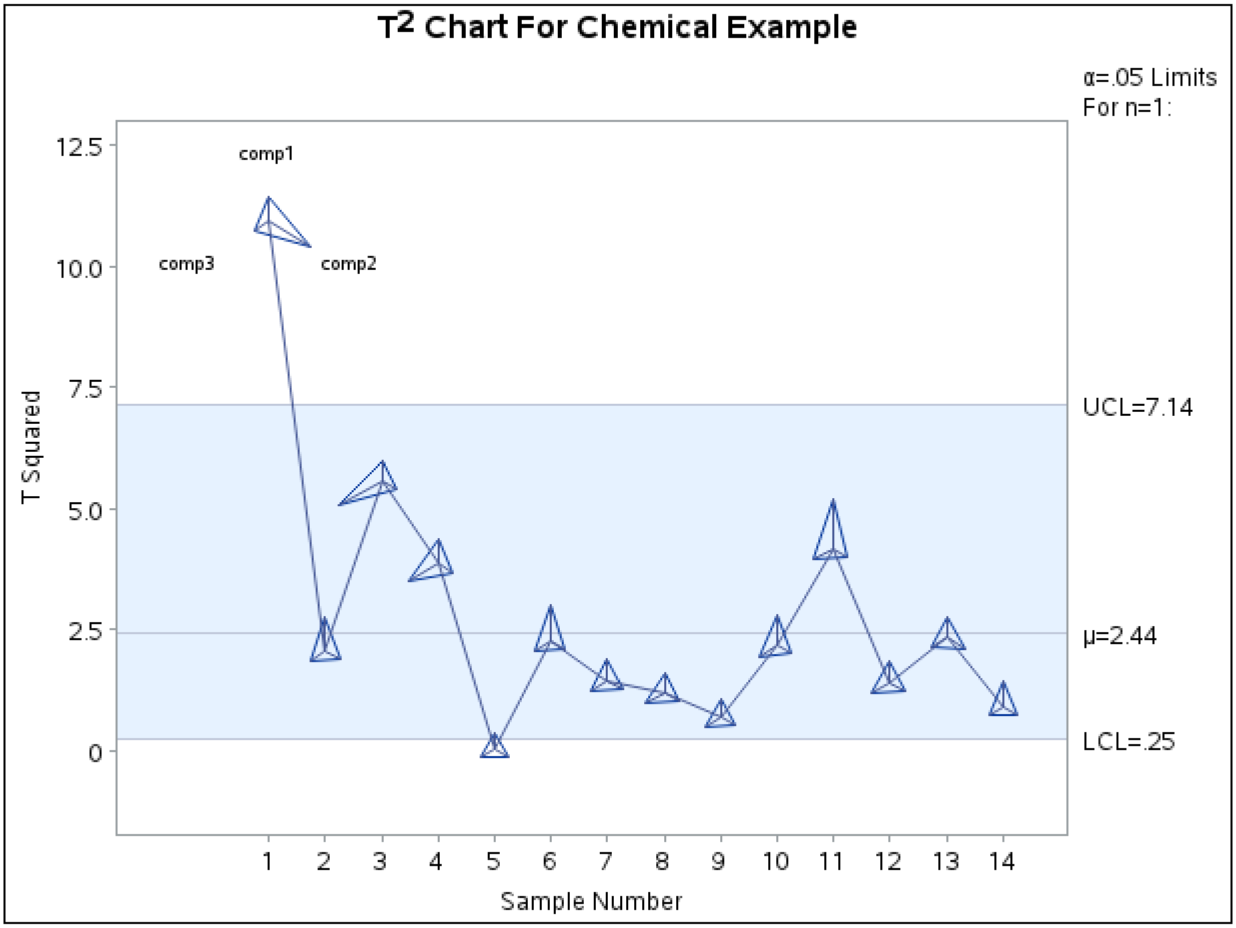
You can use the *star options* in the SHEWHART procedure to superimpose points on the chart with stars whose vertices represent standardized values of the squares of the three principal components used to determine T_i^2.

```
title 'T' m=(+0,+0.5) '2'
          m=(+0,-0.5) ' Chart For Chemical Example';
symbol value=none;
proc shewhart table=Prin;
   xchart tsquare*Sample /
      starvertices  = (comp1 comp2 comp3)
      startype      = wedge
      starlegend    = none
      starlabel     = first
      staroutradius = 4
      npanelpos     = 14
```

```
        xsymbol        = mu
        nolegend ;
run;
```

