Title

stata.com

kdensity — Univariate kernel density estimation

Description

`kdensity` produces kernel density estimates and graphs the result.

Quick start

Graph of the kernel density estimate for `v1`

```
kdensity v1
```

Add a normal curve

```
kdensity v1, normal
```

With a kernel bandwidth of 2

```
kdensity v1, bwidth(2)
```

Gaussian kernel function for `v1`

```
kdensity v1, kernel(gaussian)
```

Kernel density estimate for `v1` and `v2` in the same graph area

```
twoway kdensity v1 || kdensity v2
```

Separate graphs of kernel density estimate of `v1` for each level of `catvar`

```
twoway kdensity v1, by(catvar)
```

Kernel density estimates of `v1` for `catvar` = 0 and 1 in the same graph area

```
twoway kdensity v1 if catvar==0 || kdensity v1 if catvar==1
```

Menu

Statistics > Nonparametric analysis > Kernel density estimation

Syntax

```
kdensity varname [if] [in] [weight] [, options]
```

options	Description
Main	
`kernel(`*kernel*`)`	specify kernel function; default is `kernel(epanechnikov)`
`bwidth(#)`	half-width of kernel
`generate(`*newvar*$_x$ *newvar*$_d$`)`	store the estimation points in *newvar*$_x$ and the density estimate in *newvar*$_d$
`n(#)`	estimate density using # points; default is min(N, 50)
`at(`*var*$_x$`)`	estimate density using the values specified by *var*$_x$
`nograph`	suppress graph
Kernel plot	
cline_options	affect rendition of the plotted kernel density estimate
Density plots	
`normal`	add normal density to the graph
`normopts(`*cline_options*`)`	affect rendition of normal density
`student(#)`	add Student's t density with # degrees of freedom to the graph
`stopts(`*cline_options*`)`	affect rendition of the Student's t density
Add plots	
`addplot(`*plot*`)`	add other plots to the generated graph
Y axis, X axis, Titles, Legend, Overall	
twoway_options	any options other than `by()` documented in [G-3] ***twoway_options***

kernel	Description
`epanechnikov`	Epanechnikov kernel function; the default
`epan2`	alternative Epanechnikov kernel function
`biweight`	biweight kernel function
`cosine`	cosine trace kernel function
`gaussian`	Gaussian kernel function
`parzen`	Parzen kernel function
`rectangle`	rectangle kernel function
`triangle`	triangle kernel function

`collect` is allowed; see [U] **11.1.10 Prefix commands**.

`fweights`, `aweights`, and `iweights` are allowed; see [U] **11.1.6 weight**.

Options

Main

`kernel(`*kernel*`)` specifies the kernel function for use in calculating the kernel density estimate. The default kernel is the Epanechnikov kernel (`epanechnikov`).

`bwidth(#)` specifies the half-width of the kernel, the width of the density window around each point. If `bwidth()` is not specified, the "optimal" width is calculated and used. The optimal width is the width that would minimize the mean integrated squared error if the data were Gaussian and a Gaussian kernel were used, so it is not optimal in any global sense. In fact, for multimodal and highly skewed densities, this width is usually too wide and oversmooths the density (Silverman 1986).

`generate(`*newvar*$_x$ *newvar*$_d$`)` stores the results of the estimation. *newvar*$_x$ will contain the points at which the density is estimated. *newvar*$_d$ will contain the density estimate.

`n(#)` specifies the number of points at which the density estimate is to be evaluated. The default is $\min(N, 50)$, where N is the number of observations in memory.

`at(`*var*$_x$`)` specifies a variable that contains the values at which the density should be estimated. This option allows you to more easily obtain density estimates for different variables or different subsamples of a variable and then overlay the estimated densities for comparison.

`nograph` suppresses the graph. This option is often used with the `generate()` option.

Kernel plot

cline_options affect the rendition of the plotted kernel density estimate. See [G-3] ***cline_options***.

