SAS/STAT® 13.1 User's Guide

The KDE Procedure

This document is an individual chapter from *SAS/STAT® 13.1 User's Guide*.

The correct bibliographic citation for the complete manual is as follows: SAS Institute Inc. 2013. *SAS/STAT® 13.1 User's Guide*. Cary, NC: SAS Institute Inc.

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513-2414.

December 2013

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.

Chapter 52
The KDE Procedure

Contents

Overview: KDE Procedure

The KDE procedure performs univariate and bivariate kernel density estimation. Statistical *density estimation* involves approximating a hypothesized probability density function from observed data. *Kernel density estimation* is a nonparametric technique for density estimation in which a known density function (the *kernel*) is averaged across the observed data points to create a smooth approximation. PROC KDE uses a Gaussian density as the kernel, and its assumed variance determines the smoothness of the resulting estimate. See Silverman (1986) for a thorough review and discussion.

You can use PROC KDE to compute a variety of common statistics, including estimates of the percentiles of the hypothesized probability density function. You can produce a variety of plots, including univariate and bivariate histograms, plots of the kernel density estimates, and contour plots. You can also save kernel density estimates into SAS data sets.

Getting Started: KDE Procedure

The following example illustrates the basic features of PROC KDE. Assume that 1000 observations are simulated from a bivariate normal density with means $(0, 0)$, variances $(10, 10)$, and covariance 9. The SAS DATA step to accomplish this is as follows:

```
data bivnormal;
   seed = 1283470;
   do i = 1 to 1000;
      z1 = rannor(seed);
      z2 = rannor(seed);
      z3 = rannor(seed);
      x = 3*z1+z2;
      y = 3*z1+z3;
      output;
   end;
   drop seed;
run;
```

The following statements request a bivariate kernel density estimate for the variables `x` and `y`, with contour and surface plots:

```
ods graphics on;
proc kde data=bivnormal;
   bivar x y / plots=(contour surface);
run;
ods graphics off;
```

The contour plot and the surface plot of the estimate are displayed in Figure 52.1 and Figure 52.2, respectively. Note that the correlation of 0.9 in the original data results in oval-shaped contours. These graphs are produced by specifying the PLOTS= option in the BIVAR statement with ODS Graphics enabled. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the KDE procedure, see the section "ODS Graphics" on page 4089.

Figure 52.1 Contour Plot of Estimated Density

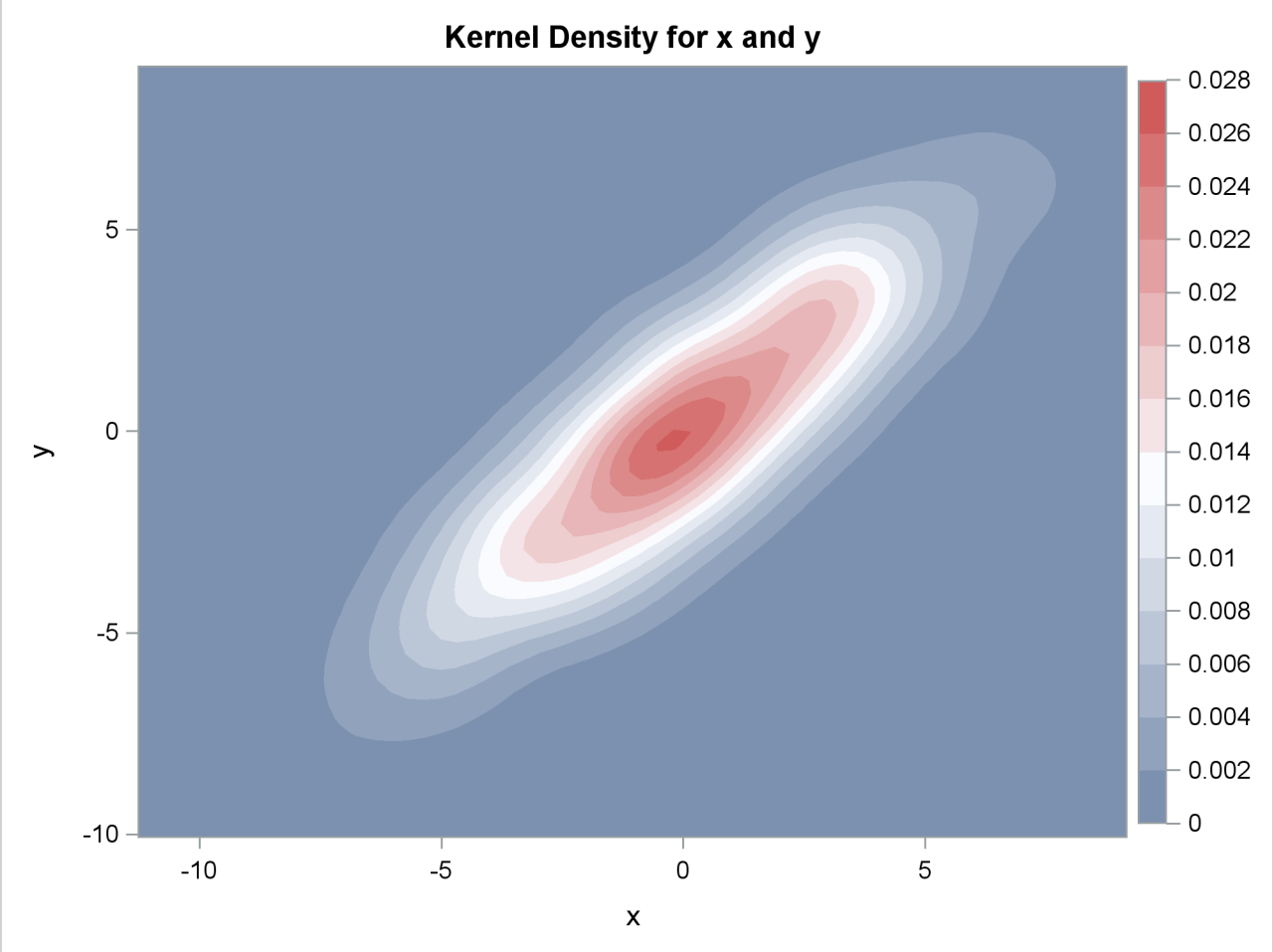
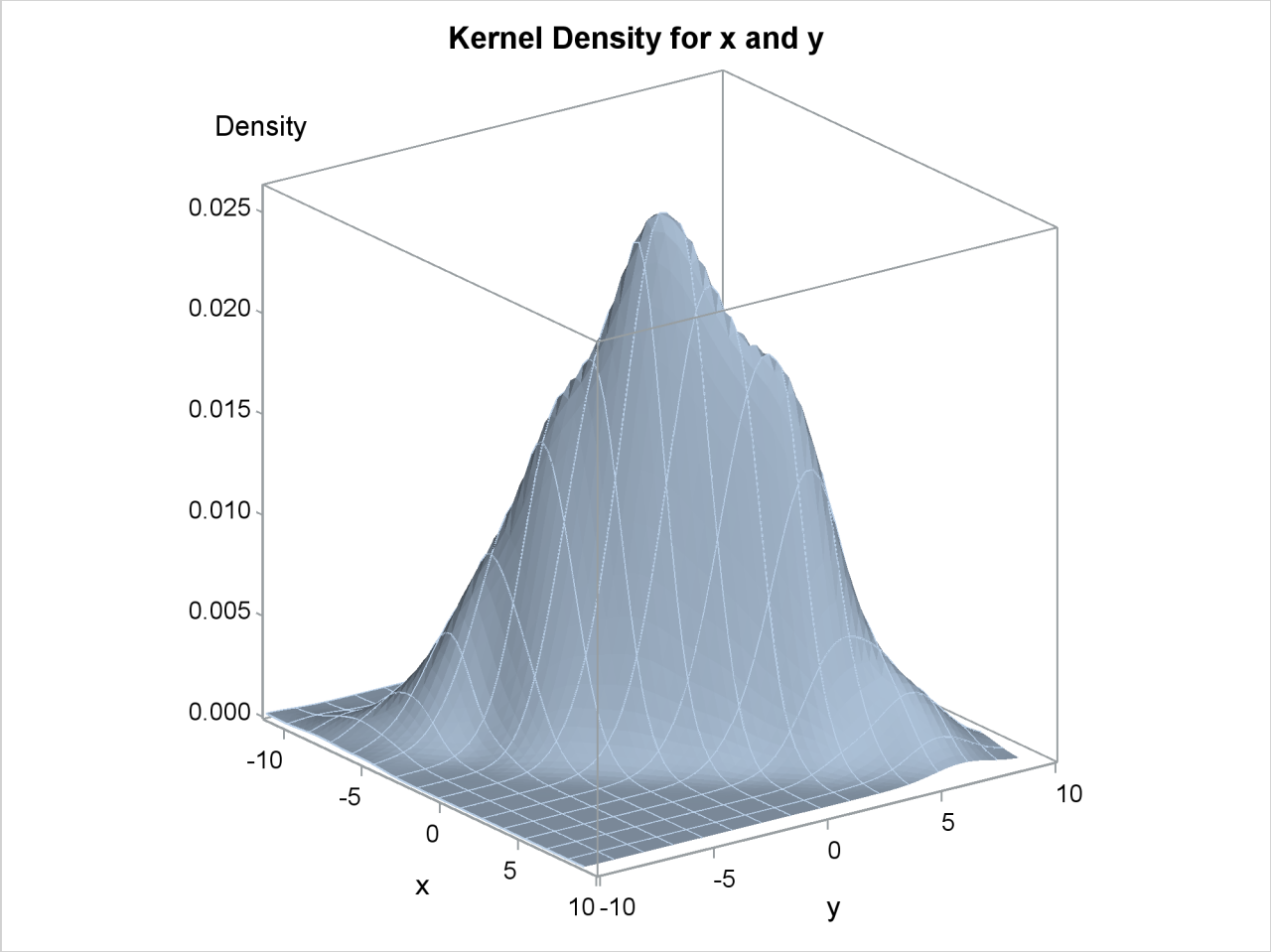


Figure 52.2 Surface Plot of Estimated Density



The default output tables for this analysis are shown in Figure 52.3.

Figure 52.3 Default Bivariate Tables

The KDE Procedure

Inputs

Data Set	WORK.BIVNORMAL
Number of Observations Used	1000
Variable 1	x
Variable 2	y
Bandwidth Method	Simple Normal Reference

Figure 52.3 *continued*

Controls

	x	y
Grid Points	60	60
Lower Grid Limit	-11.25	-10.05
Upper Grid Limit	9.1436	9.0341
Bandwidth Multiplier	1	1

The "Inputs" table lists basic information about the density fit, including the input data set, the number of observations, and the variables. The bandwidth method is the technique used to select the amount of smoothing in the estimate. A simple normal reference rule is used for bivariate smoothing.

The "Controls" table lists the primary numbers controlling the kernel density fit. Here a 60×60 grid is fit to the entire range of the data, and no adjustment is made to the default bandwidth.

Syntax: KDE Procedure

The following statements are available in the KDE procedure:

PROC KDE < *options* > **;**
BIVAR *variable-list* < / *options* > **;**
UNIVAR *variable-list* < / *options* > **;**
BY *variables* **;**
FREQ *variable* **;**
WEIGHT *variable* **;**

The PROC KDE statement invokes the procedure. The BIVAR statement requests that one or more bivariate kernel density estimates be computed. The UNIVAR statement requests one or more univariate kernel density estimates. You can specify any number of BIVAR and UNIVAR statements.

PROC KDE Statement

PROC KDE < *options* > **;**

The PROC KDE statement invokes the procedure. It also specifies the input data set.

DATA=*SAS-data-set*
specifies the input SAS data set to be used by PROC KDE. The default is the most recently created data set.

BIVAR Statement

The BIVAR statement computes bivariate kernel density estimates. Table 52.1 summarizes the *options* available in the BIVAR statement.

Table 52.1 BIVAR Statement Options

Option	Description
BIVSTATS	Produces a table for each density estimate
BWM=	Specifies the bandwidth multiplier
GRIDL=	Specifies the lower grid limit
GRIDU=	Specifies the upper grid limit
LEVELS	Requests a table of levels for contours of the bivariate density
NGRID=	Specifies the number of grid points associated with each variable
NOPRINT	Suppresses output tables
OUT=	Specifies the name of the output data set
PERCENTILES	Requests that a table of percentiles be computed
PLOTS=	Requests one or more plots
UNISTATS	Produces a table for each density estimate containing standard univariate statistics and the bandwidths

The basic syntax for the BIVAR statement specifies two variables:

BIVAR *v1* <**(***v-options***)**> *v2* <**(***v-options***)**> <**/** *options*> **;**

This statement requests a bivariate kernel density estimate for the variables v1 and v2. The *v-options* optionally specified in parentheses after a variable name apply only to that variable, and override corresponding global *options* specified following a slash (/).

You can specify a list of more than two variables:

BIVAR *v1* <**(***v-options***)**> *v2* <**(***v-options***)**> . . . *vN* <**(***v-options***)**> <**/** *options*> **;**

This statement requests a bivariate kernel density estimate for each distinct pair of variables in the list. For example, if you specify

```
bivar x y z;
```

then a bivariate kernel density estimate is computed for each of the variable pairs (x, y), (x, z), and (y, z).

Alternatively, you can specify an explicit list of variable pairs, with each pair enclosed in parentheses:

BIVAR (*v1 v2***)(***v3 v4***)**. . .**(***vN-1 vN***)**<**/** *options*> **;**

(You can also specify *v-options* following a variable name appearing in an explicit pair, but they are omitted here for clarity.) This statement requests a bivariate kernel density estimate for each pair of variables. For example, if you specify

```
bivar (x y) (y z);
```

then bivariate kernel density estimates are computed for (x, y) and (y, z).

NOTE: The VAR statement supported by PROC KDE in SAS 8 and earlier releases is now obsolete. The VAR statement has been replaced by the UNIVAR and the BIVAR statements, which enable you to produce multiple kernel density estimates with a single invocation of the procedure.

You can specify the following *options* in the BIVAR statement. As noted, some *options* can be used as *v-options*.

BIVSTATS
produces a table for each density estimate containing the covariance and correlation between the two variables.

BWM=*number*
specifies the bandwidth multiplier applied to each variable in each kernel density estimate. The default value is 1. Larger multipliers produce a smoother estimate, and smaller ones produce a rougher estimate. To specify different bandwidth multipliers for different variables, specify BWM= as a *v-option*.

GRIDL=*number*
specifies the lower grid limit applied to each variable in each kernel density estimate. The default value for a given variable is the minimum observed value of that variable. To specify different lower grid limits for different variables, specify GRIDL= as a *v-option*.

GRIDU=*number*
specifies the upper grid limit applied to each variable in each kernel density estimate. The default value for a given variable is the maximum observed value of that variable. To specify different upper grid limits for different variables, specify GRIDU= as a *v-option*.

