

SAS/ETS[®] 13.2 User's Guide

The TIMESERIES

Procedure

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

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

Copyright © 2014, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a Web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a) and DFAR 227.7202-4 and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

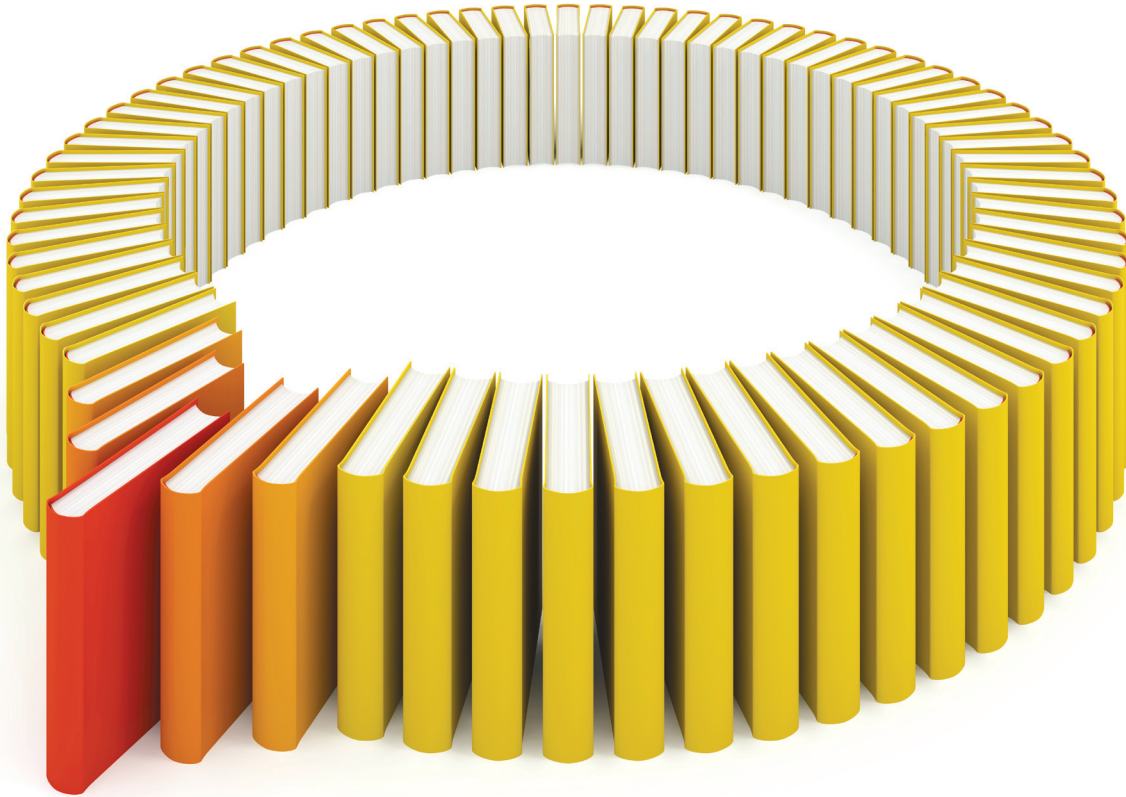
SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

August 2014

SAS provides a complete selection of books and electronic products to help customers use SAS® software to its fullest potential. For more information about our offerings, visit support.sas.com/bookstore or call 1-800-727-3228.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.



Gain Greater Insight into Your SAS[®] Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.



Chapter 32

The TIMESERIES Procedure

Contents

Overview: TIMESERIES Procedure	2220
Getting Started: TIMESERIES Procedure	2221
Syntax: TIMESERIES Procedure	2224
Functional Summary	2224
PROC TIMESERIES Statement	2227
BY Statement	2230
CORR Statement	2230
CROSSCORR Statement	2232
DECOMP Statement	2233
ID Statement	2235
SEASON Statement	2238
SPECTRA Statement	2239
SSA Statement	2241
TREND Statement	2243
VAR and CROSSVAR Statements	2244
Details: TIMESERIES Procedure	2245
Accumulation	2246
Missing Value Interpretation	2249
Time Series Transformation	2249
Time Series Differencing	2249
Descriptive Statistics	2250
Seasonal Decomposition	2250
Correlation Analysis	2252
Cross-Correlation Analysis	2253
Spectral Density Analysis	2254
Singular Spectrum Analysis	2258
Data Set Output	2260
OUT= Data Set	2260
OUTCORR= Data Set	2261
OUTCROSSCORR= Data Set	2262
OUTDECOMP= Data Set	2263
OUTPROCINFO= Data Set	2264
OUTSEASON= Data Set	2264
OUTSPECTRA= Data Set	2265
OUTSSA= Data Set	2265
OUTSUM= Data Set	2266

OUTTREND= Data Set	2267
STATUS Variable Values	2268
Printed Output	2268
ODS Table Names	2269
ODS Graphics Names	2269
Examples: TIMESERIES Procedure	2271
Example 32.1: Accumulating Transactional Data into Time Series Data	2271
Example 32.2: Trend and Seasonal Analysis	2272
Example 32.3: Illustration of ODS Graphics	2277
Example 32.4: Illustration of Spectral Analysis	2282
Example 32.5: Illustration of Singular Spectrum Analysis	2283
References	2286

Overview: TIMESERIES Procedure

The TIMESERIES procedure analyzes time-stamped transactional data with respect to time and accumulates the data into a time series format. The procedure can perform trend and seasonal analysis on the transactions. After the transactional data are accumulated, time domain and frequency domain analysis can be performed on the accumulated time series.

For seasonal analysis of the transaction data, various statistics can be computed for each season. For trend analysis of the transaction data, various statistics can be computed for each time period. The analysis is similar to applying the MEANS procedure of Base SAS software to each season or time period of concern.

After the transactional data are accumulated to form a time series and any missing values are interpreted, the accumulated time series can be functionally transformed using log, square root, logistic, or Box-Cox transformations. The time series can be further transformed using simple and/or seasonal differencing. After functional and difference transformations have been applied, the accumulated and transformed time series can be stored in an output data set. This working time series can then be analyzed further using various time series analysis techniques provided by this procedure or other SAS/ETS procedures.

Time series analyses performed by the TIMESERIES procedure include:

- descriptive (global) statistics
- seasonal decomposition/adjustment analysis
- correlation analysis
- cross-correlation analysis
- spectral analysis

All results of the transactional or time series analysis can be stored in output data sets or printed using the Output Delivery System (ODS).

The TIMESERIES procedure can process large amounts of time-stamped transactional data. Therefore, the analysis results are useful for large-scale time series analysis or (temporal) data mining. All of the results can be stored in output data sets in either a time series format (default) or in a coordinate format (transposed). The time series format is useful for preparing the data for subsequent analysis with other SAS/ETS procedures. For example, the working time series can be further analyzed, modeled, and forecast with other SAS/ETS procedures. The coordinate format is useful when using this procedure with SAS/STAT procedures or SAS Enterprise Miner. For example, clustering time-stamped transactional data can be achieved by using the results of this procedure with the clustering procedures of SAS/STAT and the nodes of SAS Enterprise Miner.

The EXPAND procedure can be used for the frequency conversion and transformations of time series output from this procedure.

Getting Started: TIMESERIES Procedure

This section outlines the use of the TIMESERIES procedure and gives a cursory description of some of the analysis techniques that can be performed on time-stamped transactional data.

Given an input data set that contains numerous transaction variables recorded over time at no specific frequency, the TIMESERIES procedure can form time series as follows:

```
PROC TIMESERIES DATA=<input-data-set>
                OUT=<output-data-set>;
  ID <time-ID-variable> INTERVAL=<frequency>
                ACCUMULATE=<statistic>;
  VAR <time-series-variables>;
RUN;
```

The TIMESERIES procedure forms time series from the input time-stamped transactional data. It can provide results in output data sets or in other output formats by using the Output Delivery System (ODS).

Time-stamped transactional data are often recorded at no fixed interval. Analysts often want to use time series analysis techniques that require fixed-time intervals. Therefore, the transactional data must be accumulated to form a fixed-interval time series.

Suppose that a bank wants to analyze the transactions associated with each of its customers over time. Further, suppose that the data set WORK.TRANSACTIONS contains four variables that are related to these transactions: CUSTOMER, DATE, WITHDRAWAL, and DEPOSITS. The following examples illustrate possible ways to analyze these transactions by using the TIMESERIES procedure.

To accumulate the time-stamped transactional data to form a daily time series based on the accumulated daily totals of each type of transaction (WITHDRAWALS and DEPOSITS), the following TIMESERIES procedure statements can be used:

```
proc timeseries data=transactions
                out=timeseries;
  by customer;
  id date interval=day accumulate=total;
  var withdrawals deposits;
run;
```

The OUT=TIMESERIES option specifies that the resulting time series data for each customer is to be stored in the data set WORK.TIMESERIES. The INTERVAL=DAY option specifies that the transactions are to be accumulated on a daily basis. The ACCUMULATE=TOTAL option specifies that the sum of the transactions is to be calculated. After the transactional data is accumulated into a time series format, many of the procedures provided with SAS/ETS software can be used to analyze the resulting time series data.

For example, the ARIMA procedure can be used to model and forecast each customer's withdrawal data by using an ARIMA(0,1,1)(0,1,1)_s model (where the number of seasons is $s=7$ days in a week) using the following statements:

```
proc arima data=timeseries;
  identify var=withdrawals(1,7) noprint;
  estimate q=(1)(7) outest=estimates noprint;
  forecast id=date interval=day out=forecasts;
quit;
```

The OUTEST=ESTIMATES data set contains the parameter estimates of the model specified. The OUT=FORECASTS data set contains forecasts based on the model specified. See the SAS/ETS ARIMA procedure for more detail.

A single set of transactions can be very large and must be summarized in order to analyze them effectively. Analysts often want to examine transactional data for trends and seasonal variation. To analyze transactional data for trends and seasonality, statistics must be computed for each time period and season of concern. For each observation, the time period and season must be determined and the data must be analyzed based on this determination.

The following statements illustrate how to use the TIMESERIES procedure to perform trend and seasonal analysis of time-stamped transactional data.

```
proc timeseries data=transactions out=out
               outseason=season outtrend=trend;
  by customer;
  id date interval=day accumulate=total;
  var withdrawals deposits;
run;
```

Since the INTERVAL=DAY option is specified, the length of the seasonal cycle is seven (7) where the first season is Sunday and the last season is Saturday. The output data set specified by the OUTSEASON=SEASON option contains the seasonal statistics for each day of the week by each customer. The output data set specified by the OUTTREND=TREND option contains the trend statistics for each day of the calendar by each customer.

Often it is desired to seasonally decompose into seasonal, trend, cycle, and irregular components or to seasonally adjust a time series. The following techniques describe how the changing seasons influence the time series.

The following statements illustrate how to use the TIMESERIES procedure to perform seasonal adjustment/decomposition analysis of time-stamped transactional data.

```
proc timeseries data=transactions
               out=out
               outdecomp=decompose;
  by customer;
  id date interval=day accumulate=total;
  var withdrawals deposits;
run;
```


The output data set specified by the `OUTDECOMP=DECOMPOSE` data set contains the decomposed/adjusted time series for each customer.

A single time series can be very large. Often, a time series must be summarized with respect to time lags in order to be efficiently analyzed using time domain techniques. These techniques help describe how a current observation is related to the past observations with respect to the time (season) lag.

The following statements illustrate how to use the `TIMESERIES` procedure to perform time domain analysis of time-stamped transactional data.

```
proc timeseries data=transactions
                out=out
                outcorr=timedomain;
    by customer;
    id date interval=day accumulate=total;
    var withdrawals deposits;
run;
```

The output data set specified by the `OUTCORR=TIMEDOMAIN` data set contains the time domain statistics, such as sample autocorrelations and partial autocorrelations, by each customer.

Sometimes time series data contain underlying patterns that can be identified using spectral analysis techniques. Two kinds of spectral analyses on univariate data can be performed using the `TIMESERIES` procedure. They are singular spectrum analysis and Fourier spectral analysis.

Singular spectrum analysis (SSA) is a technique for decomposing a time series into additive components and categorizing these components based on the magnitudes of their contributions. SSA uses a single parameter, the window length, to quantify patterns in a time series without relying on prior information about the series' structure. The window length represents the maximum lag that is considered in the analysis, and it corresponds to the dimensionality of the principle components analysis (PCA) on which SSA is based. The components are combined into groups to categorize their roles in the SSA decomposition.

Fourier spectral analysis decomposes a time series into a sum of harmonics. In the discrete Fourier transform, the contribution of components at evenly spaced frequencies are quantified in a periodogram and summarized in spectral density estimates.

The following statements illustrate how to use the `TIMESERIES` procedure to analyze time-stamped transactional data without prior information about the series' structure.

```
proc timeseries data=transactions
                outssa=ssa
                outspectra=spectra;
    by customer;
    id date interval=day accumulate=total;
    var withdrawals deposits;
run;
```

The output data set specified by the `OUTSSA=SSA` data set contains a singular spectrum analysis of the withdrawals and deposits data. The data set specified by `OUTSPECTRA=SPECTRA` contains a Fourier spectral decomposition of the same data.

By default, the `TIMESERIES` procedure produces no printed output.

Syntax: TIMESERIES Procedure

The TIMESERIES Procedure uses the following statements:

```

PROC TIMESERIES options ;
  BY variables ;
  CORR statistics-list / options ;
  CROSSCORR statistics-list / options ;
  CROSSVAR variable-list / options ;
  DECOMP component-list / options ;
  ID variable INTERVAL= interval-option ;
  SEASON statistics-list / options ;
  SPECTRA statistics-list / options ;
  SSA / options ;
  TREND statistics-list / options ;
  VAR variable-list / options ;

```

Functional Summary

Table 32.1 summarizes the statements and options that control the TIMESERIES procedure.

Table 32.1 TIMESERIES Functional Summary

Description	Statement	Option
Statements		
Specifies BY-group processing	BY	
Specifies variables to analyze	VAR	
Specifies cross variables to analyze	CROSSVAR	
Specifies the time ID variable	ID	
Specifies correlation options	CORR	
Specifies cross-correlation options	CROSSCORR	
Specifies decomposition options	DECOMP	
Specifies seasonal statistics options	SEASON	
Specifies spectral analysis options	SPECTRA	
Specifies SSA options	SSA	
Specifies trend statistics options	TREND	
Data Set Options		
Specifies the input data set	PROC TIMESERIES	DATA=
Specifies the output data set	PROC TIMESERIES	OUT=
Specifies the correlations output data set	PROC TIMESERIES	OUTCORR=
Specifies the cross-correlations output data set	PROC TIMESERIES	OUTCROSSCORR=
Specifies the decomposition output data set	PROC TIMESERIES	OUTDECOMP=
Specifies the SAS log output data set	PROC TIMESERIES	OUTPROCINFO=
Specifies the seasonal statistics output data set	PROC TIMESERIES	OUTSEASON=

Description	Statement	Option
Specifies the spectral analysis output data set	PROC TIMESERIES	OUTSPECTRA=
Specifies the SSA output data set	PROC TIMESERIES	OUTSSA=
Specifies the summary statistics output data set	PROC TIMESERIES	OUTSUM=
Specifies the trend statistics output data set	PROC TIMESERIES	OUTTREND=
Accumulation and Seasonality Options		
Specifies the accumulation frequency	ID	INTERVAL=
Specifies the length of seasonal cycle	PROC TIMESERIES	SEASONALITY=
Specifies the interval alignment	ID	ALIGN=
Specifies the interval boundary alignment	ID	BOUNDARYALIGN=
Specifies that time ID variable values not be sorted	ID	NOTSORTED
Specifies the starting time ID value	ID	START=
Specifies the ending time ID value	ID	END=
Specifies the accumulation statistic	ID, VAR, CROSSVAR	ACCUMULATE=
Specifies missing value interpretation	ID, VAR, CROSSVAR	SETMISSING=
Time-Stamped Data Seasonal Statistics Options		
Specifies the form of the output data set	SEASON	TRANSPOSE=
Fourier Spectral Analysis Options		
Specifies whether to adjust to the series mean	SPECTRA	ADJUSTMEAN=
Specifies confidence limits	SPECTRA	ALPHA=
Specifies the kernel weighting function	SPECTRA	PARZEN BARTLETT TUKEY TRUNC QS
Specifies the domain where kernel functions apply	SPECTRA	DOMAIN=
Specifies the constant kernel scale parameter	SPECTRA	C=
Specifies the exponent kernel scale parameter	SPECTRA	EXPON=
Specifies the periodogram weights	SPECTRA	WEIGHTS
Singular Spectrum Analysis Options		
Specifies whether to adjust to the series mean	SSA	ADJUSTMEAN=
Specifies the grouping of principal components	SSA	GROUPS=
Specifies the window length	SSA	LENGTH=
Specifies the number of time periods in the transposed output	SSA	NPERIODS=
Specifies the division between principal component groupings	SSA	THRESHOLDPCT
Specifies that the output be transposed	SSA	TRANSPOSE=

Description	Statement	Option
Time-Stamped Data Trend Statistics Options		
Specifies the form of the output data set	TREND	TRANSPPOSE=
Specifies the number of time periods to be stored	TREND	NPERIODS=
Time Series Transformation Options		
Specifies simple differencing	VAR, CROSSVAR	DIF=
Specifies seasonal differencing	VAR, CROSSVAR	SDIF=
Specifies transformation	VAR, CROSSVAR	TRANSFORM=
Time Series Correlation Options		
Specifies the list of lags	CORR	LAGS=
Specifies the number of lags	CORR	NLAG=
Specifies the number of parameters	CORR	NPARMS=
Specifies the form of the output data set	CORR	TRANSPPOSE=
Time Series Cross-Correlation Options		
Specifies the list of lags	CROSSCORR	LAGS=
Specifies the number of lags	CROSSCORR	NLAG=
Specifies the form of the output data set	CROSSCORR	TRANSPPOSE=
Time Series Decomposition Options		
Specifies the mode of decomposition	DECOMP	MODE=
Specifies the Hodrick-Prescott filter parameter	DECOMP	LAMBDA=
Specifies the number of time periods to be stored	DECOMP	NPERIODS=
Specifies the form of the output data set	DECOMP	TRANSPPOSE=
Printing Control Options		
Specifies the time ID format	ID	FORMAT=
Specifies which output to print	PROC TIMESERIES	PRINT=
Specifies that detailed output be printed	PROC TIMESERIES	PRINTDETAILS
Miscellaneous Options		
Specifies that analysis variables be processed in sorted order	PROC TIMESERIES	SORTNAMES
Limits error and warning messages	PROC TIMESERIES	MAXERROR=
ODS Graphics Options		
Specifies the cross-variable graphical output	PROC TIMESERIES	CROSSPLOTS=
Specifies the variable graphical output	PROC TIMESERIES	PLOTS=

PROC TIMESERIES Statement

PROC TIMESERIES *options* ;

The following options can be used in the PROC TIMESERIES statement:

DATA= *SAS-data-set*

names the SAS data set that contains the input data for the procedure to create the time series. If the DATA= option is not specified, the most recently created SAS data set is used.

CROSSPLOTS= *option* | (*options*)

specifies the cross-variable graphical output desired. By default, the TIMESERIES procedure produces no graphical output. The following plotting options are available:

SERIES plots the time series (OUT= data set).

CCF plots the cross-correlation functions (OUTCROSSCORR= data set).

ALL same as PLOTS=(SERIES CCF).

For example, CROSSPLOTS=SERIES plots the two time series. The CROSSPLOTS= option produces graphical output for these results by using the Output Delivery System (ODS). The CROSSPLOTS= option produces results similar to the data sets listed in parentheses next to the preceding options.

MAXERROR= *number*

limits the number of warning and error messages that are produced during the execution of the procedure to the specified value. The default is MAXERRORS=50. This option is particularly useful in BY-group processing where it can be used to suppress the recurring messages.

MAXVARLENGTH

specifies that processed variables be set to 8 bytes in length. This option exists principally for use when data storage may be a concern.

OUT= *SAS-data-set*

names the output data set to contain the time series variables specified in the subsequent VAR and CROSSVAR statements. If BY variables are specified, they are also included in the OUT= data set. If an ID variable is specified, it is also included in the OUT= data set. The values are accumulated based on the ID statement INTERVAL= or the ACCUMULATE= option or both. The OUT= data set is particularly useful when you want to further analyze, model, or forecast the resulting time series with other SAS/ETS procedures.

OUTCORR= *SAS-data-set*

names the output data set to contain the univariate time domain statistics.

OUTCROSSCORR= *SAS-data-set*

names the output data set to contain the cross-correlation statistics.

OUTDECOMP= *SAS-data-set*

names the output data set to contain the decomposed and/or seasonally adjusted time series.

OUTPROCINFO= *SAS-data-set*

names the output data set to contain information in the SAS log, specifically the number of notes, errors, and warnings and the number of series processed, analyses requested, and analyses failed.

OUTSEASON= *SAS-data-set*

names the output data set to contain the seasonal statistics. The statistics are computed for each season as specified by the ID statement **INTERVAL=** option or the **PROC TIMESERIES** statement **SEASONALITY=** option. The **OUTSEASON=** data set is particularly useful when analyzing transactional data for seasonal variations.

OUTSPECTRA= *SAS-data-set*

names the output data set to contain the univariate frequency domain analysis results.

OUTSSA= *SAS-data-set*

names the output data set to contain the singular spectrum analysis result series.