The chart is displayed in Figure 17.223. In situations where the principal components have a physical interpretation, the star chart can be a helpful diagnostic for determining the relative contributions of the different components.

Figure 17.223 Multivariate Control Chart Displaying Principal Components



For more information about star charts, see the section "Displaying Auxiliary Data with Stars" on page 2042, or consult the entries for the STARVERTICES= and related options in "Dictionary of Options: SHEWHART Procedure" on page 1946.

Principal components are not the only approach that can be used to interpret multivariate control charts. This problem has recently been studied by a number of authors, including Doganaksoy, Faltin, and Tucker (1991), Hawkins (1991, 1993), and Mason, Tracy, and Young (1993).

Interactive Control Charts: SHEWHART Procedure

Overview: Interactive Control Charts

This section describes two approaches for creating an interactive control chart which enables an end user to "drill down" into subgroup data points and display information not contained in the chart itself. For example, the end user might want to be able to click on a subgroup to

- list the individual measurements in the subgroup
- diagnose an out-of-control point by viewing a Pareto chart of the most common problems affecting the process
- view a list of recommended corrective actions
- trace the raw materials used to manufacture a batch of product

The two approaches for creating interactive control charts are as follows:

- saving graphics coordinate data from control charts for use in creating SAS/AF applications
- associating Uniform Resource Locators (URLs) with subgroups to produce "clickable" control charts in HTML

The options described in this section can be specified in all the chart statements available in the SHEWHART procedure.

Details: Interactive Control Charts

Saving Graphics Coordinates in a Control Chart

You can specify an WEBOUT= data set in any chart statement to save graphics coordinate information for a control chart. The WEBOUT= data set is an extension of the OUTTABLE= data set, which contains the subgroup summary statistics, control limits and related information found in an OUTTABLE= data set, as well as coordinate data. The additional coordinate variables are listed in Table 17.97.

Table 17.97 WEBOUT= Data Set

Variable	Description
_X1_	x-coordinate of lower left corner of primary chart subgroup bounding box
_Y1_	y-coordinate of lower left corner of primary chart subgroup bounding box
_X2_	x-coordinate of upper right corner of primary chart subgroup bounding box
_Y2_	y-coordinate of upper right corner of primary chart subgroup bounding box
_X*n*_	x-coordinate for point *n* of the subgroup shape
_Y*n*_	y-coordinate for point *n* of the subgroup shape

Table 17.97 (continued)

Variable	Description
_X1_2_	x-coordinate of lower left corner of secondary chart subgroup bounding box
_Y1_2_	y-coordinate of lower left corner of secondary chart subgroup bounding box
_X2_2_	x-coordinate of upper right corner of secondary chart subgroup bounding box
_Y2_2_	y-coordinate of upper right corner of secondary chart subgroup bounding box
SHAPE	shape of primary chart subgroup bounding area
NXY	number of points defining primary chart subgroup bounding area
GRAPH	name of primary chart graphics entry
GRAPH2	name of secondary chart graphics entry
DXMIN	value of lowest major tick mark on horizontal axis
DXMAX	value of highest major tick mark on horizontal axis
XMIN	x-coordinate of lowest major tick mark on horizontal axis
XMAX	x-coordinate of highest major tick mark on horizontal axis
DYMIN	value of lowest major tick mark on vertical axis
DYMAX	value of highest major tick mark on vertical axis
YMIN	y-coordinate of lowest major tick mark on vertical axis
YMAX	y-coordinate of highest major tick mark on vertical axis
XMIN2	x-coordinate of lowest major tick mark on secondary chart horizontal axis
XMAX2	x-coordinate of highest major tick mark on secondary chart horizontal axis
DYMIN2	value of lowest major tick mark on secondary chart vertical axis
DYMAX2	value of highest major tick mark on secondary chart vertical axis
YMIN2	y-coordinate of lowest major tick mark on secondary chart vertical axis
YMAX2	y-coordinate of highest major tick mark on secondary chart vertical axis

