
# SAS/STAT® 14.3 User's Guide

# The KDE Procedure

This document is an individual chapter from *SAS/STAT® 14.3 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *SAS/STAT® 14.3 User's Guide*. Cary, NC: SAS Institute Inc.

**SAS/STAT® 14.3 User's Guide**



SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

September 2017

# Chapter 68
# 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 68.1 and Figure 68.2, 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 5085.

**Figure 68.1** Contour Plot of Estimated Density

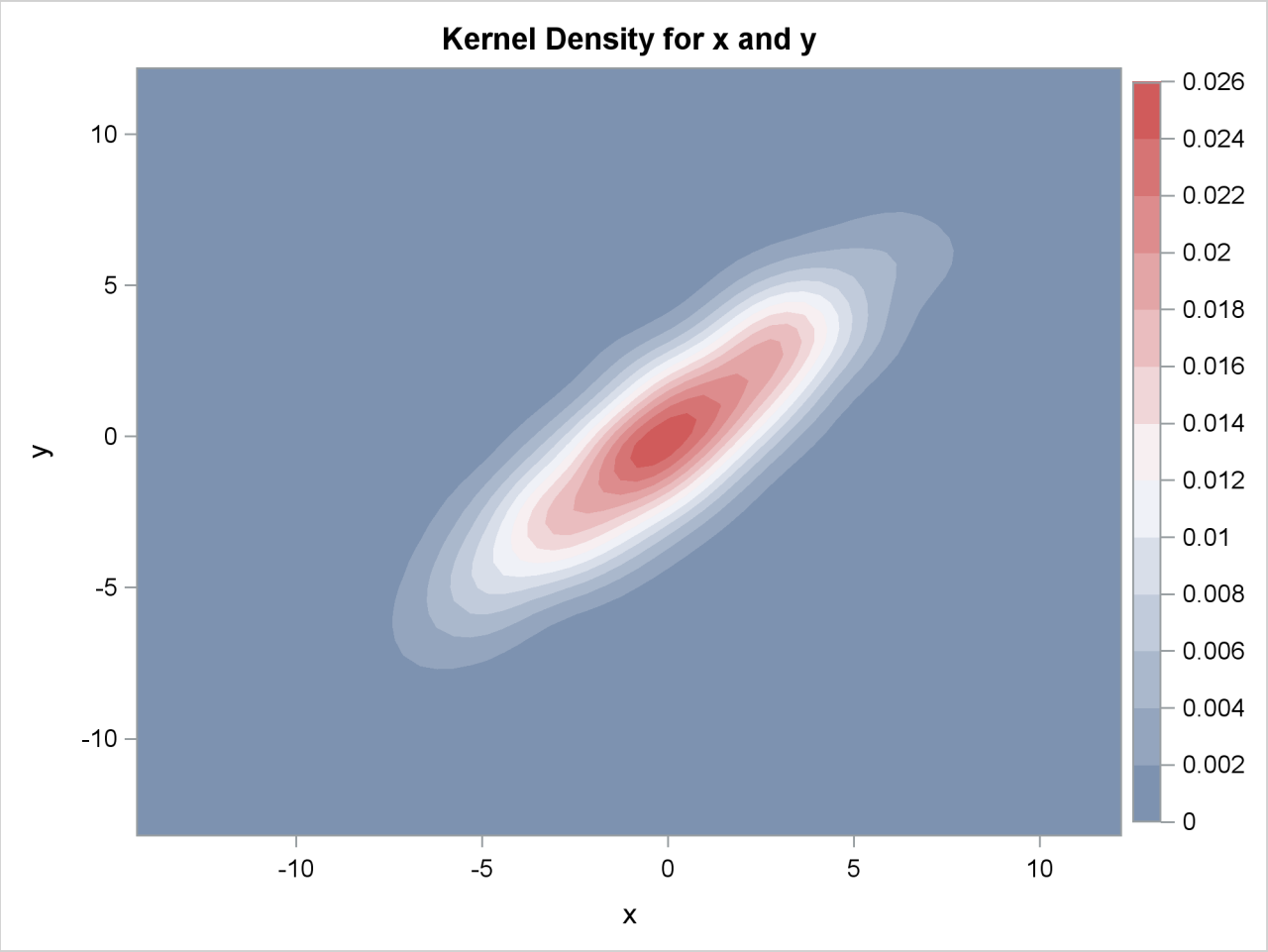
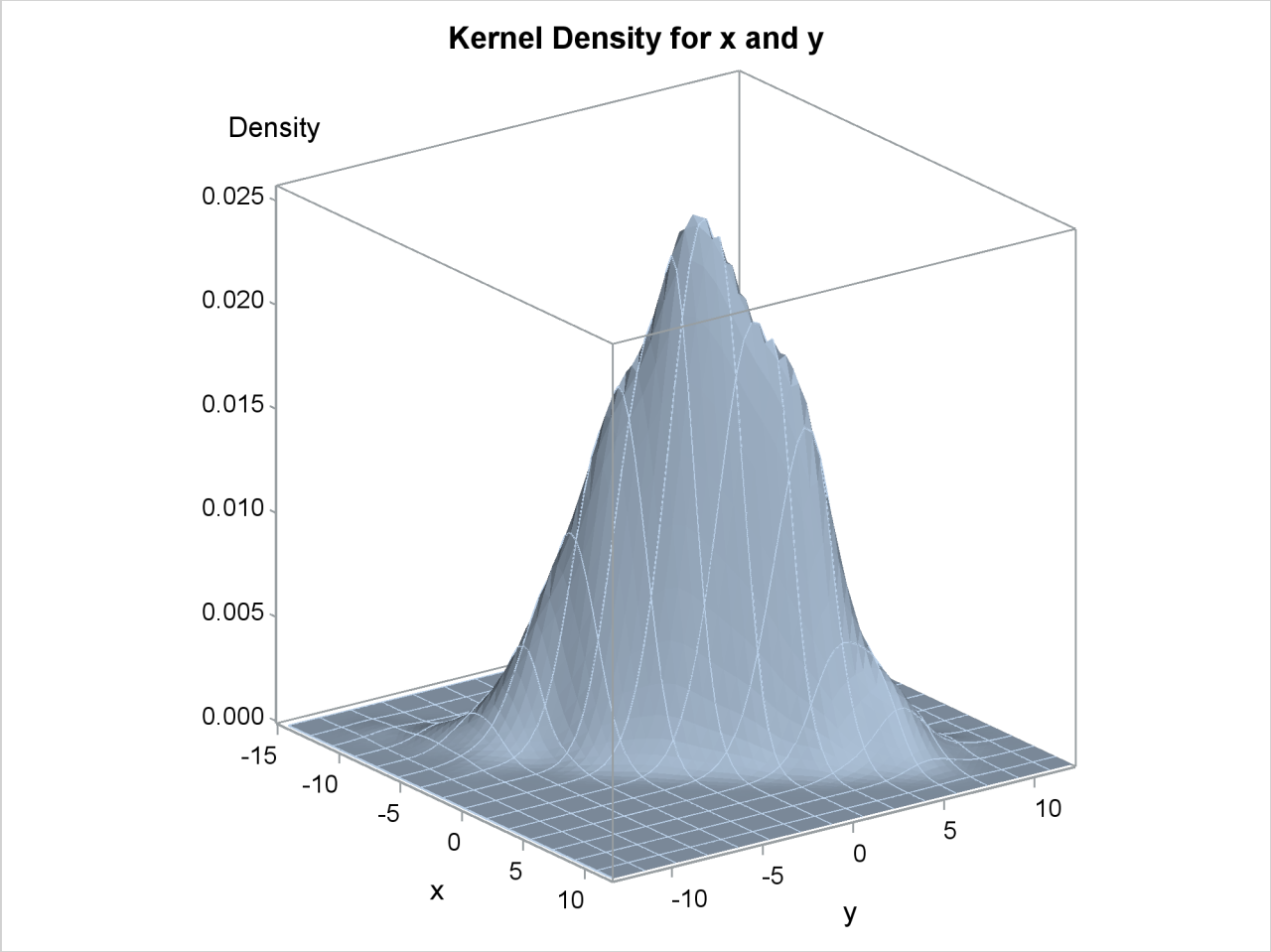


**Figure 68.2** Surface Plot of Estimated Density




The default output tables for this analysis are shown in Figure 68.3.

**Figure 68.3** 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 |

| Controls | | |
|---|---|---|
| | **x** | **y** |
| **Grid Points** | 60 | 60 |
| **Lower Grid Limit** | -14.27 | -13.18 |
| **Upper Grid Limit** | 12.17 | 12.165 |
| **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 $60 \times 60$ 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* > **;**
  **SCORE** *options* **;**
  **BY** *variables* **;**
  **FREQ** *variable* **;**
  **WEIGHT** *variable* **;**

The PROC KDE statement invokes the procedure. The BIVAR statement computes one or more bivariate kernel density estimates on a uniformly spaced grid. The UNIVAR statement computes one or more univariate kernel density estimates on a uniformly spaced grid. The SCORE statement produces kernel density estimates at arbitrary locations for an associated BIVAR or UNIVAR statement. You can specify any number of BIVAR, UNIVAR, and SCORE statements.

The following sections describe the PROC KDE statement and then describe the other statements in alphabetical order.

## PROC KDE Statement

**PROC KDE** < **DATA=***SAS-data-set* > **;**

The PROC KDE statement invokes the procedure and optionally specifies the input data set.

You can specify the following option:

**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

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

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

The BIVAR statement computes bivariate kernel density estimates for the specified variables. The *v-options* optionally specified in parentheses after a variable name apply only to that variable, and they override corresponding global *options* that are specified following a slash (/).