LEVELS

LEVELS=*numlist*
requests a table of levels for contours of the bivariate density. The contours are defined in such a way that the density has a constant level along each contour, and the volume enclosed by each contour corresponds to a specified percent. In other words, the contours correspond to slices or levels of the density surface taken along the density axis. You can specify the percents used to define the contours. The default values are 1, 5, 10, 50, 90, 95, 99, and 100. The "Levels" table also provides the minimum and maximum values for each contour along the directions of the two data variables.

NGRID=*number*

NG=*number*
specifies the number of grid points associated with each variable in each kernel density estimate. The default value is 60. To specify different numbers of grid points for different variables, specify NGRID= as a *v-option*.

NOPRINT
suppresses output tables produced by the BIVAR statement. You can use the NOPRINT option when you want to produce graphical output only.

OUT=*SAS-data-set*
specifies the name of the output data set in which kernel density estimates are saved. This output data set contains the following variables:

- var1, whose value is the name of the first variable in a bivariate kernel density estimate
- var2, whose value is the name of the second variable in a bivariate kernel density estimate
- value1, with values corresponding to grid coordinates for the first variable
- value2, with values corresponding to grid coordinates for the second variable
- density, with values equal to kernel density estimates at the associated grid point
- count, containing the number of original observations contained in the bin corresponding to a grid point

PERCENTILES

PERCENTILES=*numlist*

requests that a table of percentiles be computed for each BIVAR variable. You can specify a list of percentiles to be computed. The default percentiles are 0.5, 1, 2.5, 5, 10, 25, 50, 75, 90, 95, 97.5, 99, and 99.5.

PLOTS=*plot-request* **<(***options***)> | ALL | NONE**

PLOTS=(*plot-request* **<(***options***)> <...** *plot-request* **<(***options***)>>)**

requests one or more plots of the bivariate data and kernel density estimate. When you specify only one plot request, you can omit the parentheses around the plot request.

ODS Graphics must be enabled before plots can be requested. For example:

```
ods graphics on;

proc kde data=octane;
   bivar Rater Customer / plots=all;
run;

ods graphics off;
```

For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" on page 606 in Chapter 21, "Statistical Graphics Using ODS."

By default, if ODS Graphics is enabled and you do not specify the PLOTS= option, then the BIVAR statement creates a contour plot. If you specify the PLOTS= option, you get only the requested plots.

The following *plot-requests* are available.

ALL

produces all bivariate plots.

CONTOUR

produces a contour plot of the bivariate density estimate.

CONTOURSCATTER

produces a contour plot of the bivariate density estimate overlaid with a scatter plot of the data.

HISTOGRAM <(*view-options*)>
produces a bivariate histogram of the data. The following *view-options* can be specified:

ROTATE=*angle*
rotates the histogram *angle* degrees, where –180 < *angle* < 180. By default, *angle* = 54.

TILT=*angle*
tilts the histogram *angle* degrees, where –180 < *angle* < 180. By default, *angle* = 20.

HISTSURFACE <(*view-options*)>
produces a bivariate histogram of the data overlaid with a surface plot of the bivariate kernel density estimate. The following *view-options* can be specified:

ROTATE=*angle*
rotates the histogram and kernel density surface *angle* degrees, where –180 < *angle* < 180. By default, *angle* = 54.

TILT=*angle*
tilts the histogram and kernel density surface *angle* degrees, where –180 < *angle* < 180. By default, *angle* = 20.

NONE
suppresses all plots, including the contour plot that is produced by default when ODS Graphics is enabled and the PLOTS= option is not specified.

SCATTER
produces a scatter plot of the data.

SURFACE <(*view-options*)>
produces a surface plot of the bivariate kernel density estimate. The following *view-options* can be specified:

ROTATE=*angle*
rotates the kernel density surface *angle* degrees, where –180 < *angle* < 180. By default, *angle* = 54.

TILT=*angle*
tilts the kernel density surface *angle* degrees, where –180 < *angle* < 180. By default, *angle* = 20.

UNISTATS
produces a table for each density estimate containing standard univariate statistics for each of the two variables and the bandwidths used to compute the kernel density estimate. The statistics listed are the mean, variance, standard deviation, range, and interquartile range.

UNIVAR Statement

UNIVAR *variable* <(*v-options*)> <*. . . variable* <(*v-options*)>> </ *options*> **;**

The UNIVAR statement computes univariate kernel density estimates. You can specify various *v-options* for each variable by enclosing them in parentheses after the variable name. You can also specify global *options* among the UNIVAR statement options following a slash (/). Global *options* apply to all the variables specified in the UNIVAR statement. However, individual variable *v-options* override the global *options*.

NOTE: The VAR statement supported by PROC KDE in SAS 8 and earlier releases is now obsolete. The VAR statement has been replaced by the UNIVAR and BIVAR statements, which enable you to produce multiple kernel density estimates with a single invocation of the procedure.

Table 52.2 summarizes the *options* available in the UNIVAR statement.

Table 52.2 UNIVAR Statement Options

Option	Description
BWM=	Specifies a bandwidth multiplier
GRIDL=	Specifies a lower grid limit
GRIDU=	Specifies an upper grid limit
METHOD=	Specifies the method used to compute the bandwidth
NGRID=	Specifies a number of grid points
NOPRINT	Suppresses output tables produced
OUT=	Specifies the output SAS data set containing the kernel density estimate
PERCENTILES	Requests that a table of percentiles
PLOTS=	Requests plots of the univariate kernel density estimate
SJPIMAX=	Specifies the maximum grid value in determining the Sheather-Jones plug-in bandwidth
SJPIMIN=	Specifies the minimum grid value in determining the Sheather-Jones plug-in bandwidth
SJPINUM=	Specifies the number of grid values used in determining the Sheather-Jones plug-in bandwidth
SJPITOL=	Specifies the tolerance for termination of the bisection algorithm
UNISTATS	Produces a table for each variable containing standard univariate statistics and the bandwidth

You can specify the following *options* in the UNIVAR statement. As noted, some *options* can be used as *v-options*.

BWM=*number*

specifies a bandwidth multiplier used for each kernel density estimate. The default value is 1. Larger multipliers produce a smoother estimate, and smaller ones produce a rougher estimate. To specify different bandwidth multipliers for different variables, specify BWM= as a *v-option*.

GRIDL=*number*

specifies a lower grid limit used for each kernel density estimate. The default value for a given variable is the minimum observed value of that variable. To specify different lower grid limits for different variables, specify GRIDL= as a *v-option*.

GRIDU=*number*

specifies an upper grid limit used for each kernel density estimate. The default value for a given variable is the maximum observed value of that variable. To specify different upper grid limits for different variables, specify GRIDU= as a *v-option*.

METHOD=SJPI | SNR | SNRQ | SROT | OS

specifies the method used to compute the bandwidth. Available methods are Sheather-Jones plug-in (SJPI), simple normal reference (SNR), simple normal reference that uses the interquartile range (SNRQ), Silverman's rule of thumb (SROT), and oversmoothed (OS). See the section "Bandwidth Selection" on page 4087 and see Jones, Marron, and Sheather (1996) for a description of these methods. SJPI is the default method.

NGRID=*number*

NG=*number*

specifies a number of grid points used for each kernel density estimate. The default value is 401. To specify different numbers of grid points for different variables, specify NGRID= as a *v-option*.

NOPRINT

suppresses output tables produced by the UNIVAR statement. You can use the NOPRINT option when you want to produce graphical output only.

OUT=*SAS-data-set*

specifies the output SAS data set containing the kernel density estimate. This output data set contains the following variables:

- var, whose value is the name of the variable in the kernel density estimate
- value, with values corresponding to grid coordinates for the variable
- density, with values equal to kernel density estimates at the associated grid point
- count, containing the number of original observations contained in the bin corresponding to a grid point

PERCENTILES

PERCENTILES=*numlist*

requests that a table of percentiles be computed for each UNIVAR variable. You can specify a list of percentiles to be computed. The default percentiles are 0.5, 1, 2.5, 5, 10, 25, 50, 75, 90, 95, 97.5, 99, and 99.5.

PLOTS=*plot-request* **<(***options***)> | ALL | NONE**

PLOTS=(*plot-request* **<(***options***)> <...** *plot-request* **<(***options***)>>)**

requests plots of the univariate kernel density estimate. When you specify only one *plot-request*, you can omit the parentheses around the *plot-request*.

ODS Graphics must be enabled before plots can be requested. For example:

```
ods graphics on;

proc kde data=channel;
   univar length / plots=histdensity;
```

```
run;

ods graphics off;
```

For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" on page 606 in Chapter 21, "Statistical Graphics Using ODS."

The following table shows the available *plot-requests*.

Keyword	Description
ALL	produces all plots
DENSITY	univariate kernel density estimate curve
DENSITYOVERLAY	overlaid univariate kernel density estimate curves
HISTDENSITY	univariate histogram of data overlaid with kernel density estimate curve
HISTOGRAM	univariate histogram of data
NONE	suppresses all plots

By default, if you ODS Graphics is enabled and you do not specify the PLOTS= option, then the UNIVAR statement creates a histogram overlaid with a kernel density estimate. If you specify the PLOTS= option, you get only the requested plots.

If you specify more than one variable in the UNIVAR statement, the DENSITYOVERLAY keyword overlays the density curves for all the variables on a single plot. The other *keywords* each produce a separate plot for every variable listed in the UNIVAR statement.

SJPIMAX=*number*
specifies the maximum grid value in determining the Sheather-Jones plug-in bandwidth. The default value is two times the oversmoothed estimate.

SJPIMIN=*number*
specifies the minimum grid value in determining the Sheather-Jones plug-in bandwidth. The default value is the maximum value divided by 18.

SJPINUM=*number*
specifies the number of grid values used in determining the Sheather-Jones plug-in bandwidth. The default is 21.

SJPITOL=*number*
specifies the tolerance for termination of the bisection algorithm used in computing the Sheather-Jones plug-in bandwidth. The default value is 0.001.

UNISTATS
produces a table for each variable containing standard univariate statistics and the bandwidth used to compute its kernel density estimate. The statistics listed are the mean, variance, standard deviation, range, and interquartile range.

Examples

Suppose you have the variables x1, x2, x3, and x4 in the SAS data set MyData. You can request a univariate kernel density estimate for each of these variables with the following statements:

```
proc kde data=MyData;
   univar x1 x2 x3 x4;
run;
```

You can also specify different bandwidths and other options for each variable. For example, the following statements request kernel density estimates that use Silverman's rule of thumb (SROT) method for all variables:

```
proc kde data=MyData;
   univar x1 (bwm=2)
          x2 (bwm=0.5 ngrid=100)
          x3 x4 / ngrid=200 method=srot;
run;
```

The option NGRID=200 applies to the variables x1, x3, and x4, but the *v-option* NGRID=100 is applied to x2. Bandwidth multipliers of 2 and 0.5 are specified for the variables x1 and x2, respectively.

BY Statement

BY *variables* **;**

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

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

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

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

FREQ Statement

FREQ *variable* **;**

The FREQ statement specifies a variable that provides frequencies for each observation in the DATA= data set. Specifically, if *n* is the value of the FREQ variable for a given observation, then that observation is used *n* times. If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement specifies a variable that weights the observations in computing the kernel density estimate. Observations with higher weights have more influence in the computations. If an observation has a nonpositive or missing weight, then the entire observation is omitted from the analysis. You should be cautious in using data sets with extreme weights, because they can produce unreliable results.

Details: KDE Procedure

Computational Overview

The two main computational tasks of PROC KDE are automatic bandwidth selection and the construction of a kernel density estimate once a bandwidth has been selected. The primary computational tools used to accomplish these tasks are binning, convolutions, and the fast Fourier transform. The following sections provide analytical details on these topics, beginning with the density estimates themselves.

Kernel Density Estimates

A weighted univariate kernel density estimate involves a variable X and a weight variable W. Let $(X_i, W_i),\ i = 1, 2, \ldots, n$, denote a sample of X and W of size n. The weighted kernel density estimate of $f(x)$, the density of X, is as follows:

$$\hat{f}(x) = \frac{1}{\sum_{i=1}^{n} W_i} \sum_{i=1}^{n} W_i \varphi_h(x - X_i)$$

where h is the bandwidth and

$$\varphi_h(x) = \frac{1}{\sqrt{2\pi}h} \exp\left(-\frac{x^2}{2h^2}\right)$$

is the standard normal density rescaled by the bandwidth. If $h \to 0$ and $nh \to \infty$, then the optimal bandwidth is

$$h_{\mathrm{AMISE}} = \left[\frac{1}{2\sqrt{\pi}n \int (f'')^2}\right]^{1/5}$$

This optimal value is unknown, and so approximations methods are required. For a derivation and discussion of these results, see Silverman (1986, Chapter 3) and Jones, Marron, and Sheather (1996).

For the bivariate case, let $\mathbf{X} = (X, Y)$ be a bivariate random element taking values in R^2 with joint density function

$$f(x, y),\ (x, y) \in R^2$$

and let $\mathbf{X}_i = (X_i, Y_i),\ i = 1, 2, \ldots, n$, be a sample of size n drawn from this distribution. The kernel density estimate of $f(x, y)$ based on this sample is

$$\begin{aligned}\hat{f}(x, y) &= \frac{1}{n}\sum_{i=1}^{n}\varphi_{\mathbf{h}}(x - X_i, y - Y_i) \\ &= \frac{1}{nh_Xh_Y}\sum_{i=1}^{n}\varphi\left(\frac{x - X_i}{h_X}, \frac{y - Y_i}{h_Y}\right)\end{aligned}$$

where $(x, y) \in R^2$, $h_X > 0$ and $h_Y > 0$ are the bandwidths, and $\varphi_{\mathbf{h}}(x, y)$ is the rescaled normal density

$$\varphi_{\mathbf{h}}(x, y) = \frac{1}{h_Xh_Y}\varphi\left(\frac{x}{h_X}, \frac{y}{h_Y}\right)$$

where $\varphi(x, y)$ is the standard normal density function

$$\varphi(x, y) = \frac{1}{2\pi}\exp\left(-\frac{x^2 + y^2}{2}\right)$$

Under mild regularity assumptions about $f(x, y)$, the mean integrated squared error (MISE) of $\hat{f}(x, y)$ is

$$\begin{aligned}\text{MISE}(h_X, h_Y) &= \text{E}\int(\hat{f} - f)^2 \\ &= \frac{1}{4\pi nh_Xh_Y} + \frac{h_X^4}{4}\int\left(\frac{\partial^2 f}{\partial X^2}\right)^2 dxdy \\ &\quad + \frac{h_Y^4}{4}\int\left(\frac{\partial^2 f}{\partial Y^2}\right)^2 dxdy + O\left(h_X^4 + h_Y^4 + \frac{1}{nh_Xh_Y}\right)\end{aligned}$$

as $h_X \to 0$, $h_Y \to 0$ and $nh_Xh_Y \to \infty$.