You can use the coordinate data saved in the WEBOUT= data set to create a “clickable” control chart in a SAS/AF application. The variables _X1_, _Y1_, _X2_ and _Y2_ contain the coordinates of the lower left and upper right corners of a rectangular *bounding box* associated with each subgroup on the primary chart. This box defines the clickable area associated with the subgroup when the chart is incorporated into a SAS/AF application. It contains the symbol used to plot the subgroup data, or the junction of line segments representing the subgroup if no plotting symbol is used. The variables _X1_2_, _Y1_2_, _X2_2_ and _Y2_2_ contain coordinates of the corners of subgroup bounding boxes for a secondary chart.

If you use the BOXCHART statement, each subgroup is represented by a box-and-whisker plot rather than a single symbol. The subgroup’s bounding box is defined by the sides of the box-and-whisker plot and its lower and upper quartiles, regardless of the BOXSTYLE= value in effect.

If you specify the STARVERTICES= option, each subgroup is represented by a polygon or star with a vertex corresponding to each of the STARVERTICES= variables. The clickable area for a subgroup is the polygon with these vertices, regardless of the STARTYPE= value specified. In the WEBOUT= data set the value of the _SHAPE_ variable is POLY and the _NXY_ variable contains the number of vertices in the polygon. The variables _X*n*_ and _Y*n*_, where $n = 1$ to the value of _NXY_, contain the coordinates of the vertices of a subgroup’s polygon. When the STARVERTICES= option is not used, the value of _SHAPE_ is always RECT and the value of _NXY_ is always 2.

When a control chart spans multiple panels (pages), the panels reside in separate SAS graphics entries. The _GRAPH_ character variable records the name of the graphics entry containing the panel on which a given subgroup is plotted. This is the same name that appears in the PROC GREPLAY menu. When the SEPARATE option is used, primary and secondary charts are displayed on different graphics entries. The _GRAPH2_ variable records the name of the graphics entry containing the secondary chart panel where a subgroup appears. When the SEPARATE option is not used, the values of _GRAPH_ and _GRAPH2_ will be the same for a given subgroup.

The variables _DXMIN_, _DXMAX_, _XMIN_ and _XMAX_ provide the data values and graphics coordinates associated with the lowest and highest major tick marks on the horizontal (subgroup) axis. The variables _DYMIN_, _DYMAX_, _YMIN_ and _YMAX_ provide the analogous values for the vertical axis. Through a simple linear transformation in your SAS/AF application you can use this information to convert from percent screen units to “data” units and vice versa.

The variables _XMIN2_ and _XMAX2_ contain the graphics coordinates associated with the lowest and highest major tick marks on the horizontal axis of a secondary chart. No variables for the corresponding data values are required, since they are always identical to those for the primary chart.

The variables _DYMIN2_, _DYMAX2_, _YMIN2_ and _YMAX2_ contain the data and coordinate values for the lowest and highest tick marks on the vertical axis of a secondary chart. A SAS/AF program receives the (x,y) coordinates for the location of the cursor when the user clicks on a subgroup data point. The application can determine whether (x,y) lies within any of the boxes whose coordinates are saved in the WEBOUT= data set. If so, the program can determine which subgroup was selected on the primary or secondary chart and can check the _TESTS_ and _TESTS2_ variables included in the WEBOUT= data set to determine whether an out-of-control condition has been signaled.

