Nonparametric density estimation

Christopher F Baum

ECON 8823: Applied Econometrics

Boston College, Spring 2015

Kernel density plot

To describe a categorical variable or a continuous variable taking on discrete values, such as age measured in years, a histogram is often employed. For a continuous variable taking on many values, the *kernel density plot* is a better alternative to the histogram. This smoothed rendition connects the midpoints of the histogram, rather than forming the histogram as a step function, and it gives more weight to data that are closer to the point of evaluation.

Let $f(x)$ denote the density function of a continuous RV. The kernel density estimate of $f(x)$ at $x = x_0$ is then

$$\widehat{f}(x_0) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right)$$

where $K(\cdot)$ is a kernel function that places greater weight on points x_i that are closer to x_0.

The kernel function is symmetric around zero and integrates to one. Either $K(z) = 0$ if $|z| \geq z_0$, for some z_0, or $K(z) \rightarrow 0$ as $z \rightarrow \infty$.

A histogram with bin width $2h$ evaluated at x_0 is the special case $K(z) = 0.5$ if $|z| < 1$, $K(z) = 0$ otherwise.

Let $f(x)$ denote the density function of a continuous RV. The kernel density estimate of $f(x)$ at $x = x_0$ is then

$$\widehat{f}(x_0) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right)$$

where $K(\cdot)$ is a kernel function that places greater weight on points x_i that are closer to x_0.

The kernel function is symmetric around zero and integrates to one. Either $K(z) = 0$ if $|z| \geq z_0$, for some z_0, or $K(z) \rightarrow 0$ as $z \rightarrow \infty$.

A histogram with bin width $2h$ evaluated at x_0 is the special case $K(z) = 0.5$ if $|z| < 1$, $K(z) = 0$ otherwise.

Let $f(x)$ denote the density function of a continuous RV. The kernel density estimate of $f(x)$ at $x = x_0$ is then

$$\widehat{f}(x_0) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right)$$

where $K(\cdot)$ is a kernel function that places greater weight on points x_i that are closer to x_0.

The kernel function is symmetric around zero and integrates to one. Either $K(z) = 0$ if $|z| \geq z_0$, for some z_0, or $K(z) \rightarrow 0$ as $z \rightarrow \infty$.

A histogram with bin width $2h$ evaluated at x_0 is the special case $K(z) = 0.5$ if $|z| < 1$, $K(z) = 0$ otherwise.

A kernel density plot requires the choice of a kernel function, $K(\cdot)$ and a bandwidth h. You then evaluate the kernel density function at a number of values x_0, and plot those estimates against x_0.

In Stata, the `kdensity` command produces the kernel density estimate. The default kernel function is the Epanechnikov kernel, which sets $K(z) = (3/4)(1 - z^2/5)/\sqrt{5}$ for $|z| < \sqrt{5}$ and zero otherwise. This kernel function is said to be the most efficient in minimizing the mean integrated squared error.

A kernel density plot requires the choice of a kernel function, $K(\cdot)$ and a bandwidth h. You then evaluate the kernel density function at a number of values x_0, and plot those estimates against x_0.

In Stata, the `kdensity` command produces the kernel density estimate. The default kernel function is the Epanechnikov kernel, which sets $K(z) = (3/4)(1 - z^2/5)/\sqrt{5}$ for $|z| < \sqrt{5}$ and zero otherwise. This kernel function is said to be the most efficient in minimizing the mean integrated squared error.

Other kernel functions available include an alternative Epanechnikov kernel, as well as biweight, cosine, Gaussian, Parzen, rectangular, and triangle kernels. All but the Gaussian have a cutoff point, beyond which the kernel function is zero.

The choice of kernel bandwidth (the `bwidth()` option) determines how quickly the cutoff is reached. A small bandwidth will cause the kernel density estimate to depend only on values close to the point of evaluation, while a larger bandwidth will include more of the values in the vicinity of the point, yielding a smoother estimate. Most researchers agree that the choice of kernel is not as important as the choice of bandwidth.

Other kernel functions available include an alternative Epanechnikov kernel, as well as biweight, cosine, Gaussian, Parzen, rectangular, and triangle kernels. All but the Gaussian have a cutoff point, beyond which the kernel function is zero.

The choice of kernel bandwidth (the `bwidth()` option) determines how quickly the cutoff is reached. A small bandwidth will cause the kernel density estimate to depend only on values close to the point of evaluation, while a larger bandwidth will include more of the values in the vicinity of the point, yielding a smoother estimate. Most researchers agree that the choice of kernel is not as important as the choice of bandwidth.

If no bandwidth is specified, it is chosen according to

$$m = \min\left(sd(x), \frac{IQR(x)}{1.349}\right)$$

$$h = \frac{0.9m}{n^{1/5}}$$

where $sd(x)$ and $IQR(x)$ refer to the standard deviation and inter-quartile range of the series x, respectively.

The default number of x_0 points is 50, which may be set with the `n()` option, or a variable containing values at which the kernel density estimate is to be produced may be specified with the `at()` option. You may also use the `generate()` option to produce new variables containing the plotted coordinates.

