The Stata Journal (2012)
12, Number 3, pp. 543–548

Kernel-smoothed cumulative distribution function estimation with akdensity

Philippe Van Kerm
CEPS/INSTEAD
Esch/Alzette, Luxembourg
philippe.vankerm@ceps.lu

Abstract. In this article, I describe estimation of the kernel-smoothed cumulative distribution function with the user-written package `akdensity`, with formulas and an example.

Keywords: st0037_3, akdensity, smoothed cumulative distribution function, kernel functions

1 Introduction

`akdensity` is a user-written Stata package for (univariate) density estimation using adaptive kernel methods (Van Kerm 2003). Here I describe the recently added functionality for estimation of kernel-smoothed cumulative distribution functions in addition to density functions. I provide the syntax, formulas, and an example for `akdensity`'s new option `cdf(`*newvar*`)`, available in the latest software update.[1]

2 Methods and formulas

The adaptive kernel density estimate computed by `akdensity` is given by

$$\widehat{f}(x) = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} \frac{w_i}{h_i} K\left(\frac{x - x_i}{h_i}\right)$$

where the x_i's are data points (associated with sample weights w_i), K is a kernel function, and $h_i = h \times \lambda_i$, where h is a global bandwidth parameter and λ_i is a bandwidth adaptation factor proportional to the square root of the density of the data at each sample point (Van Kerm 2003).

The corresponding kernel-smoothed cumulative distribution function (CDF) estimate is given by

$$\begin{aligned}\widehat{F}(x) &= \int_{-\infty}^{x} \widehat{f}(v)dv \\ &= \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i I\left(\frac{x - x_i}{h_i}\right)\end{aligned}$$

1. The latest version of the `akdensity` package is 4.2 (of 2010 November 18). Stata 7.0 or later is required.

where I is the integral of the kernel function K

$$I(x) = \int_{-\infty}^{x} K(v)dv$$

(see, for example, Yamato [1973], Azzalini [1981], Reiss [1981], Kulczycki and Dawidowicz [1999], or Li and Racine [2006]).

For a Gaussian kernel, $K(z) = \phi(z)$ and $I(z) = \Phi(z)$, where ϕ and Φ are the Gaussian probability density functions (PDF) and CDF, respectively. For Epanechnikov kernel functions,

$$K(x) = \begin{cases} \frac{3}{4\sqrt{5}}\left(1 - \frac{1}{5}z^2\right) & \text{if } |z| < \sqrt{5} \\ 0 & \text{otherwise} \end{cases}$$

$$I(z) = \begin{cases} 0 & \text{if } z < -\sqrt{5} \\ \frac{1}{2} + \frac{3}{4\sqrt{5}}(z - \frac{1}{15}z^3) & \text{if } |z| < \sqrt{5} \\ 1 & \text{if } z > \sqrt{5} \end{cases}$$

or, for the "alternative" Epanechnikov kernel,

$$K(x) = \begin{cases} \frac{3}{4}(1 - z^2) & \text{if } |z| < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$I(z) = \begin{cases} 0 & \text{if } z < -1 \\ \frac{1}{2} + \frac{3}{4}(z - \frac{1}{3}z^3) & \text{if } |z| < 1 \\ 1 & \text{if } z > 1 \end{cases}$$

3 The akdensity command

3.1 Syntax

The syntax for `akdensity` follows the official `kdensity` syntax:

`akdensity` *varname* [*if*] [*in*] [*weight*] [`,` `noadaptive` `stdbands(`#`)` `cdf(`*newvar*`)` *kdensity_options*]

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

3.2 Options

`noadaptive` can be specified to obtain the standard fixed bandwidth kernel density estimate. The resulting density is as produced by `kdensity`. This may be used to obtain variability bands around the fixed kernel density estimates or kernel-smoothed CDF estimates with fixed bandwidth.

`stdbands(`#`)` requests the estimation of variability bands and specifies the number of standard errors above and below the estimates to be used (a positive number). If the

`generate()` option is specified, the estimated bands are stored in two new variables: *newvar*`_density_up` and *newvar*`_density_lo`. See Van Kerm (2003) for details.