You must specify at least two variables, *v1* and *v2*. If you specify more than two variables, PROC KDE computes a bivariate kernel density estimate for each distinct pair of variables in the list. For example, if you specify the following statement, then a bivariate kernel density estimate is computed for each of the variable pairs (x, y), (x, z), and (y, z):

```
bivar x y z;
```

Alternatively, you can specify an explicit list of variable pairs, with each pair enclosed in parentheses. This requests a bivariate kernel density for each pair of variables. For example, if you specify the following statement, then bivariate kernel density estimates are computed for (x, y) and (y, z).

```
bivar (x y) (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.

Table 68.1 summarizes the *options* available in the BIVAR statement.

**Table 68.1** BIVAR Statement Options

| Option | Description |
|---|---|
| BIVSTATS | Produces a table for each density estimate |
| BW= | Specifies the bandwidth |
| BWM= | Specifies the bandwidth multiplier |
| CDF | Produces the distribution function |
| GRIDL= | Specifies the lower grid limit |
| GRIDU= | Specifies the upper grid limit |
| LEVELS | Produces 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 | Produces a table of percentiles |
| PLOTS= | Requests one or more plots |
| TRUNCATE | Restricts the lower and upper grid limits to the minimum and maximum observed values, respectively, for each variable |
| UNISTATS | Produces, for each density estimate, a table that contains standard univariate statistics and the bandwidths |

You can specify the following *options* in the BIVAR statement. Some *options* can be used as *v-options*, as indicated in the description of the *option*.

**BIVSTATS**

produces, for each density estimate, a table that contains the covariance and correlation between the two variables.

**BW=***number*

specifies the bandwidth to apply to each variable in each kernel density estimate. Larger bandwidths produce a smoother estimate, whereas smaller bandwidths produce a rougher estimate. To specify different bandwidths for different variables, specify BW=*number* as a *v-option*. By default, the bandwidth is set automatically by the simple normal reference method (see the section "Bandwidth Selection" on page 5083).

**BWM=***number*

specifies the bandwidth multiplier to apply to the corresponding bandwidth for each variable. Values of *number* greater than 1 increase the effective bandwidth and produce a smoother estimate. Values less than 1 decrease the effective bandwidth and produce a rougher estimate. To specify different bandwidth multipliers for different variables, specify BWM=*number* as a *v-option*. By default, BWM=1.

**CDF**

computes the distribution function in addition to the density function for each pair of variables. The distribution function is obtained by a seminumerical technique as described in the section "Kernel Distribution Estimates" on page 5078.

**GRIDL=***number*

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

**GRIDU=***number*

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

**LEVELS**

**LEVELS=(***numlist***)**

computes a table of levels (called "Levels") for contours of the bivariate density. The number of contours is equal to the number of values in *numlist*, where each value in *numlist* specifies a percentage to be used in calculating the density volume that is enclosed by the contour. The contours are defined such that the density has a constant level along each contour, and the volume enclosed by each contour corresponds to the total density volume minus the specified percentage of the total volume. In other words, the contours correspond to slices or levels of the density surface that are taken along the density axis. The "Levels" table also provides the minimum and maximum values for each contour along the directions of the two data variables. By default, LEVELS=(1, 5, 10, 50, 90, 95, 99, 100).

**NGRID=***number*

**NG=***number*

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

**NOPRINT**
suppresses output tables. You can use this option when you want to produce only graphical output.

**OUT=***SAS-data-set*
names the output data set in which to save kernel density estimates. 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, whose value corresponds to grid coordinates for the first variable
- value2, whose value corresponds to grid coordinates for the second variable
- density, whose values are equal to kernel density estimate at the associated grid point
- count, whose values represent the number of original observations contained in the bin that corresponds to a grid point
- distribution, whose values are equal to the distribution estimate at the associated grid point (this variable is included only when the CDF global option is specified)

**PERCENTILES**

**PERCENTILES=***numlist*
produces a table of percentiles for each BIVAR variable. You can specify a list of percentiles to be computed in *numlist*. 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***)> <... *plot-request* <(***options***)>>)**
specifies which plots of the bivariate data and kernel density estimate to produce. 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 615 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, only the requested plots are created.

You can specify the following *plot-requests*:

**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. You can specify one or both of the following *view-options* within parentheses:

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

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

**HISTSURFACE < (*view-options*) >**
produces a bivariate histogram of the data overlaid with a surface plot of the bivariate kernel density estimate. You can specify one or both of the following *view-options* within parentheses:

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

**TILT=***angle*
tilts the histogram and kernel density surface *angle* degrees, where –180 < *angle* < 180. By default, TILT=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. You can specify one or both of the following *view-options* within parentheses:

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

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

**TRUNCATE**
sets the lower grid limit for each variable to the minimum observed for that variable, and sets the upper grid limit for each variable to the maximum observed value for that variable.

**NOTE:** The GRIDL and GRIDU options take precedence over the TRUNCATE option. If one or both are specified, the corresponding lower and upper grid limits are set accordingly.

**UNISTATS**
produces for each density estimate a table that contains standard univariate statistics for each of the variable pairs and the bandwidths that are used to compute the kernel density estimate. The statistics indexed in the table are the mean, variance, standard deviation, range, and interquartile range.

## 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 *SAS Visual Data Management and Utility 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.

## SCORE Statement

**SCORE DATA**=*SAS-data-set* **OUT**=*SAS-data-set* < / *option* > **;**

The SCORE statement produces kernel density estimates at arbitrary locations for an associated UNIVAR or BIVAR statement.

You must specify the following arguments:

**DATA=***SAS-data-set*
specifies the input SAS data set.

For a univariate density, the input data set contains the variable v1, where v1 is a variable that appears in a UNIVAR statement. The values of v1 indicate arbitrary points for which a density estimate is requested.

For a bivariate density, the input data set contains variables v1 and v2, where v1 and v2 are variables that appear in a BIVAR statement. Pairs (v1, v2) indicate arbitrary points for which a density estimate is requested.

When you specify a BY statement, the DATA= data set must not contain any of the BY variables. The entire data set is scored for each BY group.

**OUT=***SAS-data-set*
names the output SAS data set to be produced by the SCORE statement.

For a univariate density, the output data set contains the following variables:

- var, whose value is the name of the variable in the DATA= data set
- value, whose values are taken from the DATA= data set
- density, whose values are equal to the kernel density estimate
- distribution, whose values are equal to the distribution estimate (this variable is included only when the CDF global option is specified in the corresponding UNIVAR statement)

For a bivariate density, the output data set has the following variables:

- variable1, whose value is the name of the first variable in the DATA= data set
- value1, whose values are taken from the DATA= data set
- variable2, whose value is the name of the second variable in the DATA= data set
- value2, whose values are taken from the DATA= data set
- density, whose values are equal to the kernel density estimate
- distribution, whose values are equal to the distribution estimate (this variable is included only when the CDF global option is specified in the corresponding BIVAR statement)

You can provide the following *option* in the SCORE statement.

**METHOD=INTERP | EXACT**

requests a particular scoring method. You can specify the following values:

**EXACT** corresponds to the evaluation of the appropriate binned density equation from the section "Binning" on page 5077 and the analytical distribution equation from the section "Kernel Distribution Estimates" on page 5078.

**INTERP** corresponds to linear interpolation of the density estimate and seminumerical distribution estimation. The seminumerical distribution technique is described in "Kernel Distribution Estimates" on page 5078. **NOTE:** Attempting to score outside the grid results in 0 for the density estimate.

By default, METHOD=INTERP.

You can include multiple SCORE statements. Each SCORE statement applies to the first UNIVAR or BIVAR statement that specifies the same variables as are included in the DATA= data set. The order of variables matters for the bivariate case. Therefore, a SCORE statement that specifies a DATA= data set consisting of variables x and y (in that order) will match only a BIVAR statement of one of the following forms:

**BIVAR (***x y***)** *. . .* < */ options* > **;**

**BIVAR** *x y . . .* < */ options* > **;**

### Example

Suppose the data set MyData contains the variables x and y and the data sets MyScoreInX, MyScoreInY, and MyScoreInXY contain the variables x, y, and (x, y) respectively. The following statements request both the individual (univariate) and joint (bivariate) kernel density estimates and distributions and selectively score them:

```
proc kde data=MyData;
   univar x y;
   bivar x y / CDF;
   score data=MyScoreInX out=MarginalX;
   score data=MyScoreInY out=MarginalY;
   score data=MyScoreInXY out=JointXY;
run;
```

The first SCORE statement is associated with the UNIVAR statement and produces the MarginalX output data set. This data set contains variables x, value, and density, where density is the density function at value.

The second SCORE statement is associated with the UNIVAR statement and produces the MarginalY output data set. This data set contains variables y, value and density, where density is the density function at value.

The third SCORE statement is associated with the BIVAR statement and produces the JointXY output data set. This data set contains variables x, value1, y, value2, density, and distribution, where density and distribution are the density and distribution functions, respectively, at (value1, value2).

## 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* 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 68.2 summarizes the *options* available in the UNIVAR statement.

**Table 68.2** UNIVAR Statement Options

| Option | Description |
|---|---|
| BW= | Specifies a bandwidth |
| BWM= | Specifies a bandwidth multiplier |
| CDF | Produces the distribution function |
| GRIDL= | Specifies a lower grid limit |
| GRIDU= | Specifies an upper grid limit |
| LEVELS | Produces a table of levels for contours of the univariate density |
| METHOD= | Specifies which method to use to compute the bandwidth |
| NGRID= | Specifies a number of grid points |
| NOPRINT | Suppresses output tables |
| OUT= | Specifies the output SAS data set to contain the kernel density estimate |
| PERCENTILES | Produces a table of percentiles |
| PLOTS= | Produces 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 to be used in determining the Sheather-Jones plug-in bandwidth |
| SJPITOL= | Specifies the tolerance for termination of the bisection algorithm |
| TRUNCATE | Restricts the lower and upper grid limits to the minimum and maximum observed values, respectively, for each variable |
| UNISTATS | Produces, for each variable, a table that contains standard univariate statistics and the bandwidth |

You can specify the following *options* in the UNIVAR statement. Some *options* can be used as *v-options*, as indicated in the description of the *option*.

**BW=***number*

specifies the bandwidth to apply to each variable in each kernel density estimate. Larger bandwidths produce a smoother estimate, whereas smaller bandwidths produce a rougher estimate. To specify different bandwidths for different variables, specify BW=*number* as a *v-option*. By default, the bandwidth is set automatically by the Sheather-Jones plug-in method (see the section "Bandwidth Selection" on page 5083).

**BWM=***number*

specifies the bandwidth multiplier to apply to the corresponding bandwidth for each variable. Values of *number* greater than 1 increase the effective bandwidth and produce a smoother estimate. Values less than 1 decrease the effective bandwidth and produce a rougher estimate. To specify different bandwidth multipliers for different variables, specify BWM=*number* as a *v-option*. By default, BWM=1.

**CDF**

computes the distribution function in addition to the density function for each variable. The distribution function is obtained by a seminumerical technique as described in the section "Kernel Distribution Estimates" on page 5078.

**GRIDL=***number*

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

**GRIDU=***number*

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

**LEVELS**

**LEVELS=(***numlist***)**

computes a table of levels (called "Levels") for contours of the univariate density. The number of contours is equal to the number of values in *numlist*, where each value in *numlist* specifies a percentage to be used in calculating the density area that is enclosed by the contour. The contours are defined such that the density has a constant level along each contour, and the area enclosed by each contour corresponds to the total density area minus the specified percentage of the total area. In other words, the contours correspond to slices or levels of the density surface that are taken along the density axis. The "Levels" table also provides the minimum and maximum values for each contour along the direction of the data variable. By default, LEVELS=(1, 5, 10, 50, 90, 95, 99, 100).

**METHOD=SJPI | SNR | SNRQ | SROT | OS**

specifies the method for computing the bandwidth. You can specify the following values:

| | |
|---|---|
| **SJPI** | Sheather-Jones plug-in method |
| **SNR** | simple normal reference method |
| **SNRQ** | simple normal reference method using interquartile range |
| **SROT** | Silverman's rule of thumb method |
| **OS** | oversmoothed method |

For a description of these methods, see the section "Bandwidth Selection" on page 5083 and Jones, Marron, and Sheather (1996). By default, METHOD=SJPI.

**NOTE:** The BW= option takes precedence over the METHOD= option. If both are specified, the METHOD= option is ignored.

**NGRID=***number*

**NG=***number*

specifies a number of grid points to use for each kernel density estimate. To specify different numbers of grid points for different variables, specify NGRID=*number* as a *v-option*. By default, NGRID=401.

**NOPRINT**

suppresses output tables. You can use this option when you want to produce only graphical output.

**OUT=***SAS-data-set*

names the output SAS data set to contain 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, whose value corresponds to grid coordinates for the variable
- density, whose values are equal to the kernel density estimate at the associated grid point
- count, whose values indicate the number of original observations contained in the bin that corresponds to a grid point
- distribution, whose values are equal to the distribution estimate at the associated grid point (*appears only when the CDF global option is specified*)

**PERCENTILES**

**PERCENTILES=***numlist*

produces a table of percentiles for each UNIVAR variable. You can specify a list of percentiles to be computed in *numlist*. 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***) > < ... ***plot-request*** < (***options***) > >)**

specifies which plots of the univariate kernel density estimate to produce. 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 615 in Chapter 21, "Statistical Graphics Using ODS."

You can specify the following *plot-requests*, each of which (except for DENSITYOVERLAY) produces a separate plot for every variable listed in the UNIVAR statement:

**ALL** produces all plots.

**DENSITY** produces the univariate kernel density estimate curve.

**DENSITYOVERLAY** produces the overlaid univariate kernel density estimate curves. If you specify more than one variable in the UNIVAR statement, PROC KDE overlays the density curves for all the variables on a single plot.

**HISTDENSITY** produces the univariate histogram of data overlaid with kernel density estimate curve.

**HISTOGRAM** produces the 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, only the requested plots are created.

**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 to use in determining the Sheather-Jones plug-in bandwidth. By default, SJPINUM=21.

**SJPITOL=***number*
specifies the tolerance for terminating the bisection algorithm that is used in computing the Sheather-Jones plug-in bandwidth. By default, SJPITOL=0.001.

**TRUNCATE**
sets the lower grid limit for each variable to the minimum observed for that variable, and sets the upper grid limit for each variable to the maximum observed value for that variable.

**NOTE:** The GRIDL and GRIDU options take precedence over the TRUNCATE option. If one or both are specified, the corresponding lower and upper grid limits are set accordingly.

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

## Examples

Suppose the data set MyData contains the variables x1, x2, x3, and x4. The following statements request a univariate kernel density estimate for each of these variables:

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

## WEIGHT Statement

**WEIGHT** *variable* **;**

The WEIGHT statement specifies a variable that weights the observations in computing the kernel density estimate. Observations that have 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. Be cautious in using data sets that have 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 after a bandwidth has been selected. An optional task is the construction of the kernel distribution estimate. The following sections discuss density and distribution estimation, kernel bandwidth selection, and efficient computation techniques.

## 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

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

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

Now set

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

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

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

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 $c_i$s 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:

$$grid = \begin{bmatrix} (x_1, y_1) & (x_1, y_2) & \dots & (x_1, y_{g_Y}) \\ (x_2, y_1) & (x_2, y_2) & \dots & (x_2, y_{g_Y}) \\ \vdots & & & \\ (x_{g_X}, y_1) & (x_{g_X}, y_2) & \dots & (x_{g_X}, y_{g_Y}) \end{bmatrix}$$

The density estimates are then

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

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

## Kernel Distribution Estimates

The distribution function can be obtained by integrating the kernel density estimate (Azzalini 1981). PROC KDE provides both an analytical and a seminumerical integration approach, each of which involves the closed form solution to the integral of the binned density estimator from the previous section.

For the univariate case, the distribution function is

$$\begin{aligned} \hat{F}(x) &= \int_{-\infty}^{x} \hat{f}(u)du \\ &= \frac{1}{N} \sum_{i=1}^{g} c_i \Phi_h(x - x_i) \end{aligned}$$

where $h$ is the univariate kernel bandwidth and

$$\Phi_h(x) = \frac{1}{\sqrt{2\pi}h} \int_{-\infty}^{x} \exp\left(-\frac{u^2}{2h^2}\right) du$$

For the bivariate case, the distribution function is

$$\begin{aligned} \hat{F}(x, y) &= \int_{-\infty}^{x} \int_{-\infty}^{y} \hat{f}(u, v)dudv \\ &= \frac{1}{N} \sum_{j=1}^{g_Y} \Phi_{h_Y}(y - y_j) \sum_{i=1}^{g_X} c_{i,j} \Phi_{h_X}(x - x_i) \end{aligned}$$

where $h_X$ and $h_Y$ are the bivariate kernel bandwidths for variables $x$ and $y$.

The analytical integration approach is simply the direct evaluation of the appropriate distribution function equation. The seminumerical integration approach is a mixture of direct evaluation of the distribution function equation and numerical integration via the extended trapezoidal rule (Press et al. 1992). This mixture depends on whether the upper integration limits fall inside or outside the binning grid. In general, there are three cases:

- Integration limits precede leading grid edges.
- Integration limits fall within grid.
- Integration limits follow trailing grid edges.

When the integration limits lead or trail the grid edges, the seminumerical approach is identical to the analytical approach. Otherwise, the seminumerical approach splits the overall integral into integrals from minus infinity to the lower grid edges, and an integral from the leading grid edges to the upper integration limits. For the univariate case, this split becomes

$$\begin{aligned}\hat{F}(x) &= \int_{-\infty}^{x_1} \hat{f}(u)du + \int_{x_1}^{x} \hat{f}(u)du \\ &= \hat{F}(x_1) + \int_{x_1}^{x} \hat{f}(u)du\end{aligned}$$

The term $\hat{F}(x_1)$ is simply the analytical distribution function up to the leading grid edge. The second term is evaluated numerically:

- If $x$ coincides with a grid element $x_k$, the overall integral is

  $$\hat{F}(x) \approx \hat{F}(x_1) + \tilde{F}(x_k)$$

  where

  $$\tilde{F}(x_k) = \left[\sum_{m=1}^{k} \hat{f}(x_m) - \frac{1}{2}(\hat{f}(x_1) + \hat{f}(x_k))\right]\delta$$

- If $x$ does not coincide with a grid element, then the numerical integral is approximated by linear interpolation that uses the nearest grid elements $p$ and $p + 1$:

  $$\tilde{F}(x) = \tilde{F}(x_p) + \frac{(\tilde{F}(x_{p+1}) - \tilde{F}(x_p))}{x_{p+1} - x_p}(x - x_p).$$

The bivariate case is similar to the univariate case, although there are multiple analytical terms to evaluate due to the nature of the integral:

$$\hat{F}(x,y) = \hat{F}(x,y_1) + \hat{F}(x_1,y) - \hat{F}(x_1,y_1) + \int_{x_1}^{x}\int_{y_1}^{y} \hat{f}(u,v)dudv$$

As with the univariate case, the fourth term is evaluated numerically:

- If $(x, y)$ coincides with a grid element, then the overall integral is

  $$\hat{F}(x_k,y_\ell) \approx \hat{F}(x,y_1) + \hat{F}(x_1,y) - \hat{F}(x_1,y_1) + \tilde{F}(x_k,y_\ell)$$

  where $\tilde{F}(x_k, y_\ell)$ is recursively computed via

  $$\begin{aligned}
  \tilde{F}(x_k) &= \left[\sum_{n=1}^{\ell} \hat{f}(x_k,y_n) - \frac{1}{2}(\hat{f}(x_k,y_1) + \hat{f}(x_k,y_\ell))\right]\delta_Y \\
  \tilde{F}(x_k,y_\ell) &\approx \left[\sum_{m=1}^{k} \hat{F}(x_m) - \frac{1}{2}(\hat{F}(x_1) + \hat{F}(x_k))\right]\delta_X
  \end{aligned}$$

- If $(x, y)$ does not coincide with a grid element, then the numerical integral is approximated by bilinear interpolation that uses the nearest grid elements $p, p+1, q,$ and $q+1$:

  $$\begin{aligned}
  \tilde{F}(y_q) &= \tilde{F}(x_p,y_q) + \frac{(\tilde{F}(x_{p+1},y_q) - \tilde{F}(x_p,y_q))}{x_{p+1}-x_p}(x-x_p) \\
  \tilde{F}(y_{q+1}) &= \tilde{F}(x_p,y_{q+1}) + \frac{(\tilde{F}(x_{p+1},y_{q+1}) - \tilde{F}(x_p,y_{q+1}))}{x_{p+1}-x_p}(x-x_p) \\
  \tilde{F}(x,y) &= \tilde{F}(y_q) + \frac{(\tilde{F}(y_{q+1}) - \tilde{F}(y_q))}{y_{q+1}-y_q}(y-y_q)
  \end{aligned}$$

## 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

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

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 $\kappa_{i,j} = \frac{1}{n}\varphi_{\mathbf{h}}(i\delta_X, j\delta_Y)\ i = 0, 1, \ldots, l_X,\ j = 0, 1, \ldots, l_Y$.

The bin count matrix **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 68.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

**Table 68.3** 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 615 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 614 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 68.4.

**Table 68.4** 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 |
| 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.

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, only the requested plots are created.

## Univariate Plots

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

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, only the requested plots are created.

## 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 $\mathrm{Nbins}_X$ 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 68.1: 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:

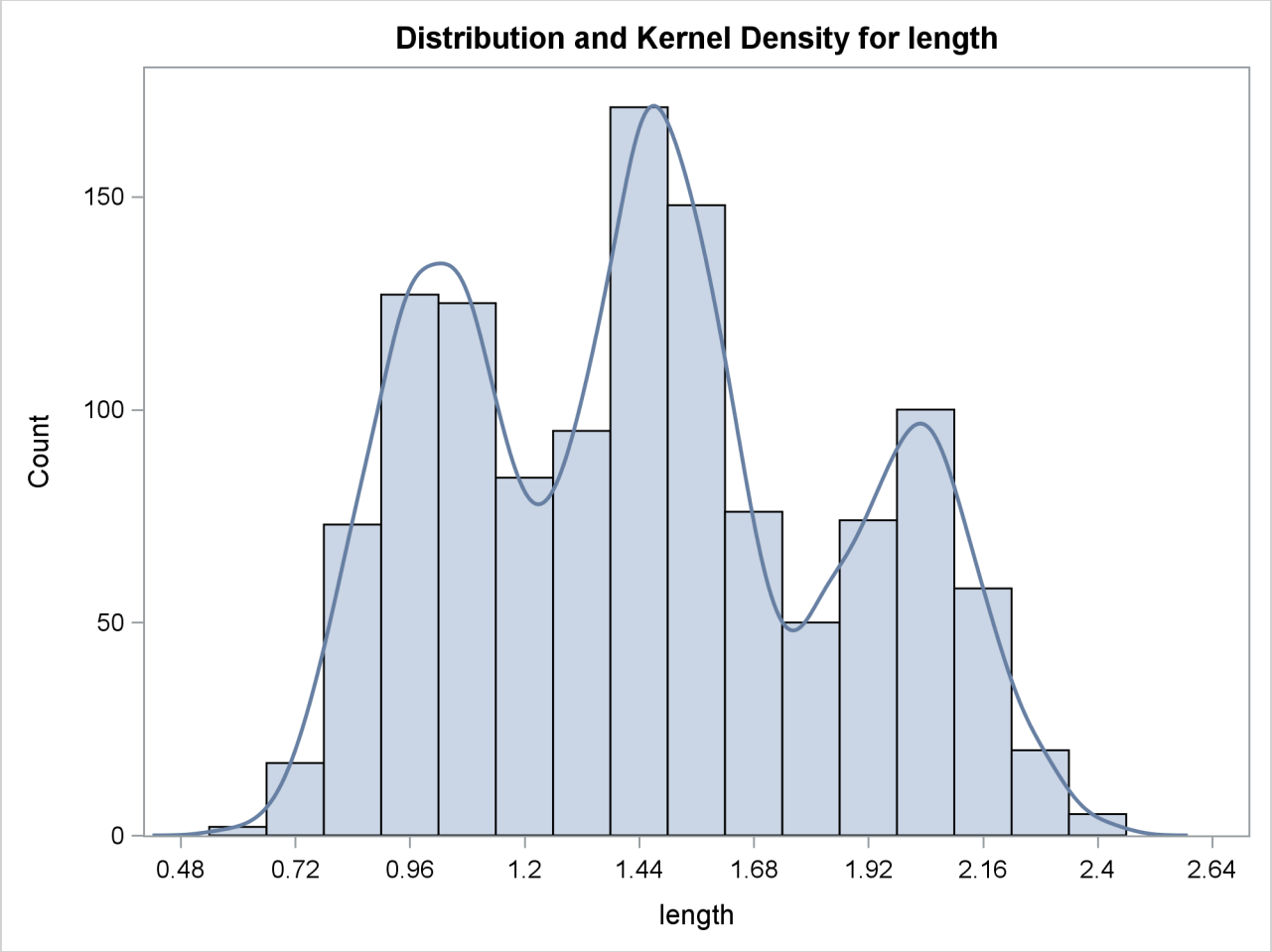
```
data channel;
   input length @@;
   datalines;
0.91 1.01 0.95 1.13 1.12 0.86 0.96 1.17 1.36 1.10
0.98 1.27 1.13 0.92 1.15 1.26 1.14 0.88 1.03 1.00
0.98 0.94 1.09 0.92 1.10 0.95 1.05 1.05 1.11 1.15
1.11 0.98 0.78 1.09 0.94 1.05 0.89 1.16 0.88 1.19