Density plots

`normal` requests that a normal density be overlaid on the density estimate for comparison.

`normopts(`*cline_options*`)` specifies details about the rendition of the normal curve, such as the color and style of line used. See [G-3] ***cline_options***.

`student(#)` specifies that a Student's t density with # degrees of freedom be overlaid on the density estimate for comparison.

`stopts(`*cline_options*`)` affects the rendition of the Student's t density. See [G-3] ***cline_options***.

Add plots

`addplot(`*plot*`)` provides a way to add other plots to the generated graph. See [G-3] ***addplot_option***.

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] ***twoway_options***, excluding `by()`. These include options for titling the graph (see [G-3] ***title_options***) and for saving the graph to disk (see [G-3] ***saving_option***).

Remarks and examples

stata.com

Kernel density estimators approximate the density $f(x)$ from observations on x. Histograms do this, too, and the histogram itself is a kind of kernel density estimate. The data are divided into nonoverlapping intervals, and counts are made of the number of data points within each interval. Histograms are bar graphs that depict these frequency counts—the bar is centered at the midpoint of each interval—and its height reflects the average number of data points in the interval.

In more general kernel density estimates, the range is still divided into intervals, and estimates of the density at the center of intervals are produced. One difference is that the intervals are allowed to overlap. We can think of sliding the interval—called a window—along the range of the data and collecting the center-point density estimates. The second difference is that, rather than merely counting the number of observations in a window, a kernel density estimator assigns a weight between

0 and 1—based on the distance from the center of the window—and sums the weighted values. The function that determines these weights is called the kernel.

Kernel density estimates have the advantages of being smooth and of being independent of the choice of origin (corresponding to the location of the bins in a histogram).

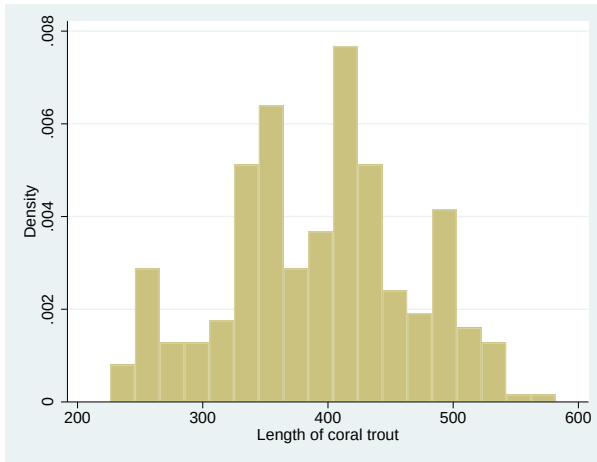
See Salgado-Ugarte, Shimizu, and Taniuchi (1993) and Fox (1990) for discussions of kernel density estimators that stress their use as exploratory data-analysis tools.

Cox (2007) gives a lucid introductory tutorial on kernel density estimation with several Stata produced examples. He provides tips and tricks for working with skewed or bounded distributions and applying the same techniques to estimate the intensity function of a point process.

▷ Example 1: Histogram and kernel density estimate

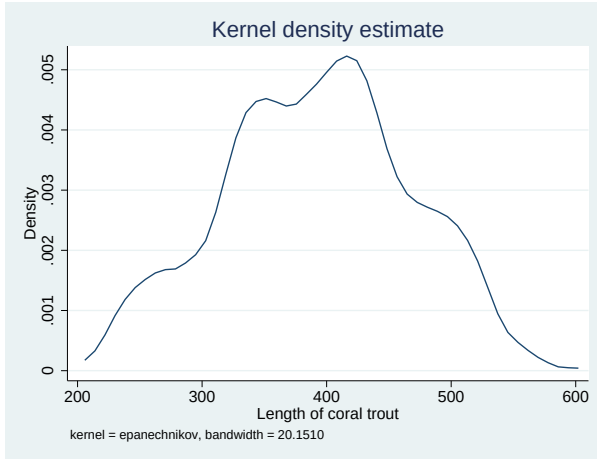
Goeden (1978) reports data consisting of 316 length observations of coral trout. We wish to investigate the underlying density of the lengths. To begin on familiar ground, we might draw a histogram. In [R] **histogram**, we suggest setting the bins to $\min(\sqrt{n}, 10 \cdot \log_{10} n)$, which for $n = 316$ is roughly 18:

```
. use https://www.stata-press.com/data/r17/trocolen
. histogram length, bin(18)
(bin=18, start=226, width=19.777778)
```



