Chapter 33
The KDE Procedure

Chapter Table of Contents

Chapter 33
The KDE Procedure

Overview

The KDE procedure performs either univariate or 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. Refer to Silverman (1986) for a thorough review and discussion.

PROC KDE uses a Gaussian density as the kernel, and its assumed variance determines the smoothness of the resulting estimate. PROC KDE outputs the kernel density estimate into a SAS data set, which you can then use with other procedures for plotting or analysis. PROC KDE also computes a variety of common statistics, including estimates of the percentiles of the hypothesized probability density function.

Getting Started

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 code to accomplish this is as follows:

```
data k;
   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 PROC KDE code computes a bivariate kernel density estimate of these data:

```
proc kde data=k out=o;
   var x y;
run;
```

The output from this analysis is as follows.

The KDE Procedure

Inputs

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

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 KDE Procedure

Controls

Description	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 "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.

The KDE Procedure

Statistics

Description	x	y
Mean	-0.075	-0.070
Variance	9.72	9.92
Standard Deviation	3.12	3.15
Range	20.39	19.09
Interquartile Range	4.46	4.51
Bandwidth	0.99	1.00

The "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 KDE Procedure

Bivariate Statistics

Description	Value
Covariance	8.88
Correlation	0.90

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 KDE Procedure

Percentiles

Percent	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

The "Percentiles" table lists percentiles for each variable.

The KDE Procedure

Levels

Percent	Density	Lower1	Lower2	Upper1	Upper2
1	0.001181	-8.14	-8.76	8.45	8.39
5	0.003028	-7.10	-7.14	7.07	6.77
10	0.004988	-6.41	-6.49	5.69	6.12
50	0.01592	-3.64	-3.58	3.96	3.86
90	0.02389	-1.22	-1.32	1.19	0.95
95	0.02525	-0.88	-0.99	0.50	0.62
99	0.02609	-0.53	-0.67	0.16	0.30
100	0.02630	-0.19	-0.35	-0.19	-0.35

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 percent 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 Lower1 and Lower2 columns), and the maximum values are 7.07 and 6.77, respectively (the Upper1 and Upper2 variables).

The output data set O from this analysis contains 3600 points containing the kernel density estimate. You can generate surface and contour plots of this estimate using SAS/GRAPH as follows:

```
proc g3d data=o;
   plot y*x=density;
run;