   ... 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 68.1.1. 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 5085.

**Output 68.1.1** Histogram with Overlaid Kernel Density Estimate




The default output tables for this analysis are the "Inputs" and "Controls" tables, shown in Output 68.1.2.

**Output 68.1.2** Univariate Inputs Table

**The KDE Procedure**

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

| Controls | |
|---|---|
| | **length** |
| **Grid Points** | 401 |
| **Lower Grid Limit** | 0.421 |
| **Upper Grid Limit** | 2.589 |
| **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.

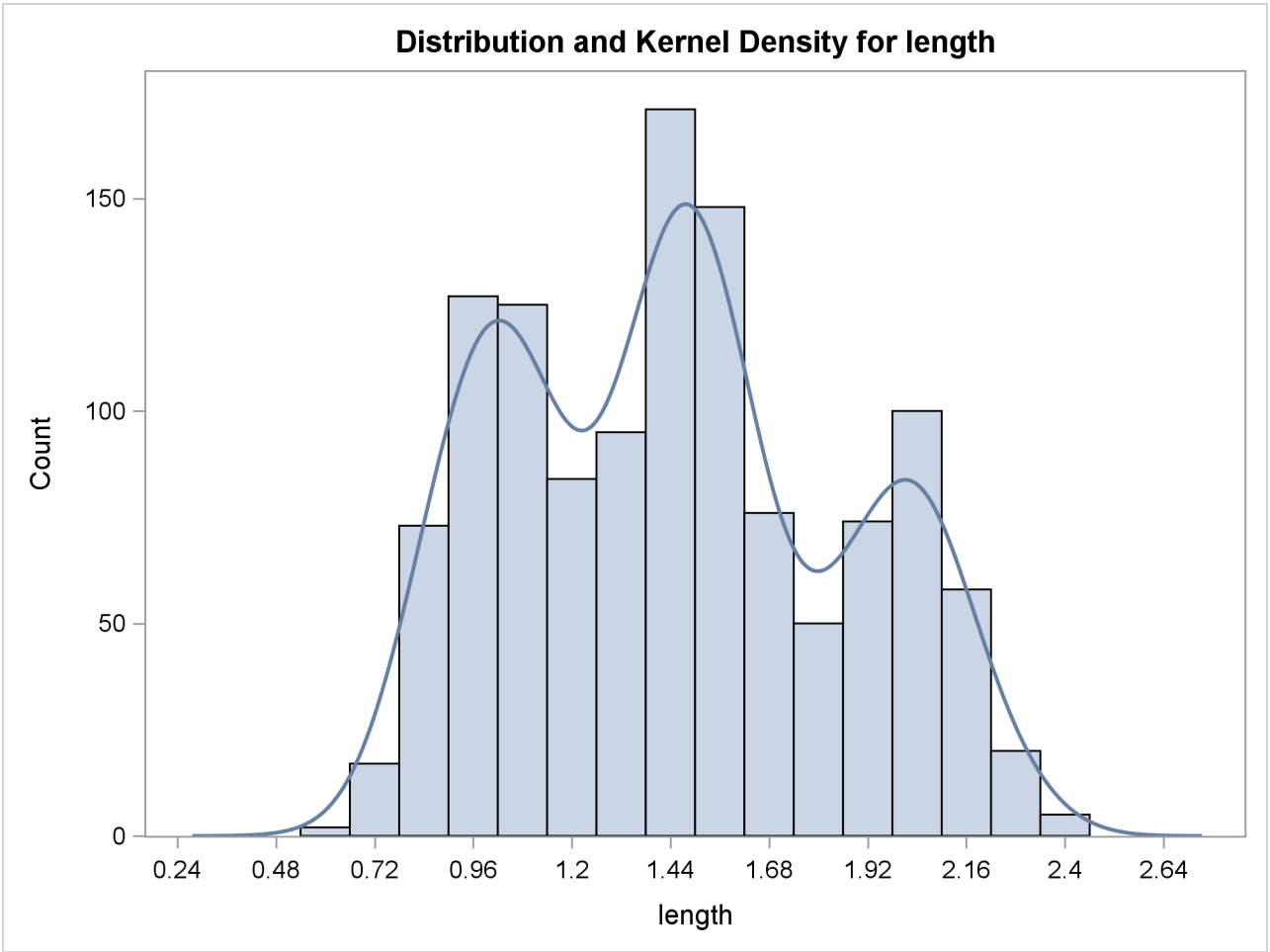
## Example 68.2: Changing the Bandwidth

Continuing with Example 68.1, 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:

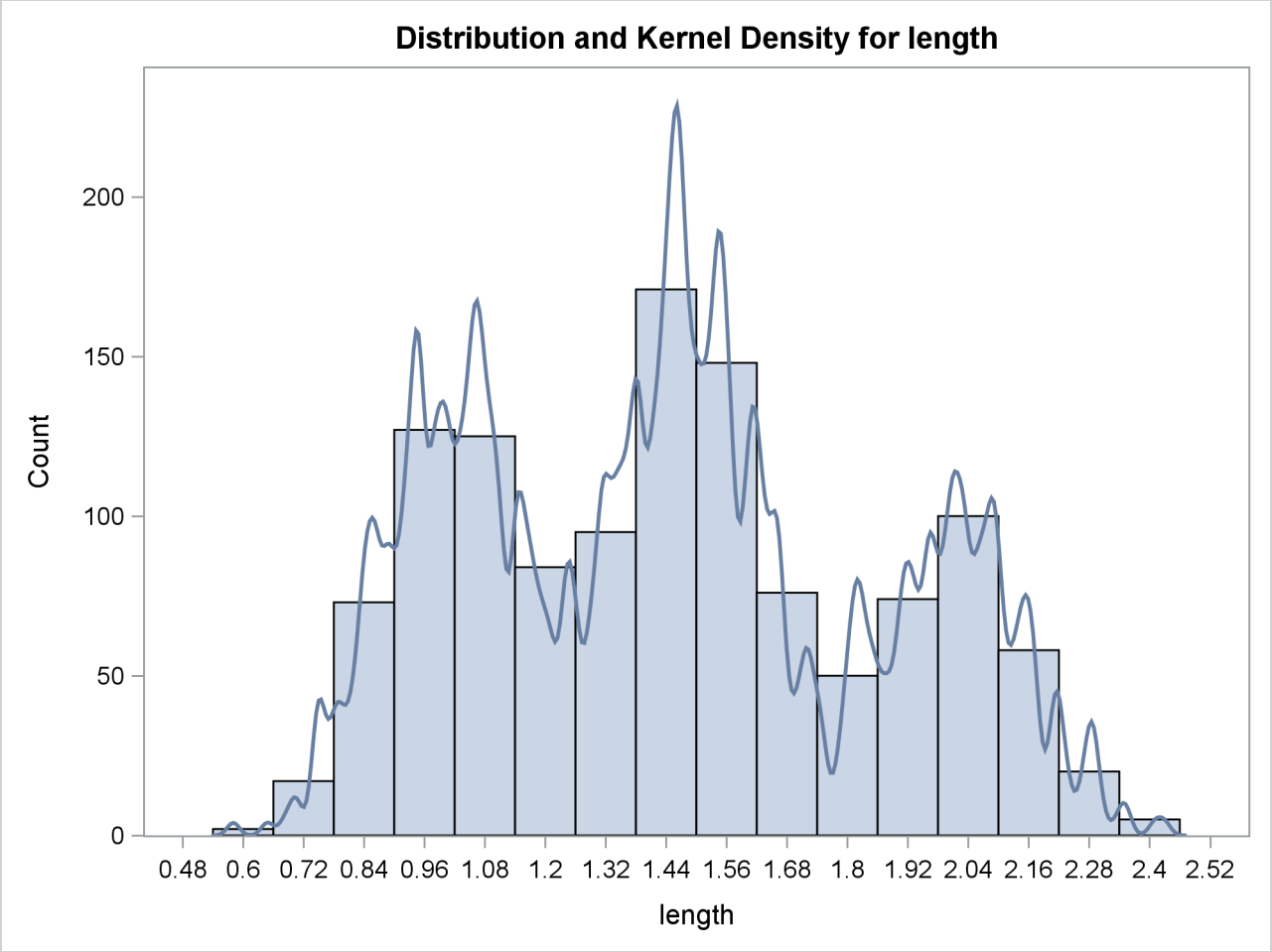
```
proc kde data=channel;
   univar length(bwm=2) length(bwm=0.25);
run;
ods graphics off;
```

Output 68.2.1 shows an oversmoothed estimate because the bandwidth multiplier is 2. Output 68.2.2 is created by specifying BWM=0.25, so it is an undersmoothed estimate.

**Output 68.2.1** Histogram with Oversmoothed Kernel Density Estimate



**Output 68.2.2** Histogram with Undersmoothed Kernel Density Estimate



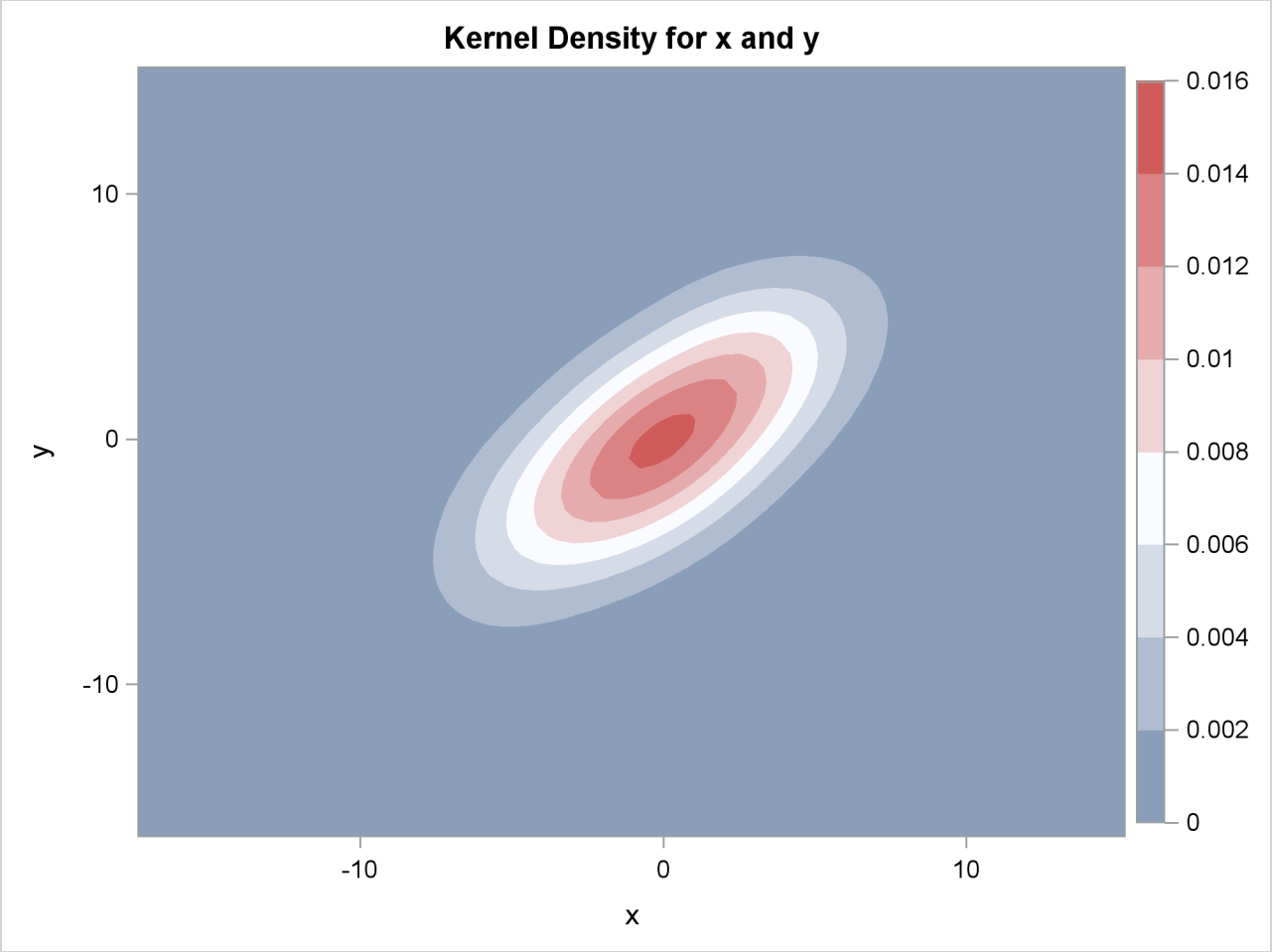

## Example 68.3: Changing the Bandwidth (Bivariate)

Recall the analysis from the section "Getting Started: KDE Procedure" on page 5060. 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 68.3.1.

**Output 68.3.1** Contour Plot of Estimated Density with Additional Smoothing



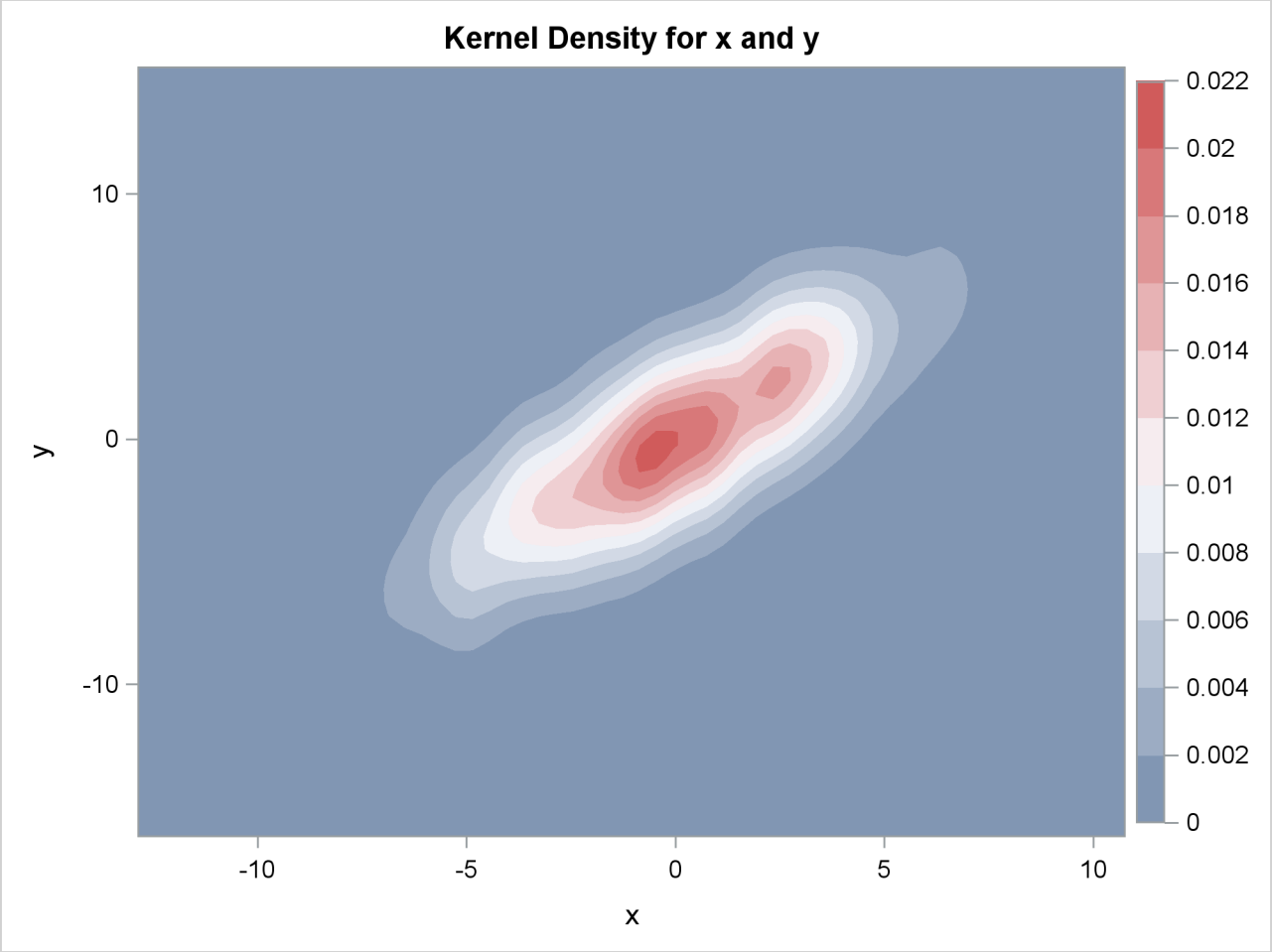

## Multiple Bandwidths

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 68.3.2.