If no bandwidth is specified, it is chosen according to

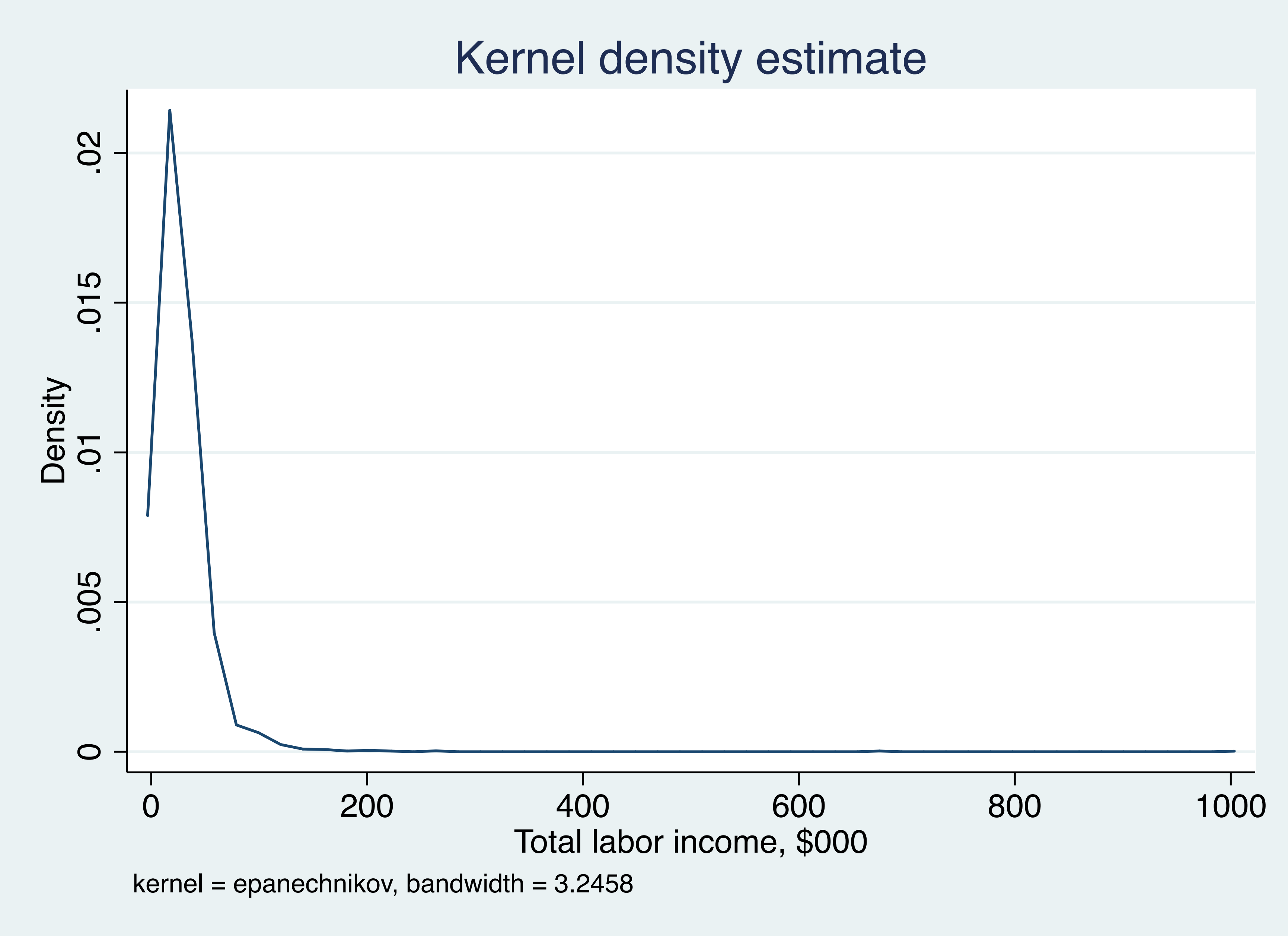
$$
\begin{aligned}
m &= \min\left(sd(x), \frac{IQR(x)}{1.349}\right) \\
h &= \frac{0.9m}{n^{1/5}}
\end{aligned}
$$

where $sd(x)$ and $IQR(x)$ refer to the standard deviation and inter-quartile range of the series x, respectively.

The default number of x_0 points is 50, which may be set with the `n()` option, or a variable containing values at which the kernel density estimate is to be produced may be specified with the `at()` option. You may also use the `generate()` option to produce new variables containing the plotted coordinates.

