Topics in Applied Econometrics for Public Policy

TA Session 2

Sergi Quintana

Barcelona School of Economics

April 25, 2023

Schedule for today

- Regression to the mean and convergence.
- lpoly
- npregress
- Post-estimation commands.
- Margins and contrast.
- In-class exercice.
- Series.. (if time)

Nonparametric Local Regression

The word "nonparametric" refers to the fact that the parameter of interest, the mean as a function of the covariates, is given by the unknown function $m(xi)$.

The regression model of outcome y_i given the k-dimensional vector of covariates x_i is given by:

$$y_i = m(x_i) + \epsilon_i$$

$$E\left(\epsilon_i | x_i\right) = 0$$

$$E(y_{ij}|x_i) = m(x_i)$$

The conditional mean function is therefore given by $m(x_i)$. By estimating $E(y_{ij}|x_i = x)$ for all points x in our data, we obtain an estimate of $E(y_{ij}|x_i)$.

Local Weighted Average

The simplest way of estimating the conditional mean is with local weighted average.

$$\hat{m}(x_0) \equiv \frac{\frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right) y_i}{\frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right)}$$

which is known as the Nadaraya-Watson estimator.

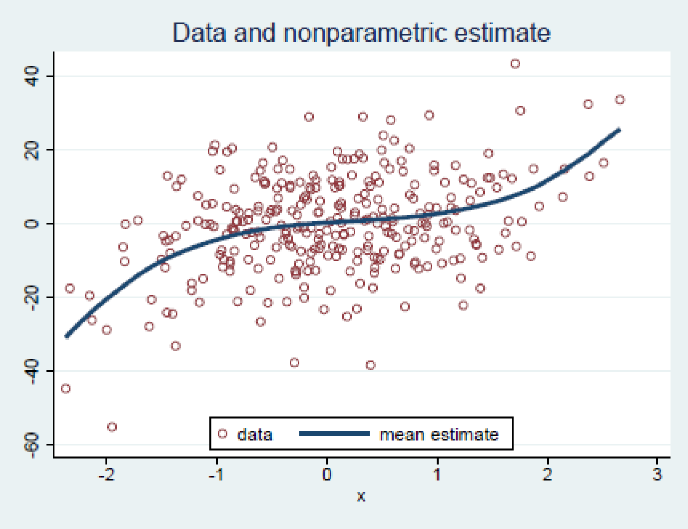
Bias introduced by bandwidth

Incorporating values of y_i for which $x_i \neq x_0$ into the weighted average introduces bias, since $E[y_i|x_i] = m(x_i) \neq m(x_0)$ for $x_i \neq x_0$. However, using these additional points reduces the variance of the estimator, since we are averaging over more data.

Consistency requires $h \to 0$, so that substantial weight is given only to x_i very close to x_0. At the same time we need many x_i close to x_0, so that many observations are used in forming the weighted average. Formally,
$\hat{m}(x_0) \xrightarrow{p} m(x_0)$ if $h \to 0$ and $Nh \to \infty$ as $N \to \infty$.

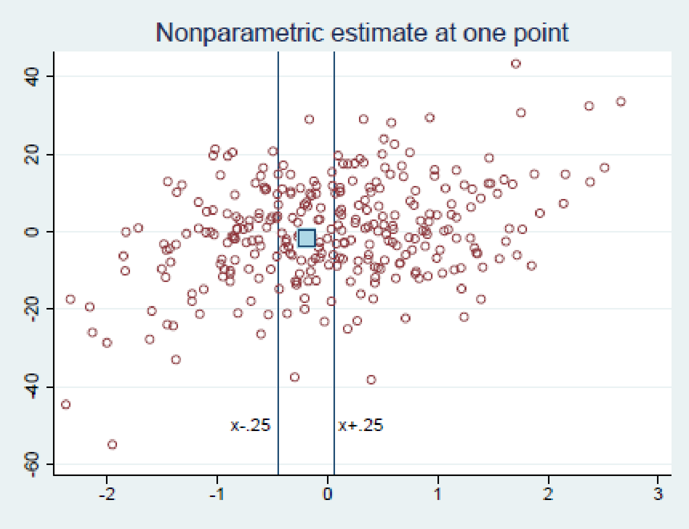
Local Weighted Average

Suppose we want to estimate the blue line.