**Output 68.3.2** Contour Plot of Estimated Density with Different Smoothing for x and y




## Example 68.4: 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 5060, 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 68.4.1.

**Output 68.4.1** 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** | -14.27 | -13.18 |
| **Upper Grid Limit** | 12.17 | 12.165 |
| **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 |

| 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 |

**Output 68.4.1** *continued*

| Levels | | | | | |
|---|---|---|---|---|---|
| Percent | Density | Lower for x | Upper for x | Lower for y | Upper for y |
| **1** | 0.001034 | -8.45 | 8.58 | -8.89 | 8.30 |
| **5** | 0.003076 | -7.10 | 7.24 | -7.17 | 7.01 |
| **10** | 0.004886 | -6.21 | 5.90 | -6.31 | 6.15 |
| **50** | 0.01586 | -3.52 | 3.65 | -3.73 | 4.00 |
| **90** | 0.02338 | -1.28 | 1.41 | -1.58 | 0.99 |
| **95** | 0.02464 | -1.28 | 0.97 | -1.15 | 0.57 |
| **99** | 0.02547 | -0.83 | 0.069 | -0.72 | 0.14 |
| **100** | 0.02564 | -0.38 | -0.38 | -0.29 | -0.29 |

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 68.4.2.

**Output 68.4.2** 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 |

**Output 68.4.2** *continued*

| | | Levels | | | |
|---|---|---|---|---|---|
| Percent | Density | Lower for x | Upper for x | Lower for y | Upper for y |
| **2.5** | 0.001990 | -7.55 | 8.14 | -8.03 | 7.44 |
| **50.0** | 0.01586 | -3.52 | 3.65 | -3.73 | 4.00 |
| **97.5** | 0.02504 | -0.83 | 0.52 | -0.72 | 0.57 |

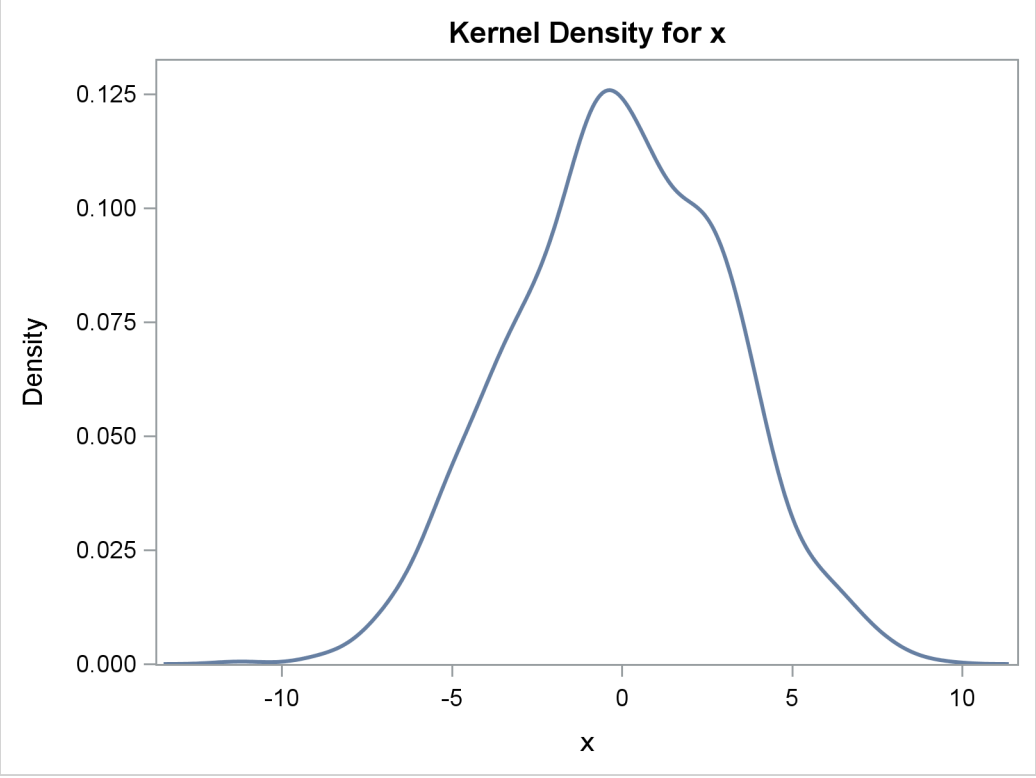
## Example 68.5: Univariate KDE Graphics

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:

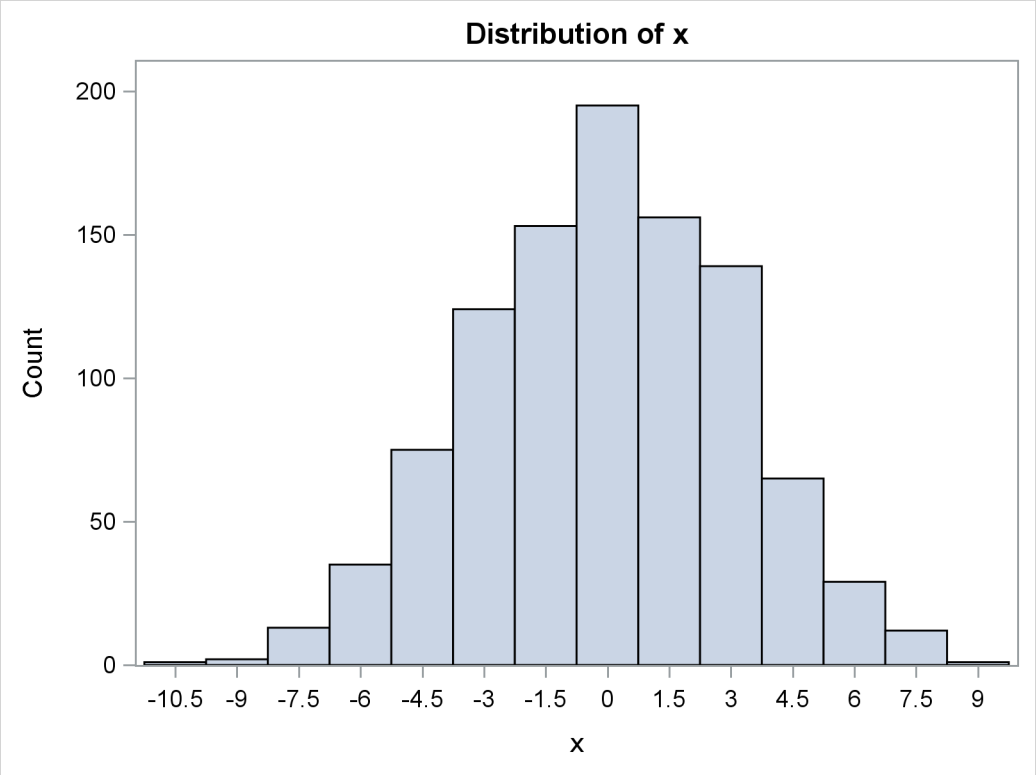
```
ods graphics on;
proc kde data=bivnormal;
   univar x / plots=(density histogram histdensity);
   univar x y / plots=densityoverlay;
run;
ods graphics off;
```

Graphs are requested by specifying the PLOTS= option in the UNIVAR statement with ODS Graphics enabled. Output 68.5.1, Output 68.5.2, and Output 68.5.3 show the kernel density estimate, histogram, and histogram with kernel density estimate overlaid, respectively, produced by the first UNIVAR statement.

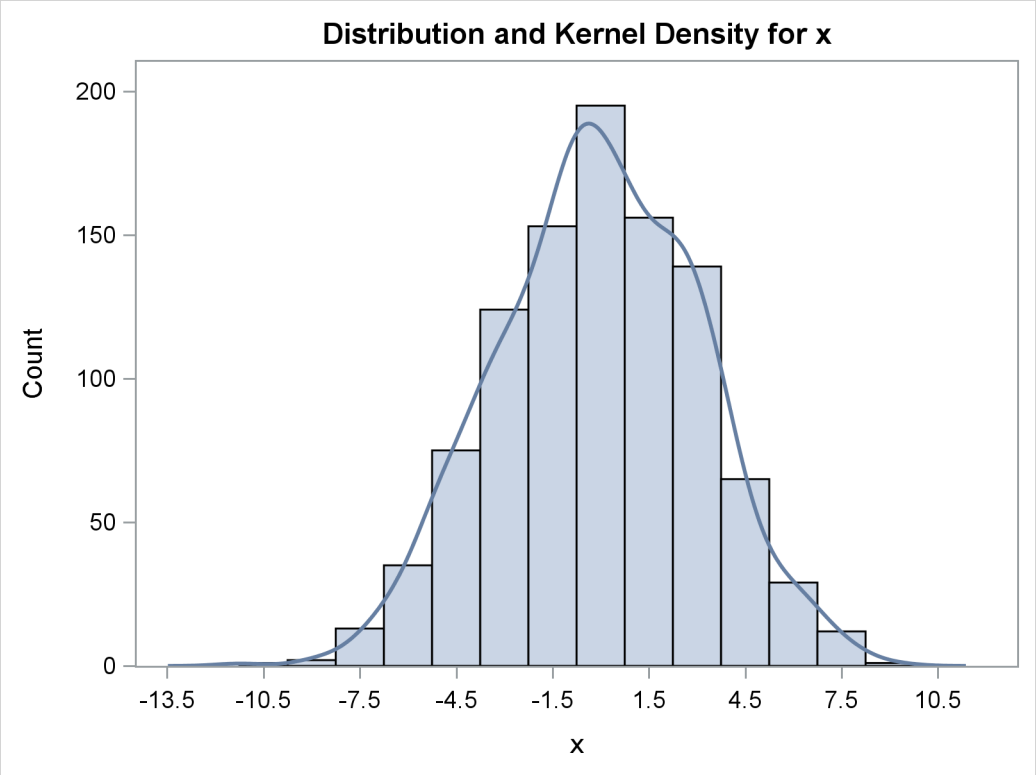
**Output 68.5.1** Kernel Density Estimate