Notes:

1. Graphics coordinates are scaled in percent screen units from 0 to 100, where (0,0) represents the lower-left corner of the screen and (100,100) represents the upper-right corner of the screen. Because SAS/AF applications define the origin of the vertical axis at the top of the screen, it will be necessary to subtract the y-coordinates from 100 in your SCL program.

2. The variables _X1_2_, _Y1_2_, _X2_2_, _Y2_2_, _GRAPH2_, _XMIN2_, _XMAX2_, _YMIN2_, _YMAX2_, _DYMIN2_ and _DYMAX2_ appear in the WEBOUT= data set only when a secondary chart is produced. A secondary chart is produced by the IRCHART, MRCHART, XRCHART and XSCHART statements and by the BOXCHART, MCHART and XCHART statements when the TRENDVAR= option is specified.

3. When the subgroup variable is a character variable, the value of _DXMIN_ is zero and the value of _DXMAX_ is the number of subgroups in the input data set minus one.

4. A bounding box circumscribes a point displayed on a chart and its dimensions depend on the size of the symbol marker used to display the point. If no symbol marker is specified, a small default size is used for the box. If a large number of subgroups are displayed on a panel, the subgroup symbols may overlap, so it is possible for a user to inadvertently select more than one point.

Associating URLs with Subgroups in HTML

You can use the Output Delivery System (ODS) to produce an HTML file containing a control chart created by the SHEWHART procedure. The HTML= option provides a way to associate Uniform Resource Locators (URLs) with subgroups plotted on a control chart. It specifies a variable in the input data set containing HTML syntax providing the URLs to be associated with different subgroups. The HTML= variable can be a character variable or a numeric variable with an associated character format.

The following statements generate an $\bar{X}$ chart that is saved to a GIF file and included in an HTML file. The formatted values of the numeric HTML= variable Web specify URLs that link subgroups in the input data set to various web pages.

```
goptions target = gif;
ods html body = "example1.html";
proc format;
   value webfmt
   1='href="http://www.sas.com/"'
   2='href="http://www.sas.com/service/techsup/faq/qc/shewproc.html"'
   3='href="http://www.sas.com/rnd/app/qc.html"'
   4='href="http://www.sas.com/rnd/app/qc/qcnew.html"'
   5='href="http://www.sas.com/rnd/app/qc/qc.html"'
   ;

data wafers;
   format Web webfmt.;
   input Batch Web @;
   do i=1 to 5;
      input Diameter @;
      output;
   end;
   drop i;
   datalines;
 1  1  35.00 34.99 34.99 34.98 35.00
 2  1  35.00 34.99 34.99 34.98 35.00
 3  1  34.99 34.99 35.00 34.99 35.00
 4  1  35.00 35.00 34.99 34.99 35.00
 5  2  35.00 34.99 34.98 34.99 35.00
 6  2  34.99 34.99 35.00 35.00 35.00
 7  2  35.01 34.98 35.00 35.00 34.99
 8  2  35.00 35.00 34.99 34.98 34.99
 9  3  34.99 34.98 34.99 35.01 35.00
10  3  34.99 35.00 35.00 34.99 35.00
11  3  35.01 35.00 35.00 34.98 34.99
12  3  34.99 34.99 35.00 34.98 35.01
13  4  35.01 34.99 34.98 34.99 34.99
14  4  35.00 35.00 34.99 35.00 34.99
15  4  34.98 35.00 34.99 35.00 34.99
16  4  34.99 35.00 35.00 35.01 35.00
17  5  34.98 34.98 34.98 34.99 34.98
18  5  35.01 35.02 35.00 34.98 35.00
19  5  34.99 34.98 35.00 34.99 34.98
20  5  34.99 35.00 35.00 34.99 34.99
;
```

```
symbol1 v=square;
proc shewhart data=wafers;
   xchart Diameter*Batch / html = ( Web );
run;

ods html close;
run;
```

In this example five different URLs are each associated with a set of four subgroup values. When you view the ODS HTML output with a browser, you can click on a subgroup data point and the browser will bring up the page specified by the subgroup's URL. These URLs happen to point to pages at SAS Institute's web site which may be of interest to SAS/QC users.