The kernel density estimate, on the other hand, is smooth.

```
. kdensity length
```



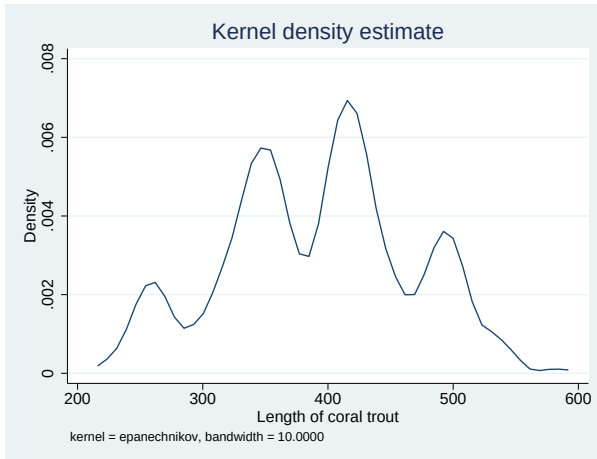
Kernel density estimators are, however, sensitive to an assumption, just as are histograms. In histograms, we specify a number of bins. For kernel density estimators, we specify a width. In the graph above, we used the default width. `kdensity` is smarter than `twoway histogram` in that its default width is not a fixed constant. Even so, the default width is not necessarily best.

`kdensity` stores the width in the returned scalar `bwidth`, so typing `display r(bwidth)` reveals it. Doing this, we discover that the width is approximately 20.

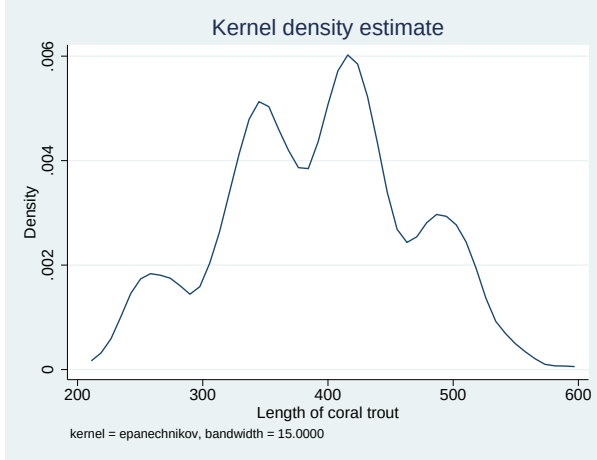
Widths are similar to the inverse of the number of bins in a histogram in that smaller widths provide more detail. The units of the width are the units of x, the variable being analyzed. The width is specified as a half-width, meaning that the kernel density estimator with half-width 20 corresponds to sliding a window of size 40 across the data.

We can specify half-widths for ourselves by using the `bwidth()` option. Smaller widths do not smooth the density as much:

```
. kdensity length, bwidth(10)
```



```
. kdensity length, bwidth(15)
```