Now set

$$\begin{aligned}\text{AMISE}(h_X, h_Y) &= \frac{1}{4\pi nh_Xh_Y} + \frac{h_X^4}{4}\int\left(\frac{\partial^2 f}{\partial X^2}\right)^2 dxdy \\ &\quad + \frac{h_Y^4}{4}\int\left(\frac{\partial^2 f}{\partial Y^2}\right)^2 dxdy\end{aligned}$$

which is the asymptotic mean integrated squared error (AMISE). For fixed n, this has a minimum at $(h_{\text{AMISE\_}X}, h_{\text{AMISE\_}Y})$ defined as

$$h_{\text{AMISE\_}X} = \left[\frac{\int(\frac{\partial^2 f}{\partial X^2})^2}{4n\pi}\right]^{1/6}\left[\frac{\int(\frac{\partial^2 f}{\partial X^2})^2}{\int(\frac{\partial^2 f}{\partial Y^2})^2}\right]^{2/3}$$

and

$$h_{\text{AMISE\_}Y} = \left[\frac{\int(\frac{\partial^2 f}{\partial Y^2})^2}{4n\pi}\right]^{1/6}\left[\frac{\int(\frac{\partial^2 f}{\partial Y^2})^2}{\int(\frac{\partial^2 f}{\partial X^2})^2}\right]^{2/3}$$

These are the optimal asymptotic bandwidths in the sense that they minimize MISE. However, as in the univariate case, these expressions contain the second derivatives of the unknown density f being estimated, and so approximations are required. See Wand and Jones (1993) for further details.

Binning

Binning, or assigning data to discrete categories, is an effective and fast method for large data sets (Fan and Marron 1994). When the sample size n is large, direct evaluation of the kernel estimate $\hat{f}$ at any point would involve n kernel evaluations, as shown in the preceding formulas. To evaluate the estimate at each point of a grid of size g would thus require ng kernel evaluations. When you use $g = 401$ in the univariate case or $g = 60 \times 60 = 3600$ in the bivariate case and $n \geq 1000$, the amount of computation can be prohibitively large. With binning, however, the computational order is reduced to g, resulting in a much quicker algorithm that is nearly as accurate as direct evaluation.

To bin a set of weighted univariate data $X_1, X_2, \ldots, X_n$ to a grid $x_1, x_2, \ldots, x_g$, simply assign each sample X_i, together with its weight W_i, to the nearest grid point x_j (also called the bin center). When binning is completed, each grid point x_i has an associated number c_i, which is the sum total of all the weights that correspond to sample points that have been assigned to x_i. These c_is are known as the *bin counts.*

This procedure replaces the data $(X_i, W_i),\ i = 1, 2, \ldots, n$, with the smaller set $(x_i, c_i),\ i = 1, 2, \ldots, g$, and the estimation is carried out with these new data. This is so-called *simple binning,* versus the finer *linear binning* described in Wand (1994). PROC KDE uses simple binning for the sake of faster and easier implementation. Also, it is assumed that the bin centers $x_1, x_2, \ldots, x_g$ are equally spaced and in increasing order. In addition, assume for notational convenience that $\sum_{i=1}^{n} W_i = n$ and, therefore, $\sum_{i=1}^{g} c_i = n$.

If you replace the data $(X_i, W_i), i = 1, 2, \ldots, n$, with $(x_i, c_i), i = 1, 2, \ldots, g$, the weighted estimator $\hat{f}$ then becomes

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{g} c_i \varphi_h(x - x_i)$$

with the same notation as used previously. To evaluate this estimator at the g points of the same grid vector $grid = (x_1, x_2, \ldots, x_g)'$ is to calculate

$$\hat{f}(x_i) = \frac{1}{n} \sum_{j=1}^{g} c_j \varphi_h(x_i - x_j)$$

for $i = 1, 2, \ldots, g$. This can be rewritten as

$$\hat{f}(x_i) = \frac{1}{n} \sum_{j=1}^{g} c_j \varphi_h(|i - j|\delta)$$

where $\delta = x_2 - x_1$ is the increment of the grid.

The same idea of binning works similarly with bivariate data, where you estimate $\hat{f}$ over the grid matrix $grid = grid_X \times grid_Y$ as follows:

$$grid = \begin{bmatrix} \mathbf{x}_{1,1} & \mathbf{x}_{1,2} & \ldots & \mathbf{x}_{1,g_Y} \\ \mathbf{x}_{2,1} & \mathbf{x}_{2,2} & \ldots & \mathbf{x}_{2,g_Y} \\ \vdots & & & \\ \mathbf{x}_{g_X,1} & \mathbf{x}_{g_X,2} & \ldots & \mathbf{x}_{g_X,g_Y} \end{bmatrix}$$

where $\mathbf{x}_{i,j} = (x_i, y_i),\ i = 1, 2, \ldots, g_X,\ j = 1, 2, \ldots, g_Y$, and the estimates are

$$\hat{f}(\mathbf{x}_{i,j}) = \frac{1}{n} \sum_{k=1}^{g_X} \sum_{l=1}^{g_Y} c_{k,l} \varphi_{\mathbf{h}}(|i - k|\delta_X, |j - l|\delta_Y)$$

where $\delta_X = x_2 - x_1$ and $\delta_Y = y_2 - y_1$ are the increments of the grid.

Convolutions

The formulas for the binned estimator $\hat{f}$ in the previous subsection are in the form of a convolution product between two matrices, one of which contains the bin counts, the other of which contains the rescaled kernels evaluated at multiples of grid increments. This section defines these two matrices explicitly, and shows that $\hat{f}$ is their convolution.

Beginning with the weighted univariate case, define the following matrices:

$$\begin{aligned} \mathbf{K} &= \frac{1}{n}(\varphi_h(0), \varphi_h(\delta), \ldots, \varphi_h((g-1)\delta))' \\ \mathbf{C} &= (c_1, c_2, \ldots, c_g)' \end{aligned}$$

The first thing to note is that many terms in $\mathbf{K}$ are negligible. The term $\varphi_h(i\delta)$ is taken to be 0 when $|i\delta/h| \geq 5$, so you can define

$$l = \min(g-1, \text{floor}(5h/\delta))$$

as the maximum integer multiple of the grid increment to get nonzero evaluations of the rescaled kernel. Here $\text{floor}(x)$ denotes the largest integer less than or equal to x.

Next, let p be the smallest power of 2 that is greater than $g + l + 1$,

$$p = 2^{\text{ceil}(\log_2(g+l+1))}$$

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

Modify $\mathbf{K}$ as follows:

$$\mathbf{K} = \frac{1}{n}(\varphi_h(0), \varphi_h(\delta), \ldots, \varphi_h(l\delta), \underbrace{0, \ldots, 0}_{p-2l-1}, \varphi_h(l\delta), \ldots, \varphi_h(\delta))'$$

Essentially, the negligible terms of $\mathbf{K}$ are omitted, and the rest are *symmetrized* (except for one term). The whole matrix is then padded to size $p \times 1$ with zeros in the middle. The dimension p is a highly composite number—that is, one that decomposes into many factors—leading to the most efficient fast Fourier transform operation (see Wand 1994).

The third operation is to pad the bin count matrix $\mathbf{C}$ with zeros to the same size as $\mathbf{K}$:

$$\mathbf{C} = (c_1, c_2, \ldots, c_g, \underbrace{0, \ldots, 0}_{p-g})'$$

The convolution $\mathbf{K} * \mathbf{C}$ is then a $p \times 1$ matrix, and the preceding formulas show that its first g entries are exactly the estimates $\hat{f}(x_i),\ i = 1, 2, \ldots, g$.

For bivariate smoothing, the matrix $\mathbf{K}$ is defined similarly as

$$\mathbf{K} = \begin{bmatrix} \kappa_{0,0} & \kappa_{0,1} & \dots & \kappa_{0,l_Y} & \mathbf{0} & \kappa_{0,l_Y} & \dots & \kappa_{0,1} \\ \kappa_{1,0} & \kappa_{1,1} & \dots & \kappa_{1,l_Y} & \mathbf{0} & \kappa_{1,l_Y} & \dots & \kappa_{1,1} \\ \vdots & & & & & & & \\ \kappa_{l_X,0} & \kappa_{l_X,1} & \dots & \kappa_{l_X,l_Y} & \mathbf{0} & \kappa_{l_X,l_Y} & \dots & \kappa_{l_X,1} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \kappa_{l_X,0} & \kappa_{l_X,1} & \dots & \kappa_{l_X,l_Y} & \mathbf{0} & \kappa_{l_X,l_Y} & \dots & \kappa_{l_X,1} \\ \vdots & & & & & & & \\ \kappa_{1,0} & \kappa_{1,1} & \dots & \kappa_{1,l_Y} & \mathbf{0} & \kappa_{1,l_Y} & \dots & \kappa_{1,1} \end{bmatrix}_{p_X \times p_Y}$$

where $l_X = \min(g_X - 1, \text{floor}(5h_X/\delta_X))$, $p_X = 2^{\text{ceil}(\log_2(g_X + l_X + 1))}$, and so forth, and $\kappa_{i,j} = \frac{1}{n}\varphi_{\mathbf{h}}(i\delta_X, j\delta_Y)$ $i = 0, 1, \dots, l_X$, $j = 0, 1, \dots, l_Y$.

The bin count matrix $\mathbf{C}$ is defined as

$$\mathbf{C} = \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,g_Y} & 0 & \dots & 0 \\ c_{2,1} & c_{2,2} & \dots & c_{2,g_Y} & 0 & \dots & 0 \\ \vdots & & & & & & \\ c_{g_X,1} & c_{g_X,2} & \dots & c_{g_X,g_Y} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & & & & & & \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix}_{p_X \times p_Y}$$

As with the univariate case, the $g_X \times g_Y$ upper-left corner of the convolution $\mathbf{K} * \mathbf{C}$ is the matrix of the estimates $\hat{f}(grid)$.

Most of the results in this subsection are found in Wand (1994).

Fast Fourier Transform

As shown in the last subsection, kernel density estimates can be expressed as a submatrix of a certain convolution. The fast Fourier transform (FFT) is a computationally effective method for computing such convolutions. For a reference on this material, see Press et al. (1988).

The *discrete Fourier transform* of a complex vector $\mathbf{z} = (z_0, \dots, z_{N-1})$ is the vector $\mathbf{Z} = (Z_0, \dots, Z_{N-1})$, where

$$Z_j = \sum_{l=0}^{N-1} z_l e^{2\pi i l j/N}, \quad j = 0, \dots, N-1$$

and i is the square root of –1. The vector $\mathbf{z}$ can be recovered from $\mathbf{Z}$ by applying the *inverse discrete Fourier transform* formula

$$z_l = N^{-1} \sum_{j=0}^{N-1} Z_j e^{-2\pi i l j/N}, \quad l = 0, \dots, N-1$$

Discrete Fourier transforms and their inverses can be computed quickly using the FFT algorithm, especially when N is *highly composite*; that is, it can be decomposed into many factors, such as a power of 2. By the *discrete convolution theorem*, the convolution of two vectors is the inverse Fourier transform of the element-by-element product of their Fourier transforms. This, however, requires certain periodicity assumptions, which explains why the vectors **K** and **C** require zero-padding. This is to avoid *wrap-around* effects (see Press et al. 1988, pp. 410–411). The vector **K** is actually mirror-imaged so that the convolution of **C** and **K** will be the vector of binned estimates. Thus, if S denotes the inverse Fourier transform of the element-by-element product of the Fourier transforms of **K** and **C**, then the first g elements of S are the estimates.

The bivariate Fourier transform of an $N_1 \times N_2$ complex matrix having $(l_1 + 1, l_2 + 1)$ entry equal to $z_{l_1 l_2}$ is the $N_1 \times N_2$ matrix with $(j_1 + 1, j_2 + 1)$ entry given by

$$Z_{j_1 j_2} = \sum_{l_1=0}^{N_1-1} \sum_{l_2=0}^{N_2-1} z_{l_1 l_2} e^{2\pi i (l_1 j_1/N_1 + l_2 j_2/N_2)}$$

and the formula of the inverse is

$$z_{l_1 l_2} = (N_1 N_2)^{-1} \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} Z_{j_1 j_2} e^{-2\pi i (l_1 j_1/N_1 + l_2 j_2/N_2)}$$

The same discrete convolution theorem applies, and zero-padding is needed for matrices **C** and **K**. In the case of **K**, the matrix is mirror-imaged twice. Thus, if S denotes the inverse Fourier transform of the element-by-element product of the Fourier transforms of **K** and **C**, then the upper-left $g_X \times g_Y$ corner of S contains the estimates.

Bandwidth Selection

Several different bandwidth selection methods are available in PROC KDE in the univariate case. Following the recommendations of Jones, Marron, and Sheather (1996), the default method follows a plug-in formula of Sheather and Jones.

This method solves the fixed-point equation

$$h = \left[\frac{R(\varphi)}{n R\left(\hat{f}''_{g(h)}\right) \left(\int x^2 \varphi(x) dx\right)^2} \right]^{1/5}$$

where $R(\varphi) = \int \varphi^2(x) dx$.

PROC KDE solves this equation by first evaluating it on a grid of values spaced equally on a log scale. The largest two values from this grid that bound a solution are then used as starting values for a bisection algorithm.

The simple normal reference rule works by assuming $\hat{f}$ is Gaussian in the preceding fixed-point equation. This results in

$$\begin{aligned} h &= \hat{\sigma}[4/(3n)]^{1/5} \\ &= 1.06\, \hat{\sigma} n^{-1/5} \end{aligned}$$

where $\hat{\sigma}$ is the sample standard deviation.

Alternatively, the bandwidth can be computed using the interquartile range, Q:

$$\begin{aligned} h &= 1.06\,\hat{\sigma} n^{-1/5} \\ &\approx 1.06\,(Q/1.34) n^{-1/5} \\ &\approx 0.785\,Q\,n^{-1/5} \end{aligned}$$

Silverman's rule of thumb (Silverman 1986, Section 3.4.2) is computed as

$$h = 0.9\min[\hat{\sigma}, Q/1.34]n^{-1/5}$$

The oversmoothed bandwidth is computed as

$$h = 3\hat{\sigma}[1/(70\sqrt{\pi}n)]^{1/5}$$

When you specify a WEIGHT variable, PROC KDE uses weighted versions of Q_3, Q_1, and $\hat{\sigma}$ in the preceding expressions. The weighted quartiles are computed as weighted order statistics, and the weighted variance takes the form

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} W_i (X_i - \bar{X})^2}{\sum_{i=1}^{n} W_i}$$

where $\bar{X} = (\sum_{i=1}^{n} W_i X_i)/(\sum_{i=1}^{n} W_i)$ is the weighted sample mean.

For the bivariate case, Wand and Jones (1993) note that automatic bandwidth selection is both difficult and computationally expensive. Their study of various ways of specifying a bandwidth matrix also shows that using two bandwidths, one in each coordinate's direction, is often adequate. PROC KDE enables you to adjust the two bandwidths by specifying a multiplier for the default bandwidths recommended by Bowman and Foster (1993):

$$\begin{aligned} h_X &= \hat{\sigma}_X n^{-1/6} \\ h_Y &= \hat{\sigma}_Y n^{-1/6} \end{aligned}$$

Here $\hat{\sigma}_X$ and $\hat{\sigma}_Y$ are the sample standard deviations of X and Y, respectively. These are the optimal bandwidths for two independent normal variables that have the same variances as X and Y. They are, therefore, conservative in the sense that they tend to oversmooth the surface.