NOTE: The value of the HTML= variable must be the same for each observation belonging to a given subgroup.

Links and Tests for Special Causes

The TESTHTML= data set provides a way to associate a link with each subgroup in a control chart for which a given test for special causes is positive:

Table 17.98 Variables Required in a TESTHTML= Data Set

Variable	Type	Description
TEST	character or numeric	test identifier
CHART	numeric	primary (1) or secondary (2) chart
URL	character	HTML specifying URL for subgroups with positive test

The variable _TEST_ identifies a test for special causes (see "Tests for Special Causes: SHEWHART Procedure" on page 2073). A standard test is identified by its number (1 to 8) and a nonstandard test is identified by the CODE= character in its pattern specification. The _TEST_ variable must be a character variable if nonstandard tests are included in the TESTHTML= data set. The value of _CHART_ is 1 or 2, specifying whether the test applies to the primary or secondary chart. The character variable _URL_ contains the HTML syntax for the link to be associated with subgroups for which the test is positive.

The following statements create a TESTHTML= data set and an $\bar{X}$ chart using the same DATA= data set as the previous example:

```
ods html body = "example2.html";

data testlink;
   length _URL_ $ 75;
   input _TEST_ _CHART_ _URL_;
   datalines;
1  1  href="http://www.sas.com/"
2  1  href="http://www.sas.com/service/techsup/faq/qc/shewproc.html"
3  1  href="http://www.sas.com/rnd/app/qc.html"
4  1  href="http://www.sas.com/rnd/app/qc/qcnew.html"
5  1  href="http://www.sas.com/products/qc/index.html"
6  1  href="http://www.sas.com/rnd/app/qc/qc/qcspc.html"
```

```
7  1  href="http://www.sas.com/software/components/qc.html"
8  1  href="http://www.sas.com/rnd/app/qc/qc.html"
;

symbol1 v=dot;
proc shewhart data=wafers testhtml=testlink;
   xchart Diameter*Batch / tests = 1 to 8;
run;

ods html close;
run;
```

In this example only subgroups triggering tests for special causes have URLs associated with them.

NOTE: If a TESTHTML= data set and an HTML= variable are both specified, the link from the TESTHTML= data set is associated with any subgroup for which the test is positive.

References

Al-Salti, M. and Statham, A. (1994), "A Review of the Literature on the Use of SPC in Batch Production," *Quality and Reliability Engineering International*, 10, 49–62.

Alt, F. (1985), "Multivariate Quality Control," in S. Kotz, N. L. Johnson, and C. B. Read, eds., *Encyclopedia of Statistical Sciences*, volume 6, 110–122, New York: John Wiley & Sons.

Alwan, L. C. and Roberts, H. V. (1988), "Time Series Modeling for Statistical Process Control," *Journal of Business and Economic Statistics*, 6, 87–95.

American Society for Testing and Materials (1976), *ASTM Manual on Presentation of Data and Control Chart Analysis*, Philadelphia: ASTM.

ASQC Automotive Division/AIAG (1990), *Fundamental Statistical Process Control: Reference Manual*, Southfield, MI: AIAG.

Austin, J. A. (1973), "Control Chart Constants for Largest and Smallest in Sampling from a Normal Distribution Using the Generalized Burr Distribution," *Technometrics*, 15, 931–933.

Bissell, A. F. (1990), "How Reliable Is Your Capability Index?" *Applied Statistics*, 39, 331–340.

Box, G. E. P. and Kramer, T. (1992), "Statistical Process Monitoring and Feedback Adjustment: A Discussion," *Technometrics*, 34, 251–285, with discussion.

Boyles, R. A. (1997), "Estimating Common-Cause Sigma in the Presence of Special Causes," *Journal of Quality Technology*, 29, 381–395.