OUTSUM= *SAS-data-set*

names the output data set to contain the descriptive statistics. The descriptive statistics are based on the accumulated time series when the **ACCUMULATE=** and/or **SETMISSING=** options are specified in the **ID** or **VAR** statements. The **OUTSUM=** data set is particularly useful when analyzing large numbers of series and a summary of the results are needed.

OUTTREND= *SAS-data-set*

names the output data set to contain the trend statistics. The statistics are computed for each time period as specified by the ID statement **INTERVAL=** option. The **OUTTREND=** data set is particularly useful when analyzing transactional data for trends.

PLOTS= *option | (options)*

specifies the univariate graphical output desired. By default, the **TIMESERIES** procedure produces no graphical output. The following plotting options are available:

SERIES	plots the time series (OUT= data set).
RESIDUAL	plots the residual time series (OUT= data set).
HISTOGRAM	plots a histogram of the time series values
CYCLES	plots the seasonal cycles (OUT= data set).
CORR	plots the correlation panel (OUTCORR= data set).
ACF	plots the autocorrelation function (OUTCORR= data set).
PACF	plots the partial autocorrelation function (OUTCORR= data set).
IACF	plots the inverse autocorrelation function (OUTCORR= data set).
WN	plots the white noise probabilities (OUTCORR= data set).
DECOMP	plots the seasonal adjustment panel (OUTDECOMP= data set).
TCS	plots the trend-cycle-seasonal component (OUTDECOMP= data set).
TCC	plots the trend-cycle component (OUTDECOMP= data set).
SIC	plots the seasonal-irregular component (OUTDECOMP= data set).
SC	plots the seasonal component (OUTDECOMP= data set).

SA	plots the seasonal adjusted component (OUTDECOMP= data set).
PCSA	plots the percent change in the seasonal adjusted component (OUTDECOMP= data set).
IC	plots the irregular component (OUTDECOMP= data set).
TC	plots the trend component (OUTDECOMP= data set).
CC	plots the cycle component (OUTDECOMP= data set).
PERIODOGRAM < (<i>option</i>)>	plots the periodogram (OUTSPECTRA= data set). The available options for modifying the periodogram are MAXFREQ=<i>number</i> specifies the maximum frequency in radians to include in the plot MINPERIOD=<i>number</i> specifies the minimum period to include in the plot
SPECTRUM < (<i>option</i>)>	plots the spectral density estimate (OUTSPECTRA= data set). The available options for modifying the spectrum plot are MAXFREQ=<i>number</i> specifies the maximum frequency in radians to include in the plot MINPERIOD=<i>number</i> specifies the minimum period to include in the plot
SSA	plots the singular spectrum analysis results (OUTSSA= data set).
ALL	same as PLOTS=(SERIES HISTOGRAM ACF PACF IACF WN SSA PERIODOGRAM SPECTRUM).
BASIC	same as PLOTS=(SERIES HISTOGRAM CYCLES CORR DECOMP)

For example, PLOTS=SERIES plots the time series. The PLOTS= option produces graphical output for these results by using the Output Delivery System (ODS). The PLOTS= option produces results similar to the data sets listed in parentheses next to the preceding options.

PRINT= *option* | (*options*)

specifies the printed output desired. By default, the TIMESERIES procedure produces no printed output. The following printing options are available:

DECOMP	prints the seasonal decomposition/adjustment table (OUTDECOMP= data set).
SEASONS	prints the seasonal statistics table (OUTSEASON= data set).
DESCSTATS	prints the descriptive statistics for the accumulated time series (OUTSUM= data set).
SUMMARY	prints the descriptive statistics table for all time series (OUTSUM= data set).
TRENDS	prints the trend statistics table (OUTTREND= data set).
SSA	prints the singular spectrum analysis results (OUTSSA= data set).
ALL	same as PRINT=(DESCSTATS SUMMARY).

For example, PRINT=SEASONS prints the seasonal statistics. The PRINT= option produces printed output for these results by using the Output Delivery System (ODS). The PRINT= option produces results similar to the data sets listed in parentheses next to the preceding options.

PRINTDETAILS

specifies that output requested with the PRINT= option be printed in greater detail.

SEASONALITY= *number*

specifies the length of the seasonal cycle. For example, SEASONALITY=3 means that every group of three time periods forms a seasonal cycle. By default, the length of the seasonal cycle is one (no seasonality) or the length implied by the INTERVAL= option specified in the ID statement. For example, INTERVAL=MONTH implies that the length of the seasonal cycle is 12.

SORTNAMES

specifies that the variables specified in the VAR and CROSSVAR statements be processed in sorted order by the variable names. This option allows the output data sets to be presorted by the variable names.

BY Statement

A BY statement can be used with PROC TIMESERIES to obtain separate dummy variable definitions for groups of observations 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 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 option NOTSORTED or DESCENDING in the BY statement for the TIMESERIES 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.

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

CORR Statement

CORR *statistics* < / *options* > ;

A CORR statement can be used with the TIMESERIES procedure to specify options related to time domain analysis of the accumulated time series. Only one CORR statement is allowed.

The following time domain statistics are available:

LAG	time lag
N	number of variance products
ACOV	autocovariances
ACF	autocorrelations
ACFSTD	autocorrelation standard errors
ACF2STD	an indicator of whether autocorrelations are less than (-1) , greater than (1) , or within (0) two standard errors of zero
ACFNORM	normalized autocorrelations
ACFPROB	autocorrelation probabilities
ACFLPROB	autocorrelation log probabilities
PACF	partial autocorrelations
PACFSTD	partial autocorrelation standard errors
PACF2STD	an indicator of whether partial autocorrelation are less than (-1) , greater than (1) , or within (0) two standard errors of zero
PACFNORM	partial normalized autocorrelations
PACFPROB	partial autocorrelation probabilities
PACFLPROB	partial autocorrelation log probabilities
IACF	inverse autocorrelations
IACFSTD	inverse autocorrelation standard errors
IACF2STD	an indicator of whether the inverse autocorrelation is less than (-1) , greater than (1) or within (0) two standard errors of zero
IACFNORM	normalized inverse autocorrelations
IACFPROB	inverse autocorrelation probabilities
IACFLPROB	inverse autocorrelation log probabilities
WN	white noise test statistics
WNPROB	white noise test probabilities
WNLPROB	white noise test log probabilities

If none of the correlation statistics are specified, the default is as follows:

```
corr lag n acov acf acfstd pacf pacfstd iacf iacfstd wn wnprob;
```

The following options can be specified in the CORR statement following the slash (/):

NLAG= *number*

specifies the number of lags to be stored in the OUTCORR= data set or to be plotted. The default is 24 or three times the length of the seasonal cycle, whichever is smaller. The LAGS= option takes precedence over the NLAG= option.

LAGS= (*numlist*)

specifies the list of lags to be stored in OUTCORR= data set or to be plotted. The list of lags must be separated by spaces or commas. For example, LAGS=(1,3) specifies the first then third lag.

NPARMS= *number*

specifies the number of parameters used in the model that created the residual time series. The number of parameters determines the degrees of freedom associated with the Ljung-Box statistics. The default is NPARMS=0. This option is useful when analyzing the residuals of a time series model with the number of parameters specified by NPARMS=*number* option.

TRANSPOSE= NO|YES

specifies which values are recorded as column names in the OUTCORR= data set. TRANSPOSE=YES specifies that lags be recorded as the column names instead of correlation statistics as the column names. The TRANSPOSE=NO option is useful for graphing the correlation results with SAS/GRAPH procedures. The TRANSPOSE=YES option is useful for analyzing the correlation results with other SAS procedures such as the CLUSTER procedure of SAS/STAT or SAS Enterprise Miner software. The default is TRANSPOSE=NO.

CROSSCORR Statement

CROSSCORR *statistics* < / *options* > ;

A CROSSCORR statement can be used with the TIMESERIES procedure to specify options that are related to cross-correlation analysis of the accumulated time series. Only one CROSSCORR statement is allowed.

The following time domain statistics are available:

LAG	time lag
N	number of variance products
CCOV	cross covariances
CCF	cross-correlations
CCFSTD	cross-correlation standard errors
CCF2STD	an indicator of whether cross-correlations are less than (−1), greater than (1), or within (0) two standard errors of zero
CCFNORM	normalized cross-correlations
CCFPROB	cross-correlation probabilities
CCFLPROB	cross-correlation log probabilities

If none of the cross-correlation statistics are specified, the default is as follows:

```
crosscorr lag n ccov ccf ccfstd;
```

The following options can be specified in the CROSSCORR statement following the slash (/):

NLAG= *number*

specifies the number of lags to be stored in the OUTCROSSCORR= data set or to be plotted. The default is 24 or three times the length of the seasonal cycle, whichever is smaller. The LAGS= option takes precedence over the NLAG= option.

LAGS=(*numlist*)

specifies a list of lags to be stored in OUTCROSSCORR= data set or to be plotted. The list of lags must be separated by spaces or commas. For example, LAGS=(1,3) specifies the first then third lag.

TRANSPPOSE= NO|YES

specifies which values are recorded as column names in the OUTCROSSCORR= data set. TRANSPPOSE=YES specifies that the lags be recorded as the column names instead of the cross-correlation statistics. The TRANSPPOSE=NO option is useful for graphing the cross-correlation results with SAS/GRAPH procedures. The TRANSPPOSE=YES option is useful for analyzing the cross-correlation results with other procedures such as the CLUSTER procedure of SAS/STAT or SAS Enterprise Miner software. The default is TRANSPPOSE=NO.

DECOMP Statement

DECOMP *components* < / *options* > ;

A DECOMP statement can be used with the TIMESERIES procedure to specify options related to classical seasonal decomposition of the time series data. Only one DECOMP statement is allowed. The options specified affect all variables listed in the VAR statements. Decomposition can be performed only when the length of the seasonal cycle specified by the PROC TIMESERIES statement SEASONALITY= option or implied by the ID statement INTERVAL= option is greater than one.

The following seasonal decomposition components are available:

ORIG ORIGINAL	original series
TCC TRENDCYCLE	trend-cycle component
SIC SEASONIRREGULAR	seasonal-irregular component
SC SEASONAL	seasonal component
SCSTD	seasonal component standard errors
TCS TRENDCYCLESEASON	trend-cycle-seasonal component
IC IRREGULAR	irregular component
SA ADJUSTED	seasonally adjusted series
PCSA	percent change seasonally adjusted series
TC	trend component
CC CYCLE	cycle component

If none of the components are specified, the default is as follows:

```
decomp orig tcc sc ic sa;
```

The following options can be specified in the DECOMP statement following the slash (/):

MODE= *option*

specifies the type of decomposition to be used to decompose the time series. The following values can be specified for the MODE= option:

ADD ADDITIVE	additive decomposition
MULT MULTIPLICATIVE	multiplicative decomposition
LOGADD LOGADDITIVE	log-additive decomposition
PSEUDOADD PSEUDOADDITIVE	pseudo-additive decomposition
MULTORADD	multiplicative or additive decomposition, depending on data

Multiplicative and log additive decomposition require strictly positive time series. If the accumulated time series contains nonpositive values and the MODE=MULT or MODE=LOGADD option is specified, an error results. Pseudo-additive decomposition requires a nonnegative-valued time series. If the accumulated time series contains negative values and the MODE=PSEUDOADD option is specified, an error results. The MODE=MULTORADD option specifies that multiplicative decomposition be used when the accumulated time series contains only positive values, that pseudo-additive decomposition be used when the accumulated time series contains only nonnegative values, and that additive decomposition be used otherwise. The default is MODE=MULTORADD.

LAMBDA= *number*

specifies the Hodrick-Prescott filter parameter for trend-cycle decomposition. The default is LAMBDA=1600. Filtering applies when the trend component or the cycle component is requested. If filtering is not specified, this option is ignored.

NPERIODS= *number*

specifies the number of time periods to be stored in the OUTDECOMP= data set when the TRANSPOSE=YES option is specified. If the TRANSPOSE=NO option is specified, the NPERIODS= option is ignored. If the NPERIODS= option is positive, the first or beginning time periods are recorded. If the NPERIODS= option is negative, the last or ending time periods are recorded. The NPERIODS= option specifies the number of OUTDECOMP= data set variables to contain the seasonal decomposition and is therefore limited to the maximum allowed number of SAS variables. If the number of time periods exceeds this limit, a warning is printed in the log and the number of periods stored is reduced to the limit.

If the NPERIODS= option is not specified, all of the periods specified between the ID statement START= and END= options are stored. If at least one of the START= or END= options is not specified, the default magnitude is the seasonality specified by the SEASONALITY= option in the PROC TIMESERIES statement or implied by the INTERVAL= option in the ID statement. If only the START= option or both the START= and END= options are specified and the seasonality is zero, the default is NPERIODS=5. If only the END= option or neither the START= nor END= option is specified and the seasonality is zero, the default is NPERIODS=-5.

TRANSPOSE= NO | YES

specifies which values are recorded as column names in the OUTDECOMP= data set. TRANSPOSE=YES specifies that the time periods be recorded as the column names instead of the statistics.

The first and last time periods stored in the OUTDECOMP= data set correspond to the period of the ID statement START= option and END= option, respectively. If only the ID statement END= option is specified, the last time ID value of each accumulated time series corresponds to the last time period column. If only the ID statement START= option is specified, the first time ID value of each accumulated time series corresponds to the first time period column. If neither the START= option nor the END= option is specified with the ID statement, the first time ID value of each accumulated time series corresponds to the first time period column. The TRANSPOSE=NO option is useful for analyzing or displaying the decomposition results with SAS/GRAPH procedures. The TRANSPOSE=YES option is useful for analyzing the decomposition results with other SAS procedures or SAS Enterprise Miner software. The default is TRANSPOSE=NO.

ID Statement

ID *variable* **INTERVAL=***interval* **< options >** ;

The ID statement names a numeric variable that identifies observations in the input and output data sets. The ID variable's values are assumed to be SAS date or datetime values. In addition, the ID statement specifies the (desired) frequency associated with the time series. The ID statement *options* also specify how the observations are accumulated and how the time ID values are aligned to form the time series. The information specified affects all variables listed in subsequent VAR statements. If the ID statement is specified, the INTERVAL= must also be used. If an ID statement is not specified, the observation number, with respect to the BY group, is used as the time ID.

The following *options* can be used with the ID statement:

ACCUMULATE= *option*

specifies how the data set observations are to be accumulated within each time period. The frequency (width of each time interval) is specified by the INTERVAL= option. The ID variable contains the time ID values. Each time ID variable value corresponds to a specific time period. The accumulated values form the time series, which is used in subsequent analysis.

The ACCUMULATE= option is useful when there are zero or more than one input observations that coincide with a particular time period (for example, time-stamped transactional data). The EXPAND procedure offers additional frequency conversions and transformations that can also be useful in creating a time series.

The following *options* determine how the observations are accumulated within each time period based on the ID variable and the frequency specified by the INTERVAL= option:

NONE	No accumulation occurs; the ID variable values must be equally spaced with respect to the frequency. This is the default option.
TOTAL	Observations are accumulated based on the total sum of their values.
AVERAGE AVG	Observations are accumulated based on the average of their values.
MINIMUM MIN	Observations are accumulated based on the minimum of their values.
MEDIAN MED	Observations are accumulated based on the median of their values.
MAXIMUM MAX	Observations are accumulated based on the maximum of their values.
N	Observations are accumulated based on the number of nonmissing observations.

NMISS	Observations are accumulated based on the number of missing observations.
NOBS	Observations are accumulated based on the number of observations.
FIRST	Observations are accumulated based on the first of their values.
LAST	Observations are accumulated based on the last of their values.
STDDEV STD	Observations are accumulated based on the standard deviation of their values.
CSS	Observations are accumulated based on the corrected sum of squares of their values.
USS	Observations are accumulated based on the uncorrected sum of squares of their values.

If the `ACCUMULATE=` option is specified, the `SETMISSING=` option is useful for specifying how accumulated missing values are to be treated. If missing values should be interpreted as zero, then `SETMISSING=0` should be used. The section “[Details: TIMESERIES Procedure](#)” on page 2245 describes accumulation in greater detail.

ALIGN= option

controls the alignment of SAS dates used to identify output observations. The `ALIGN=` option accepts the following values: `BEGINNING` | `BEG` | `B`, `MIDDLE` | `MID` | `M`, and `ENDING` | `END` | `E`. `BEGINNING` is the default.

BOUNDARYALIGN= option

controls how the `ACCUMULATE=` option is processed for the two boundary time intervals, which include the `START=` and `END=` time ID values. Some time ID values might fall inside the first and last accumulation intervals but fall outside the `START=` and `END=` boundaries. In these cases the `BOUNDARYALIGN=` option determines which values to include in the accumulation operation. You can specify the following *options*:

NONE	No values outside the <code>START=</code> and <code>END=</code> boundaries are accumulated.
START	All observations in the first time interval are accumulated.
END	All observations in the last time interval are accumulated.
BOTH	All observations in the first and last are accumulated.

If no option is specified, the default value `BOUNDARYALIGN=NONE` is used. The section “[Details: TIMESERIES Procedure](#)” on page 2245 describes the `BOUNDARYALIGN=` accumulation option in greater detail.

END= option

specifies a SAS date or datetime value that represents the end of the data. If the last time ID variable value is less than the `END=` value, the series is extended with missing values. If the last time ID variable value is greater than the `END=` value, the series is truncated. For example, `END=&sysdate` uses the automatic macro variable `SYSDATE` to extend or truncate the series to the current date. The `START=` and `END=` options can be used to ensure that data associated within each `BY` group contains the same number of observations.

FORMAT= *format*

specifies the SAS format for the time ID values. If the FORMAT= option is not specified, the default format is implied from the INTERVAL= option.

INTERVAL= *interval*

specifies the frequency of the accumulated time series. For example, if the input data set consists of quarterly observations, then INTERVAL=QTR should be used. If the PROC TIMESERIES statement SEASONALITY= option is not specified, the length of the seasonal cycle is implied from the INTERVAL= option. For example, INTERVAL=QTR implies a seasonal cycle of length 4. If the ACCUMULATE= option is also specified, the INTERVAL= option determines the time periods for the accumulation of observations. The INTERVAL= option is required and must be the first option specified in the ID statement.

NOTSORTED

specifies that the time ID values not be in sorted order. The TIMESERIES procedure sorts the data with respect to the time ID prior to analysis.

SETMISSING= *option* | *number*

specifies how missing values (either actual or accumulated) are to be interpreted in the accumulated time series. If a number is specified, missing values are set to the number. If a missing value indicates an unknown value, this option should not be used. If a missing value indicates no value, SETMISSING=0 should be used. You would typically use SETMISSING=0 for transactional data because no recorded data usually implies no activity. The following options can also be used to determine how missing values are assigned:

MISSING	Missing values are set to missing. This is the default option.
AVERAGE AVG	Missing values are set to the accumulated average value.
MINIMUM MIN	Missing values are set to the accumulated minimum value.
MEDIAN MED	Missing values are set to the accumulated median value.
MAXIMUM MAX	Missing values are set to the accumulated maximum value.
FIRST	Missing values are set to the accumulated first nonmissing value.
LAST	Missing values are set to the accumulated last nonmissing value.
PREVIOUS PREV	Missing values are set to the previous period's accumulated nonmissing value. Missing values at the beginning of the accumulated series remain missing.
NEXT	Missing values are set to the next period's accumulated nonmissing value. Missing values at the end of the accumulated series remain missing.

START= *option*

specifies a SAS date or datetime value that represents the beginning of the data. If the first time ID variable value is greater than the START= value, the series is prepended with missing values. If the first time ID variable value is less than the START= value, the series is truncated. The START= and END= options can be used to ensure that data associated with each by group contains the same number of observations.

SEASON Statement

SEASON *statistics* < / *options* > ;

A SEASON statement can be used with the TIMESERIES procedure to specify options that are related to seasonal analysis of the time-stamped transactional data. Only one SEASON statement is allowed. The options specified affect all variables specified in the VAR statements. Seasonal analysis can be performed only when the length of the seasonal cycle specified by the PROC TIMESERIES statement SEASONALITY= option or implied by the ID statement INTERVAL= option is greater than one.

The following seasonal statistics are available:

NOBS	number of observations
N	number of nonmissing observations
NMISS	number of missing observations
MINIMUM	minimum value
MAXIMUM	maximum value
RANGE	range value
SUM	summation value
MEAN	mean value
STDDEV	standard deviation
CSS	corrected sum of squares
USS	uncorrected sum of squares
MEDIAN	median value

If none of the season statistics are specified, the default is as follows:

```
season n min max mean std;
```

The following option can be specified in the SEASON statement following the slash (/):

TRANSPOSE= NO | YES

specifies which values are recorded as column names in the OUTSEASON= data set. TRANSPOSE=YES specifies that the seasonal indices be recorded as the column names instead of the statistics. The TRANSPOSE=NO option is useful for graphing the seasonal analysis results with SAS/GRAPH procedures. The TRANSPOSE=YES option is useful for analyzing the seasonal analysis results with SAS procedures or SAS Enterprise Miner software. The default is TRANSPOSE=NO.

SPECTRA Statement

SPECTRA *statistics* < / *options* > ;

A SPECTRA statement can be used with the TIMESERIES procedure to specify which statistics appear in the OUTSPECTRA= data set. The SPECTRA statement *options* are used in performing a spectral analysis on the variables listed in the VAR statement. These *options* affect values that are produced in the PROC TIMESERIES statement's OUTSPECTRA= data set, and in the periodogram and spectral density estimate. Only one SPECTRA statement is allowed.