proc gcontour data=o;
   plot y*x=density;
run;
```

Figures 33.1 and 33.2 display these plots. Note that the correlation of 0.9 in the original data results in oval-shaped contours.

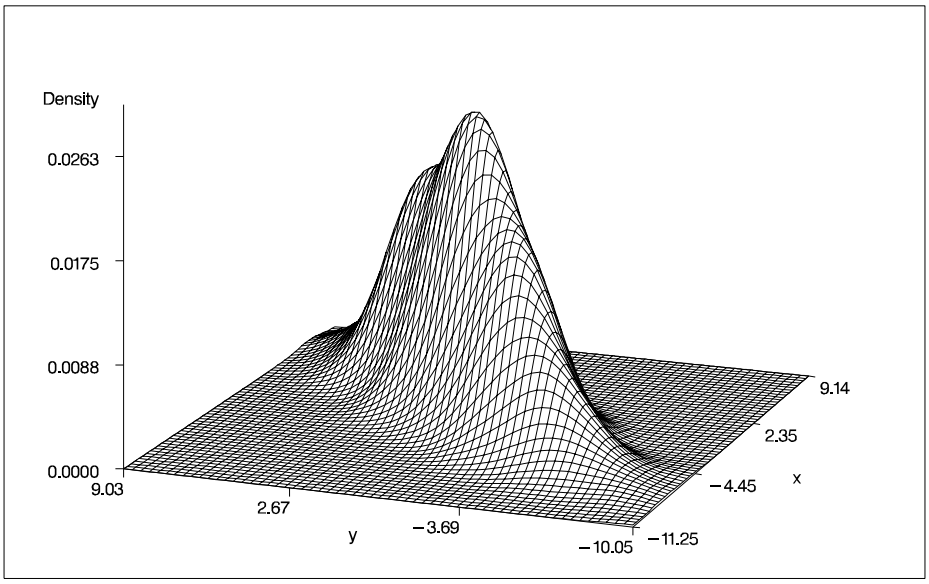

Figure 33.1. Surface plot of the bivariate kernel density estimate

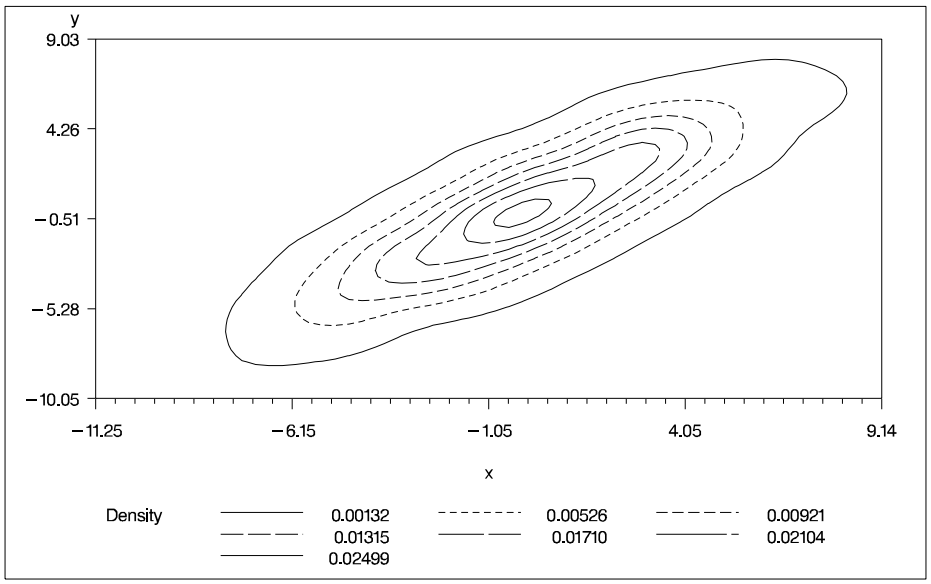

Figure 33.2. Contour plot of the bivariate kernel density estimate

Suppose, after viewing Figures 33.1 and 33.2, that you would like a slightly smoother estimate. You could then rerun the analysis with a larger bandwidth:

```
proc kde data=k out=o1 bwm=2,2;
   var x y;
run;
```

The BWM=2,2 option requests bandwidth multipliers of 2 for both X and Y. The results of this fit and a subsequent call to PROC G3D produces Figure 33.3. Note that the small flattish area behind the main mode in Figure 33.1 has disappeared in Figure 33.3.

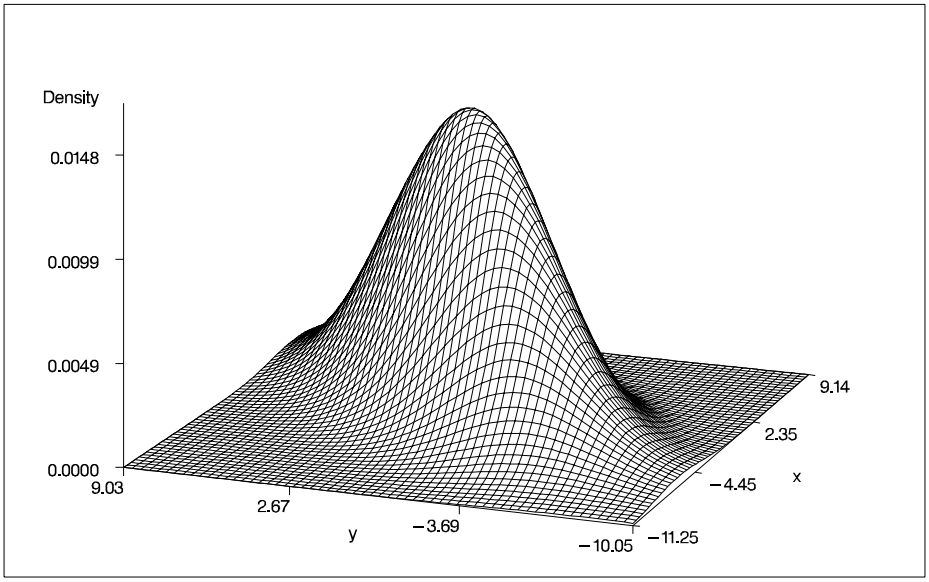

Figure 33.3. Surface plot of the bivariate kernel density estimate with additional smoothing

You can also use the results from the Levels table to plot specific contours corresponding to percentiles of the data. For example, the Levels table from the PROC KDE output using BWM=2,2 is as follows:

The KDE Procedure

Levels

Percent	Density	Lower1	Lower2	Upper1	Upper2
1	0.001238	-8.48	-8.76	8.45	8.39
5	0.003008	-7.10	-7.14	6.72	6.77
10	0.004625	-6.06	-5.85	6.03	6.12
50	0.01085	-3.30	-3.26	3.27	3.21
90	0.01430	-1.22	-1.32	1.19	0.95
95	0.01459	-0.88	-0.99	0.85	0.62
99	0.01478	-0.53	-0.67	0.50	0.30
100	0.01481	-0.19	-0.024	-0.19	-0.024

You can use the values from the Density column of this table with PROC GCONTOUR to plot the 1, 5, 10, 50, 90, 95, and 99 percent levels of the density:

```
proc gcontour data=o1;
   plot y*x=density / levels=0.0012 0.0030 0.0046 0.0109
      0.0143 0.0146 0.0148;
run;
```

This plot is displayed in Figure 33.4.

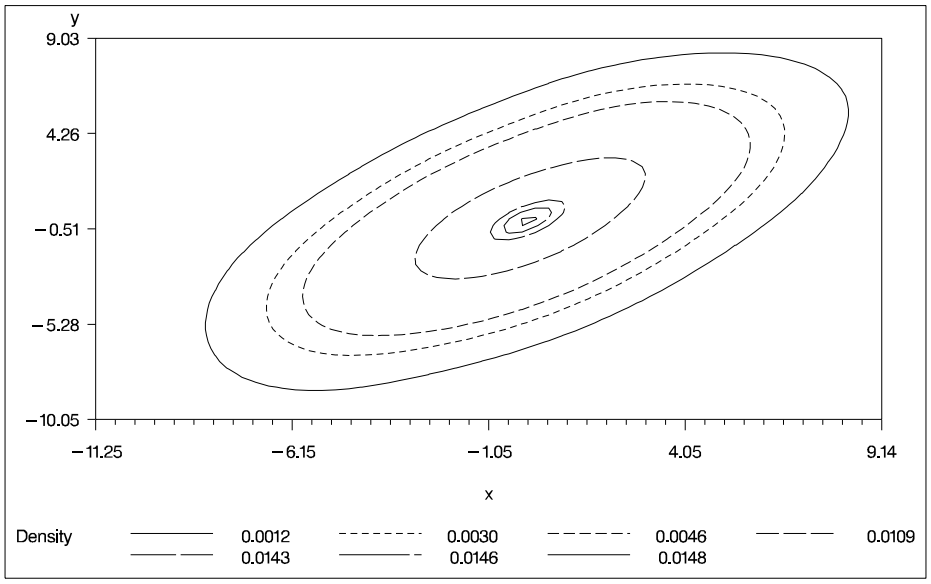

Figure 33.4. Contour plot of the bivariate kernel density estimate with additional smoothing and levels corresponding to percentiles

The next-to-outermost contour of Figure 33.4 represents an approximate 95 percent ellipsoid for X and Y.

Syntax

You can use following statements with the KDE procedure.

PROC KDE < *options* > **;**
BY *variables* **;**
FREQ *variable* **;**
VAR *variables* **;**
WEIGHT *variable* **;**

PROC KDE Statement

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

The PROC KDE statement invokes the procedure. You can specify the following options in the PROC KDE statement.

BWM=*numlist*
specifies the bandwidth multipliers for the kernel density estimate. You should specify one number for univariate smoothing and two numbers separated by a comma

for bivariate smoothing. The default values equal 1. Larger multipliers produce a smoother estimate, and smaller ones produce a rougher estimate.

GRIDL=*numlist*
specifies the lower grid limits for the kernel density estimate. You should specify one number for univariate smoothing and two numbers separated by a comma for bivariate smoothing. The default values equal the minimum observed values of the variables.

GRIDU=*numlist*
specifies the upper grid limits for the kernel density estimate. You should specify one number for univariate smoothing and two numbers separated by a comma for bivariate smoothing. The default values equal the maximum observed values of the variables.

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.

LEVELS=*numlist*
lists percentages of data for which density contours are to be computed. The default levels are 1, 5, 10, 50, 90, 95, 99, and 100.

METHOD=SJPI
METHOD=SNR
METHOD=SROT
METHOD=OS
specifies the method used to compute the bandwidth. Available methods are Sheather-Jones plug in (SJPI), simple normal reference (SNR), Silverman's rule of thumb (SROT), and oversmoothed (OS). Refer to Jones, Marron, and Sheather (1996) for a description of each of these methods. SJPI is the default for univariate smoothing, and SNR is the default and only available method for bivariate smoothing.

NGRID=*numlist*
NG=*numlist*
specifies the number of grid points associated with the variables in the VAR statement. You should specify one number for univariate smoothing and two numbers separated by a comma for bivariate smoothing. The default values are 401 when there is a single VAR variable and 60 when there are two VAR variables.

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

- variables you specify in the VAR statement, with values corresponding to grid coordinates
- 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=*numlist*
lists percentiles to be computed for each VAR variable. The default percentiles are 0.5, 1, 2.5, 5, 10, 25, 50, 75, 90, 95, 97.5, 99, and 99.5.

SJPIMAX=*number*
specifies the maximum grid value in determining the Sheather-Jones plug in bandwidth. The default value is 2 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 $1\mathrm{E}-3$.

BY Statement

BY variables ;

You can specify a BY statement with PROC KDE to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

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

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING 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 using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *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.

VAR Statement

VAR variables ;

The VAR statement lists the variables in the input data set for which a kernel density estimate is to be computed. You should specify either one or two variables. For one variable a univariate kernel density estimate is computed, and for two variables a bivariate density estimate is computed. If any VAR variable has a missing value for a particular observation, then the entire observation is omitted from the analysis.

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, as they can produce unreliable results.

Details

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

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

This optimal value is unknown, and so approximations methods are required. For a derivation and discussion of these results, refer to 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 $\Re^2$ with joint density function $f(x, y), (x, y) \in \Re^2$, 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

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

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