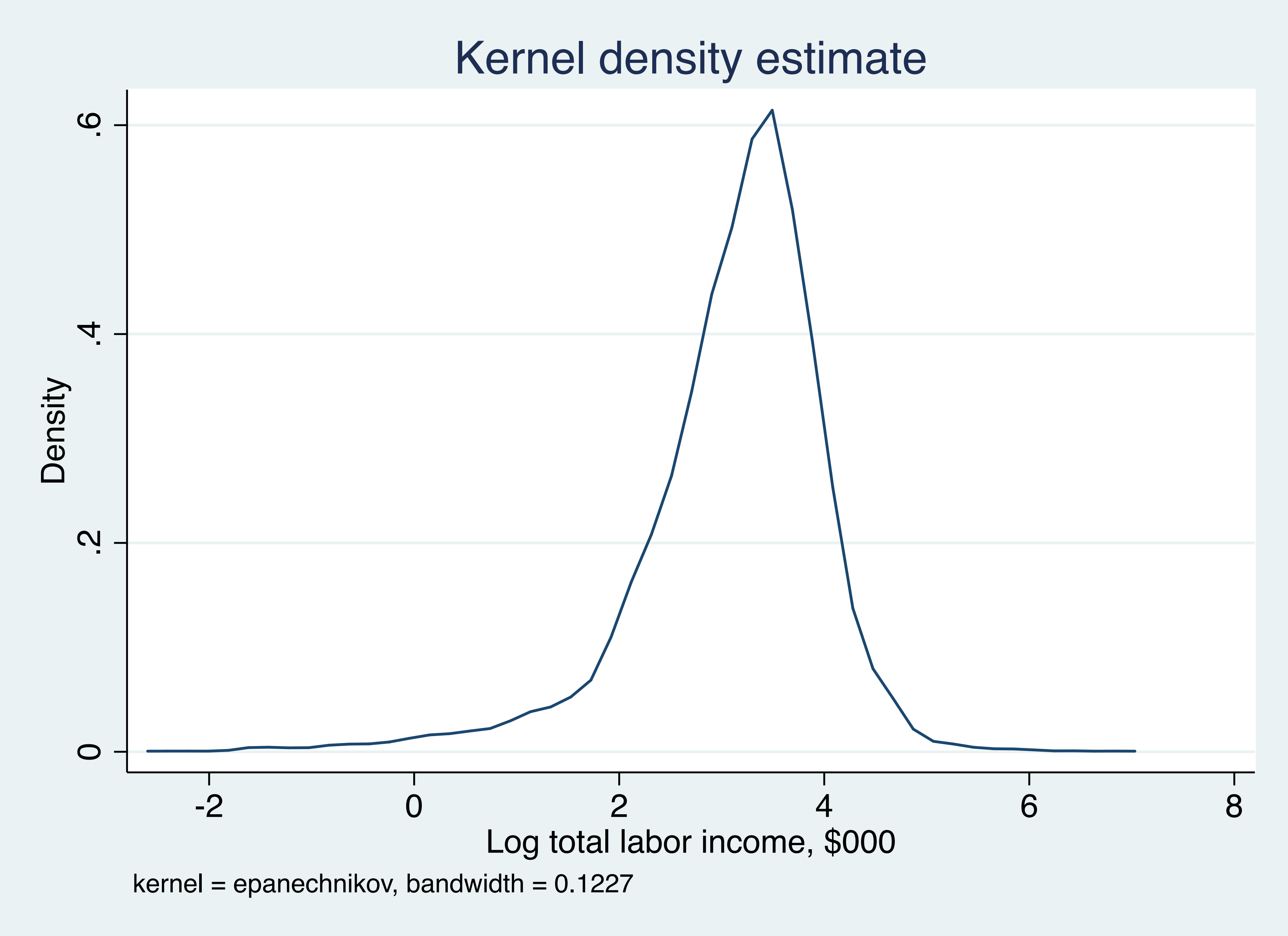
```
. use mus02psid92m.dta

. g earningsk = earnings/1000
(209 missing values generated)

. lab var earningsk "Total labor income, $000"

. kdensity earningsk
```

Kernel density of labor earnings

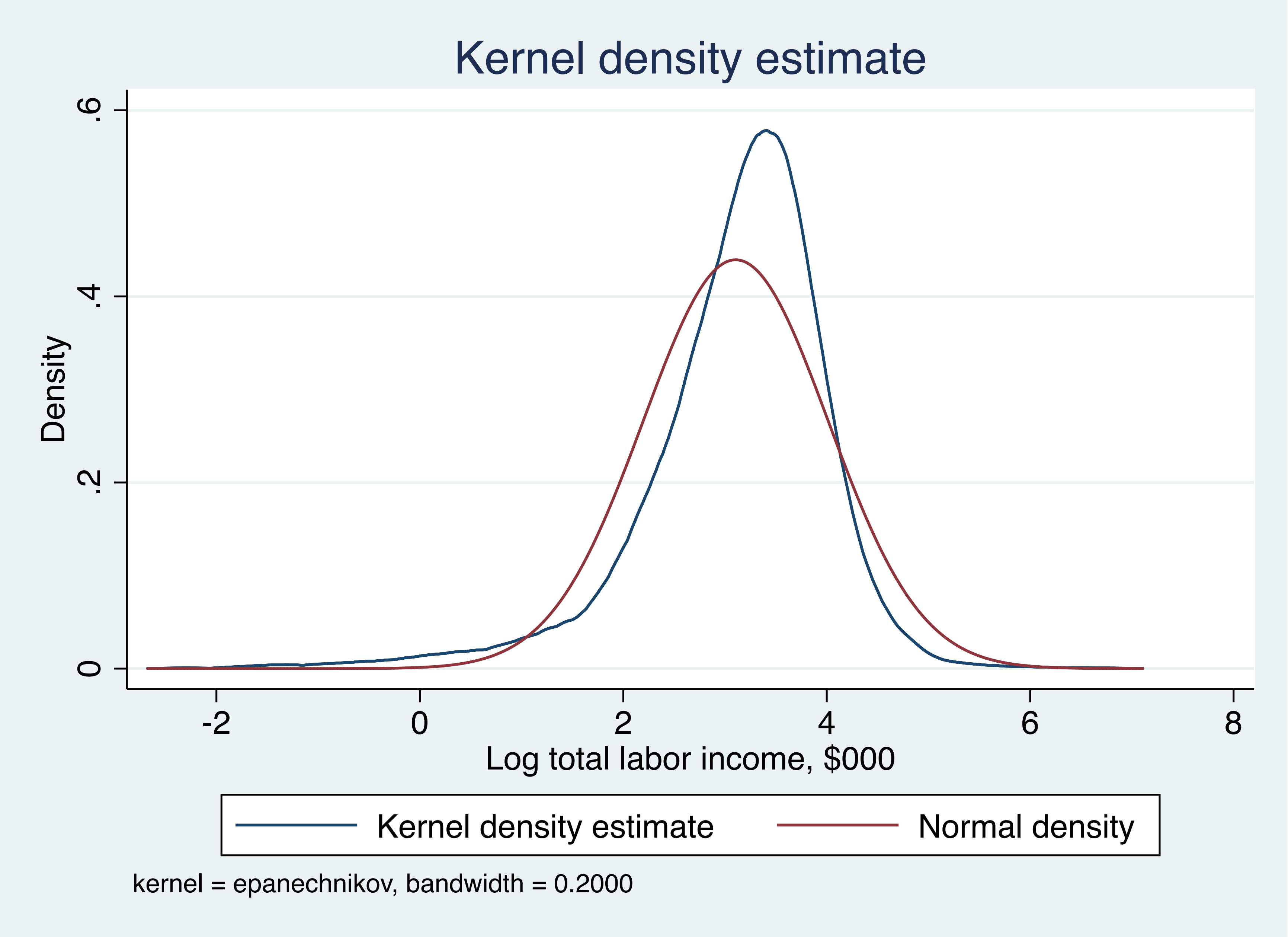


```
. g lek = log(earningsk)
(498 missing values generated)

. lab var lek "Log total labor income, $000"

. kdensity lek

. gr export 82303b.pdf, replace
(file /Users/cfbaum/Dropbox/baum/EC823 S2013/82303b.pdf written in PDF format)

. kdensity lek, bw(0.20) normal n(4000) leg(rows(1))
```