You can specify the BWM= option to adjust the aforementioned bandwidths to provide the appropriate amount of smoothing for your application.

ODS Table Names

PROC KDE assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 52.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 52.3 ODS Tables Produced in PROC KDE

ODS Table Name	Description	Statement	Option
BivariateStatistics	Bivariate statistics	BIVAR	BIVSTATS
Controls	Control variables	default	
Inputs	Input information	default	
Levels	Levels of density estimate	BIVAR	LEVELS
Percentiles	Percentiles of data	BIVAR / UNIVAR	PERCENTILES
UnivariateStatistics	Basic statistics	BIVAR / UNIVAR	UNISTATS

ODS Graphics

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."

Before you create graphs, ODS Graphics must be enabled (for example, by specifying the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section "Enabling and Disabling ODS Graphics" on page 606 in Chapter 21, "Statistical Graphics Using ODS."

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" on page 605 in Chapter 21, "Statistical Graphics Using ODS."

ODS Graph Names

PROC KDE assigns a name to each graph it creates using the Output Delivery System (ODS). You can use these names to reference the graphs when using ODS. The names are listed in Table 52.4.

Table 52.4 Graphs Produced by PROC KDE

ODS Graph Name	Plot Description	Statement	PLOTS= Option
BivariateHistogram	Bivariate histogram of data	BIVAR	HISTOGRAM
ContourPlot	Contour plot of bivariate kernel density estimate	BIVAR	CONTOUR
ContourScatterPlot	Contour plot of bivariate kernel density estimate overlaid with scatter plot	BIVAR	CONTOURSCATTER
DensityPlot	Univariate kernel density estimate curve	UNIVAR	DENSITY
DensityOverlayPlot	Overlaid univariate kernel density estimate curves	UNIVAR	DENSITYOVERLAY
HistogramDensity	Univariate histogram overlaid with kernel density estimate curve	UNIVAR	HISTDENSITY
Histogram	Univariate histogram of data	UNIVAR	HISTOGRAM
HistogramSurface	Bivariate histogram overlaid with surface plot of bivariate kernel density estimate	BIVAR	HISTSURFACE

Table 52.4 (continued)

ODS Graph Name	Plot Description	Statement	PLOTS= Option
ScatterPlot	Scatter plot of data	BIVAR	SCATTER
SurfacePlot	Surface plot of bivariate kernel density estimate	BIVAR	SURFACE

Bivariate Plots

You can specify the PLOTS= option in the BIVAR statement to request graphical displays of bivariate kernel density estimates.

PLOTS= *option1* < *option2 . . .* >
requests one or more plots of the bivariate kernel density estimate. The following table shows the available plot *options*.

Option	Description
ALL	all available displays
CONTOUR	contour plot of bivariate density estimate
CONTOURSCATTER	contour plot of bivariate density estimate overlaid with scatter plot of data
HISTOGRAM	bivariate histogram of data
HISTSURFACE	bivariate histogram overlaid with bivariate kernel density estimate
NONE	suppresses all plots
SCATTER	scatter plot of data
SURFACE	surface plot of bivariate kernel density estimate

By default, if ODS Graphics is enabled and you do not specify the PLOTS= option, then the BIVAR statement creates a contour plot. If you specify the PLOTS= option, you get only the requested plots.

Univariate Plots

You can specify the PLOTS= option in the UNIVAR statement to request graphical displays of univariate kernel density estimates.

PLOTS= *option1* < *option2 . . .* >
requests one or more plots of the univariate kernel density estimate. The following table shows the available plot *options*.

Option	Description
ALL	all available displays
DENSITY	univariate kernel density estimate curve
DENSITYOVERLAY	overlaid univariate kernel density estimate curves
HISTDENSITY	univariate histogram of data overlaid with kernel density estimate curve
HISTOGRAM	univariate histogram of data
NONE	suppresses all plots

By default, if ODS Graphics is enabled and you do not specify the PLOTS= option, then the UNIVAR statement creates a histogram overlaid with a kernel density estimate. If you specify the PLOTS= option, you get only the requested plots.

Binning of Bivariate Histogram

Let $(X_i, Y_i),\ i = 1, 2, \ldots, n$, be a sample of size n drawn from a bivariate distribution. For the marginal distribution of $X_i,\ i = 1, 2, \ldots, n$, the number of bins ($\mathrm{Nbins}_X$) in the bivariate histogram is calculated according to the formula

$$\mathrm{Nbins}_X = \mathrm{ceil}\left(\mathrm{range}_X / \mathrm{width}_X\right)$$

where $\mathrm{ceil}(x)$ denotes the smallest integer greater than or equal to x,

$$\mathrm{range}_X = \max_{1 \le i \le n}(X_i) - \min_{1 \le i \le n}(X_i)$$

and the optimal bin width is obtained, following Scott (1992, p. 84), as

$$\mathrm{width}_X = 3.504\, \hat{\sigma}_X (1 - \hat{\rho}^2)^{3/8} n^{-1/4}$$

Here, $\hat{\sigma}_X$ and $\hat{\rho}$ are the sample variance and the sample correlation coefficient, respectively. When you specify a WEIGHT variable, PROC KDE uses weighted versions of $\hat{\sigma}_X$ and $\hat{\rho}$ in the preceding expressions.

Similar formulas are used to compute the number of bins for the marginal distribution of $Y_i,\ i = 1, 2, \ldots, n$. Further details can be found in Scott (1992).

Notice that if $|\hat{\rho}| > 0.99$, then Nbins_X is calculated as in the univariate case (see Terrell and Scott 1985). In this case $\mathrm{Nbins}_Y = \mathrm{Nbins}_X$.

Examples: KDE Procedure

Example 52.1: Computing a Basic Kernel Density Estimate

This example illustrates the basic functionality of the UNIVAR statement. The effective channel length (in microns) is measured for 1225 field effect transistors. The channel lengths are saved as values of the variable length in a SAS data set named channel; see the file *kdex1.sas* in the SAS Sample Library. These statements create the channel data set:

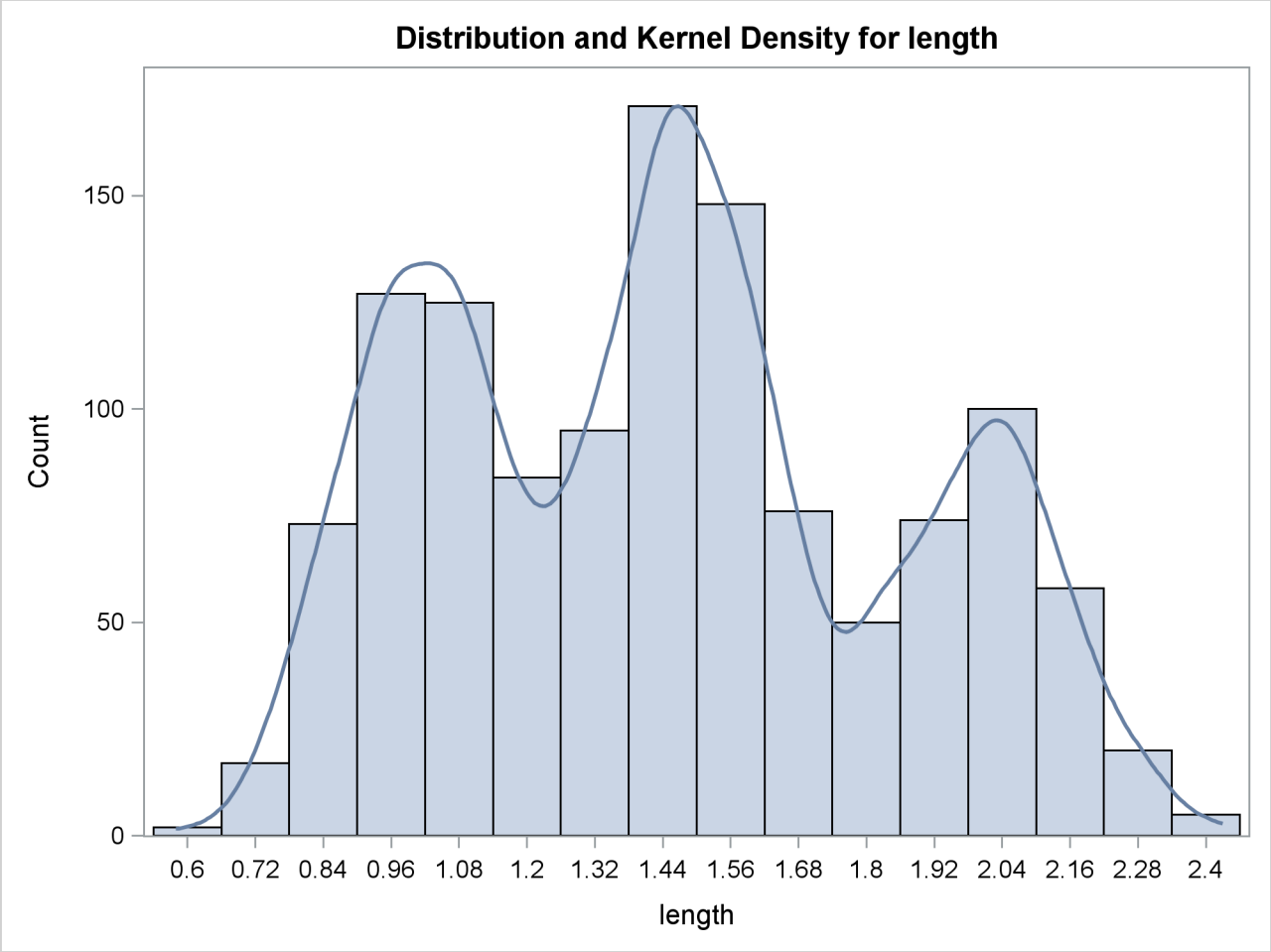
```
data channel;
   input length @@;
   datalines;
0.91 1.01 0.95 1.13 1.12 0.86 0.96 1.17 1.36 1.10
0.98 1.27 1.13 0.92 1.15 1.26 1.14 0.88 1.03 1.00
0.98 0.94 1.09 0.92 1.10 0.95 1.05 1.05 1.11 1.15
1.11 0.98 0.78 1.09 0.94 1.05 0.89 1.16 0.88 1.19

   ... more lines ...

2.13 2.05 1.90 2.07 2.15 1.96 2.15 1.89 2.15 2.04
1.95 1.93 2.22 1.74 1.91
;
```

The following statements request a kernel density estimate of the variable length:

```
ods graphics on;
proc kde data=channel;
   univar length;
run;
```

Because ODS Graphics is enabled, PROC KDE produces a histogram with an overlaid kernel density estimate by default, although the PLOTS= option is not specified. The resulting graph is shown in Output 52.1.1. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the KDE procedure, see the section "ODS Graphics" on page 4089.

Output 52.1.1 Histogram with Overlaid Kernel Density Estimate



The default output tables for this analysis are the "Inputs" and "Controls" tables, shown in Output 52.1.2.

Output 52.1.2 Univariate Inputs Table

The KDE Procedure

Inputs

Data Set	WORK.CHANNEL
Number of Observations Used	1225
Variable	length
Bandwidth Method	Sheather-Jones Plug In

Output 52.1.2 *continued*

Controls

	length
Grid Points	401
Lower Grid Limit	0.58
Upper Grid Limit	2.43
Bandwidth Multiplier	1

The "Inputs" table lists basic information about the density fit, including the input data set, the number of observations, the variable used, and the bandwidth method. The default bandwidth method is the Sheather-Jones plug-in.

The "Controls" table lists the primary numbers controlling the kernel density fit. Here the default number of grid points is used and no adjustment is made to the default bandwidth.

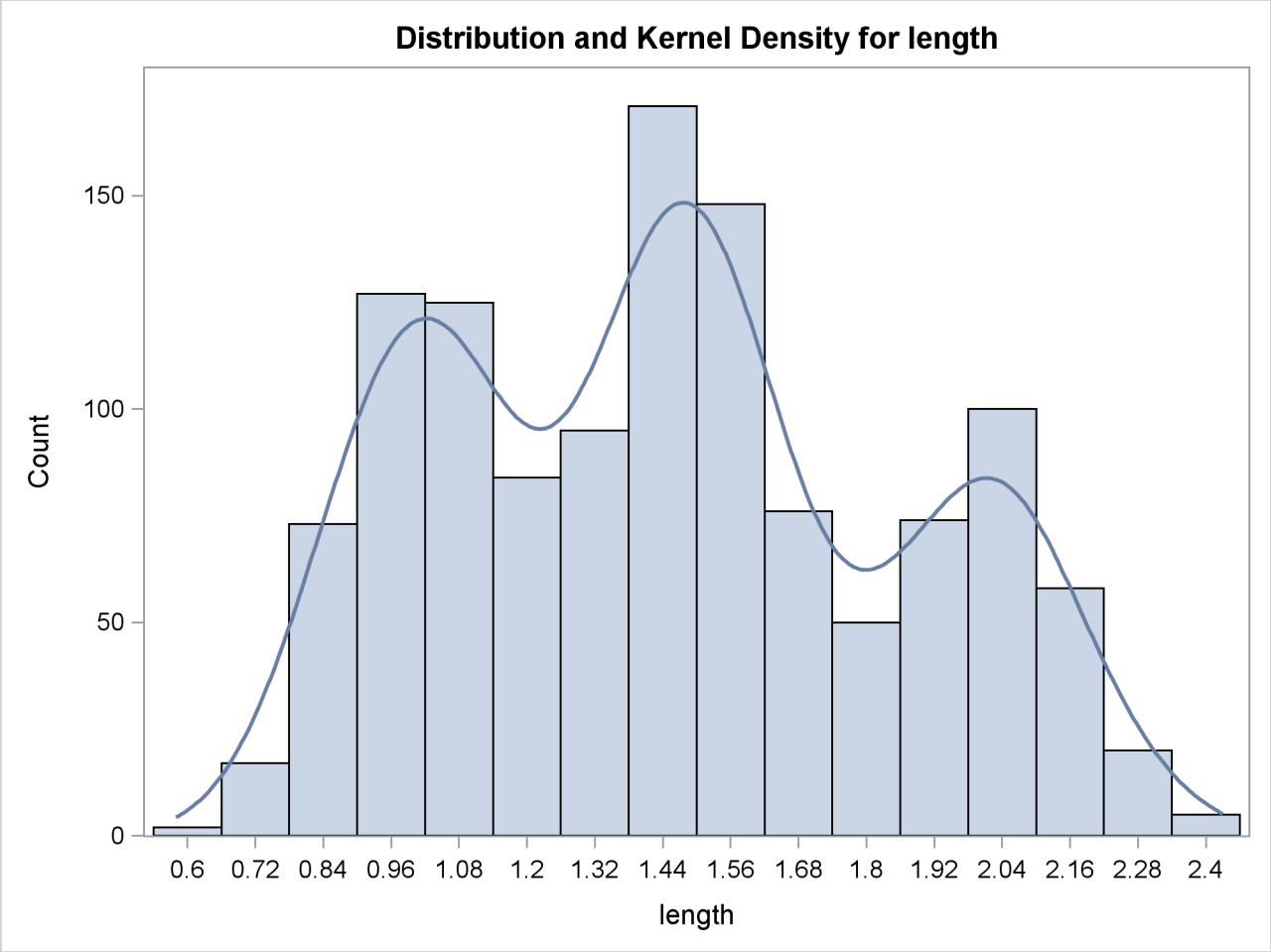
Example 52.2: Changing the Bandwidth

Continuing with Example 52.1, you can specify different bandwidth multipliers that determine the smoothness of the kernel density estimate. The following statements show kernel density estimates for the variable length by specifying two different bandwidth multipliers with the BWM= option:

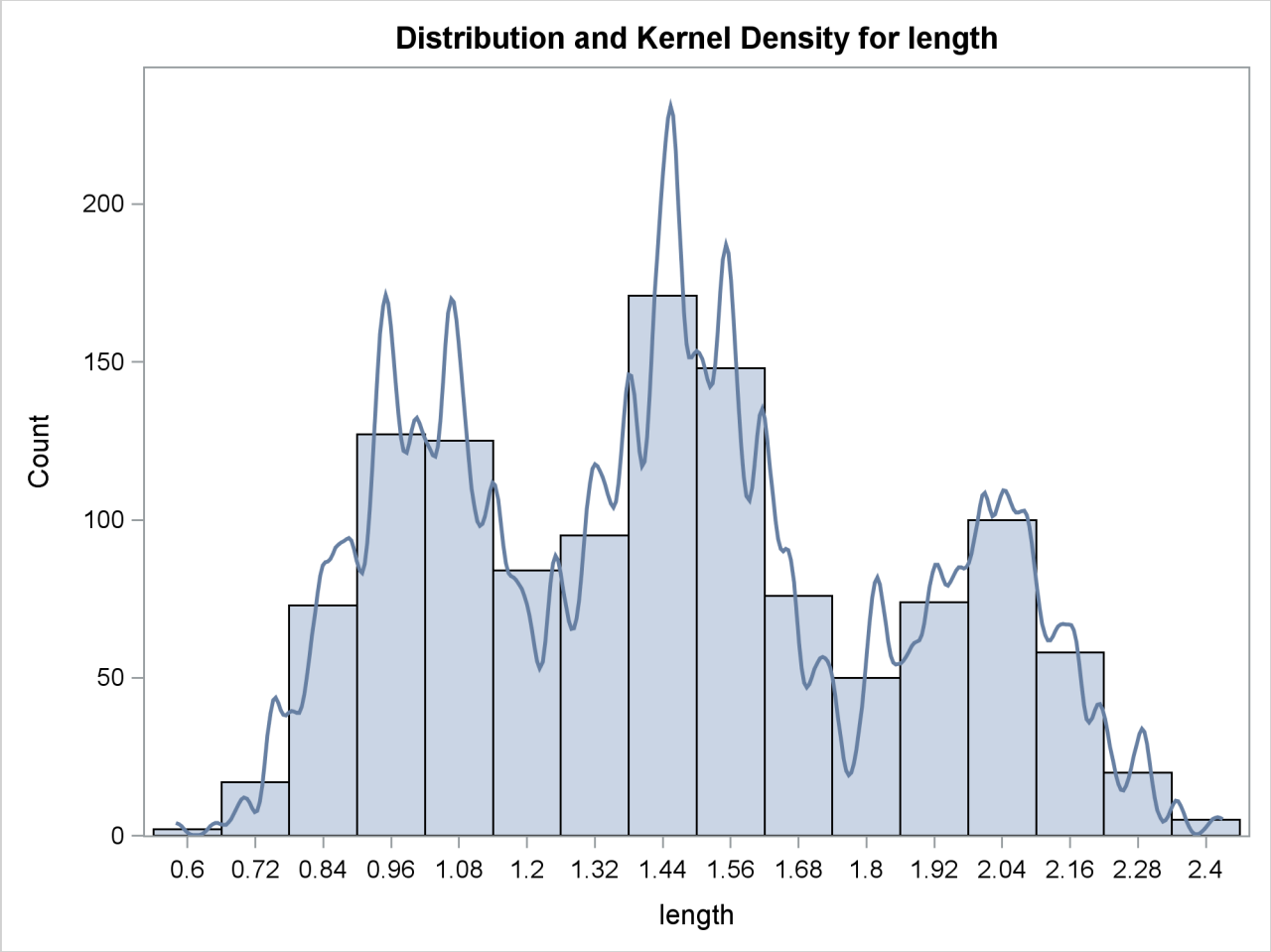
```
proc kde data=channel;
   univar length(bwm=2) length(bwm=0.25);
run;
ods graphics off;
```

Output 52.2.1 shows an oversmoothed estimate because the bandwidth multiplier is 2. Output 52.2.2 is created by specifying BWM=0.25, so it is an undersmoothed estimate.

Output 52.2.1 Histogram with Oversmoothed Kernel Density Estimate



Output 52.2.2 Histogram with Undersmoothed Kernel Density Estimate



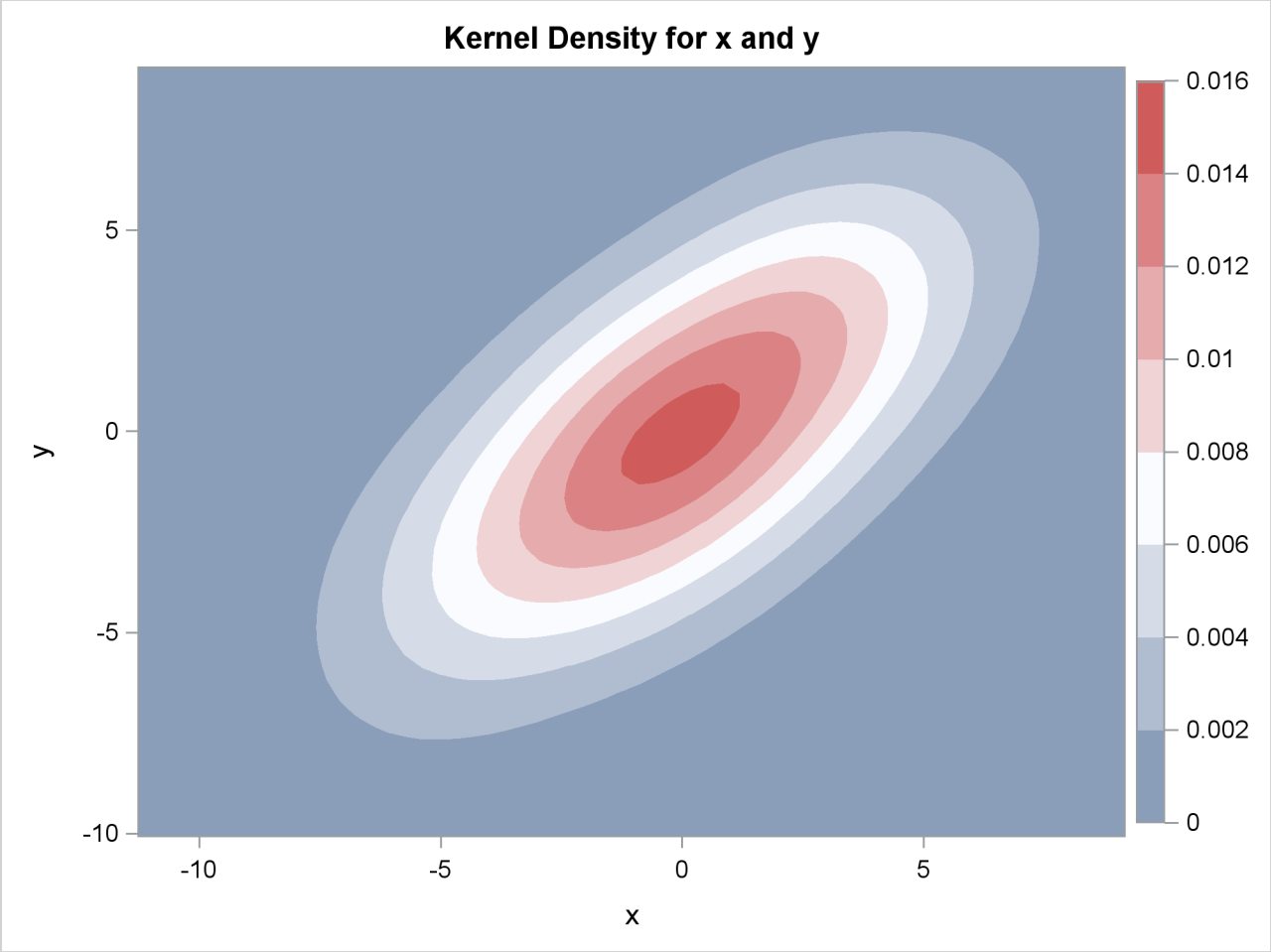
Example 52.3: Changing the Bandwidth (Bivariate)

Recall the analysis from the section "Getting Started: KDE Procedure" on page 4070. Suppose you would like a slightly smoother estimate. You could then rerun the analysis with a larger bandwidth:

```
ods graphics on;
proc kde data=bivnormal;
   bivar x y / bwm=2;
run;
```

The BWM= option requests bandwidth multipliers of 2 for both x and y. With ODS Graphics enabled, the BIVAR statement produces a contour plot, as shown in Output 52.3.1.

Output 52.3.1 Contour Plot of Estimated Density with Additional Smoothing



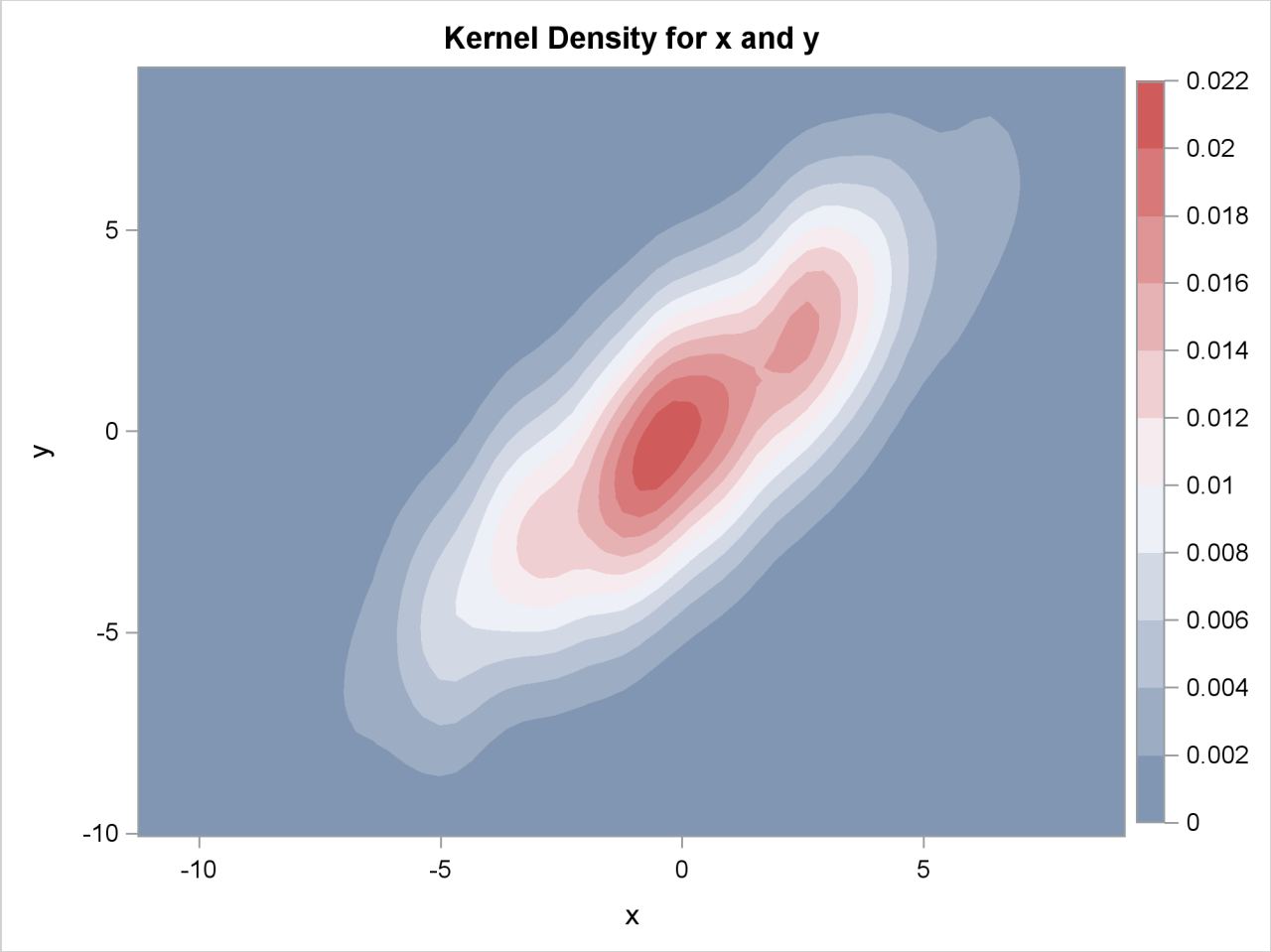
Multiple Bandwidths

You can also specify multiple bandwidths with only one run of the KDE procedure. Notice that by specifying pairs of variables inside parentheses, a kernel density estimate is computed for each pair. In the following statements the first kernel density is computed with the default bandwidth, but the second kernel density specifies a bandwidth multiplier of 0.5 for the variable x and a multiplier of 2 for the variable y:

```
proc kde data=bivnormal;
   bivar (x y) (x (bwm=0.5) y (bwm=2));
run;
ods graphics off;
```

The contour plot of the second kernel density estimate is shown in Output 52.3.2.

Output 52.3.2 Contour Plot of Estimated Density with Different Smoothing for x and y



Example 52.4: Requesting Additional Output Tables

This example illustrates how to request output tables with summary statistics in addition to the default output tables. Using the same data as in the section "Getting Started: KDE Procedure" on page 4070, the following statements request univariate and bivariate summary statistics, percentiles, and levels of the kernel density estimate:

```
proc kde data=bivnormal;
   bivar x y / bivstats levels percentiles unistats;
run;
```

The resulting output is shown in Output 52.4.1.