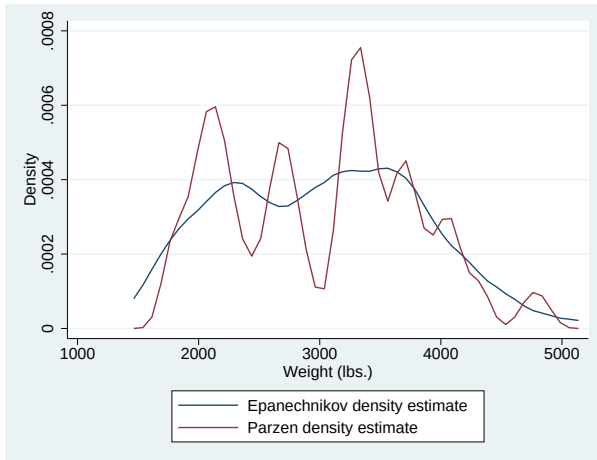


Example 2: Different kernels can produce different results

When widths are held constant, different kernels can produce surprisingly different results. This is really an attribute of the kernel and width combination; for a given width, some kernels are more sensitive than others at identifying peaks in the density estimate.

We can see this when using a dataset with lots of peaks. In the automobile dataset, we characterize the density of `weight`, the weight of the vehicles. Below, we compare the Epanechnikov and Parzen kernels.

```
. use https://www.stata-press.com/data/r17/auto
(1978 automobile data)
. kdensity weight, kernel(epanechnikov) nograph generate(x epan)
. kdensity weight, kernel(parzen) nograph generate(x2 parzen)
. label var epan "Epanechnikov density estimate"
. label var parzen "Parzen density estimate"
. line epan parzen x, sort ytitle(Density) legend(cols(1))
```

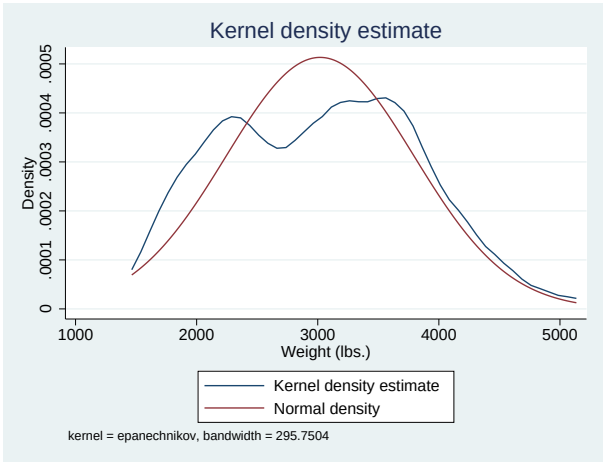


We did not specify a width, so we obtained the default width. That width is not a function of the selected kernel, but of the data. See *Methods and formulas* for the calculation of the optimal width. ◁

▷ Example 3: Density with overlaid normal density

In examining the density estimates, we may wish to overlay a normal density or a Student's t density for comparison. Using automobile weights, we can get an idea of the distance from normality by using the `normal` option.

```
. kdensity weight, kernel(epanechnikov) normal
```



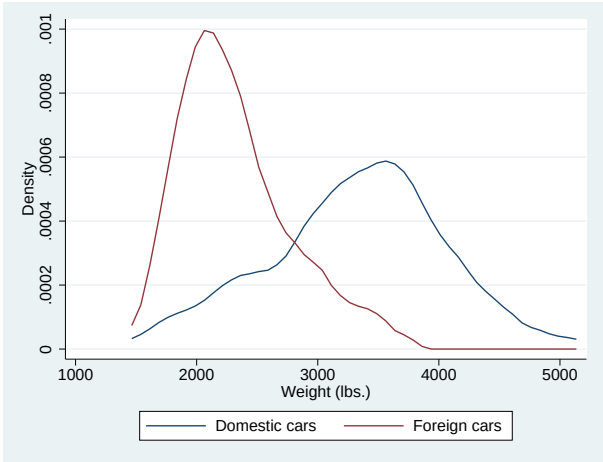
◁

▷ Example 4: Compare two densities

We also may want to compare two or more densities. In this example, we will compare the density estimates of the weights for the foreign and domestic cars.

```
. use https://www.stata-press.com/data/r17/auto, clear
(1978 automobile data)
. kdensity weight, nograph generate(x fx)
. kdensity weight if foreign==0, nograph generate(fx0) at(x)
. kdensity weight if foreign==1, nograph generate(fx1) at(x)
. label var fx0 "Domestic cars"
. label var fx1 "Foreign cars"
```

```
. line fx0 fx1 x, sort ytitle(Density)
```



◁

❑ Technical note

Although all the examples we included had densities of less than 1, the density may exceed 1.