Kernel density of log earnings, default bandwidth



Kernel density with wider bandwidth, $n \simeq N$ of sample



Bivariate kernel density estimates

We may also want to consider bivariate relationships, and analyze an empirical bivariate density using nonparametric means. The univariate kernel density estimator can be generalized to a bivariate context. Gallup and Baum's `bidensity` command, available from SSC, produces bivariate kernel density estimates and illustrates them with a `contourline`, or topographic map, plot.

Available kernels include Epanechnikov and alternative, Gaussian, rectangle and triangle, each the product of the univariate kernel functions defined in `kdensity`. The bandwidth defaults are those employed in `kdensity`. The `saving()` option allows you to create a new dataset containing the x, y, and $f(x, y)$ variables.

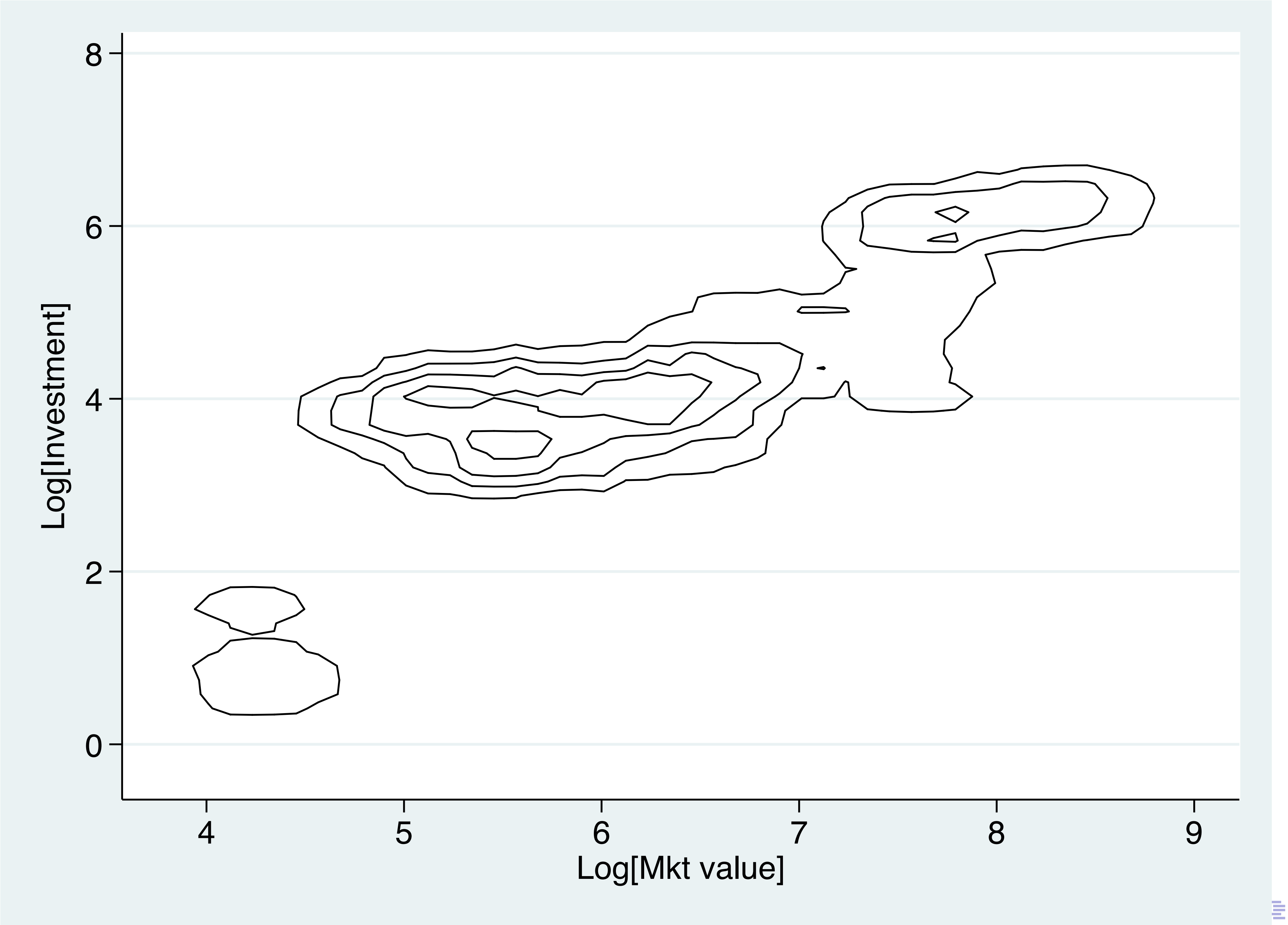
Bivariate kernel density estimates

We may also want to consider bivariate relationships, and analyze an empirical bivariate density using nonparametric means. The univariate kernel density estimator can be generalized to a bivariate context. Gallup and Baum's `bidensity` command, available from SSC, produces bivariate kernel density estimates and illustrates them with a `contourline`, or topographic map, plot.