Output 52.4.1 Bivariate Kernel Density Estimate Tables

The KDE Procedure

Inputs

Data Set	WORK.BIVNORMAL
Number of Observations Used	1000
Variable 1	x
Variable 2	y
Bandwidth Method	Simple Normal Reference

Controls

	x	y
Grid Points	60	60
Lower Grid Limit	-11.25	-10.05
Upper Grid Limit	9.1436	9.0341
Bandwidth Multiplier	1	1

Univariate Statistics

	x	y
Mean	-0.075	-0.070
Variance	9.73	9.93
Standard Deviation	3.12	3.15
Range	20.39	19.09
Interquartile Range	4.46	4.51
Bandwidth	0.99	1.00

Bivariate Statistics

Covariance	8.88
Correlation	0.90

Output 52.4.1 *continued*

Percentiles

	x	y
0.5	−7.71	−8.44
1.0	−7.08	−7.46
2.5	−6.17	−6.31
5.0	−5.28	−5.23
10.0	−4.18	−4.11
25.0	−2.24	−2.30
50.0	−0.11	−0.058
75.0	2.22	2.21
90.0	3.81	3.94
95.0	4.88	5.22
97.5	6.03	5.94
99.0	6.90	6.77
99.5	7.71	7.07

Levels

Percent	Density	Lower for x	Upper for x	Lower for y	Upper for y
1	0.001181	−8.14	8.45	−8.76	8.39
5	0.003031	−7.10	7.07	−7.14	6.77
10	0.004989	−6.41	5.69	−6.49	6.12
50	0.01591	−3.64	3.96	−3.58	3.86
90	0.02388	−1.22	1.19	−1.32	0.95
95	0.02525	−0.88	0.50	−0.99	0.62
99	0.02608	−0.53	0.16	−0.67	0.30
100	0.02629	−0.19	−0.19	−0.35	−0.35

The "Univariate Statistics" table contains standard univariate statistics for each variable, as well as statistics associated with the density estimate. Note that the estimated variances for both x and y are fairly close to the true values of 10.

The "Bivariate Statistics" table lists the covariance and correlation between the two variables. Note that the estimated correlation is equal to its true value to two decimal places.

The "Percentiles" table lists percentiles for each variable.

The "Levels" table lists contours of the density corresponding to percentiles of the bivariate data, and the minimum and maximum values of each variable on those contours. For example, 5% of the observed data have a density value less than 0.0030. The minimum x and y values on this contour are –7.10 and –7.14, respectively (the Lower for x and Lower for y columns), and the maximum values are 7.07 and 6.77, respectively (the Upper for x and Upper for y columns).

You can also request "Percentiles" or "Levels" tables with specific percentiles:

```
proc kde data=bivnormal;
   bivar x y / levels=2.5, 50, 97.5
               percentiles=2.5, 25, 50, 75, 97.5;
run;
```

The resulting "Percentiles" and "Levels" tables are shown in Output 52.4.2.

Output 52.4.2 Customized Percentiles and Levels Tables

The KDE Procedure

Percentiles

	x	y
2.5	-6.17	-6.31
25.0	-2.24	-2.30
50.0	-0.11	-0.058
75.0	2.22	2.21
97.5	6.03	5.94

Levels

Percent	Density	Lower for x	Upper for x	Lower for y	Upper for y
2.5	0.001914	-7.79	8.11	-7.79	7.74
50.0	0.01591	-3.64	3.96	-3.58	3.86
97.5	0.02573	-0.88	0.50	-0.99	0.30

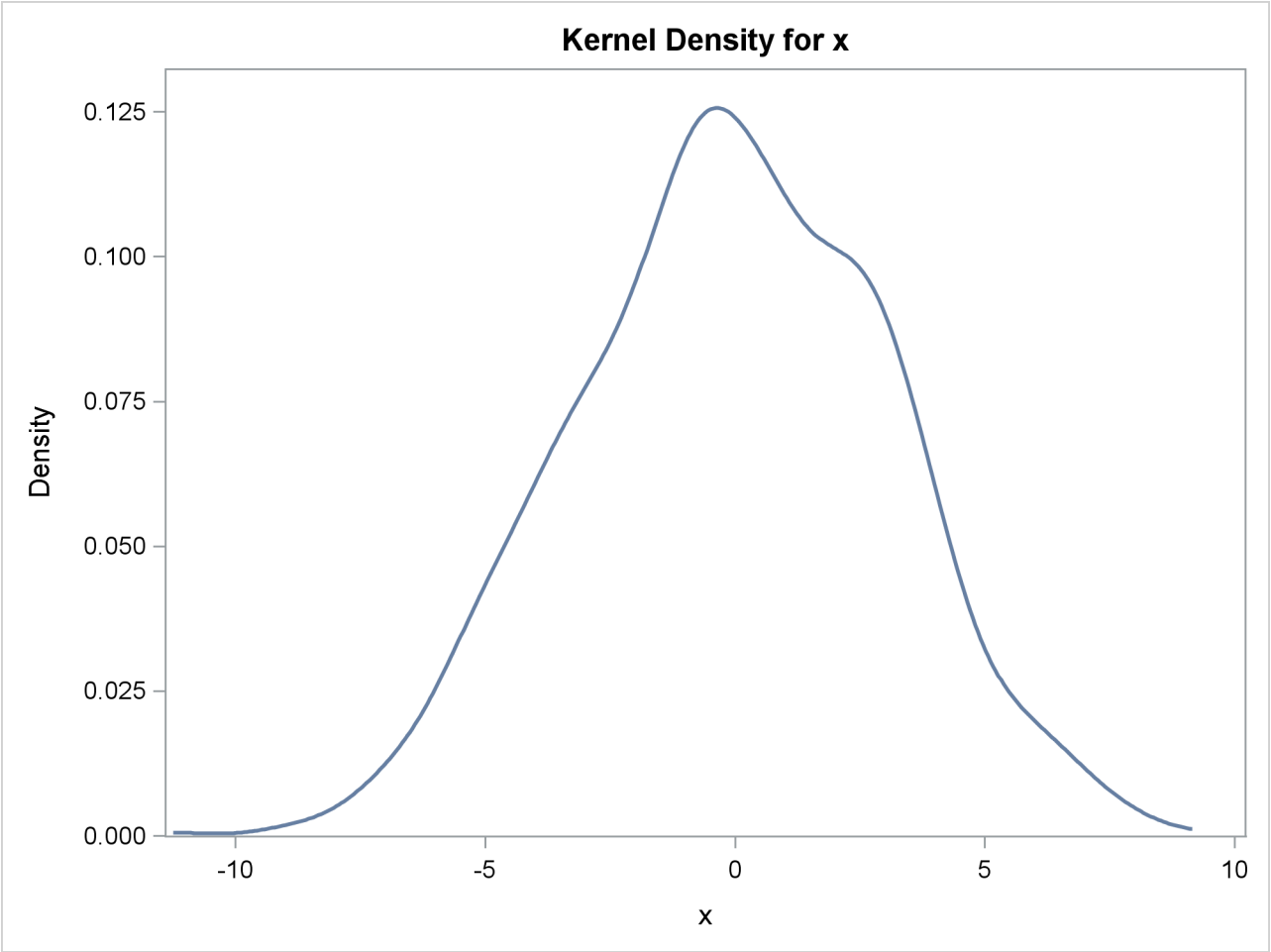
Example 52.5: Univariate KDE Graphics

This example uses data from the section "Getting Started: KDE Procedure" to illustrate the use of ODS Graphics. The following statements request the available univariate plots in PROC KDE:

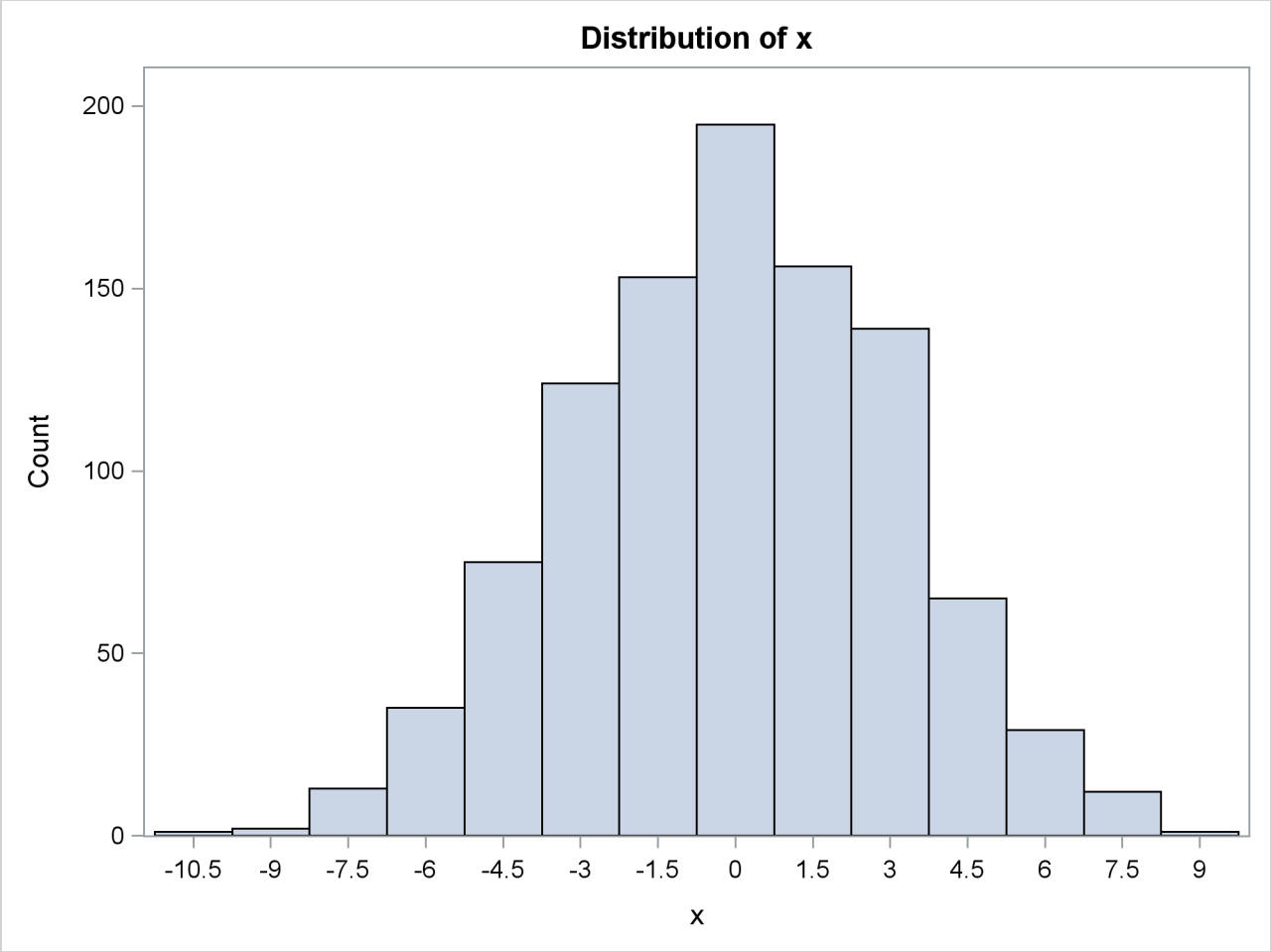
```
ods graphics on;
proc kde data=bivnormal;
   univar x / plots=(density histogram histdensity);
   univar x y / plots=densityoverlay;
run;
ods graphics off;
```

Graphs are requested by specifying the PLOTS= option in the UNIVAR statement with ODS Graphics enabled. Output 52.5.1, Output 52.5.2, and Output 52.5.3 show the kernel density estimate, histogram, and histogram with kernel density estimate overlaid, respectively, produced by the first UNIVAR statement.