**Output 68.5.2** Histogram

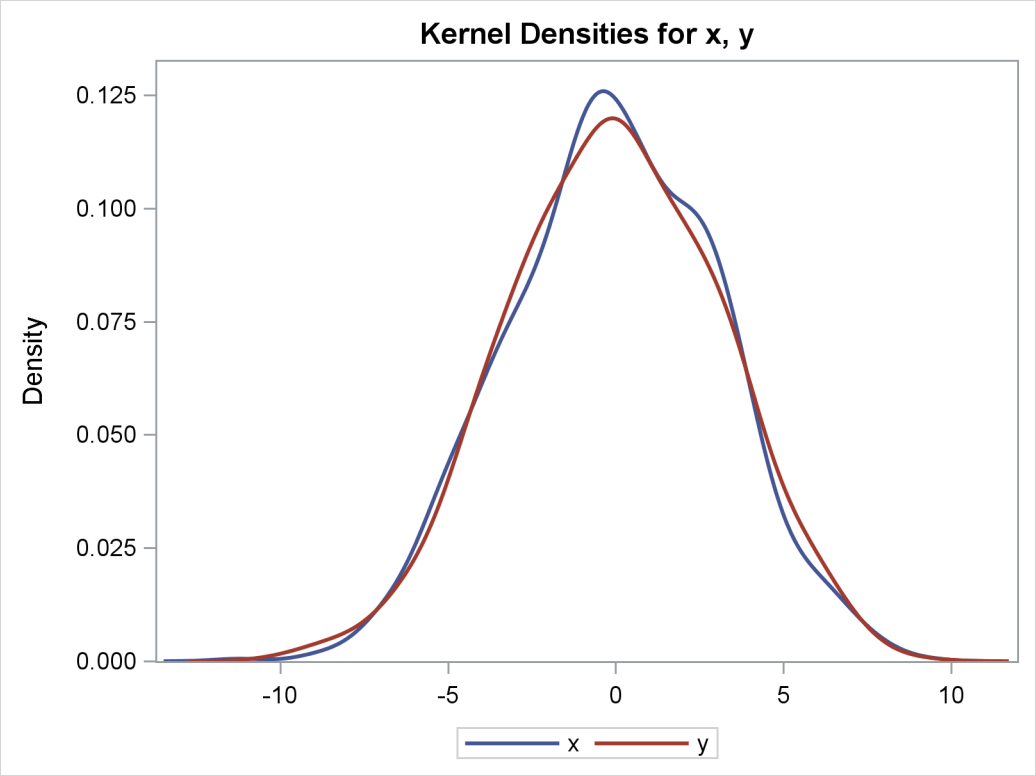



**Output 68.5.3** Histogram with Overlaid Kernel Density Estimate



Output 68.5.4 shows the plot produced by the second UNIVAR statement, in which the kernel density estimates for x and y are overlaid.

**Output 68.5.4** 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 5085.

## Example 68.6: 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:

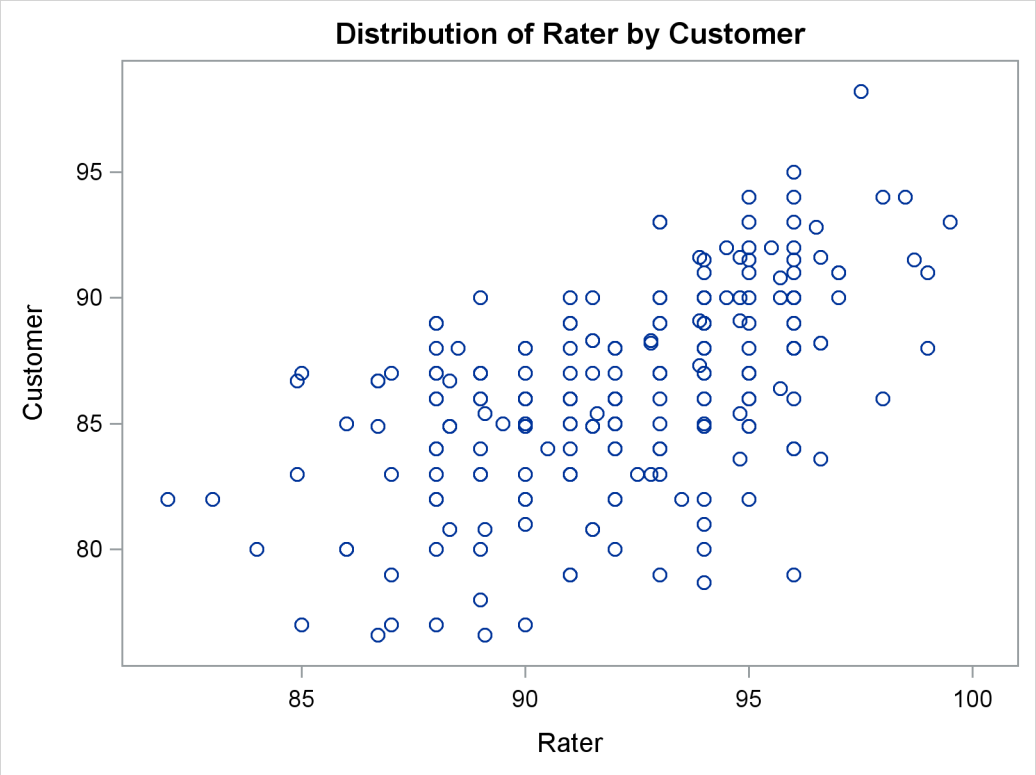
```
data octane;
   input Rater Customer;
   label Rater    = 'Rater'
         Customer = 'Customer';
   datalines;
94.5 92.0
94.0 88.0
94.0 90.0
93.0 93.0

   ... more lines ...

88.0 84.0
  .H 90.0
;
```

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

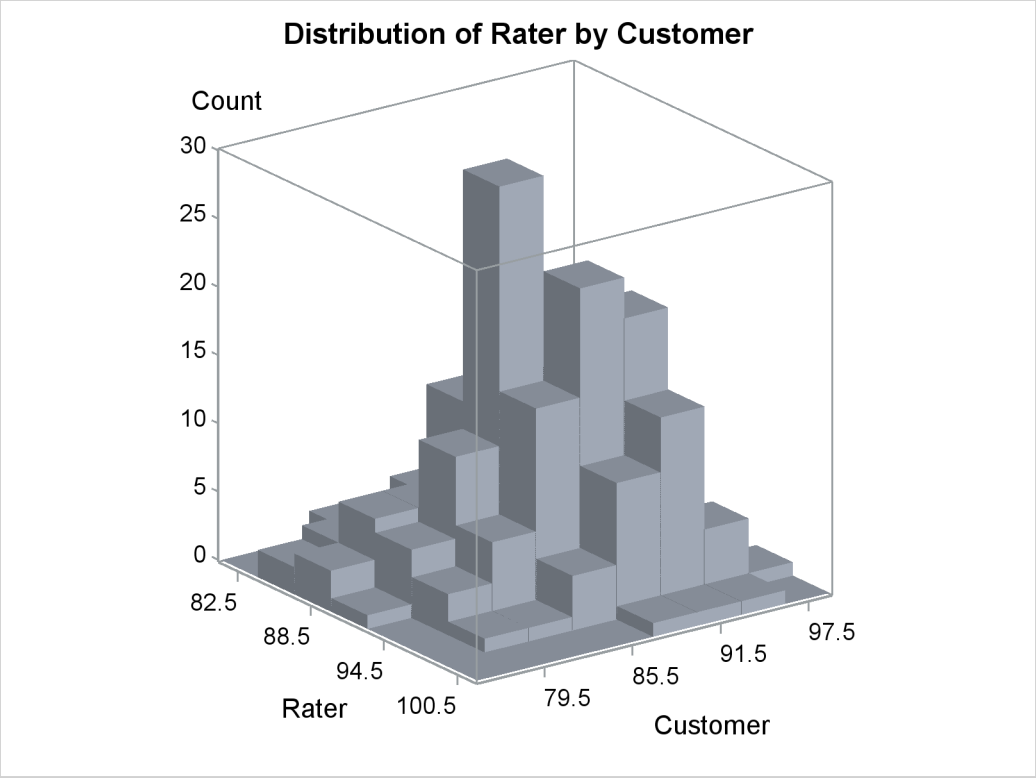
```
ods graphics on;
proc kde data=octane;
   bivar Rater Customer / plots=all;
run;
ods graphics off;
```

Output 68.6.1 shows a scatter plot of the data, Output 68.6.2 shows a bivariate histogram of the data, Output 68.6.3 shows a contour plot of bivariate density estimate, Output 68.6.4 shows a contour plot of bivariate density estimate overlaid with a scatter plot of data, Output 68.6.5 shows a surface plot of bivariate kernel density estimate, and Output 68.6.6 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 5085.