The following univariate frequency domain statistics are available:

FREQ	frequency in radians from 0 to π
PERIOD	period or wavelength
COS	cosine transform
SIN	sine transform
P	periodogram
S	spectral density estimates

If none of the frequency domain statistics are specified, the default is as follows:

spectra period p;

The following *options* can be specified in the SPECTRA statement following the slash (/):

C=coefficient

specifies the scale coefficient for the kernel function. See the section “[Kernel Option Details](#)” on page 2240 for more information.

E=exponent

EXP=exponent

EXPON=exponent

specifies the exponent for the kernel function. See the section “[Kernel Option Details](#)” on page 2240 for more information.

ADJUSTMEAN=NO | YES

CENTER=NO | YES

specifies whether or not the series should be adjusted by its mean prior to performing the Fourier decomposition. This sets the first periodogram ordinate to 0 rather than to $2n$ times the squared mean. This option is commonly used when the periodograms are to be plotted to prevent a large first periodogram ordinate from distorting the scale of the plot. ADJUSTMEAN=YES specifies that the series be transformed by subtracting its mean. ADJUSTMEAN=NO specifies that no adjustment of the series be performed. The default is ADJUSTMEAN=NO.

ALPHA=num

specifies the width of a window that is drawn around the spectral density estimate in a spectral density versus frequency plot. Based on approximations proposed by Brockwell and Davis (1991), periodogram ordinates fall within this window with a confidence level of $1 - num$. The value *num* must be between 0 and 1; the default is 0.5.

DOMAIN=domain

specifies how the smoothing function is interpreted. You can specify the following *domain* values:

FREQUENCY	smooths the periodogram ordinates.
TIME	applies the kernel as a filter to the time series autocovariance function.

By default DOMAIN=FREQUENCY, and smoothing is applied in the same manner as weights are applied when you specify the WEIGHTS= option.

kernel

specifies the smoothing function to use to calculate a spectral density estimate as the moving average of periodogram ordinates. The kernel function is an alternative smoothing method to using the WEIGHTS= option. You can specify the following *kernel* values:

PARZEN	Parzen kernel
BARTLETT	Bartlett kernel
TUKEY	Tukey-Hanning kernel
TRUNC TRUNCAT	truncated kernel
QS QUADR	quadratic spectral kernel

If neither a WEIGHTS= option nor a kernel function is specified, the spectral density estimate is identical to the unmodified periodogram.

WEIGHTS=numlist

specifies the relative weights to use to compute a spectral density estimate as the moving average smoothing of periodogram ordinates. If neither a WEIGHTS= option nor a kernel function is specified, the spectral density estimate is identical to the unmodified periodogram. The following SPECTRA statement uses the WEIGHTS= option to specify equal weighting for each of the three adjacent periodogram ordinates centered on each spectral density estimate:

```
spectra / weights 1 1 1;
```

For information about how the weights are applied, see the section [“Using Specification of Weight Constants”](#) on page 2258.

Kernel Option Details

You can further parameterize each of the kernel functions with a kernel scale factor by using the C= and E= options. The default values of the kernel scale parameters, *c* and *e*, that are associated with each of the kernel functions together with their kernel scale factor values, *M*, for a series with 100 periodogram ordinates are listed in [Table 32.2](#). The formula that is used to generate the table entries is $M = cK^e$, where *K* is the number of Fourier component frequencies.

Table 32.2 Default Kernel Scale Factor Parameters

Kernel	<i>c</i>	<i>e</i>	<i>M</i>
Bartlett	1/2	1/3	2.32
Parzen	1	1/5	2.51
Quadratic	1/2	1/5	1.26
Tukey-Hanning	2/3	1/5	1.67
Truncated	1/4	1/5	0.63

For example, to apply the truncated kernel by using default scale factor parameters in the frequency domain, you could use the following SPECTRA statement:

```
spectra / truncat;
```

For more information about the kernel function parameterization and the DOMAIN= option, see the section “Using Kernel Specifications” on page 2256.

SSA Statement

SSA < / *options* > ;

An SSA statement can be used with the TIMESERIES procedure to specify *options* that are related to singular spectrum analysis (SSA) of the accumulated time series. Only one SSA statement is allowed.

The following *options* can be specified in the SSA statement following the slash (/).

ADJUSTMEAN=NO | YES

CENTER=NO | YES

specifies whether or not the series should be adjusted by its mean prior to performing the singular spectrum analysis. ADJUSTMEAN=YES specifies that the series be transformed by subtracting its mean. ADJUSTMEAN=NO specifies that no adjustment of the series be performed. The default is ADJUSTMEAN=NO.

GROUPS= (*numlist*)...(*numlist*)

specifies the lists that categorize window lags into groups. The window lags must be separated by spaces or commas. For example, GROUPS=(1,3) (2,4) specifies that the first and third window lags form the first group and the second and fourth window lags form the second group. If no GROUPS= option is specified, the window lags are divided into two groups based on the THRESHOLDPCT= value.

For example, the following SSA statement specifies three groups:

```
ssa / groups=(1 3) (2 4 5) (6) ;
```

The first group contains the first and third principal components; the second group contains the second, fourth, and fifth principal components; and the third group contains the sixth principal component.

By default, the first group contains the principal components whose contributions to the series sum to greater than the THRESHOLDPCT= value of 90%, and the second group contains the remaining components.

LENGTH = number

specifies the window length to be used in the analysis. It represents the maximum lag used in the SSA autocovariance calculations. The number specified by the LENGTH= option must be greater than one. When the SEASONALITY= option is provided or inferred by the INTERVAL= option in the ID statement the default window length is the smaller of two times the length of the seasonal cycle and one half the length of the time series. When no seasonality value is available the default window length is the smaller of 12 and one half the length of the time series.

For example, the following SSA statement specifies a window length of 10:

```
ssa / length=10;
```

If no window length option is specified and the INTERVAL=MONTH or SEASONALITY=12 options are specified, a window length of 24 is used. If the specified window length is greater than one-half the length of the accumulated time series, the window length is reduced and a warning message is printed to the log.

NPERIODS= number

specifies the number of time periods to be stored in the OUTSSA= data set when the TRANSPOSE=YES option is specified. If the TRANSPOSE option is not specified, the NPERIODS= option is ignored. The NPERIODS= option specifies the number of OUTSSA= data set variables to contain the groups.

If the NPERIODS= option is not specified, all of the periods specified between the ID statement START= and END= options are stored. If at least one of the START= or END= options is not specified, the default magnitude is the seasonality specified by the SEASONALITY= option in the PROC TIMESERIES statement or implied by the INTERVAL= option in the ID statement. If only the START= option or both the START= and END= options are specified and the seasonality is zero, the default is NPERIODS=5. If only the END= option or neither the START= nor END= option is specified and the seasonality is zero, the default is NPERIODS=-5.

THRESHOLDPCT= percent

specifies a percentage used to divide the SSA components into two groups based on the cumulative percentage of their singular values. The percentage specified by the THRESHOLDPCT= option must be greater than zero and less than 100. The default is THRESHOLDPCT=90.

For example, the following SSA statement specifies 80%:

```
ssa / THRESHOLDPCT=80;
```

The size of the second group must be at least one, and it must be less than the window length. The percentage is adjusted to achieve this requirement.

For example, the following SSA statement specifies a THRESHOLDPCT= of 0%, which effectively sets the size of the second group to one less than the window length:

```
ssa / THRESHOLDPCT = 0;
```

The following SSA statement specifies a THRESHOLDPCT= of 100%, which implies that the size of the last group is one:

```
ssa / THRESHOLDPCT= 100;
```

TRANSPOSE= NO | YES

specifies which values are recorded as column names in the OUTSSA= data set. TRANSPOSE=YES specifies that the time periods be recorded as the column names instead of the specified groups as the column names. The first and last time period stored in the OUTSSA= data set corresponds to the period of the ID statement START= and END= options, respectively. If only the ID statement END= option is specified, the last time ID value of each accumulated time series corresponds to the last time period column. If only the ID statement START= option is specified, the first time ID value of each accumulated time series corresponds to the first time period column. If neither the START= option nor the END= option is specified with the ID statement, the first time ID value of each accumulated time series corresponds to the first time period column. The TRANSPOSE=NO option is useful for displaying the SSA results. The TRANSPOSE=YES option is useful for analyzing the SSA results using SAS Enterprise Miner software. The default is TRANSPOSE=NO.

TREND Statement

TREND *statistics* < / *options* > ;

A TREND statement can be used with the TIMESERIES procedure to specify options related to trend analysis of the time-stamped transactional data. Only one TREND statement is allowed. The options specified affect all variables specified in the VAR statements.

The following trend statistics are available:

NOBS	number of observations
N	number of nonmissing observations
NMISS	number of missing observations
MINIMUM	minimum value
MAXIMUM	maximum value
RANGE	range value
SUM	summation value
MEAN	mean value
STDDEV	standard deviation
CSS	corrected sum of squares
USS	uncorrected sum of squares
MEDIAN	median value

If none of the trend statistics are specified, the default is as follows:

```
trend n min max mean std;
```

The following options can be specified in the TREND statement following the slash (/):

NPERIODS= *number*

specifies the number of time periods to be stored in the OUTTREND= data set when the TRANSPPOSE=YES option is specified. If the TRANSPPOSE option is not specified, the NPERIODS= option is ignored. The NPERIODS= option specifies the number of OUTTREND= data set variables to contain the trend statistics and is therefore limited to the maximum allowed number of SAS variables.

If the NPERIODS= option is not specified, all of the periods specified between the ID statement START= and END= options are stored. If at least one of the START= or END= options is not specified, the default magnitude is the seasonality specified by the SEASONALITY= option in the PROC TIMESERIES statement or implied by the INTERVAL= option in the ID statement. If only the START= option or both the START= and END= options are specified and the seasonality is zero, the default is NPERIODS=5. If only the END= option or neither the START= nor END= option is specified and the seasonality is zero, the default is NPERIODS=-5.

TRANSPPOSE= NO | YES

specifies which values are recorded as column names in the OUTTREND= data set. TRANSPPOSE=YES specifies that the time periods be recorded as the column names instead of the statistics as the column names. The first and last time periods stored in the OUTTREND= data set correspond to the period of the ID statement START= and END= options, respectively. If only the ID statement END= option is specified, the last time ID value of each accumulated time series corresponds to the last time period column. If only the ID statement START= option is specified, the first time ID value of each accumulated time series corresponds to the first time period column. If neither the START= option nor the END= option is specified with the ID statement, the first time ID value of each accumulated time series corresponds to the first time period column. The TRANSPPOSE=NO option is useful for analyzing or displaying the trend analysis results with SAS/GRAPH procedures. The TRANSPPOSE=YES option is useful for analyzing the trend analysis results with other SAS procedures or SAS Enterprise Miner software. The default is TRANSPPOSE=NO.

VAR and CROSSVAR Statements

```
VAR variable-list < / options > ;
```

```
CROSSVAR variable-list < / options > ;
```

The VAR and CROSSVAR statements list the numeric variables in the DATA= data set whose values are to be accumulated to form the time series.

An input data set variable can be specified in only one VAR or CROSSVAR statement. Any number of VAR and CROSSVAR statements can be used. The following options can be used with the VAR and CROSSVAR statements:

ACCUMULATE= *option*

specifies how the data set observations are to be accumulated within each time period for the variables listed in the VAR or CROSSVAR statement. If the ACCUMULATE= option is not specified in the VAR or CROSSVAR statement, accumulation is determined by the ACCUMULATE= option of the ID statement. See the ID statement ACCUMULATE= option for more details.

DIF=(*numlist*)

specifies the differencing to be applied to the accumulated time series. The list of differencing orders must be separated by spaces or commas. For example, DIF=(1,3) specifies first then third order differencing. Differencing is applied after time series transformation. The TRANSFORM= option is applied before the DIF= option.

SDIF=(*numlist*)

specifies the seasonal differencing to be applied to the accumulated time series. The list of seasonal differencing orders must be separated by spaces or commas. For example, SDIF=(1,3) specifies first then third order seasonal differencing. Differencing is applied after time series transformation. The TRANSFORM= option is applied before the SDIF= option.

SETMISS= *option* | *number***SETMISSING=** *option* | *number*

specifies how missing values (either actual or accumulated) are to be interpreted in the accumulated time series for variables listed in the VAR or CROSSVAR statement. If the SETMISSING= option is not specified in the VAR or CROSSVAR statement, missing values are set based on the SETMISSING= option of the ID statement. See the ID statement SETMISSING= option for more details.

TRANSFORM= *option*

specifies the time series transformation to be applied to the accumulated time series. The following transformations are provided:

NONE	No transformation is applied. This option is the default.
LOG	logarithmic transformation
SQRT	square-root transformation
LOGISTIC	logistic transformation
BOXCOX (<i>n</i>)	Box-Cox transformation with parameter number where the number is between -5 and 5

When the TRANSFORM= option is specified, the time series must be strictly positive.

Details: TIMESERIES Procedure

The TIMESERIES procedure can be used to perform trend and seasonal analysis on transactional data. For trend analysis, various sample statistics are computed for each time period defined by the time ID variable and INTERVAL= option. For seasonal analysis, various sample statistics are computed for each season defined by the INTERVAL= or the SEASONALITY= option. For example, if the transactional data ranges from June 1990 to January 2000 and the data are to be accumulated on a monthly basis, then the trend statistics are

computed for every month: June 1990, July 1990, . . . , January 2000. The seasonal statistics are computed for each season: January, February, . . . , December.

The TIMESERIES procedure can be used to form time series data from transactional data. The accumulated time series can then be analyzed using time series techniques. The data is analyzed in the following order:

- | | |
|---------------------------------|--|
| 1. accumulation | ACCUMULATE= option in the ID, VAR, or CROSSVAR statement |
| 2. missing value interpretation | SETMISSING= option in the ID, VAR, or CROSSVAR statement |
| 3. time series transformation | TRANSFORM= option in the VAR or CROSSVAR statement |
| 4. time series differencing | DIF= and SDIF= options in the VAR or CROSSVAR statement |
| 5. descriptive statistics | OUTSUM= option and the PRINT=DESCSTATS option |
| 6. seasonal decomposition | DECOMP statement or the OUTDECOMP= option in the PROC TIME-SERIES statement |
| 7. correlation analysis | CORR statement or the OUTCORR= option in the PROC TIME-SERIES statement |
| 8. singular spectrum analysis | SSA statement or the OUTSSA= option in the PROC TIMESERIES statement |
| 9. Fourier spectral analysis | SPECTRA statement or the OUTSPECTRA= option in the PROC TIMESERIES statement |
| 10. cross-correlation analysis | CROSSCORR statement or the OUTCROSSCORR= option in the PROC TIMESERIES statement |

Accumulation

If the ACCUMULATE= option in the ID, VAR, or CROSSVAR statement is specified, data set observations are accumulated within each time period. The frequency (width of each time interval) is specified by the ID statement INTERVAL= option. The ID variable contains the time ID values. Each time ID value corresponds to a specific time period. Accumulation is useful when the input data set contains transactional data, whose observations are not spaced with respect to any particular time interval. The accumulated values form the time series, which is used in subsequent analyses.

For example, suppose a data set contains the following observations:

```

19MAR1999    10
19MAR1999    30
11MAY1999    50
12MAY1999    20
23MAY1999    20

```


If the INTERVAL=MONTH is specified, all of the above observations fall within a three-month period of time between March 1999 and May 1999. The observations are accumulated within each time period as follows:

If the ACCUMULATE=NONE option is specified, an error is generated because the ID variable values are not equally spaced with respect to the specified frequency (MONTH).

If the ACCUMULATE=TOTAL option is specified, the resulting time series is:

O1MAR1999	40
O1APR1999	.
O1MAY1999	90

If the ACCUMULATE=AVERAGE option is specified, the resulting time series is:

O1MAR1999	20
O1APR1999	.
O1MAY1999	30

If the ACCUMULATE=MINIMUM option is specified, the resulting time series is:

O1MAR1999	10
O1APR1999	.
O1MAY1999	20

If the ACCUMULATE=MEDIAN option is specified, the resulting time series is:

O1MAR1999	20
O1APR1999	.
O1MAY1999	20

If the ACCUMULATE=MAXIMUM option is specified, the resulting time series is:

O1MAR1999	30
O1APR1999	.
O1MAY1999	50

If the ACCUMULATE=FIRST option is specified, the resulting time series is:

O1MAR1999	10
O1APR1999	.
O1MAY1999	50

If the ACCUMULATE=LAST option is specified, the resulting time series is:

O1MAR1999	30
O1APR1999	.
O1MAY1999	20

If the ACCUMULATE=STDDEV option is specified, the resulting time series is:

```
01MAR1999    14.14
01APR1999    .
01MAY1999    17.32
```

As can be seen from the above examples, even though the data set observations contain no missing values, the accumulated time series can have missing values.

Boundary Alignment

When the BOUNDARYALIGN= option is used to qualify the START= or END= options, additional time series values can be incorporated into the accumulation operation. For instance, if a data set contains the following observations

```
01JAN1999    10
01FEB1999    10
01MAR1999    10
01APR1999    10
01MAY1999    10
01JUN1999    10
```

and the options START='01FEB1999'd, END='01APR1999'd, INTERVAL=QUARTER, and ACCUMULATE=TOTAL are specified, using the BOUNDARYALIGN= option results in the following accumulated time series:

If BOUNDARYALIGN=START is specified, the accumulated time series is

```
01JAN1999    30
01APR1999    10
```

If BOUNDARYALIGN=END is specified, the accumulated time series is

```
01JAN1999    20
01APR1999    30
```

If BOUNDARYALIGN=BOTH is specified, the accumulated time series is

```
01JAN1999    30
01APR1999    30
```

If BOUNDARYALIGN=NONE is specified, the accumulated time series is

```
01JAN1999    20
01APR1999    10
```

Missing Value Interpretation

Sometimes missing values should be interpreted as unknown values. But sometimes missing values are known, such as when missing values are created from accumulation and no observations should be interpreted as no value—that is, zero. In the former case, the SETMISSING= option can be used to interpret how missing values are treated. The SETMISSING=0 option should be used when missing observations are to be treated as no (zero) values. In other cases, missing values should be interpreted as global values, such as minimum or maximum values of the accumulated series. The accumulated and interpreted time series is used in subsequent analyses.

Time Series Transformation

There are four transformations available for strictly positive series only. Let $y_t > 0$ be the original time series, and let w_t be the transformed series. The transformations are defined as follows:

Log is the logarithmic transformation.

$$w_t = \ln(y_t)$$

Logistic is the logistic transformation.

$$w_t = \ln(cy_t/(1 - cy_t))$$

where the scaling factor c is

$$c = (1 - 10^{-6})10^{-\text{ceil}(\log_{10}(\max(y_t)))}$$

and $\text{ceil}(x)$ is the smallest integer greater than or equal to x .

Square root is the square root transformation.

$$w_t = \sqrt{y_t}$$

Box Cox is the Box-Cox transformation.

$$w_t = \begin{cases} \frac{y_t^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \ln(y_t), & \lambda = 0 \end{cases}$$

More complex time series transformations can be performed by using the EXPAND procedure of SAS/ETS.

Time Series Differencing

After optionally transforming the series, the accumulated series can be simply or seasonally differenced by using the VAR and CROSSVAR statement DIF= and SDIF= options. For example, suppose y_t is a monthly time series. The following examples of the DIF= and SDIF= options demonstrate how to simply and seasonally difference the time series.

```

dif=(1) sdif=(1)
dif=(1,12)

```

Additionally, when y_t is strictly positive and the TRANSFORM=, DIF=, and SDIF= options are combined in the VAR and CROSSVAR statements, the transformation operation is performed before the differencing operations.

Descriptive Statistics

Descriptive statistics can be computed from the working series by specifying the OUTSUM= option or PRINT=DESCSTATS.

Seasonal Decomposition

Seasonal decomposition/analysis can be performed on the working series by specifying the OUTDECOMP= option, the PRINT=DECOMP option, or one of the PLOTS= options associated with decomposition in the PROC TIMESERIES statement. The DECOMP statement enables you to specify options related to decomposition. The TIMESERIES procedure uses classical decomposition. More complex seasonal decomposition/adjustment analysis can be performed by using the X11 or the X12 procedure of SAS/ETS.

The DECOMP statement MODE= option determines the mode of the seasonal adjustment decomposition to be performed. There are four modes: multiplicative (MODE=MULT), additive (MODE=ADD), pseudo-additive (MODE=PSEUDOADD), and log-additive (MODE=LOGADD) decomposition. The default is MODE=MULTORADD which specifies MODE=MULT for series that are strictly positive, MODE=PSEUDOADD for series that are nonnegative, and MODE=ADD for series that are not nonnegative.

When MODE=LOGADD is specified, the components are exponentiated to the original metric.

The DECOMP statement LAMBDA= option specifies the Hodrick-Prescott filter parameter (Hodrick and Prescott 1980). The default is LAMBDA=1600. The Hodrick-Prescott filter is used to decompose the trend-cycle component into the trend component and cycle component in an additive fashion. A smaller parameter assigns less significance to the cycle; that is, LAMBDA=0 implies no cycle component.

The notation and keywords associated with seasonal decomposition/adjustment analysis are defined in [Table 32.3](#).