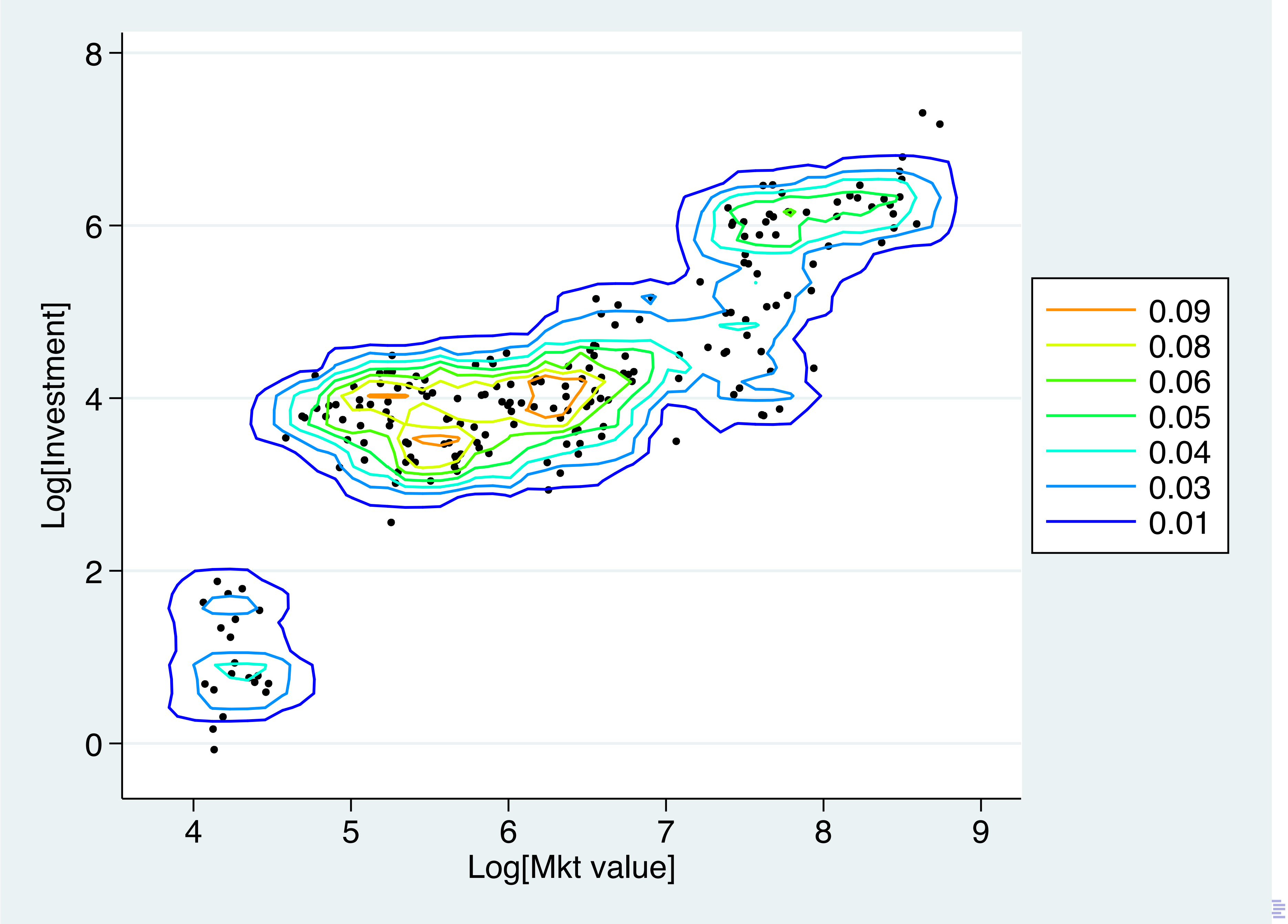
Available kernels include Epanechnikov and alternative, Gaussian, rectangle and triangle, each the product of the univariate kernel functions defined in `kdensity`. The bandwidth defaults are those employed in `kdensity`. The `saving()` option allows you to create a new dataset containing the x, y, and $f(x, y)$ variables.

```
. webuse grunfeld, clear
. gen linv = log(invest)
. lab var linv "Log[Investment]"
. gen lmkt = log(mvalue)
. lab var lmkt "Log[Mkt value]"
. bidensity linv lmkt
. gr export 82303d.pdf, replace
(file /Users/cfbaum/Dropbox/baum/EC823 S2013/82303d.pdf written in PDF format)
. bidensity linv lmkt, scatter(msize(vsmall) mcolor(black)) ///
> colorlines levels(8) format(%3.2f)
```

Bivariate kernel density



Bivariate kernel density with scatterplot overlay



Local polynomial regression

While the bivariate density provides a nonparametric estimate of the joint density of x and y, it does not presume any causal relationship among those variables. A variety of local linear regression techniques may be employed to flexibly model the relationship between explanatory variable x and outcome variable y.

The local linear aspect of these techniques refers to the concept that the relationship is modeled as linear in the neighborhood, but may vary across values of x.