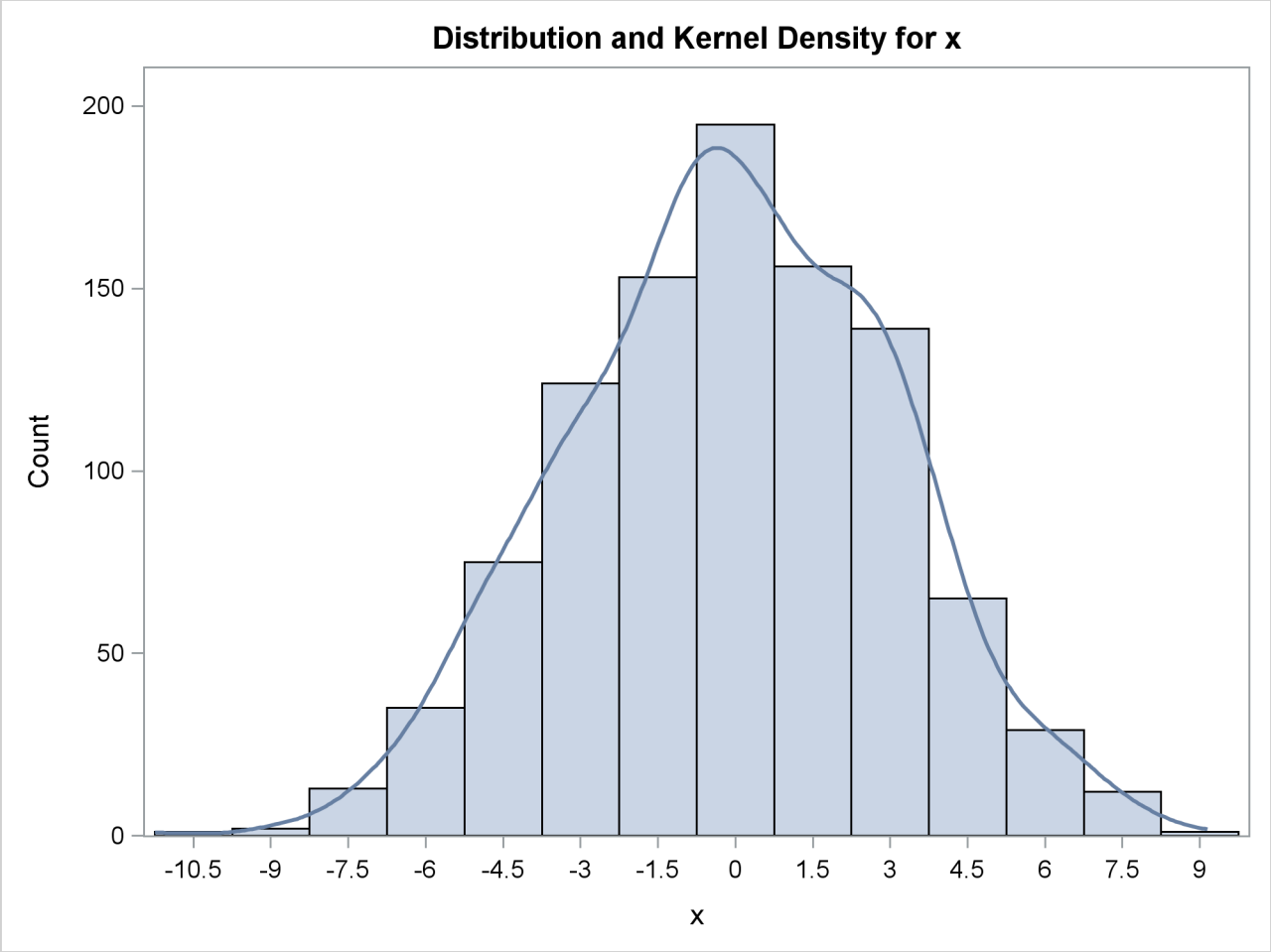
Output 52.5.1 Kernel Density Estimate



Output 52.5.2 Histogram

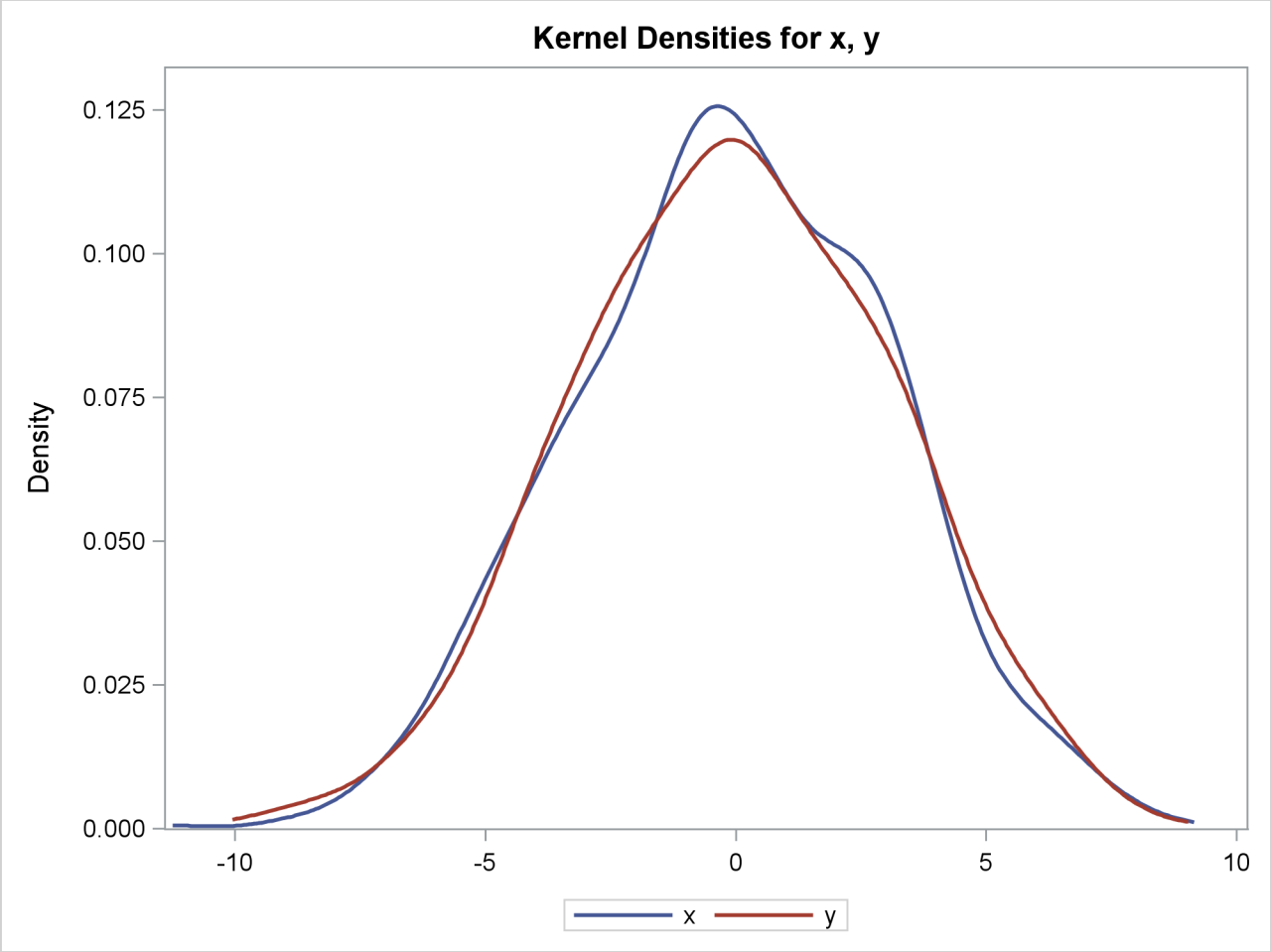


Output 52.5.3 Histogram with Overlaid Kernel Density Estimate



Output 52.5.4 shows the plot produced by the second UNIVAR statement, in which the kernel density estimates for x and y are overlaid.

Output 52.5.4 Overlaid Kernel Density Estimates



For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the KDE procedure, see the section "ODS Graphics" on page 4089.

Example 52.6: Bivariate KDE Graphics

This example illustrates the available bivariate graphics in PROC KDE. The octane data set comes from Rodriguez and Taniguchi (1980), where it is used for predicting customer octane satisfaction by using trained-rater observations. The variables in this data set are Rater and Customer. Either variable might have missing values. See the file *kdex3.sas* in the SAS Sample Library. The following statements create the octane data set:

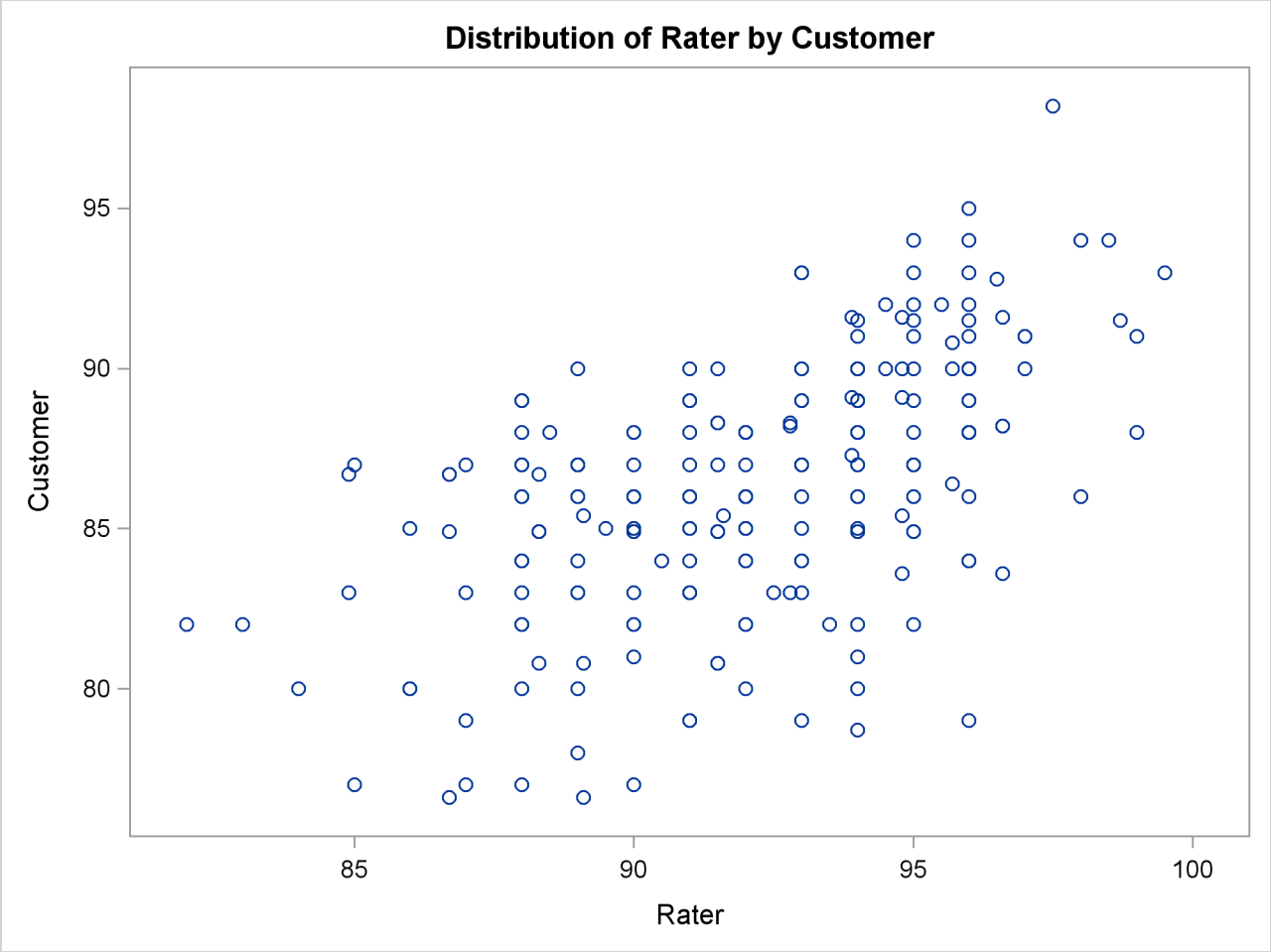
```
data octane;
   input Rater Customer;
   label Rater    = 'Rater'
         Customer = 'Customer';
   datalines;
94.5 92.0
94.0 88.0
94.0 90.0
93.0 93.0

   ... more lines ...

88.0 84.0
  .H 90.0
;
```

The following statements request all the available bivariate plots in PROC KDE:

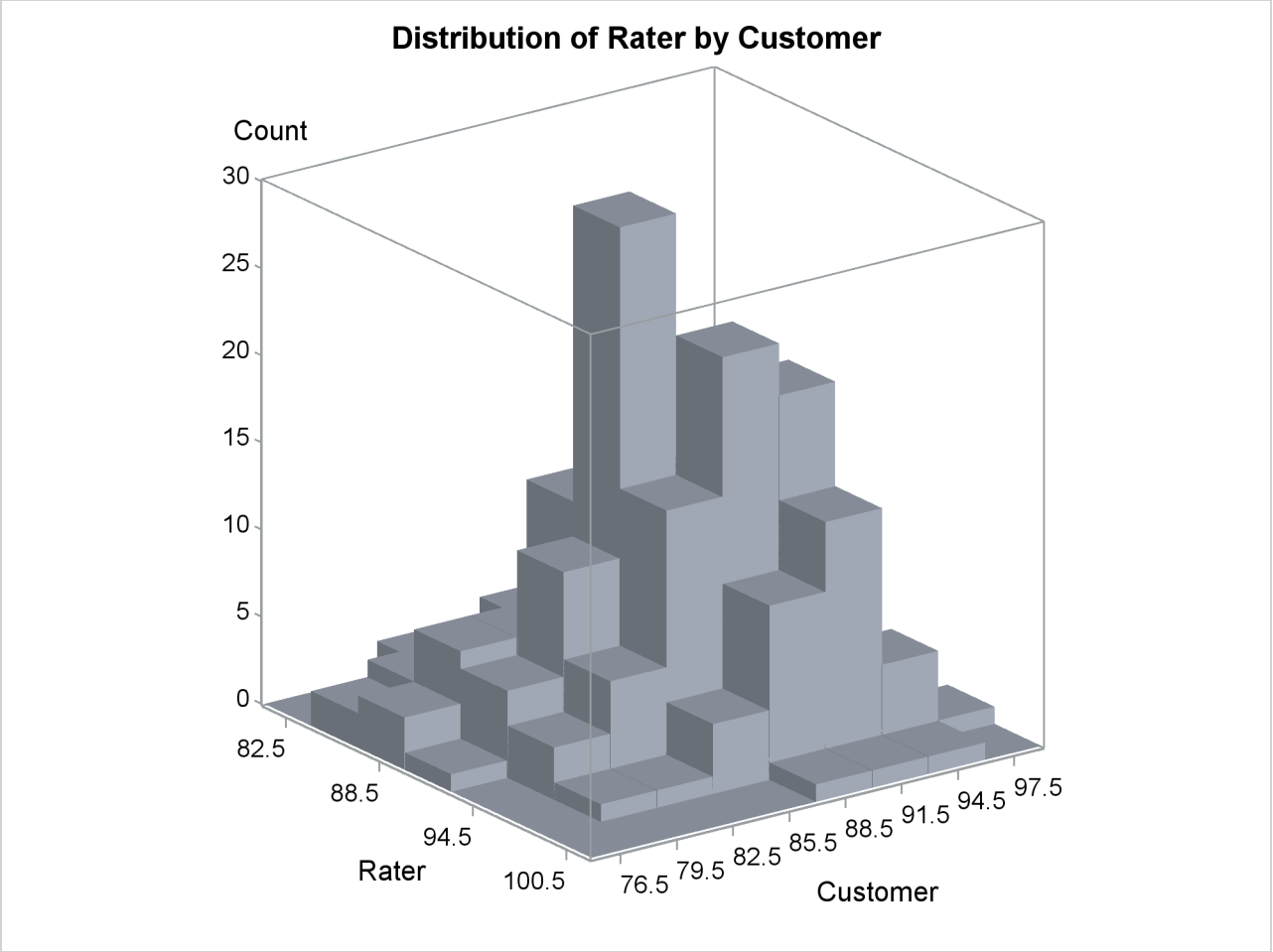
```
ods graphics on;
proc kde data=octane;
   bivar Rater Customer / plots=all;
run;
ods graphics off;
```

Output 52.6.1 shows a scatter plot of the data, Output 52.6.2 shows a bivariate histogram of the data, Output 52.6.3 shows a contour plot of bivariate density estimate, Output 52.6.4 shows a contour plot of bivariate density estimate overlaid with a scatter plot of data, Output 52.6.5 shows a surface plot of bivariate kernel density estimate, and Output 52.6.6 shows a bivariate histogram overlaid with a bivariate kernel density estimate. These graphical displays are requested by specifying the PLOTS= option in the BIVAR statement with ODS Graphics enabled. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the KDE procedure, see the section "ODS Graphics" on page 4089.