Table 32.3 Seasonal Adjustment Formulas

Component	Keyword	MODE= Option	Formula
original series	ORIGINAL	MULT	$O_t = TC_t S_t I_t$
		ADD	$O_t = TC_t + S_t + I_t$
		LOGADD	$\log(O_t) = TC_t + S_t + I_t$
		PSEUDOADD	$O_t = TC_t(S_t + I_t - 1)$
trend-cycle component	TCC	MULT	centered moving average of O_t
		ADD	centered moving average of O_t
		LOGADD	centered moving average of $\log(O_t)$
		PSEUDOADD	centered moving average of O_t
seasonal-irregular component	SIC	MULT	$SI_t = S_t I_t = O_t / TC_t$
		ADD	$SI_t = S_t + I_t = O_t - TC_t$
		LOGADD	$SI_t = S_t + I_t = \log(O_t) - TC_t$
		PSEUDOADD	$SI_t = S_t + I_t - 1 = O_t / TC_t$
seasonal component	SC	MULT	seasonal Averages of SI_t
		ADD	seasonal Averages of SI_t
		LOGADD	seasonal Averages of SI_t
		PSEUDOADD	seasonal Averages of SI_t
irregular component	IC	MULT	$I_t = SI_t / S_t$
		ADD	$I_t = SI_t - S_t$
		LOGADD	$I_t = SI_t - S_t$
		PSEUDOADD	$I_t = SI_t - S_t + 1$
trend-cycle-seasonal component	TCS	MULT	$TCS_t = TC_t S_t = O_t / I_t$
		ADD	$TCS_t = TC_t + S_t = O_t - I_t$
		LOGADD	$TCS_t = TC_t + S_t = O_t - I_t$
		PSEUDOADD	$TCS_t = TC_t S_t$
trend component	TC	MULT	$T_t = TC_t - C_t$
		ADD	$T_t = TC_t - C_t$
		LOGADD	$T_t = TC_t - C_t$
		PSEUDOADD	$T_t = TC_t - C_t$
cycle component	CC	MULT	$C_t = TC_t - T_t$
		ADD	$C_t = TC_t - T_t$
		LOGADD	$C_t = TC_t - T_t$
		PSEUDOADD	$C_t = TC_t - T_t$
seasonally adjusted series	SA	MULT	$SA_t = O_t / S_t = TC_t I_t$
		ADD	$SA_t = O_t - S_t = TC_t + I_t$
		LOGADD	$SA_t = O_t / \exp(S_t) = \exp(TC_t + I_t)$
		PSEUDOADD	$SA_t = TC_t I_t$

When s is odd the trend-cycle component is computed from the s -period centered moving average as follows:

$$TC_t = \sum_{k=-\lfloor s/2 \rfloor}^{\lfloor s/2 \rfloor} y_{t+k} / s$$

When s is even the trend-cycle component is computed from the s -period centered moving average as follows:

$$TC_t = \sum_{k=-s/2}^{s/2-1} (y_{t+k} + y_{t+1+k}) / 2s$$

The seasonal component is obtained by averaging the seasonal-irregular component for each season.

$$S_{k+js} = \sum_{t=k \bmod s} \frac{SI_t}{T/s}$$

where $0 \leq j \leq T/s$ and $1 \leq k \leq s$. The seasonal components are normalized to sum to one (multiplicative) or zero (additive).

Correlation Analysis

Correlation analysis can be performed on the working series by specifying the OUTCORR= option or one of the PLOTS= options that are associated with correlation. The CORR statement enables you to specify options that are related to correlation analysis.

Autocovariance Statistics

LAGS	$h \in \{0, \dots, H\}$
N	N_h is the number of observed products at lag h , ignoring missing values
ACOV	$\hat{\gamma}(h) = \frac{1}{T} \sum_{t=h+1}^T (y_t - \bar{y})(y_{t-h} - \bar{y})$
ACOV	$\hat{\gamma}(h) = \frac{1}{N_h} \sum_{t=h+1}^T (y_t - \bar{y})(y_{t-h} - \bar{y})$ when embedded missing values are present

Autocorrelation Statistics

ACF	$\hat{\rho}(h) = \hat{\gamma}(h) / \hat{\gamma}(0)$
ACFSTD	$Std(\hat{\rho}(h)) = \sqrt{\frac{1}{T} \left(1 + 2 \sum_{j=1}^{h-1} \hat{\rho}(j)^2 \right)}$
ACFNORM	$Norm(\hat{\rho}(h)) = \hat{\rho}(h) / Std(\hat{\rho}(h))$
ACFPROB	$Prob(\hat{\rho}(h)) = 2(1 - \Phi(Norm(\hat{\rho}(h))))$
ACFLPROB	$LogProb(\hat{\rho}(h)) = -\log_{10}(Prob(\hat{\rho}(h)))$
ACF2STD	$Flag(\hat{\rho}(h)) = \begin{cases} 1 & \hat{\rho}(h) > 2Std(\hat{\rho}(h)) \\ 0 & -2Std(\hat{\rho}(h)) < \hat{\rho}(h) < 2Std(\hat{\rho}(h)) \\ -1 & \hat{\rho}(h) < -2Std(\hat{\rho}(h)) \end{cases}$

Partial Autocorrelation Statistics

PACF	$\hat{\phi}(h) = \Gamma_{(0,h-1)}\{\gamma_j\}_{j=1}^h$
PACFSTD	$Std(\hat{\phi}(h)) = 1/\sqrt{N_0}$
PCFNORM	$Norm(\hat{\phi}(h)) = \hat{\phi}(h)/Std(\hat{\phi}(h))$
PACFPROB	$Prob(\hat{\phi}(h)) = 2(1 - \Phi(Norm(\hat{\phi}(h))))$
PACFLPROB	$LogProb(\hat{\phi}(h)) = -\log_{10}(Prob(\hat{\phi}(h)))$
PACF2STD	$Flag(\hat{\phi}(h)) = \begin{cases} 1 & \hat{\phi}(h) > 2Std(\hat{\phi}(h)) \\ 0 & -2Std(\hat{\phi}(h)) < \hat{\phi}(h) < 2Std(\hat{\phi}(h)) \\ -1 & \hat{\phi}(h) < -2Std(\hat{\phi}(h)) \end{cases}$

Inverse Autocorrelation Statistics

IACF	$\hat{\theta}(h)$
IACFSTD	$Std(\hat{\theta}(h)) = 1/\sqrt{N_0}$
IACFNORM	$Norm(\hat{\theta}(h)) = \hat{\theta}(h)/Std(\hat{\theta}(h))$
IACFPROB	$Prob(\hat{\theta}(h)) = 2\left(1 - \Phi\left(Norm(\hat{\theta}(h)) \right)\right)$
IACFLPROB	$LogProb(\hat{\theta}(h)) = -\log_{10}(Prob(\hat{\theta}(h)))$
IACF2STD	$Flag(\hat{\theta}(h)) = \begin{cases} 1 & \hat{\theta}(h) > 2Std(\hat{\theta}(h)) \\ 0 & -2Std(\hat{\theta}(h)) < \hat{\theta}(h) < 2Std(\hat{\theta}(h)) \\ -1 & \hat{\theta}(h) < -2Std(\hat{\theta}(h)) \end{cases}$

White Noise Statistics

WN	$Q(h) = T(T+2) \sum_{j=1}^h \rho(j)^2 / (T-j)$
WN	$Q(h) = \sum_{j=1}^h N_j \rho(j)^2$ when embedded missing values are present
WNPROB	$Prob(Q(h)) = \chi_{\max(1,h-p)}(Q(h))$
WNLPROB	$LogProb(Q(h)) = -\log_{10}(Prob(Q(h)))$

Cross-Correlation Analysis

Cross-correlation analysis can be performed on the working series by specifying the OUTCROSSCORR= option or one of the CROSSPLOTS= options that are associated with cross-correlation. The CROSSCORR statement enables you to specify options that are related to cross-correlation analysis.

Cross-Correlation Statistics

The cross-correlation statistics for the variable x supplied in a VAR statement and variable y supplied in a CROSSVAR statement are:

LAGS	$h \in \{0, \dots, H\}$
N	N_h is the number of observed products at lag h , ignoring missing values
CCOV	$\hat{\gamma}_{x,y}(h) = \frac{1}{T} \sum_{t=h+1}^T (x_t - \bar{x})(y_{t-h} - \bar{y})$
CCOV	$\hat{\gamma}_{x,y}(h) = \frac{1}{N_h} \sum_{t=h+1}^T (x_t - \bar{x})(y_{t-h} - \bar{y})$ when embedded missing values are present
CCF	$\hat{\rho}_{x,y}(h) = \hat{\gamma}_{x,y}(h) / \sqrt{\hat{\gamma}_x(0)\hat{\gamma}_y(0)}$
CCFSTD	$Std(\hat{\rho}_{x,y}(h)) = 1/\sqrt{N_0}$
CCFNORM	$Norm(\hat{\rho}_{x,y}(h)) = \hat{\rho}_{x,y}(h) / Std(\hat{\rho}_{x,y}(h))$
CCFPROB	$Prob(\hat{\rho}_{x,y}(h)) = 2(1 - \Phi(Norm(\hat{\rho}_{x,y}(h))))$
CCFLPROB	$LogProb(\hat{\rho}_{x,y}(h)) = -\log_{10}(Prob(\hat{\rho}_{x,y}(h)))$
CCF2STD	$Flag(\hat{\rho}_{x,y}(h)) = \begin{cases} 1 & \hat{\rho}_{x,y}(h) > 2Std(\hat{\rho}_{x,y}(h)) \\ 0 & -2Std(\hat{\rho}_{x,y}(h)) < \hat{\rho}_{x,y}(h) < 2Std(\hat{\rho}_{x,y}(h)) \\ -1 & \hat{\rho}_{x,y}(h) < -2Std(\hat{\rho}_{x,y}(h)) \end{cases}$

Spectral Density Analysis

Spectral analysis can be performed on the working series by specifying the OUTSPECTRA= option or by specifying the PLOTS=PERIODOGRAM or PLOTS=SPECTRUM option in the PROC TIMESERIES statement. PROC TIMESERIES uses the finite Fourier transform to decompose data series into a sum of sine and cosine terms of different amplitudes and wavelengths. The finite Fourier transform decomposition of the series x_t is

$$x_t = \frac{a_0}{2} + \sum_{k=1}^{K-1} f_k(a_k \cos \omega_k t + b_k \sin \omega_k t)$$

$$f_k = \begin{cases} 1/2 & \text{if } T \text{ is even and } k = K - 1 \\ 1 & \text{otherwise} \end{cases}$$

where

t	is the time subscript, $t = 0, 1, 2, \dots, T - 1$
x_t	are the equally spaced time series data
T	is the number of observations in the time series
K	is the number of frequencies in the Fourier decomposition: $K = \frac{T+2}{2}$ if T is even, $K = \frac{T+1}{2}$ if T is odd
k	is the frequency subscript, $k = 0, 1, 2, \dots, K - 1$
a_0	is the mean term: $a_0 = 2\bar{x}$
a_k	are the cosine coefficients
b_k	are the sine coefficients
ω_k	are the Fourier frequencies: $\omega_k = \frac{2\pi k}{T}$

The Fourier decomposition is performed after the ACCUMULATE=, DIF=, SDIF= and TRANSFORM= options in the ID and VAR statements have been applied.

Functions of the Fourier coefficients a_k and b_k can be plotted against frequency or against wavelength to form *periodograms*. The amplitude periodogram I_k is defined as follows:

$$I_k = \frac{T}{2}(a_k^2 + b_k^2)$$

Since the Fourier transform is an even, periodic function of frequency which repeats every T ordinates the periodogram is also. Values of I_k for all k therefore can be mapped to the unique values $I_k : k = 0 \dots K-1$ using the equations

$$\begin{aligned} I_k &= I_{-k} && \text{for all } k \\ I_k &= I_{k+nT} && \text{for } n = \pm 1, \pm 2, \pm 3, \dots \\ I_k &= I_{T-k} && \text{for } 0 \leq k \leq K-1 \end{aligned}$$

The periodogram, I_k , is an estimate at the discrete frequencies ω_k of the spectral density function which characterizes the series x_t . By smoothing the periodogram an improved spectral density estimate with reduced variance and bias can be achieved at these points. Smoothing can be accomplished either through use of a spectral window smoothing function or by applying a lag window filter to the series autocovariance function.

When the SPECTRA statement's DOMAIN=FREQUENCY option is in effect spectral density estimates are computed by smoothing the periodogram ordinates using the equation

$$S_k(M) = \sum_{\tau=K-T}^{K-1} w\left(\frac{\tau}{M}\right) I_{k+\tau}$$

where $w(\theta)$ is the spectral window function whose form is specified by either the KERNEL= option or the WEIGHTS option. M is the kernel scale parameter which acts as a frequency scaling factor in the spectral window smoothing function. Values of $I_{k+\tau}$ that fall outside of $0 \leq k + \tau \leq K-1$ are mapped to values inside this range by the equations presented previously.

When the DOMAIN=TIME option is specified spectral density values are estimated by applying a lag window filter, $\lambda(h, M)$, to the series autocovariance function. The spectral density estimate then can be computed from the filtered autocovariance function using the equation

$$S_k(M) = \sum_{h=-(T-1)}^{T-1} \lambda(h, M) \hat{\gamma}(h) \cos h\omega_k.$$

In this case the kernel scale parameter, M , serves as a scale factor for the lag length, h , in the time domain. In the lag window formulation the spectral density estimate is a consistent estimator as $T, M \rightarrow \infty$ under the conditions $\lambda(h, M) = 0$ for $|h| > M$, and $\lim_{T \rightarrow \infty} M/T = 0$. These conditions lead to the following parameterization of M provided by the SPECTRA statement:

$$M = cK^e$$

where the values $c > 0$ and $0 < e < 1$ satisfy the consistency conditions. To specify the kernel scale parameter explicitly, set $c =$ to the desired scale factor and $e = 0$.

For uniformity and computational efficiency all spectral density estimates are calculated using a spectral window weighting function, $w(\theta)$, applied to the periodogram ordinates. In the case where the DOMAIN=TIME option is specified the effective spectral window weighting function is computed by the equation

$$w_{\text{TIME}}(\theta) = \sum_{h=-(T-1)}^{T-1} \lambda(h, M) \cos h\theta.$$

Because the kernel scale parameter, M , serves as a lag scale factor in the time domain and bandwidth scale factor in the frequency domain the impact of M on spectral density estimates depends on the value of the DOMAIN= option. When DOMAIN=FREQUENCY increasing values of M decrease variance and increase bias in the spectral density estimates whereas when DOMAIN=TIME increasing values of M increase variance and decrease bias.

Using Kernel Specifications

You can specify one of ten different kernel smoothing functions in the SPECTRA statement. Five smoothing functions are available as KERNEL= options and five complementary smoothing functions which correspond to lag window filters are available when the KERNEL= option is used in conjunction with the DOMAIN=TIME option.

For example, a Parzen kernel with a support of 11 periodogram ordinates in the frequency domain can be specified using the kernel option:

```
spectra / parzen c=5 expon=0;
```

The TIMESERIES procedure supports the following spectral window kernel functions in the frequency domain where $x = \tau/M$:

BARTLETT: Bartlett kernel

$$w(x) = \begin{cases} 1 - |x| & |x| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

PARZEN: Parzen kernel

$$w(x) = \begin{cases} 1 - 6|x|^2 + 6|x|^3 & 0 \leq |x| \leq \frac{1}{2} \\ 2(1 - |x|)^3 & \frac{1}{2} \leq |x| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

QS: quadratic spectral kernel

$$w(x) = \frac{3}{(2\pi x)^2} \left(\frac{\sin 2\pi x}{2\pi x} - \cos 2\pi x \right)$$

TUKEY: Tukey-Hanning kernel

$$w(x) = \begin{cases} (1 + \cos(\pi x))/2 & |x| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

TRUNCAT: truncated kernel

$$w(x) = \begin{cases} 1 & |x| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

When the DOMAIN=TIME option is specified the five kernel functions above are interpreted as lag window filters on the autocovariance function. The lag window kernel functions correspond to the following spectral window smoothing functions where $\theta = 2\pi\tau/T$:

BARTLETT: Bartlett equivalent lag window filter

$$w(\theta) = \frac{1}{2\pi M} \left(\frac{\sin(M\theta/2)}{\sin(\theta/2)} \right)^2$$

PARZEN: Parzen equivalent lag window filter

$$w(\theta) = \frac{6}{\pi M^3} \left(\frac{\sin(M\theta/4)}{\sin(\theta/2)} \right)^4 \left(1 - \frac{2}{3} \sin^2(\theta/2) \right)$$

QS: quadratic spectral equivalent lag window filter

$$w(\theta) = \begin{cases} \frac{3M}{4\pi} (1 - (M\theta/\pi)^2) & |\theta| \leq \pi/M \\ 0 & |\theta| > \pi/M \end{cases}$$

TUKEY: Tukey-Hanning equivalent lag window filter

$$w(\theta) = \frac{1}{4} D_M(\theta - \pi/M) + \frac{1}{2} D_M(\theta) + \frac{1}{4} D_M(\theta + \pi/M)$$

$$D_M(\theta) = \frac{1}{2\pi} \frac{\sin[(M + 1/2)\theta]}{\sin(\theta/2)}$$

TRUNC: truncated equivalent lag window filter

$$w(\theta) = D_M(\theta)$$

Using Specification of Weight Constants

Any number of weighting constants can be specified. The constants are interpreted symmetrically about the middle weight. The middle constant (or the constant to the right of the middle if an even number of weight constants is specified) is the relative weight of the current periodogram ordinate. The constant immediately following the middle one is the relative weight of the next periodogram ordinate, and so on. The actual weights used in the smoothing process are the weights specified in the WEIGHTS option scaled so that they sum to 1.

The moving average calculation reflects at each end of the periodogram to accommodate the periodicity of the periodogram function.

For example, a simple triangular weighting can be specified using the following WEIGHTS option:

```
spectra / weights 1 2 3 2 1;
```

Computational Method

If the number of observations, T , factors into prime integers that are less than or equal to 23, and the product of the square-free factors of T is less than 210, then the procedure uses the fast Fourier transform developed by Cooley and Tukey (1965) and implemented by Singleton (1969). If T cannot be factored in this way, then the procedure uses a Chirp-Z algorithm similar to that proposed by Monro and Branch (1977).

Missing Values

Missing values are replaced with an estimate of the mean to perform spectral analyses. This treatment of a series with missing values is consistent with the approach used by Priestley (1981).

Singular Spectrum Analysis

Given a time series, y_t , for $t = 1, \dots, T$, and a window length, $2 \leq L < T/2$, singular spectrum analysis Golyandina, Nekrutkin, and Zhigljavsky (2001) decompose the time series into spectral groupings using the following steps:

Embedding Step

Using the time series, form a $K \times L$ trajectory matrix, $\mathbf{X}$, with elements

$$\mathbf{X} = \{x_{k,l}\}_{k=1, l=1}^{K,L}$$

such that $x_{k,l} = y_{k-l+1}$ for $k = 1, \dots, K$ and $l = 1, \dots, L$ and where $K = T - L + 1$. By definition $L \leq K < T$, because $2 \leq L < T/2$.

Decomposition Step

Using the trajectory matrix, $\mathbf{X}$, apply singular value decomposition to the trajectory matrix

$$\mathbf{X} = \mathbf{U}\mathbf{Q}\mathbf{V}$$

where $\mathbf{U}$ represents the $K \times L$ matrix that contains the left-hand-side (LHS) eigenvectors, where $\mathbf{Q}$ represents the diagonal $L \times L$ matrix that contains the singular values, and where $\mathbf{V}$ represents the $L \times L$ matrix that contains the right-hand-side (RHS) eigenvectors.

Therefore,

$$\mathbf{X} = \sum_{l=1}^L \mathbf{X}^{(l)} = \sum_{l=1}^L \mathbf{u}_l q_l \mathbf{v}_l^T$$

where $\mathbf{X}^{(l)}$ represents the $K \times L$ principal component matrix, $\mathbf{u}_l$ represents the $K \times 1$ left-hand-side (LHS) eigenvector, q_l represents the singular value, and $\mathbf{v}_l$ represents the $L \times 1$ right-hand-side (RHS) eigenvector associated with the l th window index.

Grouping Step

For each group index, $m = 1, \dots, M$, define a group of window indices $I_m \subset \{1, \dots, L\}$. Let

$$\mathbf{X}_{I_m} = \sum_{l \in I_m} \mathbf{X}^{(l)} = \sum_{l \in I_m} \mathbf{u}_l q_l \mathbf{v}_l^T$$

represent the grouped trajectory matrix for group I_m . If groupings represent a spectral partition,

$$\bigcup_{m=1}^M I_m = \{1, \dots, L\} \quad \text{and} \quad I_m \cap I_n = \emptyset \quad \text{for} \quad m \neq n$$

then according to the singular value decomposition theory,

$$\mathbf{X} = \sum_{m=1}^M \mathbf{X}_{I_m}$$

Averaging Step

For each group index, $m = 1, \dots, M$, compute the diagonal average of $\mathbf{X}_{I_m}$,

$$\tilde{x}_t^{(m)} = \frac{1}{n_t} \sum_{l=s_t}^{e_t} x_{t-l+1,l}^{(m)}$$

where

$$\begin{aligned} s_t &= 1, & e_t &= t, & n_t &= t & \text{for} & 1 \leq t < L \\ s_t &= 1, & e_t &= L, & n_t &= L & \text{for} & L \leq t \leq T - L + 1 \\ s_t &= T - t - 1, & e_t &= L, & n_t &= T - t + 1 & \text{for} & T - L + 1 < t \leq T \end{aligned}$$

If the groupings represent a spectral partition, then by definition

$$y_t = \sum_{m=1}^M \tilde{x}_t^{(m)}$$

Hence, singular spectrum analysis additively decomposes the original time series, y_t , into m component series $\tilde{x}_t^{(m)}$ for $m = 1, \dots, M$.