Local polynomial regression

While the bivariate density provides a nonparametric estimate of the joint density of x and y, it does not presume any causal relationship among those variables. A variety of local linear regression techniques may be employed to flexibly model the relationship between explanatory variable x and outcome variable y.

The local linear aspect of these techniques refers to the concept that the relationship is modeled as linear in the neighborhood, but may vary across values of x.

Local linear regression techniques model $y = m(x) + u$, where the conditional mean function $m(\cdot)$ is not specified. The estimate of $m(x)$ at $x = x_0$ is a local weighted average of y_i where high weight is placed on observations for which x_i is close to x_0 and little or no weight is placed on observations with x_i far from x_0. Formally,

$$\widehat{m}(x_0) = \sum_{i=1}^{N} w(x_i, x_0, h) y_i$$

where the weights $w(\cdot)$ sum to one and decrease as the distance $|x_i - x_0|$ increases.

As in the kernel density estimator, the bandwidth parameter h controls the process. A narrower bandwidth (smaller h) causes more weight to be placed on nearby observations.

Local linear regression techniques model $y = m(x) + u$, where the conditional mean function $m(\cdot)$ is not specified. The estimate of $m(x)$ at $x = x_0$ is a local weighted average of y_i where high weight is placed on observations for which x_i is close to x_0 and little or no weight is placed on observations with x_i far from x_0. Formally,

$$\widehat{m}(x_0) = \sum_{i=1}^{N} w(x_i, x_0, h) y_i$$

where the weights $w(\cdot)$ sum to one and decrease as the distance $|x_i - x_0|$ increases.

As in the kernel density estimator, the bandwidth parameter h controls the process. A narrower bandwidth (smaller h) causes more weight to be placed on nearby observations.

After defining a kernel function $K(\cdot)$, a local linear regression estimate at $x = x_0$ can be obtained by minimizing

$$\sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right)(y_i - \alpha - \beta(x_i - x_0))^2$$

This may be generalized to the local polynomial regression estimate produced by Stata's `lpoly`, where the term $\beta(x_i - x_0)$ becomes $\beta(x_i - x_0)^d$, where d is an integer power. If $d = 0$, this becomes local mean smoothing. For $d = 1$, we have a locally weighted least squares model. An estimate fit with higher powers of d has better bias properties than the zero-degree local polynomial. Odd-order degrees are preferable.

After defining a kernel function $K(\cdot)$, a local linear regression estimate at $x = x_0$ can be obtained by minimizing

$$\sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right)(y_i - \alpha - \beta(x_i - x_0))^2$$

This may be generalized to the local polynomial regression estimate produced by Stata's `lpoly`, where the term $\beta(x_i - x_0)$ becomes $\beta(x_i - x_0)^d$, where d is an integer power. If $d = 0$, this becomes local mean smoothing. For $d = 1$, we have a locally weighted least squares model. An estimate fit with higher powers of d has better bias properties than the zero-degree local polynomial. Odd-order degrees are preferable.

The bandwidth, if not specified, is chosen by the rule-of-thumb method, which provides the asymptotically optimal constant bandwidth: that which minimizes the conditional weighted mean integrated squared error. As with `kdensity`, a bandwidth may also be specified. The same set of kernel functions is available, as are options to alter the number of evaluation points (`n()`) and save the results as new variables with `generate()`.

Confidence bands for the local polynomial regression estimate may also be produced with option `ci`, and the sequence of standard errors saved as a new variable with `se()`.

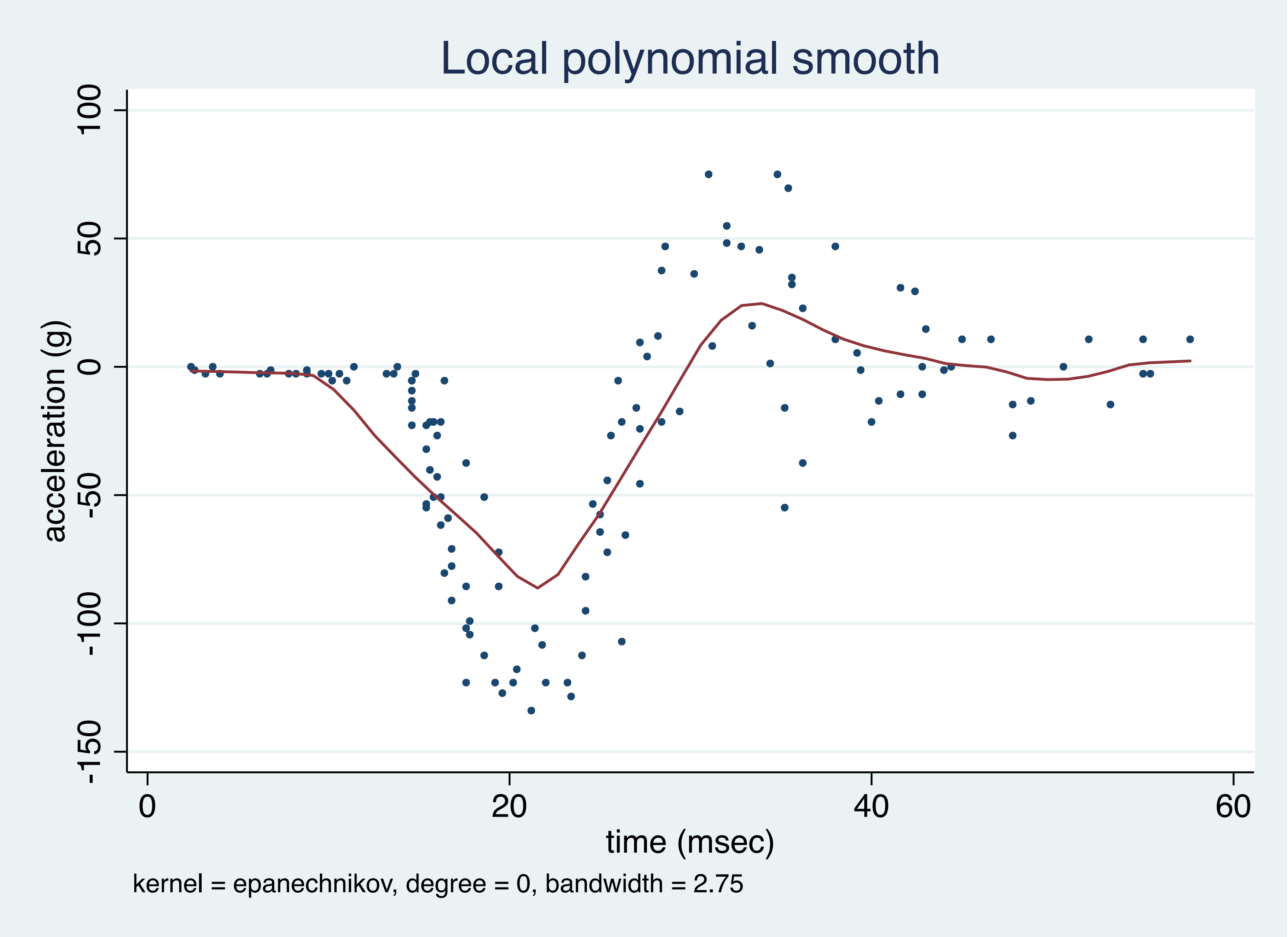
The bandwidth, if not specified, is chosen by the rule-of-thumb method, which provides the asymptotically optimal constant bandwidth: that which minimizes the conditional weighted mean integrated squared error. As with `kdensity`, a bandwidth may also be specified. The same set of kernel functions is available, as are options to alter the number of evaluation points (`n()`) and save the results as new variables with `generate()`.