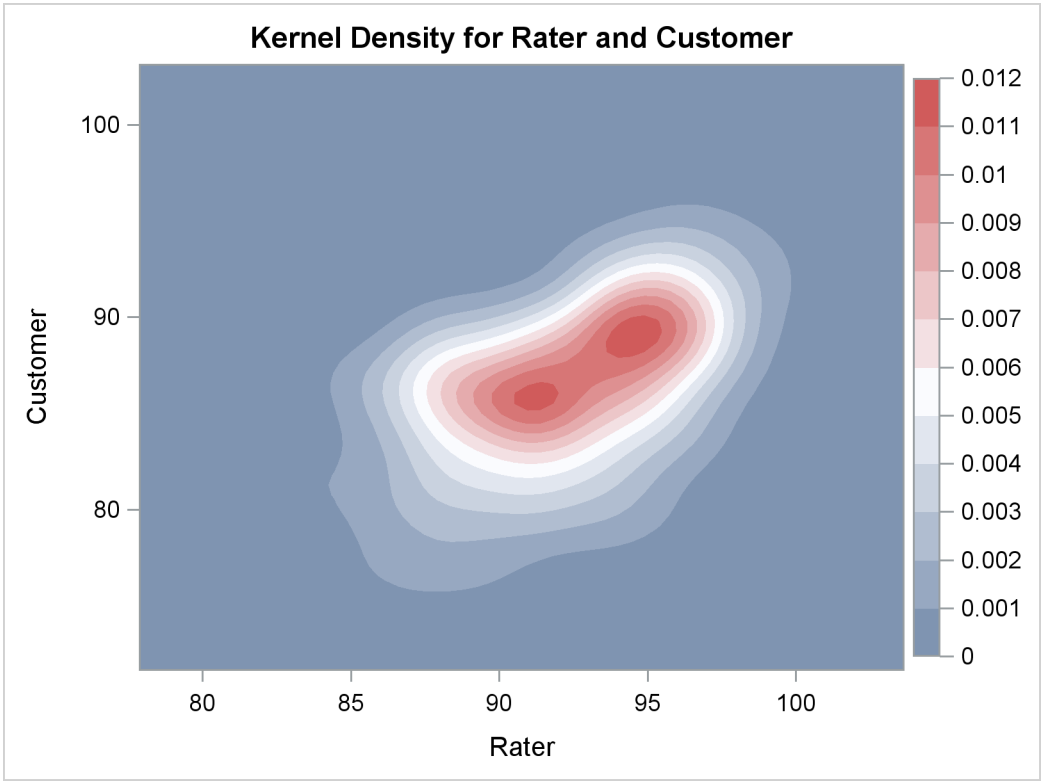
**Output 68.6.1** Scatter Plot



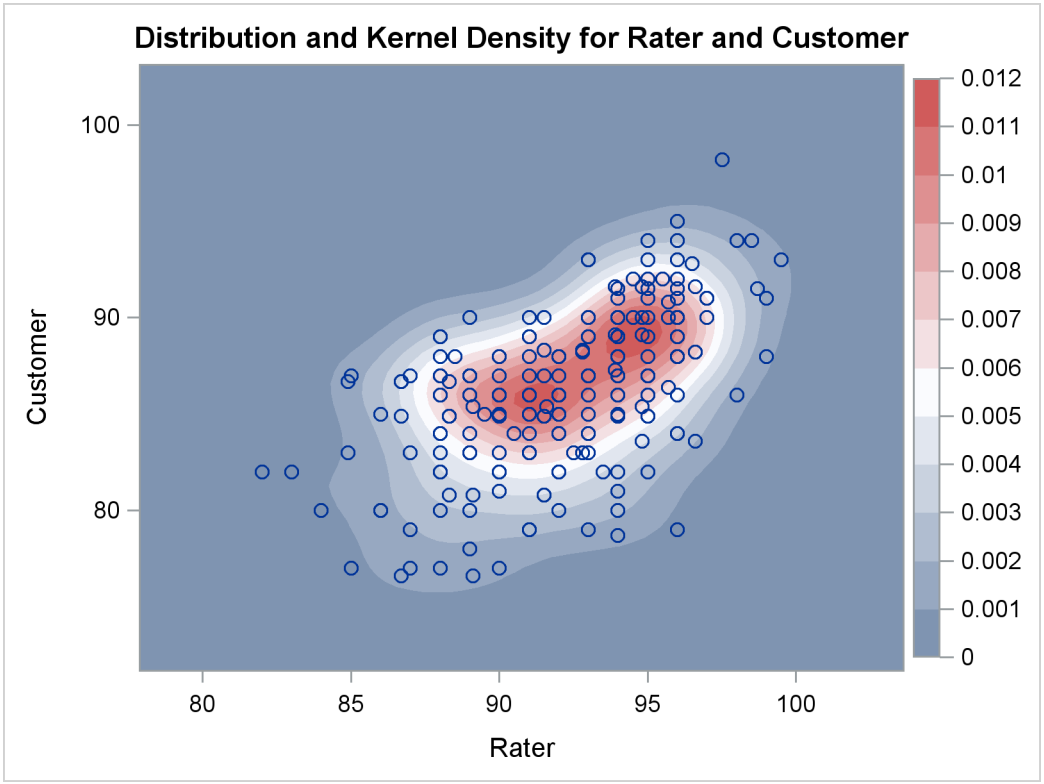

**Output 68.6.2** Bivariate Histogram



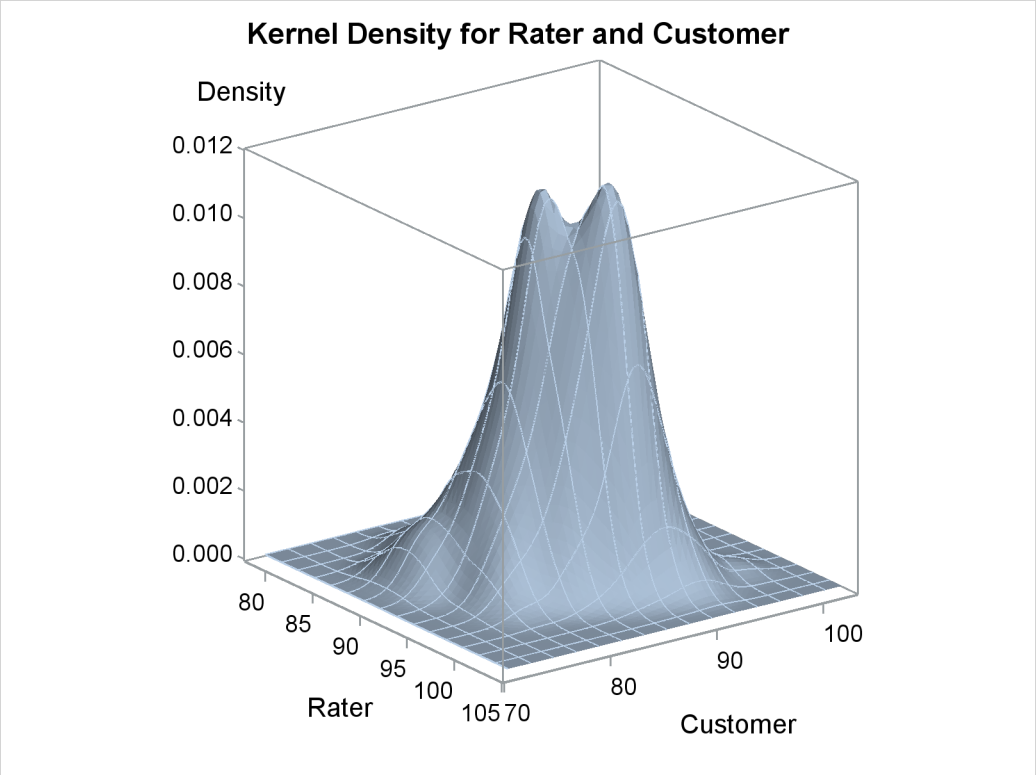
**Output 68.6.3** Contour Plot



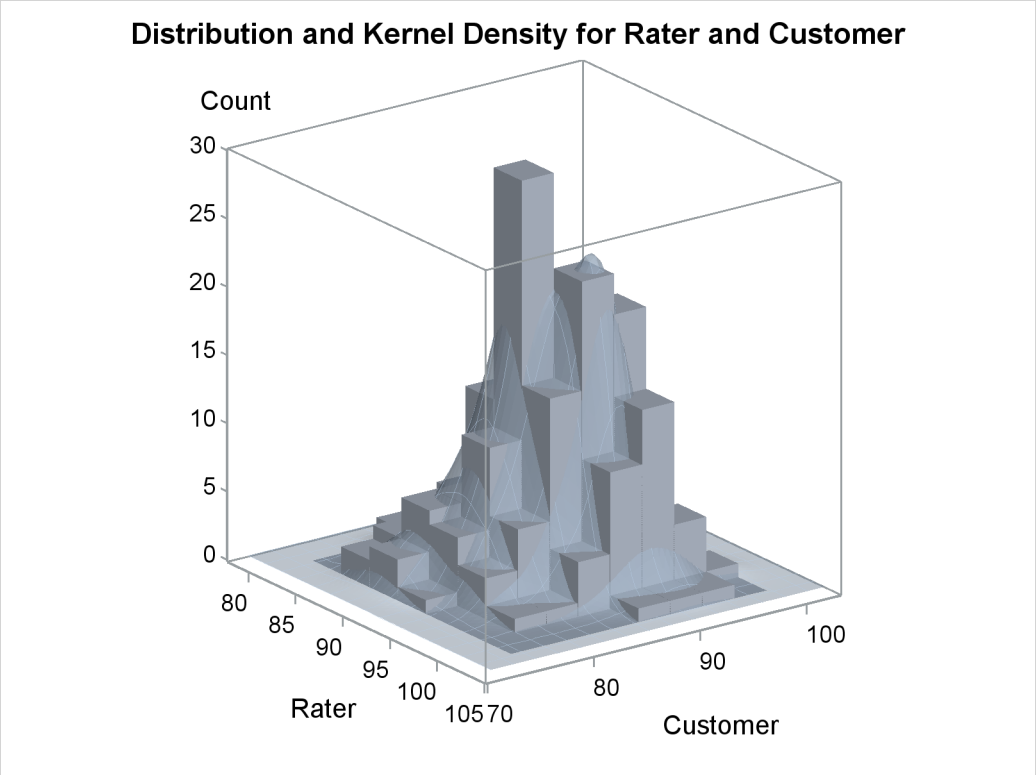

**Output 68.6.4** Contour Plot with Overlaid Scatter Plot



**Output 68.6.5** Surface Plot




**Output 68.6.6** Bivariate Histogram with Overlaid Surface Plot



# References

Azzalini, A. (1981). "A Note on the Estimation of a Distribution Function and Quantiles by a Kernel Method." *Biometrika* 68:326–328.

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.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C: The Art of Scientific Computing*. 2nd ed. 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