Specifying the Window Length

You can explicitly specify the maximum window length, $2 \leq L \leq 1000$, using the LENGTH= option or implicitly specify the window length using the INTERVAL= option in the ID statement or the SEASONALITY= option in the PROC TIMESERIES statement.

Either way the window length is reduced based on the accumulated time series length, T , to enforce the requirement that $2 \leq L \leq T/2$.

Specifying the Groups

You can use the GROUPS= option to explicitly specify the composition and number of groups, $I_m \subset \{1, \dots, L\}$ or use the THRESHOLDPCT= option in the SSA statement to implicitly specify the grouping. The THRESHOLDPCT= option is useful for removing noise or less dominant patterns from the accumulated time series.

Let $0 < \alpha < 1$ be the cumulative percent singular value THRESHOLDPCT=. Then the last group, $I_M = \{l_\alpha, \dots, L\}$, is determined by the smallest value such that

$$\left(\sum_{l=1}^{l_\alpha-1} q_l / \sum_{l=1}^L q_l \right) \geq \alpha \quad \text{where } 1 < l_\alpha \leq L$$

Using this rule, the last group, I_M , describes the least dominant patterns in the time series and the size of the last group is at least one and is less than the window length, $L \geq 2$.

The magnitudes of the principal components which are plotted using the PLOT=SSA option and selected by the THRESHOLDPCT= option are based on the singular values which appear on the diagonal of $\mathbf{Q}$. Alternatively, each principal component's contribution to variation in the series can be quantified by using the squares of the singular values. The relative contributions of the principal components to variation in the series are included in the printed tabular output produced by the PRINT=SSA option.

Data Set Output

The TIMESERIES procedure can create the OUT=, OUTCORR=, OUTCROSSCORR=, OUTDECOMP=, OUTSEASON=, OUTSPECTRA=, OUTSSA=, OUTSUM=, and OUTTREND= data sets. In general, these data sets contain the variables listed in the BY statement. If an analysis step that is related to an output data step fails, the values of this step are not recorded or are set to missing in the related output data set and appropriate error and/or warning messages are recorded in the log.

OUT= Data Set

The OUT= data set contains the variables specified in the BY, ID, VAR, and CROSSVAR statements. If the ID statement is specified, the ID variable values are aligned and extended based on the ALIGN= and INTERVAL= options. The values of the variables specified in the VAR and CROSSVAR statements are accumulated based on the ACCUMULATE= option, and missing values are interpreted based on the SETMISSING= option.

OUTCORR= Data Set

The OUTCORR= data set contains the variables specified in the BY statement as well as the variables listed below. The OUTCORR= data set records the correlations for each variable specified in a VAR statement (not the CROSSVAR statement).

When the CORR statement TRANSPOSE=NO option is omitted or specified explicitly, the variable *names* are related to correlation statistics specified in the CORR statement options and the variable *values* are related to the NLAG= or LAGS= option.

NAME	variable name
LAG	time lag
N	number of variance products
ACOV	autocovariances
ACF	autocorrelations
ACFSTD	autocorrelation standard errors
ACF2STD	an indicator of whether autocorrelations are less than (–1), greater than (1), or within (0) two standard errors of zero
ACFNORM	normalized autocorrelations
ACFPROB	autocorrelation probabilities
ACFLPROB	autocorrelation log probabilities
PACF	partial autocorrelations
PACFSTD	partial autocorrelation standard errors
PACF2STD	an indicator of whether partial autocorrelations are less than (–1), greater than (1), or within (0) two standard errors of zero
PACFNORM	partial normalized autocorrelations
PACFPROB	partial autocorrelation probabilities
PACFLPROB	partial autocorrelation log probabilities
IACF	inverse autocorrelations
IACFSTD	an indicator of whether inverse autocorrelations are less than (–1), greater than (1), or within (0) two standard errors of zero
IACF2STD	two standard errors beyond inverse autocorrelation
IACFNORM	normalized inverse autocorrelations
IACFPROB	inverse autocorrelation probabilities
IACFLPROB	inverse autocorrelation log probabilities
WN	white noise test Statistics
WNPROB	white noise test probabilities
WNLPROB	white noise test log probabilities

The preceding correlation statistics are computed for each specified time lag.

When the CORR statement TRANSPOSE=YES option is specified, the variable *values* are related to correlation statistics specified in the CORR statement and the variable *names* are related to the NLAG= or LAGS= options.

NAME	variable name
STAT	correlation statistic name
LABEL	correlation statistic label
LAG h	correlation statistics for lag h

OUTCROSSCORR= Data Set

The OUTCROSSCORR= data set contains the variables specified in the BY statement as well as the variables listed below. The OUTCROSSCORR= data set records the cross-correlations for each variable specified in a VAR and the CROSSVAR statements.

When the CROSSCORR statement TRANSPOSE=NO option is omitted or specified explicitly, the variable *names* are related to cross-correlation statistics specified in the CROSSCORR statement options and the variable *values* are related to the NLAG= or LAGS= option.

NAME	variable name
CROSS	cross variable name
LAG	time lag
N	number of variance products
CCOV	cross-covariances
CCF	cross-correlations
CCFSTD	cross-correlation standard errors
CCF2STD	an indicator of whether cross-correlations are less than (-1) , greater than (1) , or within (0) two standard errors of zero
CCFNORM	normalized cross-correlations
CCFPROB	cross-correlation probabilities
CCFLPROB	cross-correlation log probabilities

The preceding cross-correlation statistics are computed for each specified time lag.

When the CROSSCORR statement TRANSPOSE=YES option is specified, the variable *values* are related to cross-correlation statistics specified in the CROSSCORR statement and the variable *names* are related to the NLAG= or LAGS= options.

NAME	variable name
CROSS	cross variable name
STAT	cross-correlation statistic name

<code>_LABEL_</code>	cross-correlation statistic label
<code>LAGh</code>	cross-correlation statistics for lag h

OUTDECOMP= Data Set

The OUTDECOMP= data set contains the variables specified in the BY statement as well as the variables listed below. The OUTDECOMP= data set records the seasonal decomposition/adjustments for each variable specified in a VAR statement (not the CROSSVAR statement).

When the DECOMP statement TRANSPOSE=NO option is omitted or specified explicitly, the variable *names* are related to decomposition/adjustments specified in the DECOMP statement and the variable *values* are related to the ID statement INTERVAL= option and the PROC TIMESERIES statement SEASONALITY= option.

<code>_NAME_</code>	variable name
<code>_MODE_</code>	mode of decomposition
<code>_TIMEID_</code>	time ID values
<code>_SEASON_</code>	seasonal index
<code>ORIGINAL</code>	original series values
<code>TCC</code>	trend-cycle component
<code>SIC</code>	seasonal-irregular component
<code>SC</code>	seasonal component
<code>SCSTD</code>	seasonal component standard errors
<code>TCS</code>	trend-cycle-seasonal component
<code>IC</code>	irregular component
<code>SA</code>	seasonally adjusted series
<code>PCSA</code>	percent change seasonally adjusted series
<code>TC</code>	trend component
<code>CC</code>	cycle component

The preceding decomposition components are computed for each time period.

When the DECOMP statement TRANSPOSE=YES option is specified, the variable *values* are related to decomposition/adjustments specified in the DECOMP statement and the variable *names* are related to the ID statement INTERVAL= option, the PROC TIMESERIES statement SEASONALITY= option, and the DECOMP statement NPERIODS= option.

<code>_NAME_</code>	variable name
<code>_MODE_</code>	mode of decomposition name
<code>_COMP_</code>	decomposition component name
<code>_LABEL_</code>	decomposition component label
<code>PERIODt</code>	decomposition component value for time period t

OUTPROCINFO= Data Set

The OUTPROCINFO= data set contains information about the run of the TIMESERIES procedure. The following variables are present:

SOURCE	set to the name of the procedure, in this case TIMESERIES
NAME	name of the item being reported
LABEL	descriptive label for the item in _NAME_
STAGE	set to the current stage of the procedure; for TIMESERIES this is set to ALL
VALUE	value of the item specified in _NAME_

OUTSEASON= Data Set

The OUTSEASON= data set contains the variables specified in the BY statement as well as the variables listed below. The OUTSEASON= data set records the seasonal statistics for each variable specified in a VAR statement (not the CROSSVAR statement).

When the SEASON statement TRANSPOSE=NO option is omitted or specified explicitly, the variable *names* are related to seasonal statistics specified in the SEASON statement and the variable *values* are related to the ID statement INTERVAL= option or the PROC TIMESERIES statement SEASONALITY= option.

NAME	variable name
TIMEID	time ID values
SEASON	seasonal index
NOBS	number of observations
N	number of nonmissing observations
NMISS	number of missing observations
MINIMUM	minimum value
MAXIMUM	maximum value
RANGE	range value
SUM	summation value
MEAN	mean value
STDDEV	standard deviation
CSS	corrected sum of squares
USS	uncorrected sum of squares
MEDIAN	median value

The preceding statistics are computed for each season.

When the SEASON statement TRANSPOSE=YES option is specified, the variable *values* are related to seasonal statistics specified in the SEASON statement and the variable *names* are related to the ID statement INTERVAL= option or the PROC TIMESERIES statement SEASONALITY= option.

<code>_NAME_</code>	variable name
<code>_STAT_</code>	season statistic name
<code>_LABEL_</code>	season statistic name
<code>SEASONs</code>	season statistic value for season s

OUTSPECTRA= Data Set

The OUTSPECTRA= data set contains the variables that are specified in the BY statement in addition to the variables listed below. The OUTSPECTRA= data set records the frequency domain analysis for each variable specified in a VAR statement (not the CROSSVAR statement).

The following variable names are related to correlation statistics specified in the SPECTRA statement options.

<code>_NAME_</code>	variable name
<code>FREQ</code>	frequency in radians from 0 to π
<code>PERIOD</code>	period or wavelength
<code>COS</code>	cosine transform
<code>SIN</code>	sine transform
<code>P</code>	periodogram
<code>S</code>	spectral density estimates

OUTSSA= Data Set

The OUTSSA= data set contains the variables that are specified in the BY statement in addition to the variables listed below. The OUTSSA= data set records the singular spectrum analysis (SSA) for each variable specified in a VAR statement (not the CROSSVAR statement).

When the SSA statement TRANSPOSE=NO option is omitted or specified explicitly, the variable *names* are related to singular spectrum analysis specified in the SSA statement, and the variable *values* are related to the INTERVAL= option in the ID statement and the SEASONALITY= option in the PROC TIMESERIES statement.

<code>_NAME_</code>	variable name
<code>_TIMEID_</code>	time ID values
<code>_CYCLE_</code>	cycle index
<code>_SEASON_</code>	seasonal index
<code>ORIGINAL</code>	original series values
<code>_GROUPi_</code>	SSA result groups

The `_GROUPi_` decomposition components are computed for each time period.

When the SSA statement `TRANSPPOSE=YES` option is specified, the variable *values* are related to singular spectrum analysis specified in the SSA statement, and the variable *names* are related to the `INTERVAL=` option in the ID statement, the `SEASONALITY=` option in the PROC TIMESERIES statement, or the `NPERIODS=` option in the SSA statement. The following variables are written to a transposed `OUTSSA=` data set:

<code>_NAME_</code>	variable name
<code>_GROUP_</code>	group number
<code>PERIOD_t</code>	SSA group value for time period <i>t</i>

OUTSUM= Data Set

The `OUTSUM=` data set contains the variables specified in the BY statement as well as the variables listed below. The `OUTSUM=` data set records the descriptive statistics for each variable specified in a `VAR` statement (not the `CROSSVAR` statement).

Variables related to descriptive statistics are based on the `ACCUMULATE=` and `SETMISSING=` options in the ID and VAR statements:

<code>_NAME_</code>	variable name
<code>_STATUS_</code>	status flag that indicates whether the requested analyses were successful
<code>NOBS</code>	number of observations
<code>N</code>	number of nonmissing observations
<code>NMISS</code>	number of missing observations
<code>START</code>	the starting date of the time series
<code>END</code>	the ending date of the time series
<code>STARTOBS</code>	the beginning observation of the time series
<code>ENDOBS</code>	the ending observation of the time series
<code>MINIMUM</code>	minimum value
<code>MAXIMUM</code>	maximum value
<code>AVG</code>	average value
<code>STDDEV</code>	standard deviation

The `OUTSUM=` data set contains the descriptive statistics of the (accumulated) time series.

OUTTREND= Data Set

The OUTTREND= data set contains the variables specified in the BY statement as well as the variables listed below. The OUTTREND= data set records the trend statistics for each variable specified in a VAR statement (not the CROSSVAR statement).

When the TREND statement TRANSPOSE=NO option is omitted or explicitly specified, the variable *names* are related to trend statistics specified in the TREND statement and the variable *values* are related to the INTERVAL= option in the ID statement or the SEASONALITY= option in the PROC TIMESERIES statement.

NAME	variable name
TIMEID	time ID values
SEASON	seasonal index
NOBS	number of observations
N	number of nonmissing observations
NMISS	number of missing observations
MINIMUM	minimum value
MAXIMUM	maximum value
RANGE	range value
SUM	summation value
MEAN	mean value
STDDEV	standard deviation
CSS	corrected sum of squares
USS	uncorrected sum of squares
MEDIAN	median value

The preceding statistics are computed for each time period.

When the TREND statement TRANSPOSE=YES option is specified, the variable *values* related to trend statistics specified in the TREND statement and the variable *names* are related to the INTERVAL= option in the ID statement, the SEASONALITY= option in the PROC TIMESERIES statement, or the NPERIODS= option in the TREND statement. The following variables are written to the OUTTREND= data set:

NAME	variable name
STAT	trend statistic name
LABEL	trend statistic name
PERIOD t	trend statistic value for time period t

STATUS Variable Values

The `_STATUS_` variable that appears in the `OUTSUM=` data set contains a code that specifies whether the analysis has been successful or not. The `_STATUS_` variable can take the following values:

0	success
1000	transactional trend statistics failure
2000	transactional seasonal statistics failure
3000	accumulation failure
4000	missing value interpretation failure
6000	series is all missing
7000	transformation failure
8000	differencing failure
9000	unable to compute descriptive statistics
10000	seasonal decomposition failure
11000	correlation analysis failure
15000	singular spectrum analysis failure
16000	spectral analysis failure

Printed Output

The TIMESERIES procedure optionally produces printed output by using the Output Delivery System (ODS). By default, the procedure produces no printed output. All output is controlled by the `PRINT=` and `PRINTDETAILS` options associated with the `PROC TIMESERIES` statement. In general, if an analysis step related to printed output fails, the values of this step are not printed and appropriate error or warning messages or both are recorded in the log. The printed output is similar to the output data set, and these similarities are described below.

<code>PRINT=DECOMP</code>	prints the seasonal decomposition similar to the <code>OUTDECOMP=</code> data set.
<code>PRINT=DESCSTATS</code>	prints a table of descriptive statistics for each variable.
<code>PRINT=SEASONS</code>	prints the seasonal statistics similar to the <code>OUTSEASON=</code> data set.
<code>PRINT=SSA</code>	prints the singular spectrum analysis similar to the <code>OUTSSA=</code> data set.
<code>PRINT=SUMMARY</code>	prints the summary statistics similar to the <code>OUTSUM=</code> data set.
<code>PRINT=TRENDS</code>	prints the trend statistics similar to the <code>OUTTREND=</code> data set.
<code>PRINTDETAILS</code>	prints each table with greater detail.

If `PRINT=SEASONS` and the `PRINTDETAILS` options are both specified, all seasonal statistics are printed.

ODS Table Names

Table 32.4 relates the PRINT= options to ODS tables:

Table 32.4 ODS Tables Produced in PROC TIMESERIES

ODS Table Name	Description	Statement	Option
SeasonalDecomposition	Seasonal decomposition	PRINT	DECOMP
DescStats	Descriptive statistics	PRINT	DESCSTATS
GlobalStatistics	Global statistics	PRINT	SEASONS
SeasonStatistics	Season statistics	PRINT	SEASONS
StatisticsSummary	Statistics summary	PRINT	SUMMARY
TrendStatistics	Trend statistics	PRINT	TRENDS
GlobalStatistics	Global statistics	PRINT	TRENDS
SSASingularValues	SSA singular values	PRINT	SSA
SSAResults	SSA results	PRINT	SSA

The tables are related to a single series within a BY group.

ODS Graphics Names

Statistical procedures use ODS Graphics to create graphs as part of their output. ODS Graphics is described in detail in Chapter 21, “Statistical Graphics Using ODS” (*SAS/STAT User’s Guide*).

Before you create graphs, ODS Graphics must be enabled (for example, with the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section “Enabling and Disabling ODS Graphics” in that chapter.

The overall appearance of graphs is controlled by ODS styles. Styles and other aspects of using ODS Graphics are discussed in the section “A Primer on ODS Statistical Graphics” in that chapter.

This section describes the graphical output produced by the TIMESERIES procedure. PROC TIMESERIES assigns a name to each graph it creates. These names are listed in Table 32.5.

Table 32.5 ODS Graphics Produced by PROC TIMESERIES

ODS Graph Name	Plot Description	Statement	Option
ACFPlot	Autocorrelation function	PLOTS	ACF
ACFNORMPlot	Normalized autocorrelation function	PLOTS	ACF
CCFNORMPlot	Normalized cross-correlation function	CROSSPLOTS	
			CCF
CCFPlot	Cross-correlation function	CROSSPLOTS	
			CCF

Table 32.5 (continued)

ODS Graph Name	Plot Description	Statement	Option
CorrelationPlots	Correlation graphics panel	PLOTS	CORR
CrossSeriesPlot	Cross series plot	CROSSPLOTS	SERIES
CycleComponentPlot	Cycle component	PLOTS	CC
CyclePlot	Seasonal cycles plot	PLOTS	CYCLES
DecompositionPlots	Decomposition graphics panel	PLOTS	DECOMP
IACFPlot	Inverse autocorrelation function	PLOTS	IACF
IACFNORMPlot	Normalized inverse autocorrelation function	PLOTS	IACF
IrregularComponentPlot	Irregular component	PLOTS	IC
PACFPlot	Partial autocorrelation function	PLOTS	PACF
PACFNORMPlot	Standardized partial autocorrelation function	PLOTS	PACF
PercentChangeAdjustedPlot	Percent-change seasonally adjusted	PLOTS	PCSA
Periodogram	Periodogram versus period	PLOTS	PERIODOGRAM
ResidualPlot	Residual time series plot	PLOTS	RESIDUAL
SeasonallyAdjustedPlot	Seasonally adjusted	PLOTS	SA
SeasonalComponentPlot	Seasonal component	PLOTS	SC
SeasonalIrregularComponentPlot	Seasonal-irregular component	PLOTS	SIC
SeriesHistogram	Histogram of series values	PLOTS	HISTOGRAM
SeriesPlot	Time series plot	PLOTS	SERIES
SpectralDensityPlot	Spectral density versus period	PLOTS	SPECTRUM
SSASingularValuesPlot	SSA singular values	PLOTS	SSA
SSAResultsPlot	SSA results	PLOTS	SSA
TrendComponentPlot	Trend component	PLOTS	TC
TrendCycleComponentPlot	Trend-cycle component	PLOTS	TCC
TrendCycleSeasonalPlot	Trend-cycle-seasonal component	PLOTS	TCS

Table 32.5 (continued)

ODS Graph Name	Plot Description	Statement	Option
WhiteNoiseLogProbabilityPlot			
	White noise log probability	PLOTS	WN
WhiteNoiseProbabilityPlot			
	White noise probability	PLOTS	WN

Examples: TIMESERIES Procedure

Example 32.1: Accumulating Transactional Data into Time Series Data

This example illustrates using the TIMESERIES procedure to accumulate time-stamped transactional data that has been recorded at no particular frequency into time series data at a specific frequency. After the time series is created, the various SAS/ETS procedures related to time series analysis, seasonal adjustment/decomposition, modeling, and forecasting can be used to further analyze the time series data.

Suppose that the input data set WORK.RETAIL contains variables STORE and TIMESTAMP and numerous other numeric transaction variables. The BY variable STORE contains values that break up the transactions into groups (BY groups). The time ID variable TIMESTAMP contains SAS date values recorded at no particular frequency. The other data set variables contain the numeric transaction values to be analyzed. It is further assumed that the input data set is sorted by the variables STORE and TIMESTAMP. The following statements form monthly time series from the transactional data based on the median value (ACCUMULATE=MEDIAN) of the transactions recorded with each time period. Also, the accumulated time series values for time periods with no transactions are set to zero instead of to missing (SETMISS=0) and only transactions recorded between the first day of 1998 (START='01JAN1998'D) and last day of 2000 (END='31DEC2000'D) are considered and, if needed, extended to include this range.

```
proc timeseries data=retail out=mseries;
  by store;
  id timestamp interval=month
      accumulate=median
      setmiss=0
      start='01jan1998'd
      end  ='31dec2000'd;
  var item1-item8;
run;
```

The monthly time series data are stored in the data WORK.MSERIES. Each BY group associated with the BY variable STORE contains an observation for each of the 36 months associated with the years 1998, 1999, and 2000. Each observation contains the variable STORE, TIMESTAMP, and each of the analysis variables in the input data set.