Burr, I. W. (1969), "Control Charts for Measurements with Varying Sample Sizes," *Journal of Quality Technology*, 1, 163–167.

Burr, I. W. (1976), *Statistical Quality Control Methods*, New York: Marcel Dekker.

Champ, S. W. and Woodall, W. H. (1987), "Exact Results for Shewhart Control Charts with Supplementary Runs Rules," *Technometrics*, 29, 393–401.

Champ, S. W. and Woodall, W. H. (1990), "A Program to Evaluate the Run Length Distribution of a Shewhart Control Chart with Supplementary Run Rules," *Journal of Quality Technology*, 29, 393–399.

Deming, W. E. (1982), *Out of the Crisis*, Cambridge, MA: Massachusetts Institute of Technology, Center for Advanced Engineering Study.

Doganaksoy, N., Faltin, F. W., and Tucker, W. T. (1991), "Identification of Out-of-Control Quality Characteristics in a Multivariate Manufacturing Environment," *Communications in Statistics—Theory and Methods*, 20, 2775–2790.

Gnanadesikan, R. and Kettenring, J. R. (1972), "Robust Estimates, Residuals, and Outlier Detection with Multiresponse Data," *Biometrics*, 28, 81–124.

Grant, E. L. and Leavenworth, R. S. (1988), *Statistical Quality Control*, 6th Edition, New York: McGraw-Hill.

Hawkins, D. M. (1991), "Multivariate Quality Control Based on Regression-Adjusted Variables," *Technometrics*, 33, 61–75.

Hawkins, D. M. (1993), "Regression Adjustment for Variables in Multivariate Quality Control," *Journal of Quality Technology*, 25, 170–182.

Hillier, F. S. (1969), "$\bar{X}$- and *R*-Chart Control Limits Based on a Small Number of Subgroups," *Journal of Quality Technology*, 1, 17–26.

Hunter, J. S. (1986), "The Exponentially Weighted Moving Average," *Journal of Quality Technology*, 18, 203–210.

Hunter, J. S. (1988), "The Digidot Plot," *American Statistician*, 24, 54.

Iglewicz, B. and Hoaglin, D. C. (1987), "Use of Boxplots for Process Evaluation," *Journal of Quality Technology*, 19, 180–190.

Jackson, J. E. (1980), "Principal Components and Factor Analysis, Part 1: Principal Components," *Journal of Quality Technology*, 12, 201–213.

Jackson, J. E. (1991), *A User's Guide to Principal Components*, New York: John Wiley & Sons.

Johnson, N. L., Kotz, S., and Kemp, A. W. (1992), *Univariate Discrete Distributions*, 2nd Edition, New York: John Wiley & Sons.

Kume, H. (1985), *Statistical Methods for Quality Improvement*, Tokyo: AOTS Chosakai.

MacGregor, J. F. (1987), "Interfaces between Process Control and Online Statistical Process Control," *Computing and Systems Technology Division Communications*, 10, 9–20.

MacGregor, J. F. (1990), "A Different View of the Funnel Experiment," *Journal of Quality Technology*, 22, 255–259.

MacGregor, J. F., Hunter, J. S., and Harris, T. (1988), "SPC Interfaces," Short course notes.

Mason, R. L., Tracy, N. D., and Young, J. C. (1993), "Use of Hotelling's T^2 Statistic in Multivariate Control Charts," Unpublished paper.

McGill, R., Tukey, J. W., and Larsen, W. A. (1978), "Variations of Box Plots," *American Statistician*, 32, 12–16.

Montgomery, D. C. (1996), *Introduction to Statistical Quality Control*, 3rd Edition, New York: John Wiley & Sons.

Montgomery, D. C., Keats, J. B., Runger, G. C., and Messina, W. S. (1994), "Integrating Statistical Process Control and Engineering Process Control," *Journal of Quality Technology*, 26, 79–87.

Montgomery, D. C. and Mastrangelo, C. M. (1991), "Some Statistical Process Control Methods for Autocorrelated Data," *Journal of Quality Technology*, 23, 179–204, with discussion.