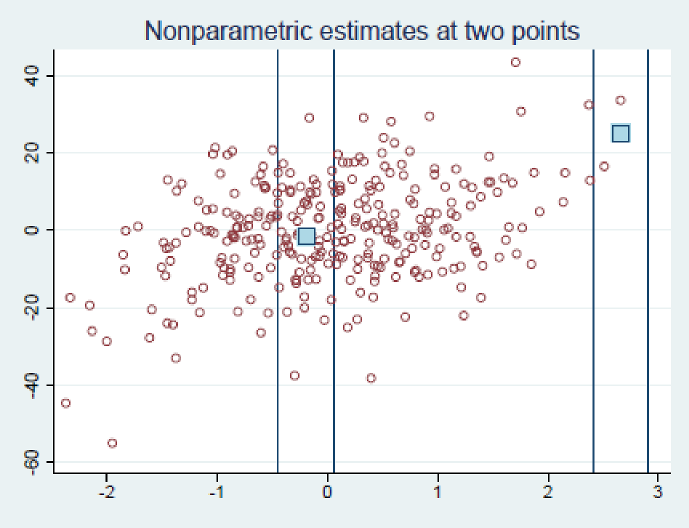
Local Weighted Average

The light square is the estimate of the conditional mean at that interval.



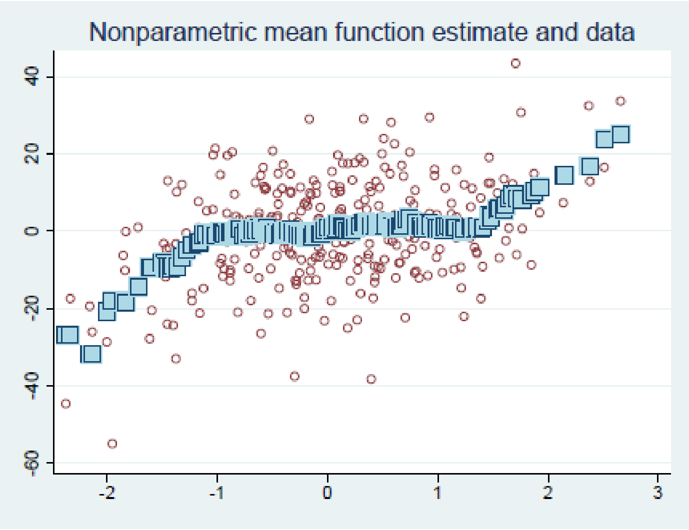
Local Weighted Average

We can do this for other intervals.



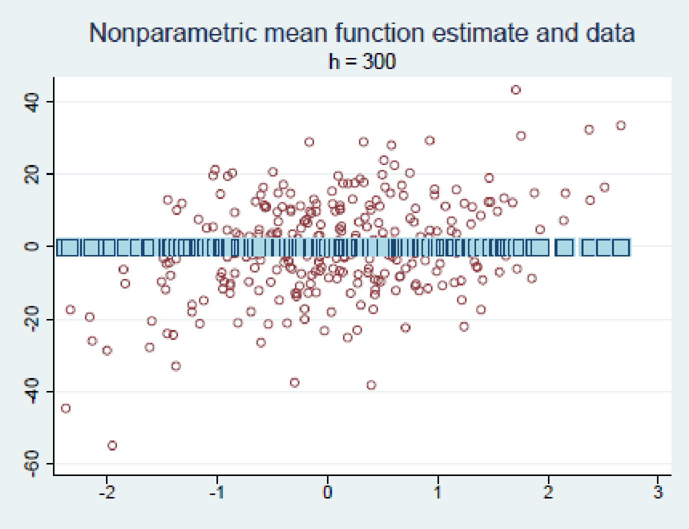
Local Weighted Average

Doing this estimation for each point in our data produces a nonparametric estimate of the mean for a given value of the covariate.