After each set of transactions has been accumulated to form corresponding time series, accumulated time series can be analyzed using various time series analysis techniques. For example, exponentially weighted moving averages can be used to smooth each series. The following statements use the EXPAND procedure to smooth the analysis variable named STOREITEM.

```
proc expand data=mseries out=smoothed from=month;
  by store;
  id date;
  convert storeitem=smooth / transform=(ewma 0.1);
run;
```

The smoothed series are stored in the data set WORK.SMOOTHED. The variable SMOOTH contains the smoothed series.

If the time ID variable TIMESTAMP contains SAS datetime values instead of SAS date values, the INTERVAL=, START=, and END= options must be changed accordingly and the following statements could be used:

```
proc timeseries data=retail out=tseries;
  by store;
  id timestamp interval=dtmonth
      accumulate=median
      setmiss=0
      start='01jan1998:00:00:00'dt
      end  ='31dec2000:00:00:00'dt;
  var _numeric_;
run;
```

The monthly time series data are stored in the data WORK.TSERIES, and the time ID values use a SAS datetime representation.

Example 32.2: Trend and Seasonal Analysis

This example illustrates using the TIMESERIES procedure for trend and seasonal analysis of time-stamped transactional data.

Suppose that the data set SASHELP.AIR contains two variables: DATE and AIR. The variable DATE contains sorted SAS date values recorded at no particular frequency. The variable AIR contains the transaction values to be analyzed.

The following statements accumulate the transactional data on an average basis to form a quarterly time series and perform trend and seasonal analysis on the transactions.

```

proc timeseries data=sashelp.air
    out=series
    outtrend=trend
    outseason=season print=seasons;
    id date interval=qtr accumulate=avg;
    var air;
run;

```

The time series is stored in the data set WORK.SERIES, the trend statistics are stored in the data set WORK.TREND, and the seasonal statistics are stored in the data set WORK.SEASON. Additionally, the seasonal statistics are printed (PRINT=SEASONS) and the results of the seasonal analysis are shown in [Output 32.2.1](#).

Output 32.2.1 Seasonal Statistics Table

The TIMESERIES Procedure

Season Statistics for Variable AIR							
Season							
Index	N	Minimum	Maximum	Sum	Mean	Standard	Deviation
1	36	112.0000	419.0000	8963.00	248.9722	95.65189	
2	36	121.0000	535.0000	10207.00	283.5278	117.61839	
3	36	136.0000	622.0000	12058.00	334.9444	143.97935	
4	36	104.0000	461.0000	9135.00	253.7500	101.34732	

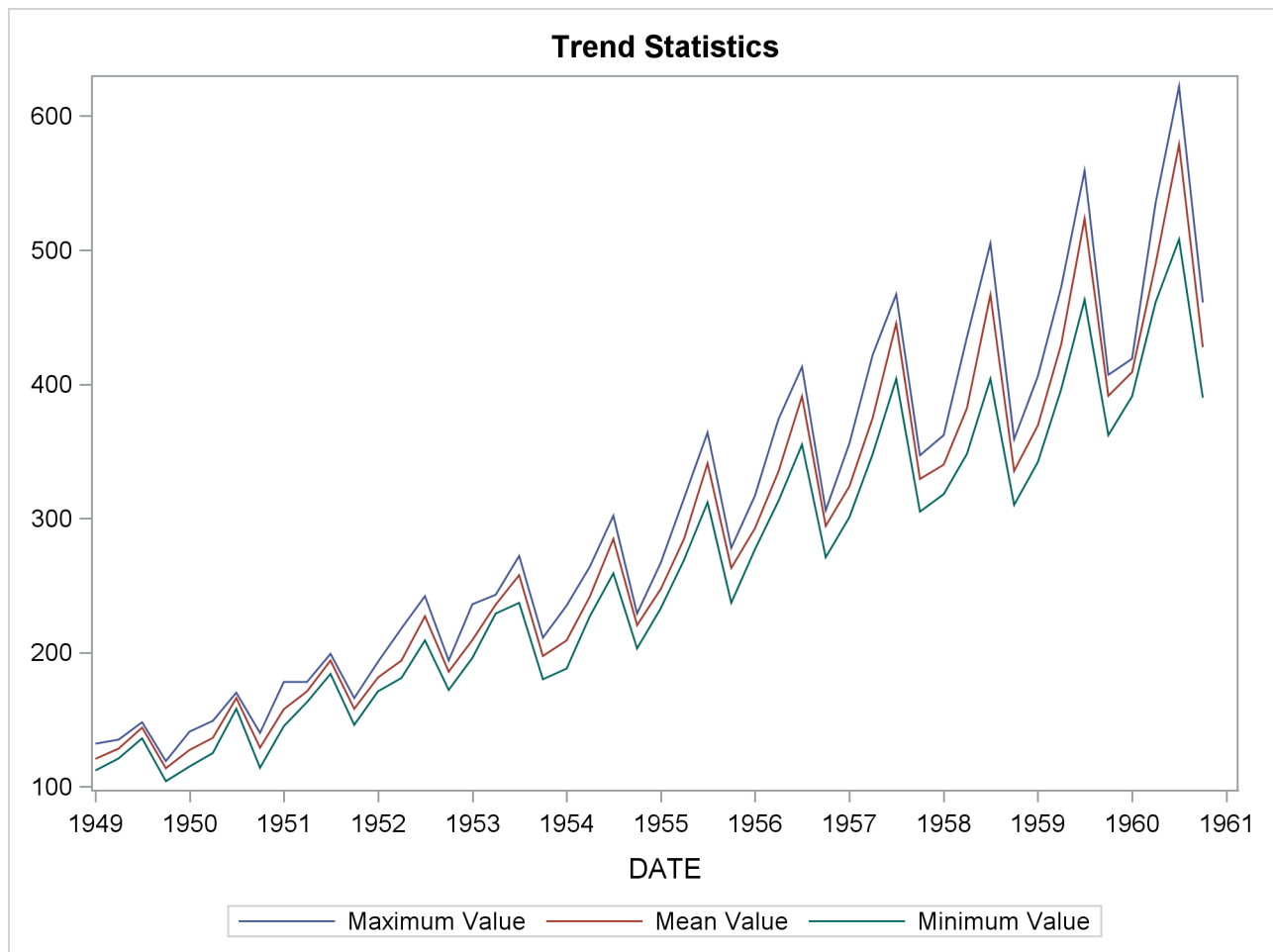
Using the trend statistics stored in the WORK.TREND data set, the following statements plot various trend statistics associated with each time period over time.

```

title1 "Trend Statistics";
proc sgplot data=trend;
    series x=date y=max / lineattrs=(pattern=solid);
    series x=date y=mean / lineattrs=(pattern=solid);
    series x=date y=min / lineattrs=(pattern=solid);
    yaxis display=(nolabel);
    format date year4.;
run;

```

The results of this trend analysis are shown in [Output 32.2.2](#).

Output 32.2.2 Trend Statistics Plot

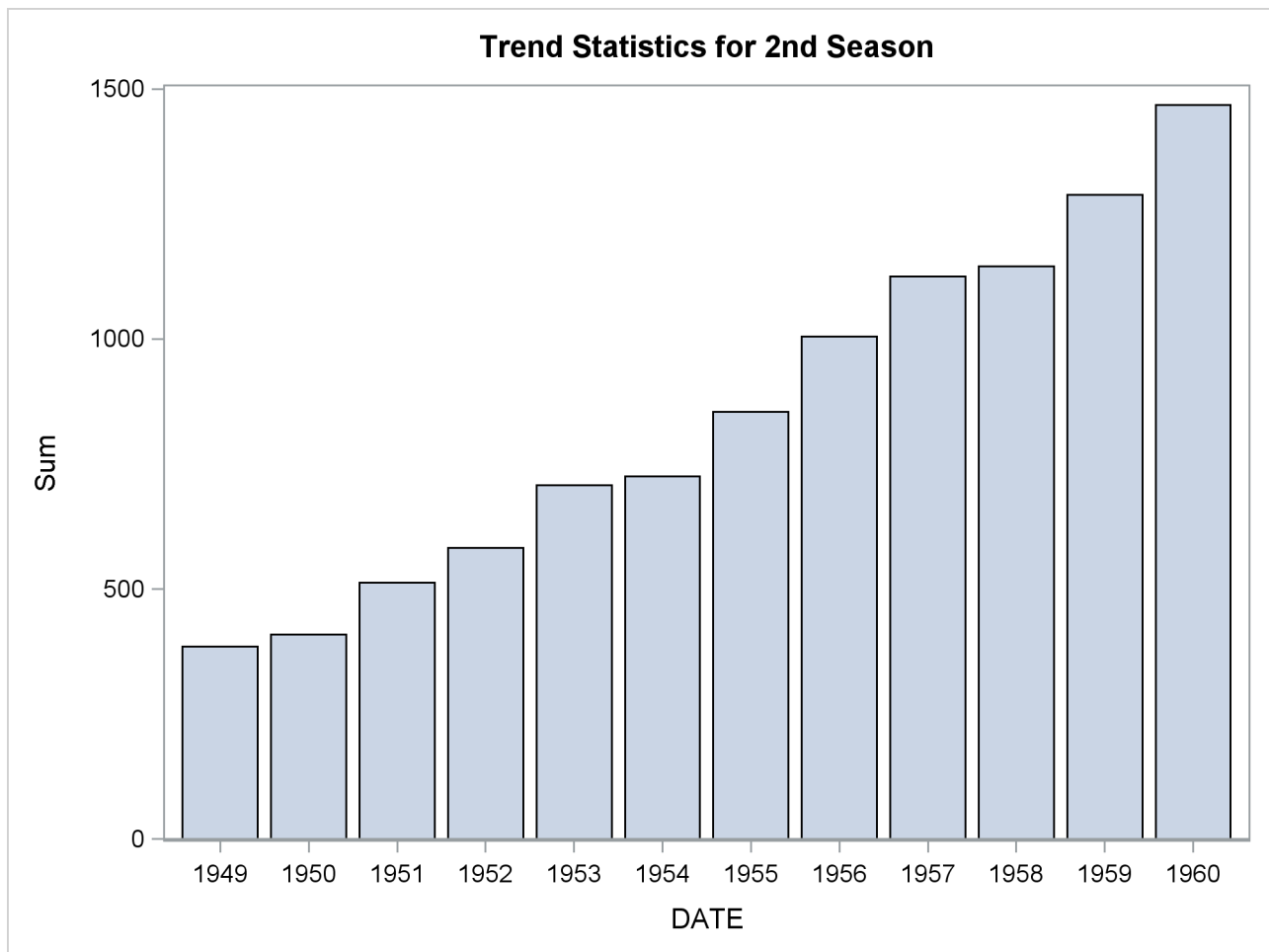
Using the trend statistics stored in the WORK.TREND data set, the following statements chart the sum of the transactions associated with each time period for the second season over time.

```

title1 "Trend Statistics for 2nd Season";
proc sgplot data=trend;
  where _season_ = 2;
  vbar date / freq=sum;
  format date year4.;
  yaxis label='Sum';
run;

```

The results of this trend analysis are shown in [Output 32.2.3](#).

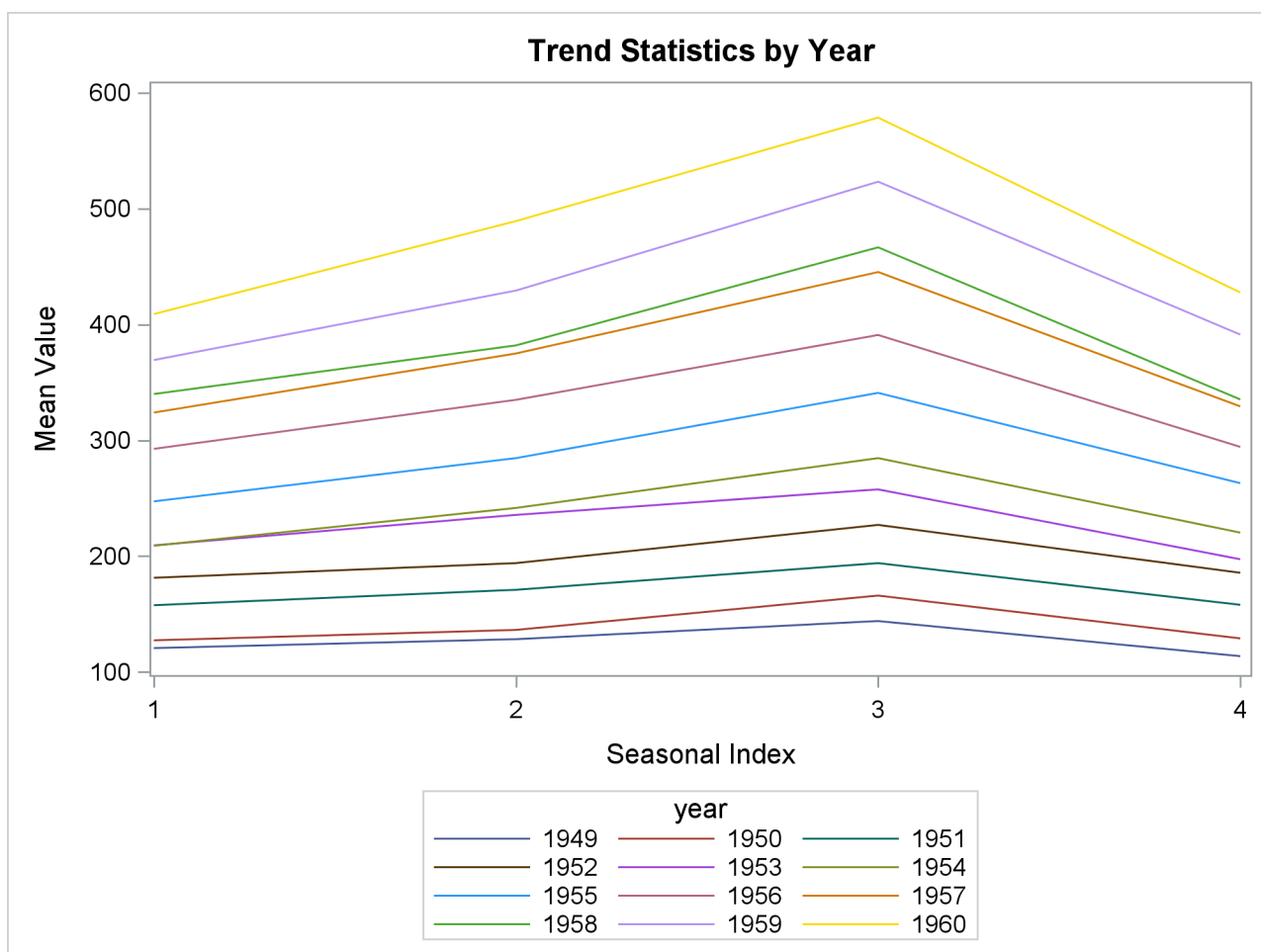
Output 32.2.3 Trend Statistics Bar Chart

Using the trend statistics stored in the WORK.TREND data set, the following statements plot the mean of the transactions associated with each time period by each year over time.

```
data trend;
  set trend;
  year = year(date);
run;

title1 "Trend Statistics by Year";
proc sgplot data=trend;
  series x=_season_ y=mean / group=year lineattrs=(pattern=solid);
  xaxis values=(1 to 4 by 1);
run;
```

The results of this trend analysis are shown in [Output 32.2.4](#).

Output 32.2.4 Trend Statistics

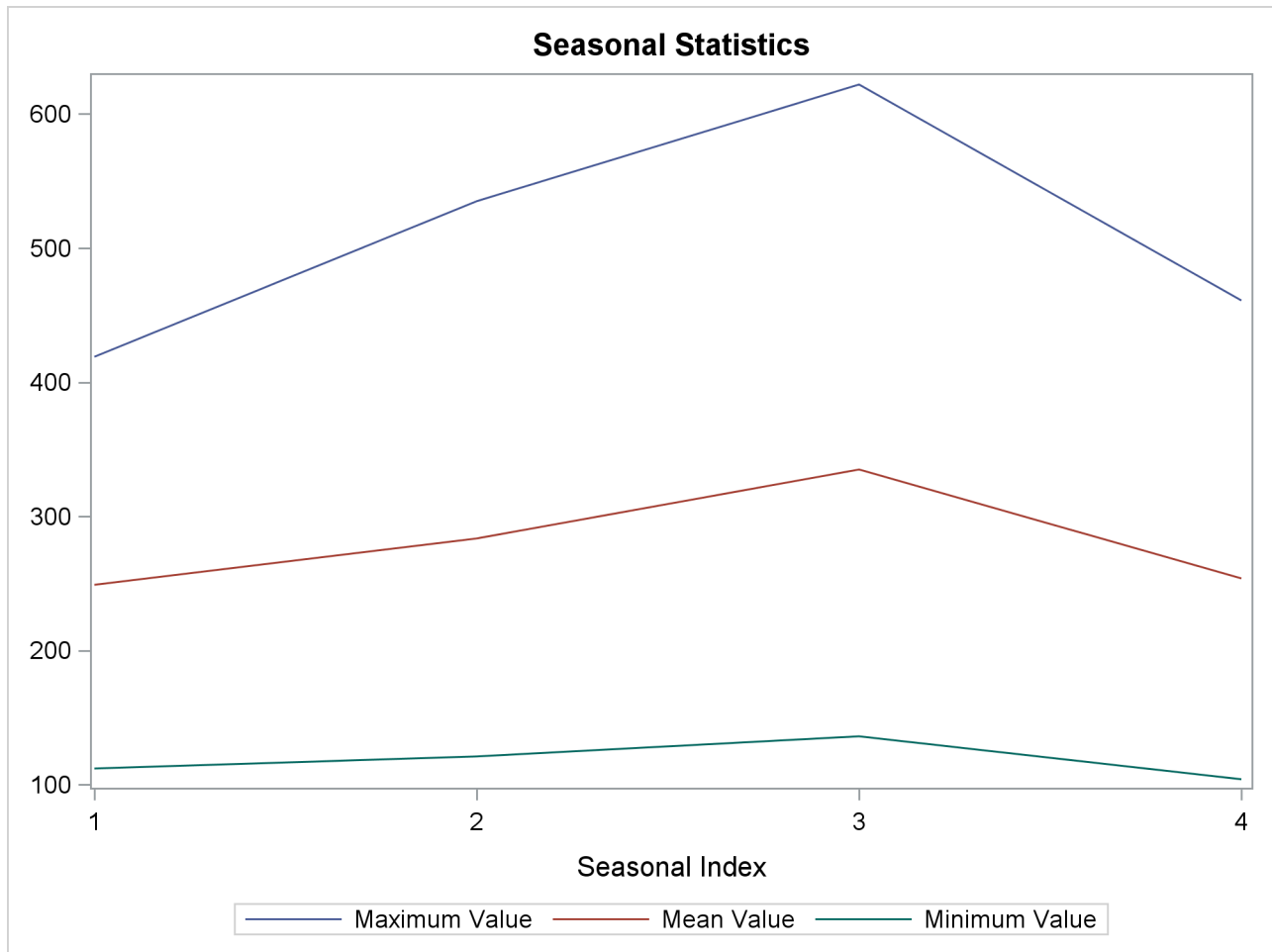
Using the season statistics stored in the WORK.SEASON data set, the following statements plot various season statistics for each season.

```

title1 "Seasonal Statistics";
proc sgplot data=season;
  series x=_season_ y=max / lineattrs=(pattern=solid);
  series x=_season_ y=mean / lineattrs=(pattern=solid);
  series x=_season_ y=min / lineattrs=(pattern=solid);
  yaxis display=(nolabel);
  xaxis values=(1 to 4 by 1);
run;

```

The results of this seasonal analysis are shown in [Output 32.2.5](#).

Output 32.2.5 Seasonal Statistics Plot

Example 32.3: Illustration of ODS Graphics

This example illustrates the use of ODS graphics.

The following statements use the SASHELP.WORKERS data set to study the time series of electrical workers and its interaction with the simply differenced series of masonry workers. The series plot, the correlation panel, the seasonal adjustment panel, and all cross-series plots are requested. [Output 32.3.1](#) through [Output 32.3.4](#) show a selection of the plots created.