Nelson, L. S. (1982), "Control Charts for Individual Measurements," *Journal of Quality Technology*, 14, 172–174.

Nelson, L. S. (1984), "The Shewhart Control Chart—Tests for Special Causes," *Journal of Quality Technology*, 15, 237–239.

Nelson, L. S. (1985), "Interpreting Shewhart $\bar{X}$ Control Charts," *Journal of Quality Technology*, 17, 114–116.

Nelson, L. S. (1989), "Standardization of Shewhart Control Charts," *Journal of Quality Technology*, 21, 287–289.

Nelson, L. S. (1994), "Shewhart Control Charts with Unequal Subgroup Sizes," *Journal of Quality Technology*, 26, 64–67.

Quesenberry, C. P. (1991a), "SPC Q Charts for a Binomial Parameter p: Short or Long Runs," *Journal of Quality Technology*, 23, 239–246.

Quesenberry, C. P. (1991b), "SPC Q Charts for Start-Up Processes and Short or Long Runs," *Journal of Quality Technology*, 23, 213–224.

Quesenberry, C. P. (1993), "The Effect of Sample Size on Estimated Effects," *Journal of Quality Technology*, 25, 237–247.

Rocke, D. M. (1989), "Robust Control Charts," *Technometrics*, 31, 173–184.

Rodriguez, R. N. and Bynum, R. A. (1992), "Examples of Short Run Process Control Methods with the SHEWHART Procedure in SAS/QC Software," Unpublished manuscript available from the authors.

Ryan, T. P. (1989), *Statistical Methods for Quality Improvement*, New York: John Wiley & Sons.

Schilling, E. G. and Nelson, P. R. (1976), "The Effect of Non-normality on the Control Limits of $\bar{X}$ Charts," *Journal of Quality Technology*, 8, 183–187.

Schneider, H. and Pruett, J. M. (1994), "Control Charting Issues in the Process Industries," *Quality Engineering*, 6, 347–373.

Shewhart, W. A. (1931), *Economic Control of Quality Manufactured Product*, New York: D. Van Nostrand; republished in 1980 by the American Society of Quality Control.

Snedecor, G. W. and Cochran, W. G. (1980), *Statistical Methods*, 7th Edition, Ames: Iowa State University Press.

Teichroew, D. (1962), "Tables of Expected Values of Order Statistics and Products of Order Statistics for Samples of Size 20 and Less from the Normal Distribution," in *Contributions to Order Statistics*, 190–205, New York: John Wiley & Sons.

Tracy, N. D., Young, J. C., and Mason, R. L. (1992), “Multivariate Control Charts for Individual Observations,” *Journal of Quality Technology*, 24, 88–95.

Tukey, J. W. (1977), *Exploratory Data Analysis*, Reading, MA: Addison-Wesley.

Western Electric Company (1956), *Statistical Quality Control Handbook*, Indianapolis: Western Electric Company.

Westgard, J. O. (2002), *Basic QC Practices: Training in Statistical Quality Control for Healthcare Laboratories*, Madison, WI: Westgard QC.

Wetherill, G. B. and Brown, D. B. (1991), *Statistical Process Control: Theory and Practice*, London: Chapman & Hall.

Wheeler, D. J. (1991), “Shewhart’s Chart: Myths, Facts, and Competitors,” *Annual Quality Congress Transactions*, 45, 533–538.

Wheeler, D. J. (1995), *Advanced Topics in Statistical Process Control*, Knoxville, TN: SPC Press.

Wheeler, D. J. and Chambers, D. S. (1986), *Understanding Statistical Process Control*, Knoxville, TN: SPC Press.

Wilks, S. S. (1962), *Mathematical Statistics*, New York: John Wiley & Sons.

Woodall, W. H. (1993), “Autocorrelated Data and SPC,” *ASQC Statistics Division Newsletter*, 13, 18–21.

Subject Index

Syntax Index