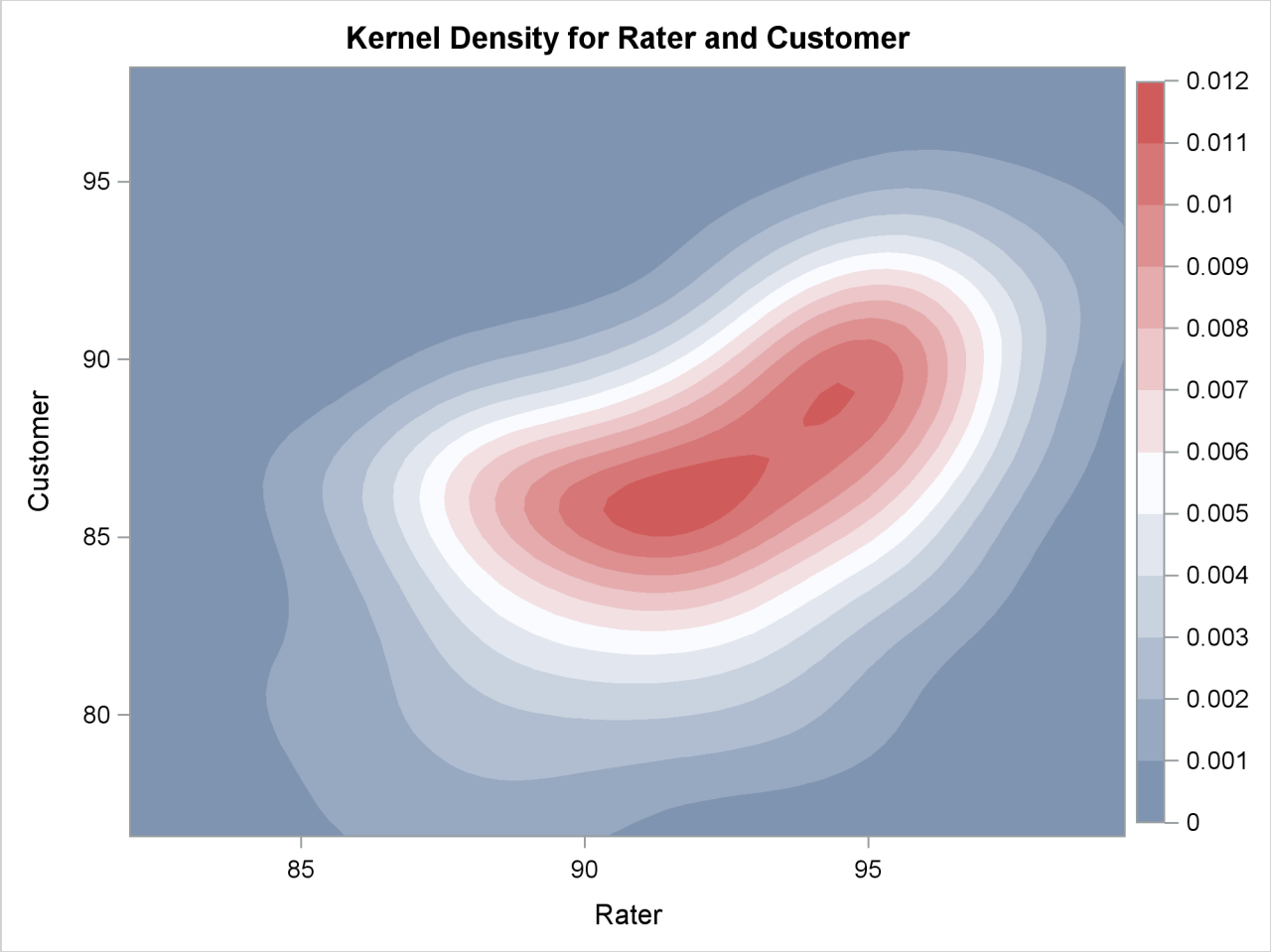
Output 52.6.1 Scatter Plot



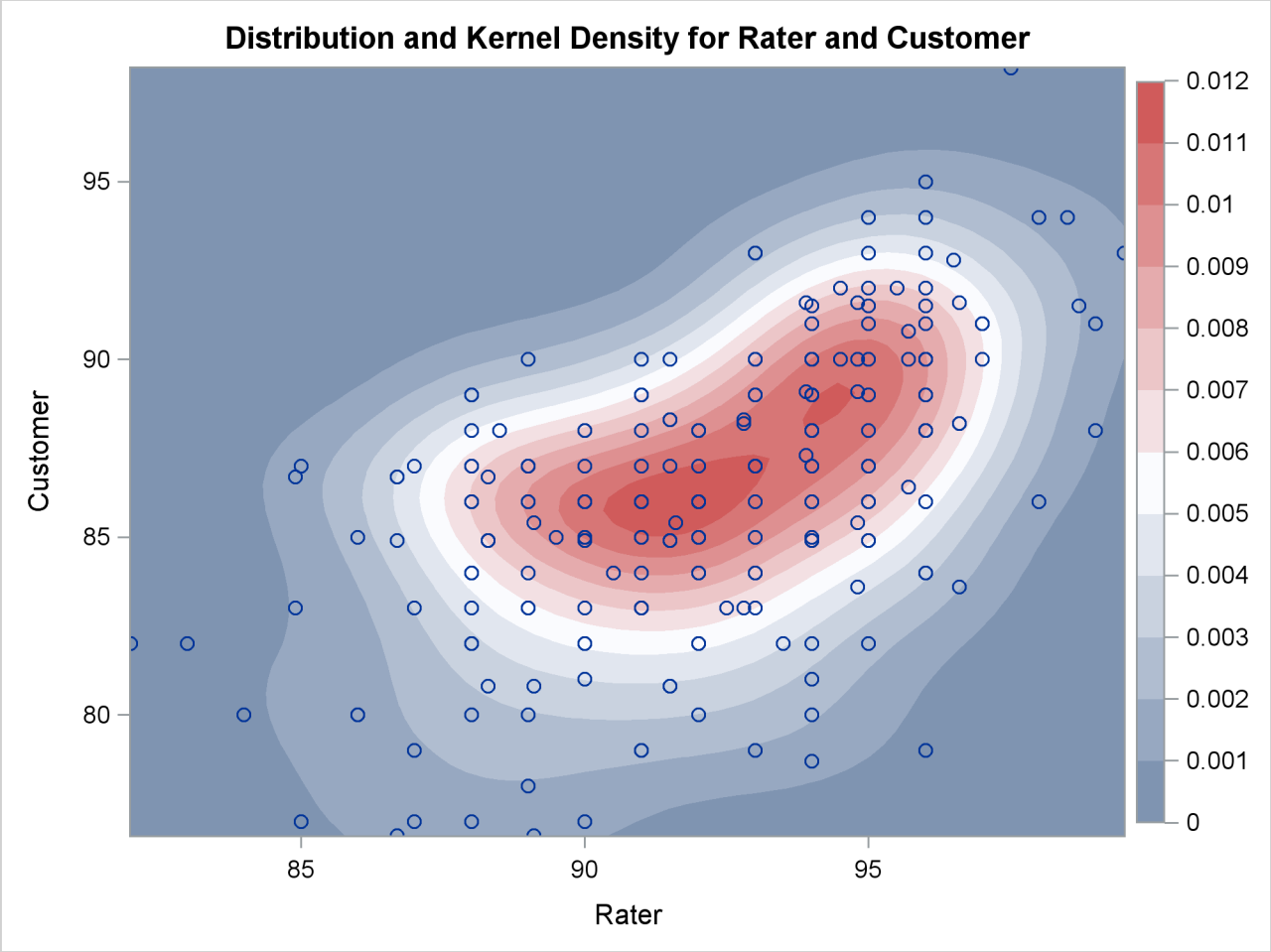
Output 52.6.2 Bivariate Histogram



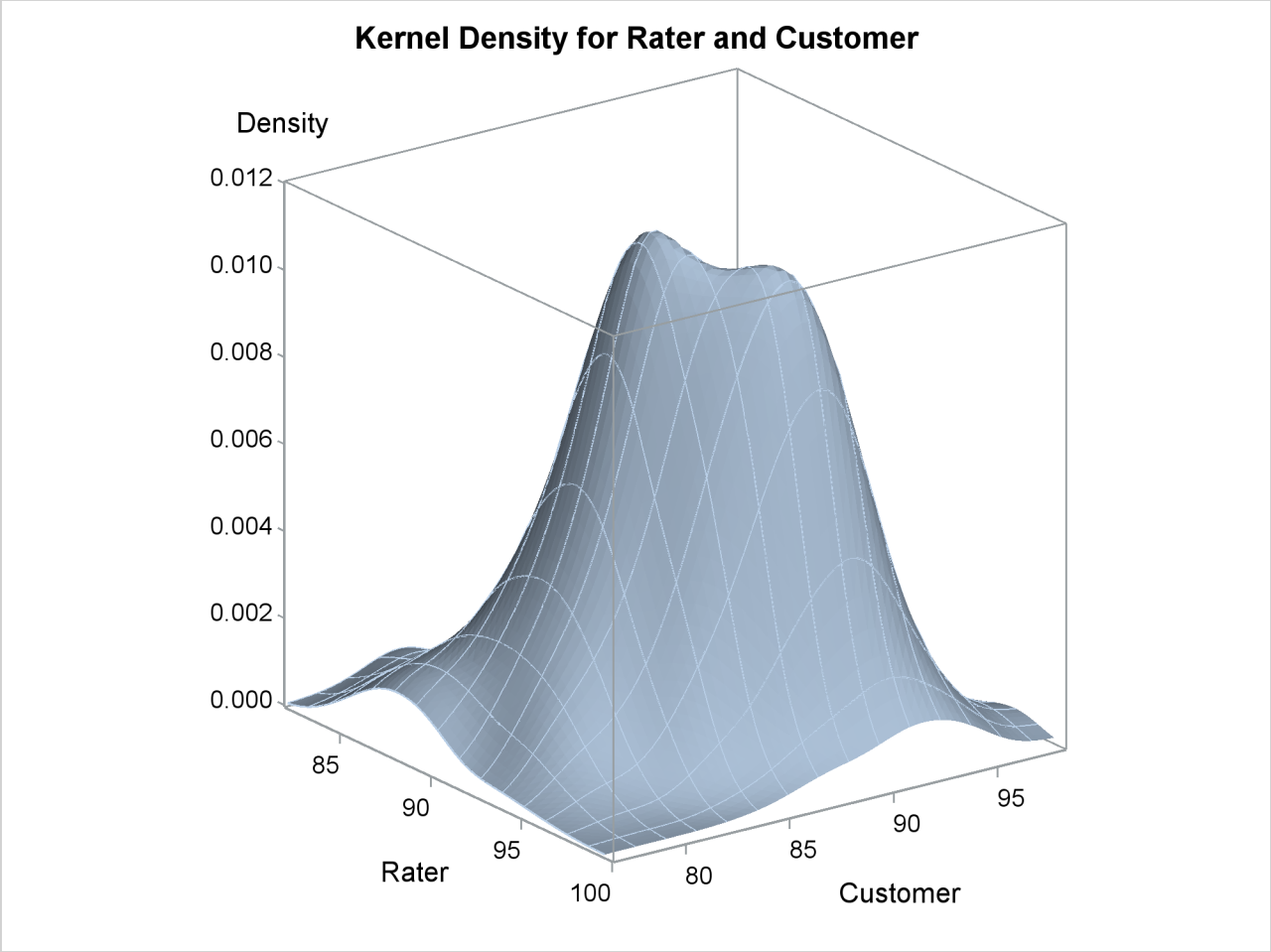
Output 52.6.3 Contour Plot



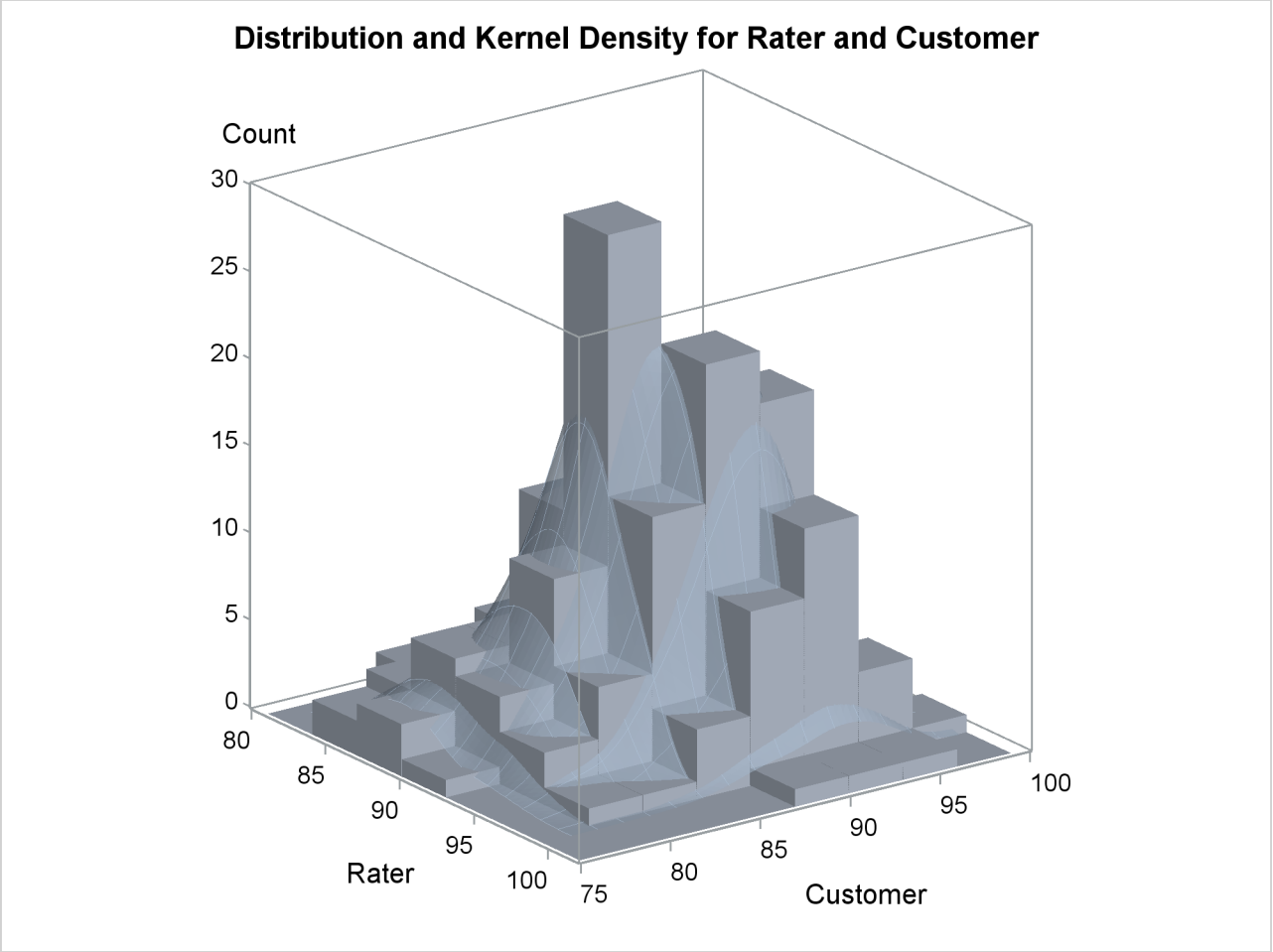
Output 52.6.4 Contour Plot with Overlaid Scatter Plot



Output 52.6.5 Surface Plot



Output 52.6.6 Bivariate Histogram with Overlaid Surface Plot



References

Bowman, A. W. and Foster, P. J. (1993), "Density Based Exploration of Bivariate Data," *Statistics and Computing*, 3, 171–177.

Fan, J. and Marron, J. S. (1994), "Fast Implementations of Nonparametric Curve Estimators," *Journal of Computational and Graphical Statistics*, 3, 35–56.

Jones, M. C., Marron, J. S., and Sheather, S. J. (1996), "A Brief Survey of Bandwidth Selection for Density Estimation," *Journal of the American Statistical Association*, 91, 401–407.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1988), *Numerical Recipes: The Art of Scientific Computing*, Cambridge: Cambridge University Press.

Rodriguez, R. N. and Taniguchi, B. Y. (1980), "A New Statistical Model for Predicting Customer Octane Satisfaction Using Trained-Rater Observations," *Transactions of the Society of Automotive Engineers*, 4213–4235.

Scott, D. W. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, New York: John Wiley & Sons.

Silverman, B. W. (1986), *Density Estimation for Statistics and Data Analysis*, New York: Chapman & Hall.

Terrell, G. R. and Scott, D. W. (1985), "Oversmoothed Nonparametric Density Estimates," *Journal of the American Statistical Association*, 80, 209–214.

Wand, M. P. (1994), "Fast Computation of Multivariate Kernel Estimators," *Journal of Computational and Graphical Statistics*, 3, 433–445.

Wand, M. P. and Jones, M. C. (1993), "Comparison of Smoothing Parameterizations in Bivariate Kernel Density Estimation," *Journal of the American Statistical Association*, 88, 520–528.

Subject Index

Syntax Index