The probability density $f(x)$ of a continuous variable, x, has the units and dimensions of the reciprocal of x. If x is measured in meters, $f(x)$ has units $1/$meter. Thus, the density is not measured on a probability scale, so it is possible for $f(x)$ to exceed 1.

To see this, think of a uniform density on the interval 0 to 1. The area under the density curve is 1: this is the product of the density, which is constant at 1, and the range, which is 1. If the variable is then transformed by doubling, the area under the curve remains 1 and is the product of the density, constant at 0.5, and the range, which is 2. Conversely, if the variable is transformed by halving, the area under the curve also remains at 1 and is the product of the density, constant at 2, and the range, which is 0.5. (Strictly, the range is measured in certain units, and the density is measured in the reciprocal of those units, so the units cancel on multiplication.)

❑

Stored results

`kdensity` stores the following in `r()`:

Scalars

`r(bwidth)`	kernel bandwidth
`r(n)`	number of points at which the estimate was evaluated
`r(scale)`	density bin width

Macros

`r(kernel)`	name of kernel

Methods and formulas

A kernel density estimate is formed by summing the weighted values calculated with the kernel function K, as in

$$\widehat{f}_K = \frac{1}{qh}\sum_{i=1}^{n} w_i K\left(\frac{x - X_i}{h}\right)$$

where $q = \sum_i w_i$ if weights are frequency weights (`fweight`) or analytic weights (`aweight`), and $q = 1$ if weights are importance weights (`iweights`). Analytic weights are rescaled so that $\sum_i w_i = n$ (see [U] **11 Language syntax**). If weights are not used, then $w_i = 1$, for $i = 1, \ldots, n$. `kdensity` includes seven different kernel functions. The Epanechnikov is the default function if no other kernel is specified and is the most efficient in minimizing the mean integrated squared error.

Kernel	Formula	
Biweight	$K[z] = \begin{cases} \frac{15}{16}(1-z^2)^2 & \\ 0 & \end{cases}$	if $\lvert z\rvert < 1$ otherwise
Cosine	$K[z] = \begin{cases} 1+\cos(2\pi z) & \\ 0 & \end{cases}$	if $\lvert z\rvert < 1/2$ otherwise
Epanechnikov	$K[z] = \begin{cases} \frac{3}{4}(1-\frac{1}{5}z^2)/\sqrt{5} & \\ 0 & \end{cases}$	if $\lvert z\rvert < \sqrt{5}$ otherwise
Epan2	$K[z] = \begin{cases} \frac{3}{4}(1-z^2) & \\ 0 & \end{cases}$	if $\lvert z\rvert < 1$ otherwise
Gaussian	$K[z] = \frac{1}{\sqrt{2\pi}}e^{-z^2/2}$	
Parzen	$K[z] = \begin{cases} \frac{4}{3} - 8z^2 + 8\lvert z\rvert^3 & \\ 8(1-\lvert z\rvert)^3/3 & \\ 0 & \end{cases}$	if $\lvert z\rvert \le 1/2$ if $1/2 < \lvert z\rvert \le 1$ otherwise
Rectangular	$K[z] = \begin{cases} 1/2 & \\ 0 & \end{cases}$	if $\lvert z\rvert < 1$ otherwise
Triangular	$K[z] = \begin{cases} 1-\lvert z\rvert & \\ 0 & \end{cases}$	if $\lvert z\rvert < 1$ otherwise

From the definitions given in the table, we can see that the choice of h will drive how many values are included in estimating the density at each point. This value is called the *window width* or *bandwidth*. If the window width is not specified, it is determined as

$$m = \min\left(\sqrt{\text{variance}_x},\ \frac{\text{interquartile range}_x}{1.349}\right)$$
$$h = \frac{0.9m}{n^{1/5}}$$

where x is the variable for which we wish to estimate the kernel and n is the number of observations.