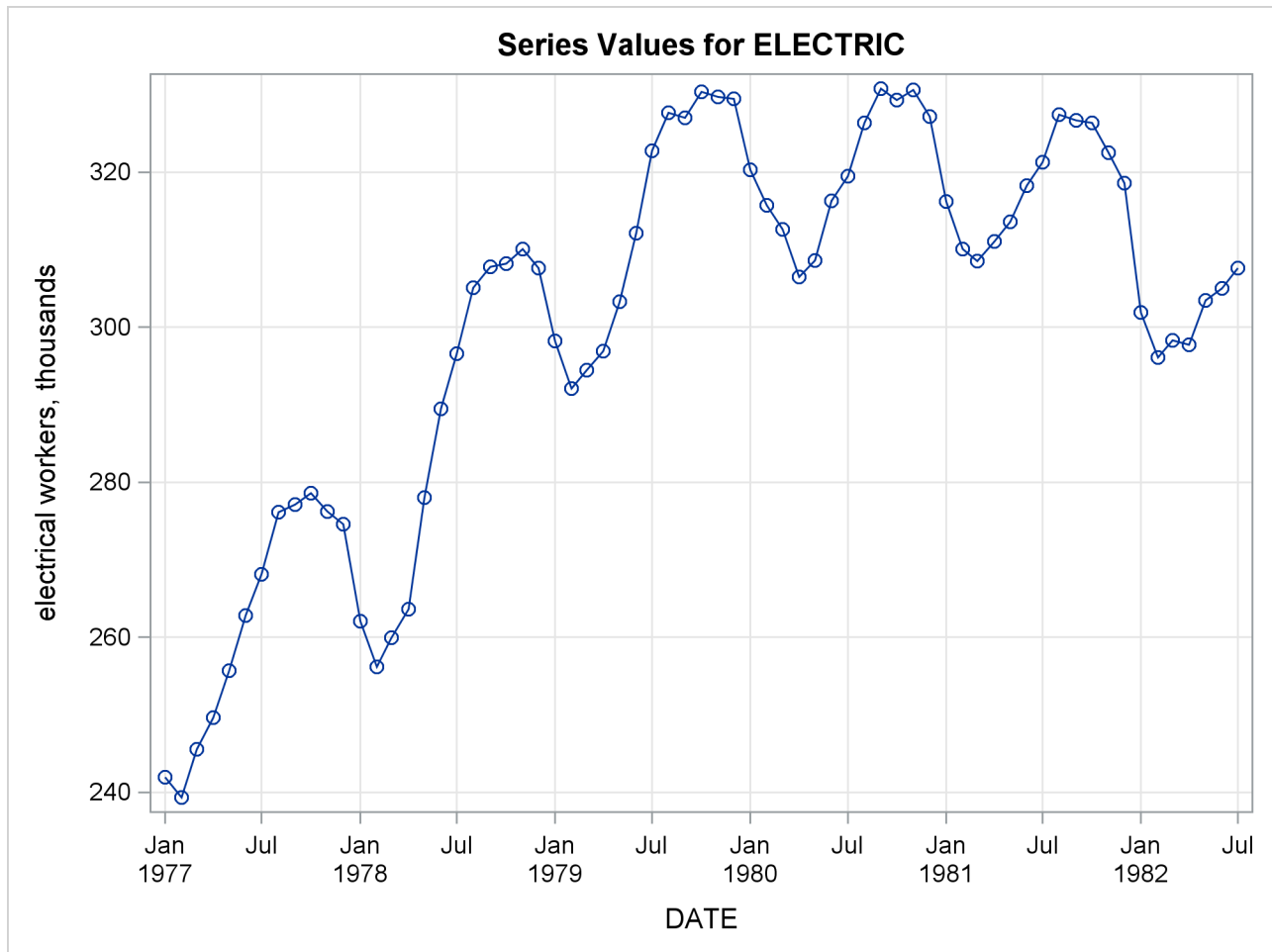
The graphical displays are requested by specifying the **PLOTS=** or **CROSSPLOTS=** options in the PROC TIMESERIES statement. For information about the graphics available in the TIMESERIES procedure, see the section “[ODS Graphics Names](#)” on page 2269.

```

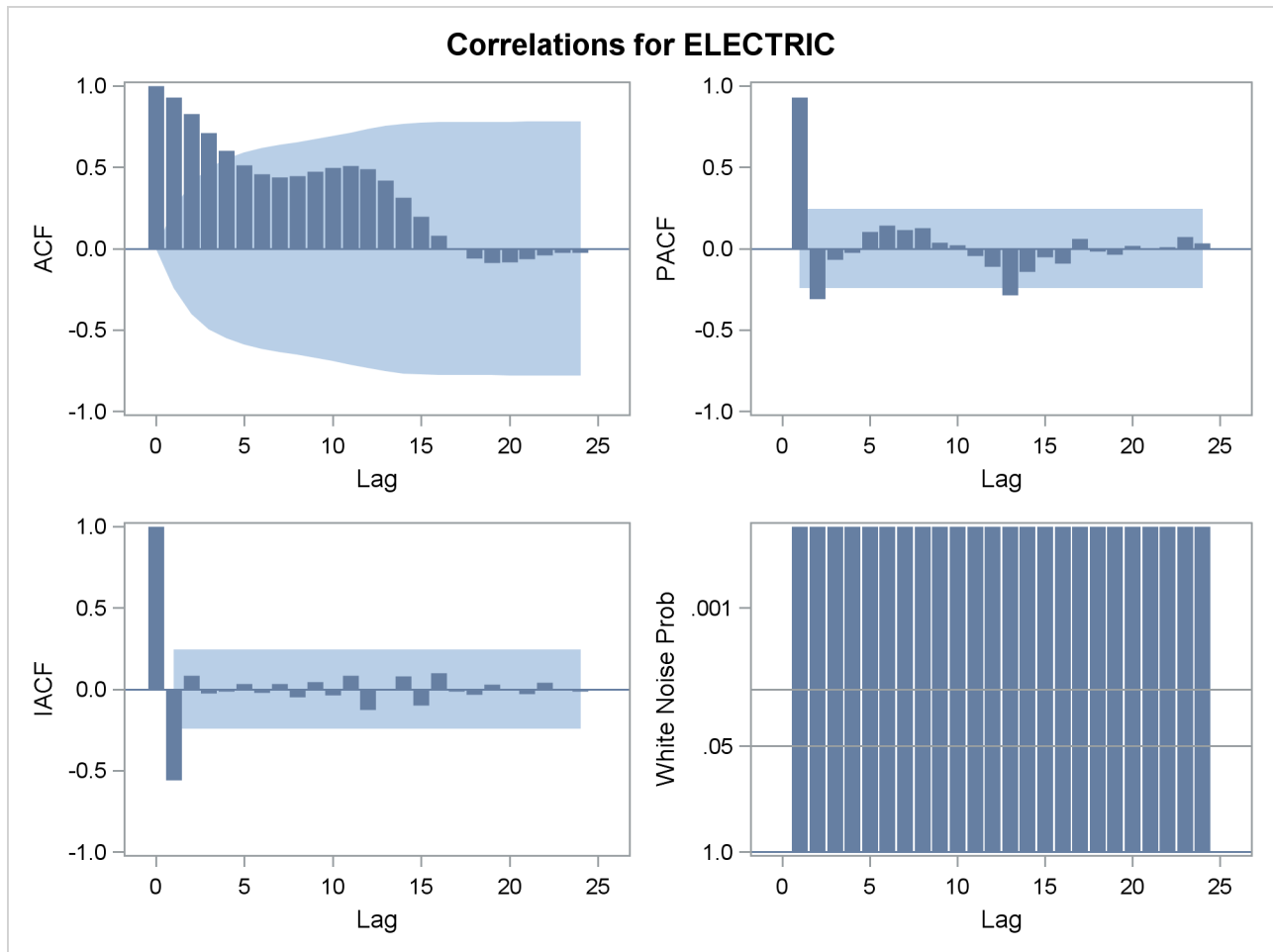
title "Illustration of ODS Graphics";
proc timeseries data=sashelp.workers out=_null_
    plots=(series corr decomp)
    crossplots=all;
    id date interval=month;
    var electric;
    crossvar masonry / dif=(1);
run;

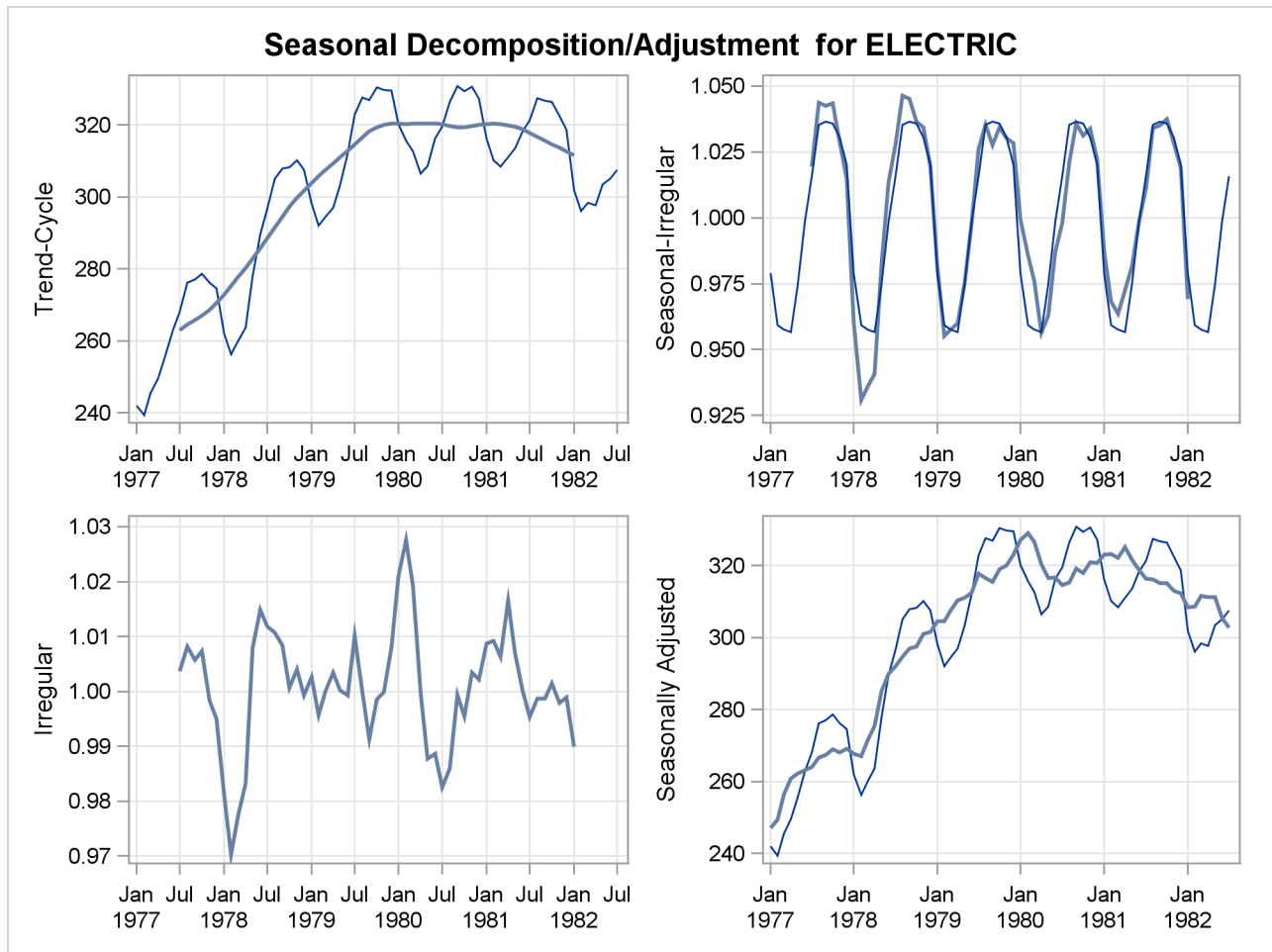
```

Output 32.3.1 Series Plot

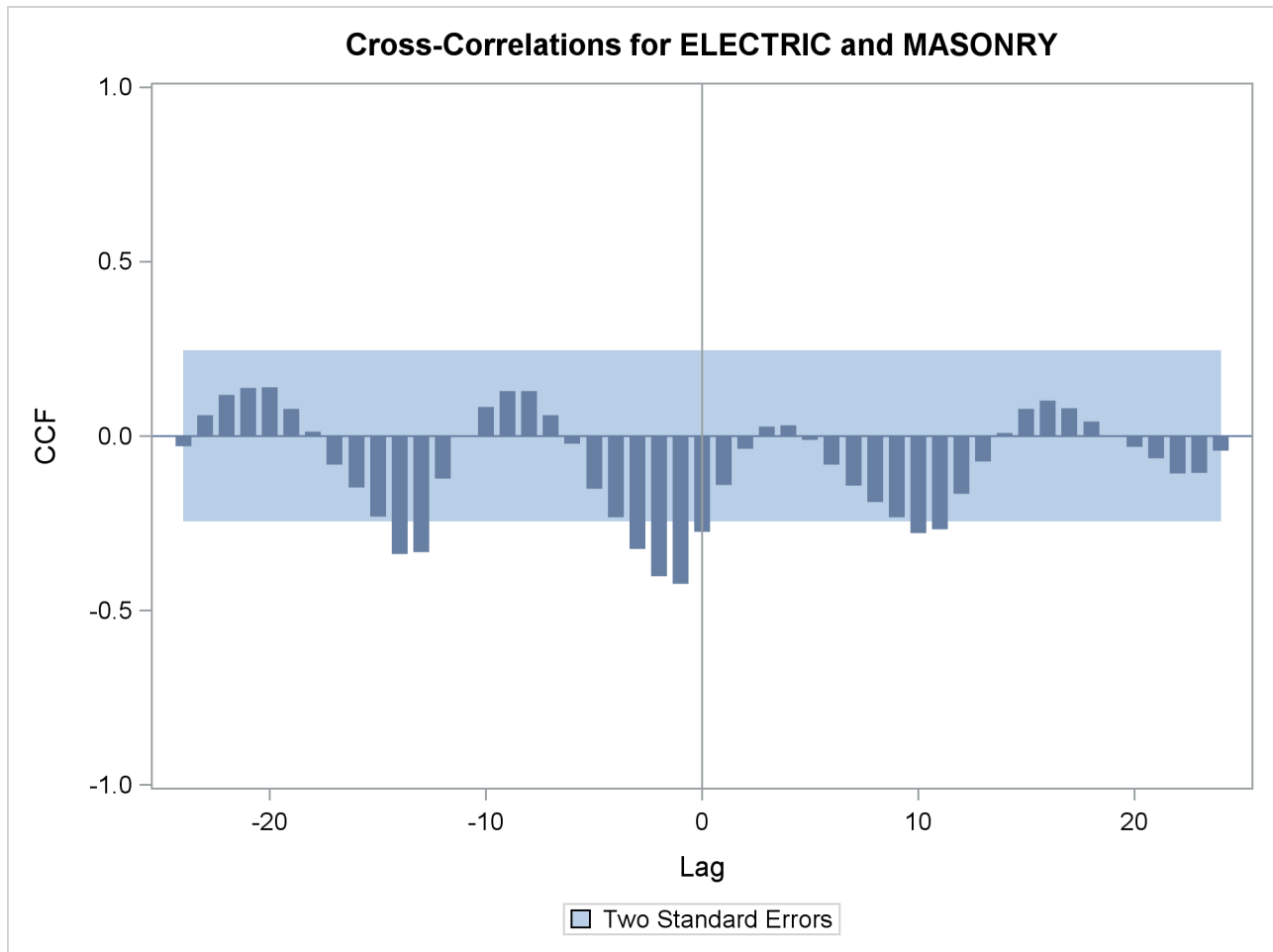


Output 32.3.2 Correlation Panel



Output 32.3.3 Seasonal Decomposition Panel

Output 32.3.4 Cross-Correlation Plot



Example 32.4: Illustration of Spectral Analysis

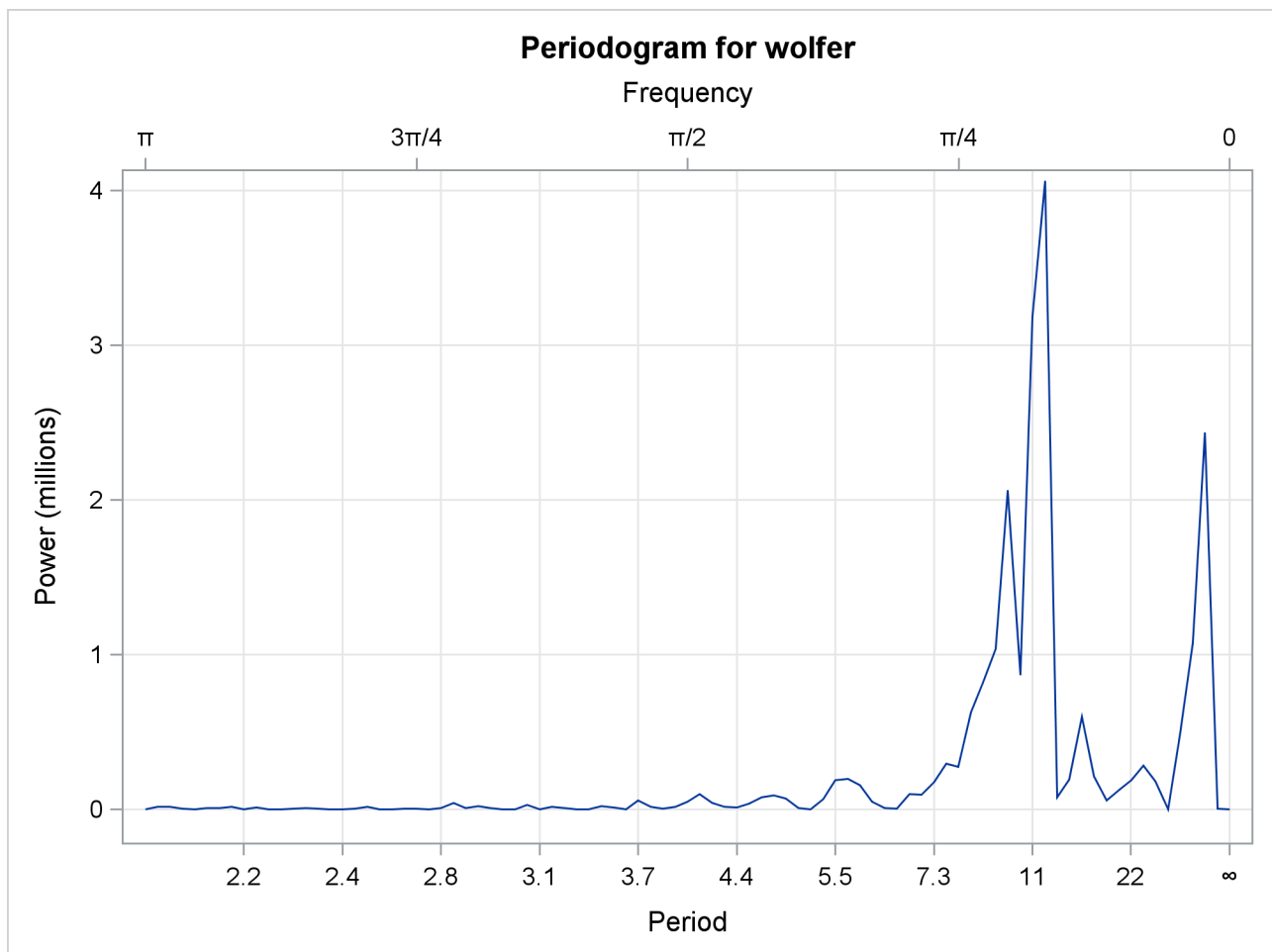
This example illustrates the use of spectral analysis.