`cdf(`*newvar*`)` is a new option and requests estimation of the kernel-smoothed CDF in addition to the density function. Both function estimates are based on identical (adaptive) bandwidth and kernel function specifications. CDF estimates for each point on the grid specified by the `at()` or `n()` option are stored in *newvar*. If `stdbands()` is specified, estimates of pointwise variability bands for $\widehat{F}$ are also constructed and stored in variables *newvar*`_lo` and *newvar*`_up`.[2]

kdensity_options are the official `kdensity` options, with the exception of `kernel(`*kernel*`)`, which here only accepts three possible kernel functions (`epanechnikov`, `epan2`, or `gaussian`); see [R] **kdensity**.[3]

4 The akdensity0 command

4.1 Syntax

The syntax for the companion command `akdensity0` is

`akdensity0` *varname* [*if*] [*in*] [*weight*]`,` `bwidth(`*#* | *varname*`)`
`generate(`*newvar*`)` `at(`*var_x*`)` [`stdbands(`*#*`)` `cdf(`*newvar*`)` `lambda(`*newvar*`)`
`kernel(`*kernel*`)` `double`]

`fweight`s and `aweight`s are allowed; see [U] **11.1.6 weight**.

4.2 Options

`bwidth(`*#* | *varname*`)`, `generate(`*newvar*`)`, and `at(`*var_x*`)` are required. These options are as in `kdensity`; see [R] **kdensity**. Note, however, that the `bwidth()` option can here be either a scalar or a variable name containing observation-specific bandwidths. Also `generate()` must specify a single new variable name to store the estimated value of the density function at the grid points.

`stdbands(`*#*`)`, `cdf(`*newvar*`)`, and `kernel(`*kernel*`)`; see *Options for akdensity* above.

`lambda(`*newvar*`)` requests the estimation of local bandwidth factors based on the estimated density function and specifies a new variable name where these values are to be stored.

2. These variability bands are constructed for consistency with PDF bands as $\widehat{F}(x) \pm b \times V\{\widehat{F}(x)\}^{0.5}$ with $V\{\widehat{F}(x)\} = \{\sum_{i=1}^{n} w_i^2/(\sum_{i=1}^{n} w_i)^2\}[\widehat{F}(x)\{1-\widehat{F}(x)\} - \widehat{f}(x)h\lambda(x)\alpha(K)]$, where $\alpha(K)$ is a kernel-specific constant (Van Kerm 2003; Li and Racine 2006).
3. `akdensity` options have been updated to conform to Stata 11 syntax. The command remains backward-compatible with earlier releases, though some options have become undocumented.

`double` requests the use of double precision in the estimation of the density functions and standard error bands.

5 Example

The following example illustrates `akdensity`'s `cdf()` option by using the coral trout length data used in Van Kerm (2003).

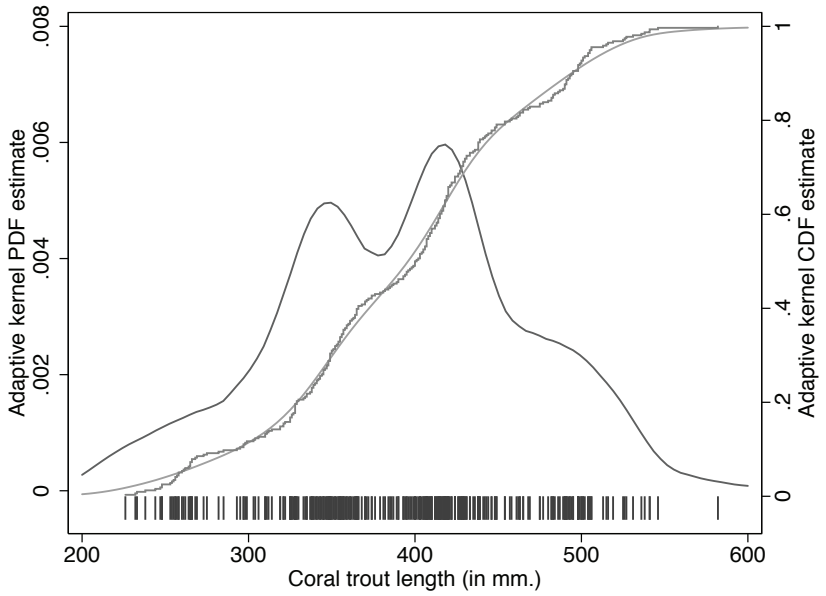
```
. use trocolen
```

First, `cdf()` is used with the default bandwidth selected by `akdensity`. The first plot (top panel in figure 1, below) illustrates the resulting estimates of both the PDF and CDF estimates along with the empirical CDF estimate.