Most researchers agree that the choice of kernel is not as important as the choice of bandwidth. There is a great deal of literature on choosing bandwidths under various conditions; see, for example, Parzen (1962) or Tapia and Thompson (1978). Also see Newton (1988) for a comparison with sample spectral density estimation in time-series applications.

Acknowledgments

We gratefully acknowledge the previous work by Isaías H. Salgado-Ugarte of Universidad Nacional Autónoma de México, and Makoto Shimizu and Toru Taniuchi of the University of Tokyo; see Salgado-Ugarte, Shimizu, and Taniuchi (1993). Their article provides a good overview of the subject of univariate kernel density estimation and presents arguments for its use in exploratory data analysis.

References

Cox, N. J. 2005. Speaking Stata: Density probability plots. *Stata Journal* 5: 259–273.

——. 2007. Kernel estimation as a basic tool for geomorphological data analysis. *Earth Surface Processes and Landforms* 32: 1902–1912. https://doi.org/10.1002/esp.1518.

Fiorio, C. V. 2004. Confidence intervals for kernel density estimation. *Stata Journal* 4: 168–179.

Fox, J. 1990. Describing univariate distributions. In *Modern Methods of Data Analysis*, ed. J. Fox and J. S. Long, 58–125. Newbury Park, CA: SAGE.

Goeden, G. B. 1978. A monograph of the coral trout, *Plectropomus leopardus* (Lacépède). *Queensland Fisheries Services Research Bulletin* 1: 1–42.

Kohler, U., and F. Kreuter. 2012. *Data Analysis Using Stata*. 3rd ed. College Station, TX: Stata Press.

López-de-Ullibarri, I. 2015. Bandwidth selection in kernel distribution function estimation. *Stata Journal* 15: 784–795.

Newton, H. J. 1988. *TIMESLAB: A Time Series Analysis Laboratory*. Belmont, CA: Wadsworth.

Parzen, E. 1962. On estimation of a probability density function and mode. *Annals of Mathematical Statistics* 33: 1065–1076. https://doi.org/10.1214/aoms/1177704472.

Royston, P., and N. J. Cox. 2005. A multivariable scatterplot smoother. *Stata Journal* 5: 405–412.

Salgado-Ugarte, I. H., and M. A. Pérez-Hernández. 2003. Exploring the use of variable bandwidth kernel density estimators. *Stata Journal* 3: 133–147.

Salgado-Ugarte, I. H., M. Shimizu, and T. Taniuchi. 1993. snp6: Exploring the shape of univariate data using kernel density estimators. *Stata Technical Bulletin* 16: 8–19. Reprinted in *Stata Technical Bulletin Reprints*, vol. 3, pp. 155–173. College Station, TX: Stata Press.

Scott, D. W. 2015. *Multivariate Density Estimation: Theory, Practice, and Visualization*. 2nd ed. Hoboken, NJ: Wiley.

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

Simonoff, J. S. 1996. *Smoothing Methods in Statistics*. New York: Springer.

Tapia, R. A., and J. R. Thompson. 1978. *Nonparametric Probability Density Estimation*. Baltimore: Johns Hopkins University Press.

Van Kerm, P. 2003. Adaptive kernel density estimation. *Stata Journal* 3: 148–156.

——. 2012. Kernel-smoothed cumulative distribution function estimation with akdensity. *Stata Journal* 12: 543–548.

Wand, M. P., and M. C. Jones. 1995. *Kernel Smoothing*. London: Chapman & Hall.

Also see

[R] **histogram** — Histograms for continuous and categorical variables

[R] **npregress kernel** — Nonparametric kernel regression

[R] **npregress series** — Nonparametric series regression