Confidence bands for the local polynomial regression estimate may also be produced with option `ci`, and the sequence of standard errors saved as a new variable with `se()`.

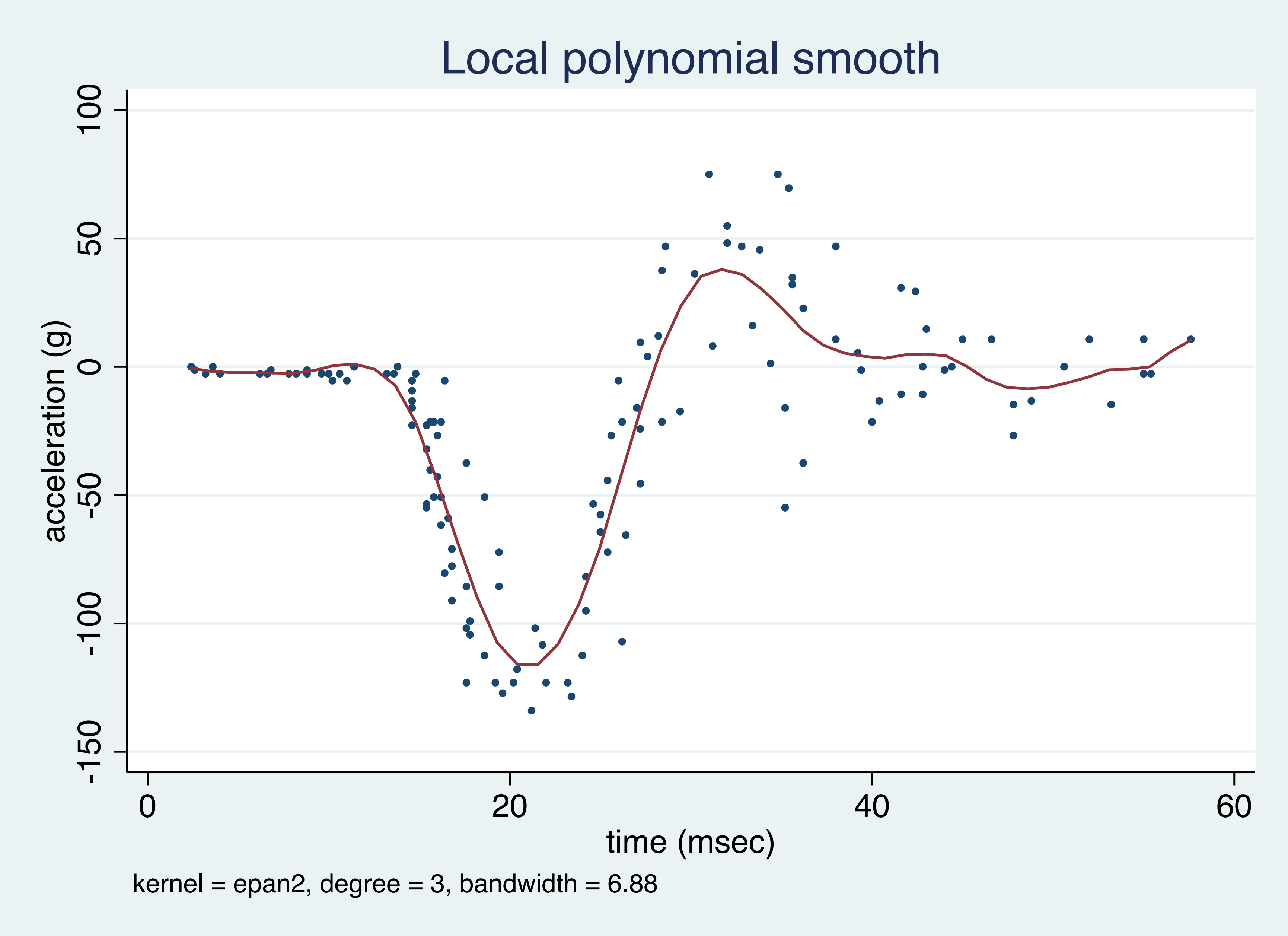
An example of local polynomial regression:

```
. webuse motorcycle, clear
(Motorcycle data from Fan & Gijbels (1996))
. lpoly accel time, msize(vsmall)
. gr export 82303f.pdf, replace
(file /Users/cfbaum/Dropbox/baum/EC823 S2013/82303f.pdf written in PDF format)
. lpoly accel time, degree(3) kernel(epan2) msize(vsmall)
. gr export 82303g.pdf, replace
(file /Users/cfbaum/Dropbox/baum/EC823 S2013/82303g.pdf written in PDF format)
. lpoly accel time, degree(3) kernel(gaussian) msize(vsmall) ci
```