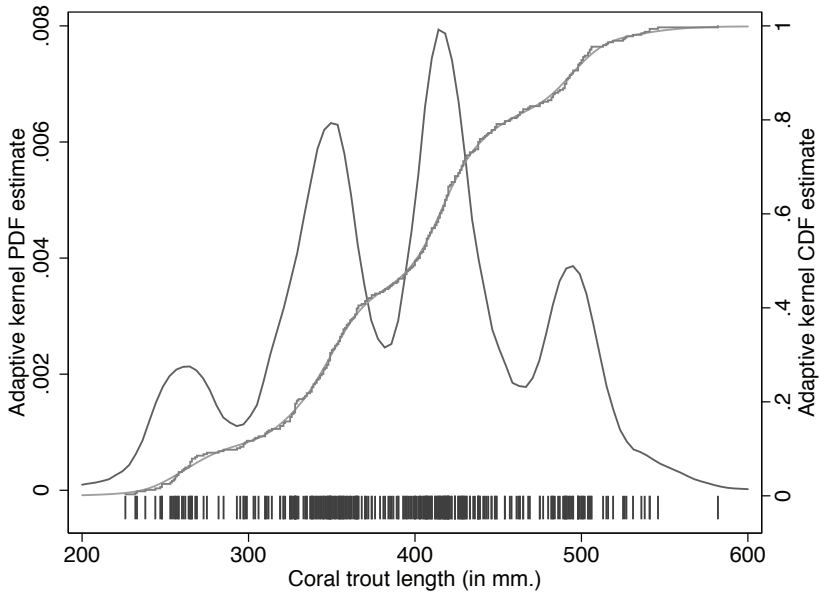
```
. generate to = - 0.05
. local spikes "(dropline to length, msymbol(none) yaxis(2))"
. local gropts `spikes´, scheme(s1mono) legend(off)
> yscale(range(-0.0005 0.008) axis(1)) ylabel(0(.002)0.008, axis(1))
> xtitle("Coral trout length (in mm.)")
> ytitle("Adaptive kernel PDF estimate", axis(1))
> ytitle("Adaptive kernel CDF estimate", axis(2))
. range x 200 600 100
(216 missing values generated)
. label var x "Length"
. akdensity length, nograph generate(fx) at(x) cdf(Fx)
Two-stage adaptive kernel density estimation
Step 1: Pilot density and local bandwidth factors estimation
Step 2: Adaptive kernel density estimation
. cumul length, generate(ecdf)
. twoway (line fx x) (line Fx x, yaxis(2)) (line ecdf length, connect(stairstep)
> sort yaxis(2)) `gropts´
```

Note that preference over bandwidth size may differ according to focus on smoothing the PDF or the CDF. For consistency, however, `akdensity` will estimate both PDF and CDF using identical bandwidths. The second plot (bottom panel in figure 1) illustrates PDF and CDF estimates with a smaller bandwidth.

```
. akdensity length, nograph generate(fx2) at(x) cdf(Fx2) bwidth(10)
Two-stage adaptive kernel density estimation
Step 1: Pilot density and local bandwidth factors estimation
Step 2: Adaptive kernel density estimation
. twoway (line fx2 x) (line Fx2 x, yaxis(2))
> (line ecdf length, connect(stairstep) sort yaxis(2)) `gropts´
```



(a) Default bandwidth



(b) Bandwidth set to 10 mm.

Figure 1. Adaptive kernel PDF and CDF estimates and empirical CDF

6 Acknowledgments

Implementation of the new `cdf()` option and preparation of this article was financially supported by the World Bank Knowledge for Change Program (KCP II-TF094570). Support from the Luxembourg FNR is also gratefully acknowledged (FNR/06/15/08).

7 References

Azzalini, A. 1981. A note on the estimation of a distribution function and quantiles by a kernel method. *Biometrika* 68: 326–328.

Kulczycki, P., and A. L. Dawidowicz. 1999. Kernel estimator of quantile. *Universitatis Iagellonicae Acta Mathematica* 37: 325–336.

Li, Q., and J. S. Racine. 2006. *Nonparametric Econometrics: Theory and Practice*. Princeton, NJ: Princeton University Press.

Reiss, R.-D. 1981. Nonparametric estimation of smooth distribution functions. *Scandinavian Journal of Statistics* 8: 116–119.

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

Yamato, H. 1973. Uniform convergence of an estimator of a distribution function. *Bulletin of Mathematical Statistics* 15: 69–78.

About the author

Philippe Van Kerm is a research economist at CEPS/INSTEAD (G.-D. Luxembourg). His research focuses on applied microeconometrics, with particular interest in income distribution issues.