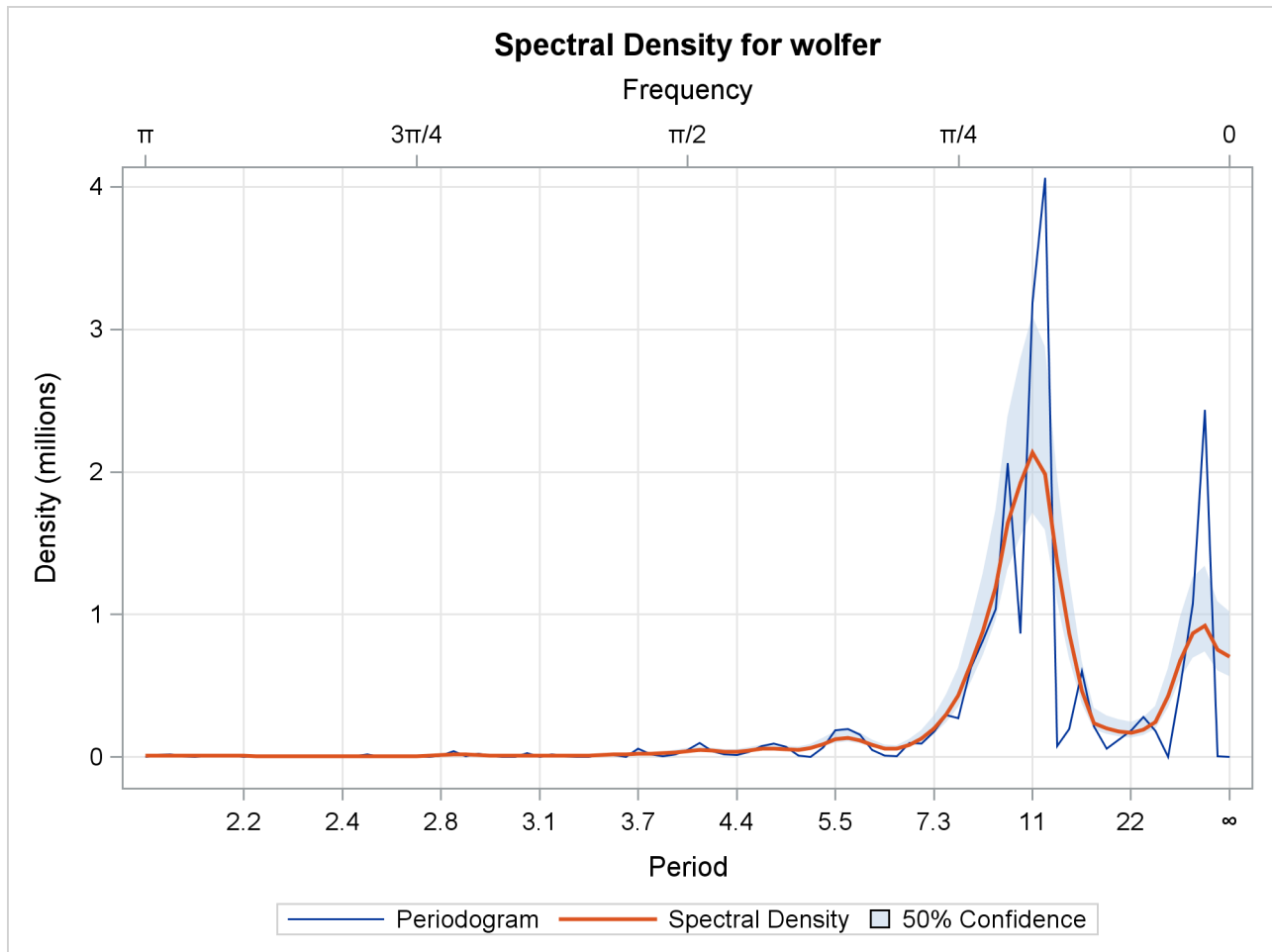
The following statements perform a spectral analysis on the SUNSPOT dataset. The periodogram is displayed as a function of the period and frequency in [Output 32.4.1](#). The estimated spectral density together with its 50% confidence limits are displayed in [Output 32.4.2](#).

```
title "Wolfer's Sunspot Data";

proc timeseries data=sunspot plot=(series periodogram spectrum);
  var wolfer;
  id year interval=year;
  spectra freq period p s / adjmean bart c=1.5 expon=0.2;
run;
```

Output 32.4.1 Periodogram



Output 32.4.2 Spectral Density Plot

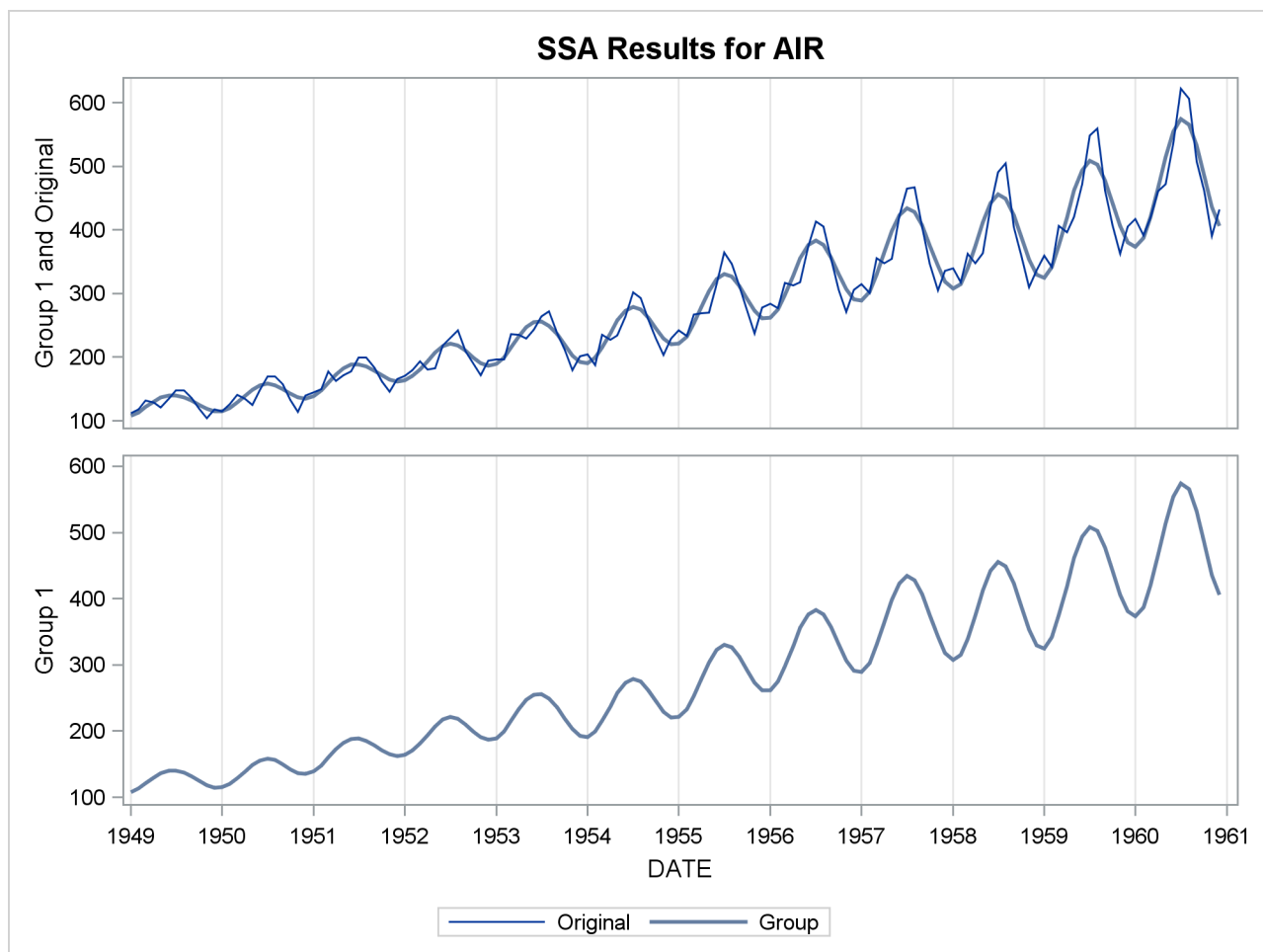
Example 32.5: Illustration of Singular Spectrum Analysis

This example illustrates the use of singular spectrum analysis.

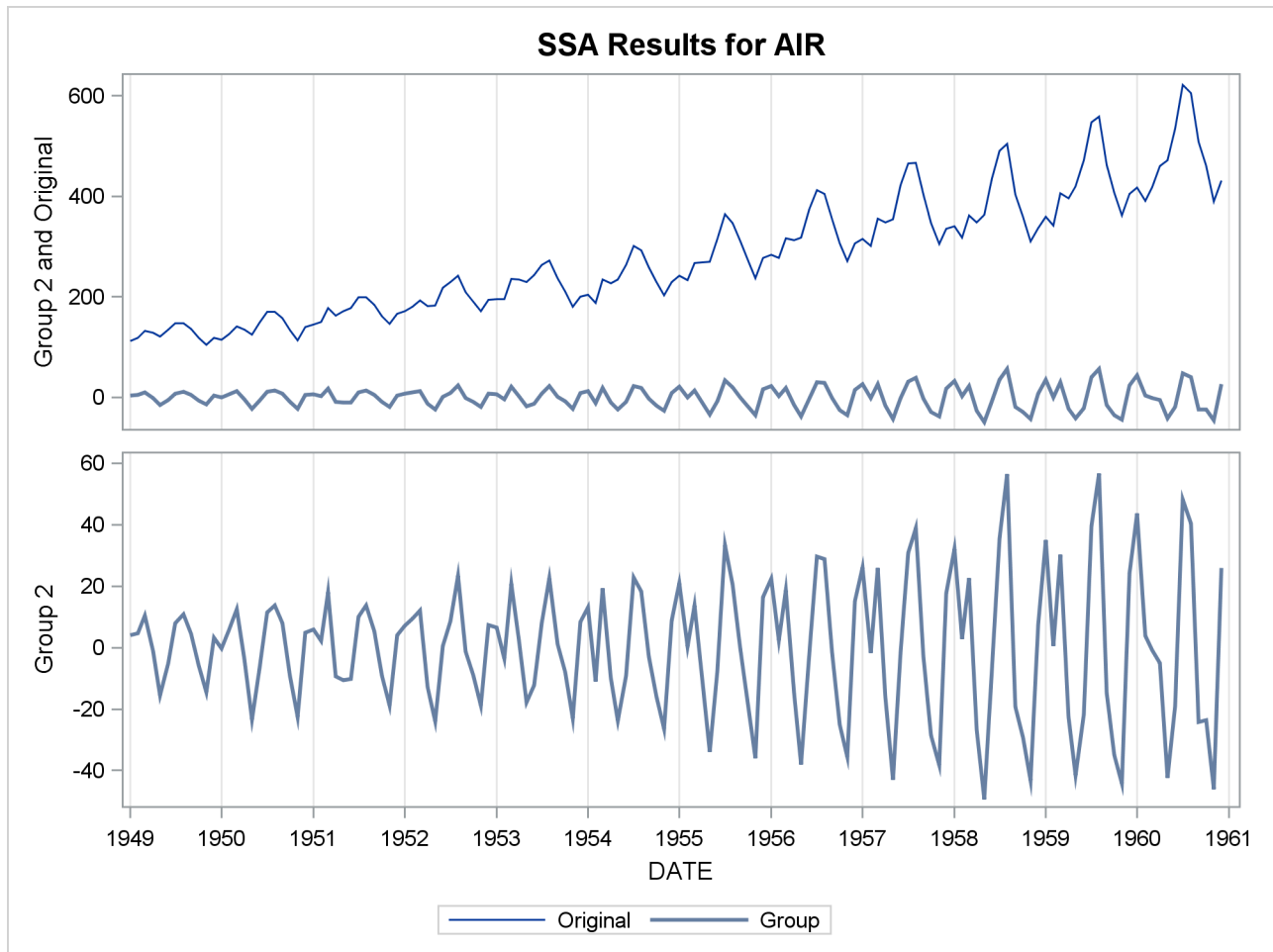
The following statements extract two additive components from the SASHELP.AIR time series by using the THRESHOLDPCT= option to specify that the first component represent 80% of the variability in the series. The resulting groupings, consisting of the first three and remaining nine singular value components, are presented in [Output 32.5.1](#) through [Output 32.5.3](#).

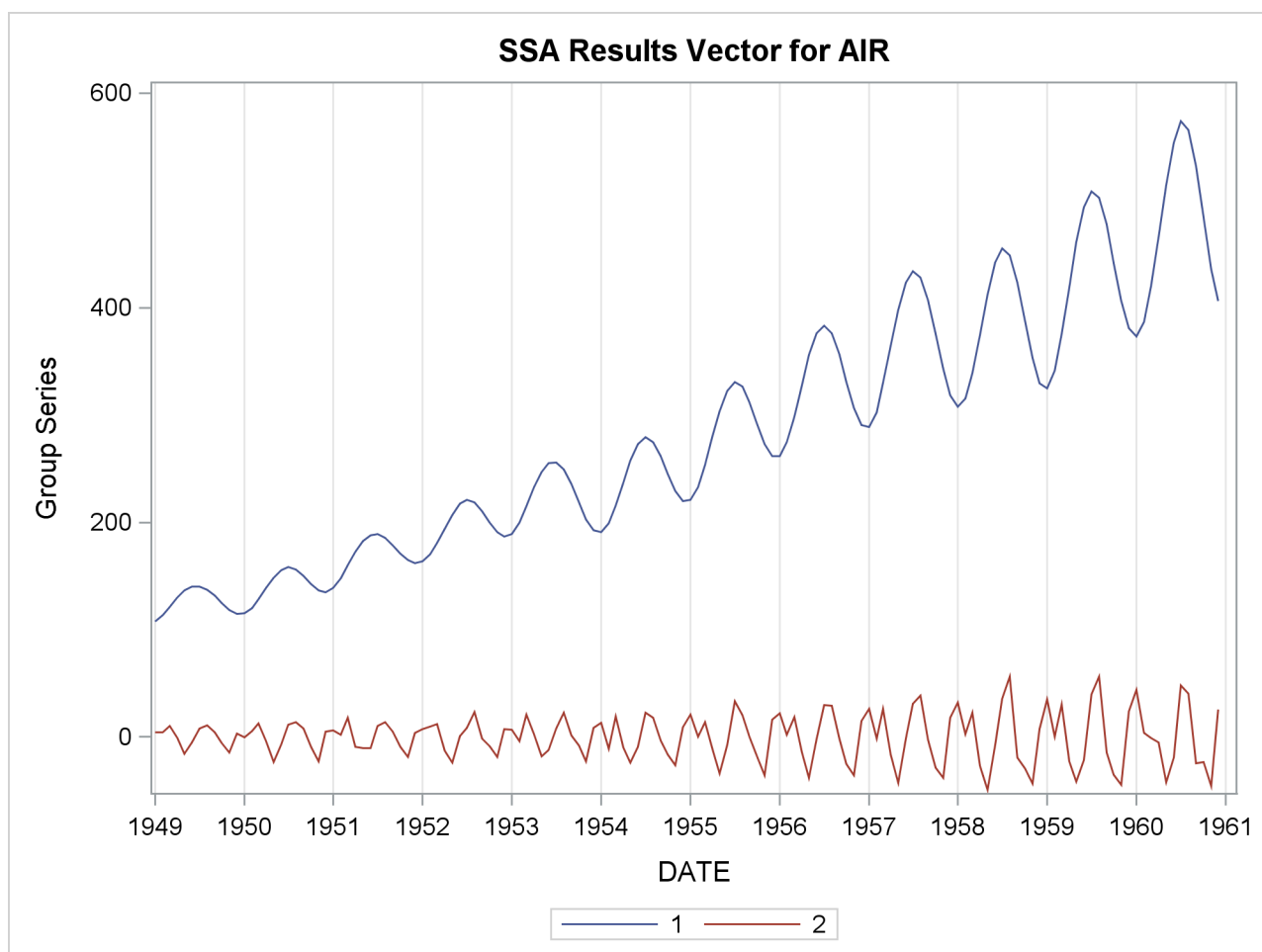
```
title "SSA of AIR data";

proc timeseries data=sashelp.air plot=ssa;
  id date interval=month;
  var air;
  ssa / length=12 THRESHOLDPCT=80;
run;
```

Output 32.5.1 Singular Value Grouping #1 Plot

Output 32.5.2 Singular Value Grouping #2 Plot



Output 32.5.3 Singular Value Components Plot

References

- Brockwell, P. J. and Davis, R. A. (1991), *Time Series: Theory and Methods*, 2nd Edition, New York: Springer-Verlag.
- Cooley, J. W. and Tukey, J. W. (1965), "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, 19, 297–301.
- Golyandina, N., Nekrutkin, V., and Zhigljavsky, A. (2001), *Analysis of Time Series Structure SSA and Related Techniques*, Boca Raton, FL: CRC Press.
- Greene, W. H. (2000), *Econometric Analysis*, 4th Edition, Upper Saddle River, NJ: Prentice-Hall.
- Hodrick, R. J. and Prescott, E. C. (1980), "Post-war U.S. Business Cycles: An Empirical Investigation," Discussion Paper 451, Carnegie Mellon University.
- Makridakis, S. G. and Wheelwright, S. C. (1978), *Interactive Forecasting: Univariate and Multivariate Methods*, 2nd Edition, San Francisco: Holden-Day.

- Monro, D. M. and Branch, J. L. (1977), "Algorithm AS 117: The Chirp Discrete Fourier Transform of General Length," *Journal of the Royal Statistical Society, Series C*, 26, 351–361.
- Priestley, M. B. (1981), *Spectral Analysis and Time Series*, London: Academic Press.
- Pyle, D. (1999), *Data Preparation for Data Mining*, San Francisco: Morgan Kaufmann.
- Singleton, R. C. (1969), "An Algorithm for Computing the Mixed Radix Fast Fourier Transform," *IEEE Transactions on Audio and Electroacoustics*, 17, 93–103.
- Stoffer, D. S. and Tolo, C. M. C. (1992), "A Note on the Ljung-Box-Pierce Portmanteau Statistic with Missing Data," *Statistics and Probability Letters*, 13, 391–396.
- Wheelwright, S. C. and Makridakis, S. G. (1973), *Forecasting Methods for Management*, 3rd Edition, New York: Wiley-Interscience.

Subject Index

BY groups

 TIMESERIES procedure, [2230](#)

ODS graph names

 TIMESERIES procedure, [2269](#)

TIMESERIES procedure

 BY groups, [2230](#)

 ODS graph names, [2269](#)

Syntax Index

- ACCUMULATE= option
 - ID statement (TIMESERIES), [2235](#)
 - VAR statement (TIMESERIES), [2245](#)
- ADJUSTMEAN
 - SPECTRA statement (TIMESERIES), [2239](#)
 - SSA statement (TIMESERIES), [2241](#)
- ALIGN= option
 - ID statement (TIMESERIES), [2236](#)
- ALPHA option
 - SPECTRA statement (TIMESERIES), [2240](#)
- BOUNDARYALIGN= option
 - ID statement (TIMESERIES), [2236](#)
- BY statement
 - TIMESERIES procedure, [2230](#)
- C
 - SPECTRA statement (TIMESERIES), [2239](#)
- CENTER
 - SPECTRA statement (TIMESERIES), [2239](#)
 - SSA statement (TIMESERIES), [2241](#)
- CORR statement
 - TIMESERIES procedure, [2230](#)
- CROSSCORR statement
 - TIMESERIES procedure, [2232](#)
- CROSSPLOTS= option
 - PROC TIMESERIES statement, [2227](#)
- CROSSVAR statement
 - TIMESERIES procedure, [2244](#)
- DATA= option
 - PROC TIMESERIES statement, [2227](#)
- DECOMP statement
 - TIMESERIES procedure, [2233](#)
- DIF= option
 - VAR statement (TIMESERIES), [2245](#)
- DOMAIN option
 - SPECTRA statement (TIMESERIES), [2240](#)
- E
 - SPECTRA statement (TIMESERIES), [2239](#)
- END= option
 - ID statement (TIMESERIES), [2236](#)
- FORMAT= option
 - ID statement (TIMESERIES), [2237](#)
- GROUPS option
 - SSA statement (TIMESERIES), [2241](#)
- ID statement
 - TIMESERIES procedure, [2235](#)
- INTERVAL= option
 - ID statement (TIMESERIES), [2237](#)
- Kernel option values
 - SPECTRA statement (TIMESERIES), [2240](#)
- LAGS= option
 - CORR statement (TIMESERIES), [2232](#)
 - CROSSCORR statement (TIMESERIES), [2233](#)
- LAMBDA= option
 - DECOMP statement (TIMESERIES), [2234](#)
- LENGTH
 - SSA statement (TIMESERIES), [2242](#)
- MAXERROR= option
 - PROC TIMESERIES statement, [2227](#)
- MAXVARLENGTH option
 - PROC TIMESERIES statement, [2227](#)
- MODE= option
 - DECOMP statement (TIMESERIES), [2234](#)
- NLAG= option
 - CORR statement (TIMESERIES), [2231](#)
 - CROSSCORR statement (TIMESERIES), [2233](#)
- NOTSORTED option
 - ID statement (TIMESERIES), [2237](#)
- NPARMS= option
 - CORR statement (TIMESERIES), [2232](#)
- NPERIODS option
 - SSA statement (TIMESERIES), [2242](#)
- NPERIODS= option
 - DECOMP statement (TIMESERIES), [2234](#)
 - TREND statement (TIMESERIES), [2244](#)
- OUT= option
 - PROC TIMESERIES statement, [2227](#)
- OUTCORR= option
 - PROC TIMESERIES statement, [2227](#)
- OUTCROSSCORR= option
 - PROC TIMESERIES statement, [2227](#)
- OUTDECOMP= option
 - PROC TIMESERIES statement, [2227](#)
- OUTPROCINFO= option
 - PROC TIMESERIES statement, [2228](#)
- OUTSEASON= option
 - PROC TIMESERIES statement, [2228](#)
- OUTSPECTRA= option

- PROC TIMESERIES statement, [2228](#)
- OUTSSA= option
 - PROC TIMESERIES statement, [2228](#)
- OUTSUM= option
 - PROC TIMESERIES statement, [2228](#)
- OUTTREND= option
 - PROC TIMESERIES statement, [2228](#)
- PLOTS= option
 - PROC TIMESERIES statement, [2228](#)
- PRINT= option
 - PROC TIMESERIES statement, [2229](#)
- PRINTDETAILS option
 - PROC TIMESERIES statement, [2230](#)
- PROC TIMESERIES statement, [2227](#)
- SDIF= option
 - VAR statement (TIMESERIES), [2245](#)
- SEASON statement
 - TIMESERIES procedure, [2238](#)
- SEASONALITY= option
 - PROC TIMESERIES statement, [2230](#)
- SETMISSING= option
 - ID statement (TIMESERIES), [2237](#)
 - VAR statement (TIMESERIES), [2245](#)
- SORTNAMES option
 - PROC TIMESERIES statement, [2230](#)
- SPECTRA statement
 - TIMESERIES procedure, [2239](#)
- SSA statement
 - TIMESERIES procedure, [2241](#)
- START= option
 - ID statement (TIMESERIES), [2237](#)
- THRESHOLDPCT option
 - SSA statement (TIMESERIES), [2242](#)
- TIMESERIES procedure, [2224](#)
 - syntax, [2224](#)
- TRANSFORM= option
 - VAR statement (TIMESERIES), [2245](#)
- TRANSPPOSE option
 - SSA statement (TIMESERIES), [2243](#)
- TRANSPPOSE= option
 - CORR statement (TIMESERIES), [2232](#)
 - CROSSCORR statement (TIMESERIES), [2233](#)
 - DECOMP statement (TIMESERIES), [2234](#)
 - SEASON statement (TIMESERIES), [2238](#)
 - TREND statement (TIMESERIES), [2244](#)
- TREND statement
 - TIMESERIES procedure, [2243](#)
- VAR statement
 - TIMESERIES procedure, [2244](#)
- WEIGHTS option
 - SPECTRA statement (TIMESERIES), [2240](#)