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

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 of $\hat{f}(x,y)$ is

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

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

Now set

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

which is the asymptotic mean integrated squared error. For fixed n, this has minimum at $(h_{AMISE\_X}, h_{AMISE\_Y})$ defined as

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

and

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

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. Refer to 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 practically 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 this new data. This is so-called "simple binning," as versus the finer "linear binning" described in Wand (1993). 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.

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

where $\mathbf{x}_{i,j} = (x_i, y_i), i = 1, 2, \dots, g_X, j = 1, 2, \dots, g_Y$, 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:

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

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

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

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

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

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

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

Modify K as follows:

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

Essentially, the negligible terms of 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 fastest Fast Fourier Transform operation (refer to Wand 1993).

The third operation is to pad the bin count matrix C with zeros to the same size as K:

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

The convolution $K * C$ 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 K is defined similarly as

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

where $l_X = \min(g_X - 1, floor(5h_X/\delta_X))$, $p_X = 2^{ceil(\log_2(g_X + l_X + 1))}$, 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

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

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

Most of the results in this subsection are found in Wand (1993).

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, refer to Press et al. (1988).

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

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

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

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

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 (refer to 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

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

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

$$h = \hat{\sigma}[4/(3n)]^{1/5}$$

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

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

$$h = 0.9 \min[\hat{\sigma}, (Q_3 - Q_1)/1.34]n^{-1/5}$$

where Q_3 and Q_1 are the third and first sample quartiles, respectively.

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

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

where $\overline{X} = (\sum_{i=1}^n W_i X_i)/(\sum_{i=1}^n W_i)$ 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 (1992):

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

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 the following table. For more information on ODS, see Chapter 15, "Using the Output Delivery System."

Table 33.1. ODS Tables Produced in PROC KDE

ODS Table Name	Description	Statement
BivariateStatistics	Bivariate statistics	default for two variables
Controls	Control variables	default
Inputs	Input information	default
Levels	Levels of density estimate	default
Percentiles	Percentiles of data	default
Statistics	Basic statistics	default

References

Bowman, A. and Foster, P. (1992), "Density Based Exploration of Bivariate Data," Department of Statistics, University of Glasgow, Technical Report No. 92-1.

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.

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

Wand, M.P. (1993), "Fast Computation of Multivariate Kernel Estimators," University of New South Wales, Australian Graduate School of Management, Working Paper Series 93-007.

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.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/STAT® User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/STAT® User's Guide, Version 8

ISBN 1–58025–494–2

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

1st printing, October 1999