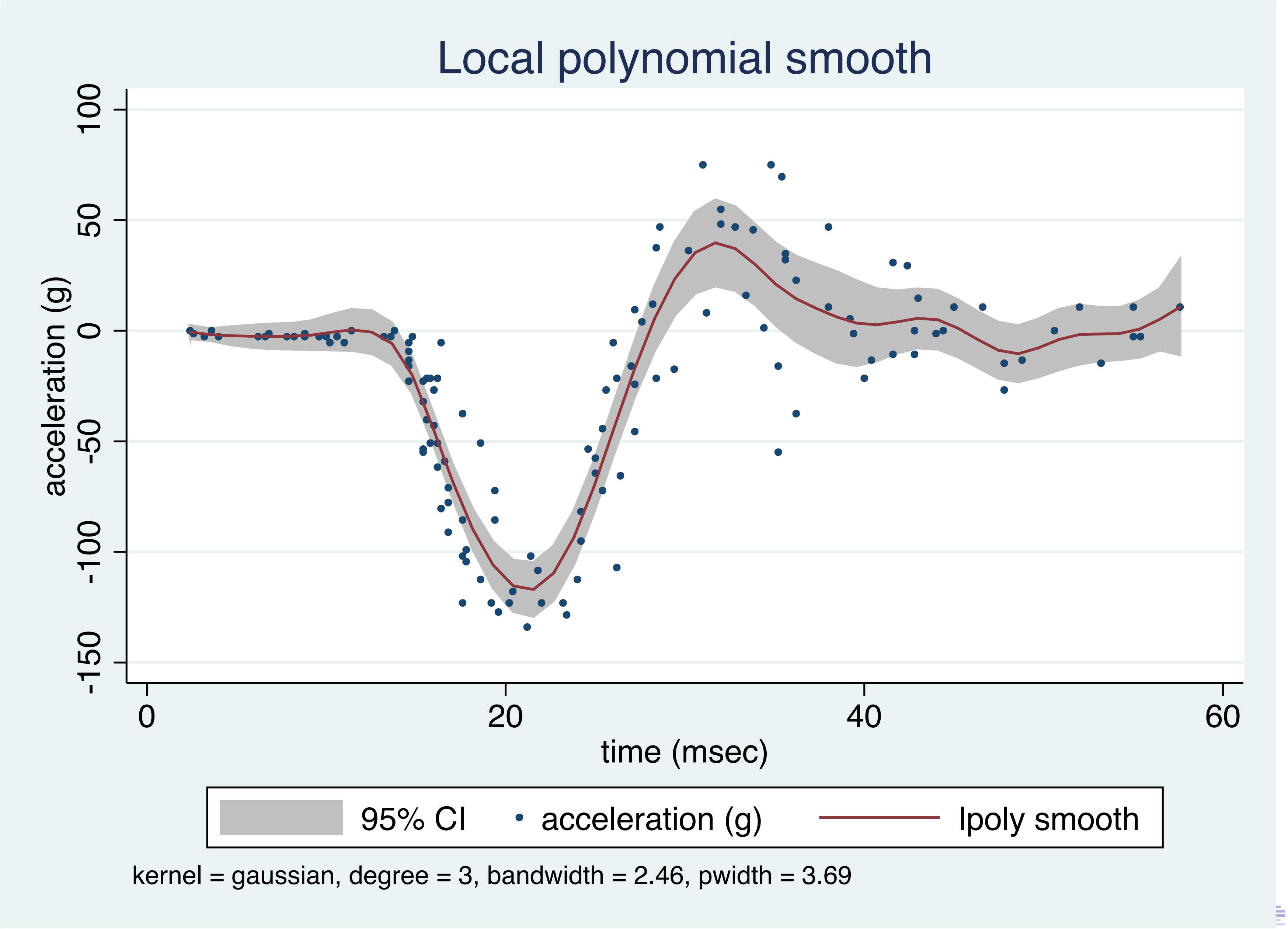
The default kernel and bandwidth do not work very well:



Choosing a different kernel and higher degree improve the fit:



We may also examine the confidence bands around the estimate, now computed with a Gaussian kernel:



A similar methodology is provided by `lowess` (Cleveland, *JASA* 1979), which makes use of a variable bandwidth, with evaluation points near the extrema are smoothed using a narrower bandwidth. Observations with large residuals in the local linear regression are also downweighted, making this method more computationally demanding than local polynomial regression.