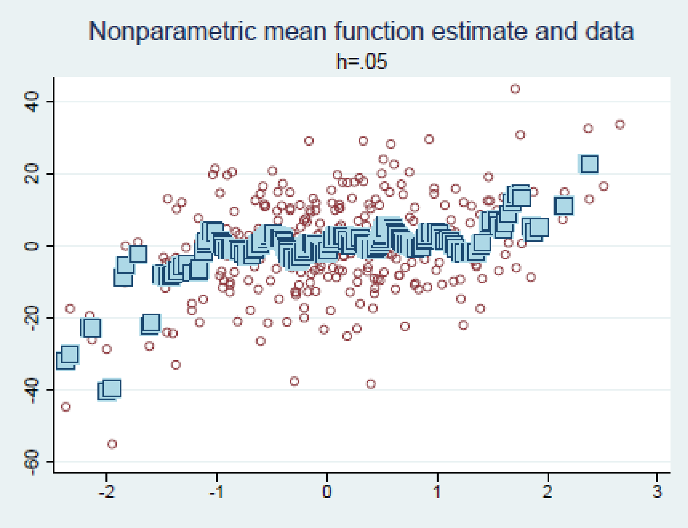
Local Weighted Average

The bandwidth is very important. Suppose we choose a very large one.



Local Weighted Average

On the other hand, too small will introduce variability.



Nonparametric Regression

There are other ways of estimating the conditional mean of y_i non parametrically.

Local-linear estimation. One can let $m(x)$ be linear in the neighborhood of x_0, so that $m(x) = a_0 + b_0(xx_0)$ in the neighborhood of x_0.

Stata Implementation

The main two functions that we will use are lpoly and npregress.

One of the main differences is that the later uses cross-validation to obtain the optimal bandwidth, whereas the former minimizes the mean squared error.

lpoly

lpoly performs a kernel-weighted local polynomial regression of *yvar* on *xvar* and displays a graph of the smoothed values with (optional) confidence bands.

We can estimate the polynomial that we want:

$$\widehat{m(x_0)} = argmin_\gamma \sum_i W\left(\frac{x_i - x_0}{h}\right)(y_i - \gamma_0 - \gamma_1(x_i - x_0) ... - \gamma_p(x_i - x_0)$$

npregress

By default estimates a local-linear regression which solves:

$$\min_{\gamma} \sum_{i=1}^{n} \left(y_i - \gamma_0 - \gamma_1'(x_i - x)\right)^2 K(x_i, x, h)$$

where $\gamma = (\gamma_0, \gamma_1')'$ and $K(x_i, x, h)$ is the product of the kernels for each covariate.

$$K(x_i, x, h) = \prod_{j=1}^{k} K_j(x_{ij}, x_j, h_j)$$

Notice that the solution to the previous problem is given by:

$$\hat{\gamma} = (Z'WZ)^{-1}Z'Wy$$

where

- $\hat{\gamma} = (\hat{\gamma}_0, \hat{\gamma}_1{}')'$.
- Z is an $nx(k+1)$ matrix.
- W is an nxn diagonal matrix with the kernel weights as elements.

Importantly, if $(Z'WZ)$ is not full rank, the parameter γ is not identified. The observations for which this is true are dropped from the estimation sample.

npregress

```
Bandwidth
```

	Mean	Effect
fines	.5631079	.924924

```
Local-linear regression                    Number of obs     =        500
Kernel   : epanechnikov                    E(Kernel obs)     =        282
Bandwidth: cross-validation                R-squared         =     0.4380
```

citations	Estimate
Mean	
citations	22.33999
Effect	
fines	-7.692388

Note: Effect estimates are averages of derivatives.
Note: You may compute standard errors using **vce(bootstrap)** or **reps()**.

npregress

We can have different bandwidths for the mean and the derivative.

Bandwidth

	Mean	Effect
fines	.5631079	.924924

```
Local-linear regression                    Number of obs     =        500
Kernel   : epanechnikov                    E(Kernel obs)     =        282
Bandwidth: cross-validation                R-squared         =     0.4380
```

citations	Estimate
Mean	
citations	22.33999
Effect	
fines	-7.692388

Note: Effect estimates are averages of derivatives.
Note: You may compute standard errors using vce(bootstrap) or reps().

npregress

Bandwidth

	Mean	Effect
fines	.5631079	.924924

Local-linear regression
Kernel : **epanechnikov**
Bandwidth: **cross-validation**

Number of obs = **500**
E(Kernel obs) = **282**
R-squared = **0.4380**

citations	Estimate
Mean	
citations	**22.33999**
Effect	
fines	**-7.692388**

Note: Effect estimates are averages of derivatives.
Note: You may compute standard errors using **vce(bootstrap)** or **reps()**.

npregress

The output reports averages of the mean function and the effects of the mean function. An average effect is 1) an average marginal effect or 2) the mean of contrasts for discrete covariates.

```
Bandwidth

             |      Mean     Effect
-------------+----------------------
       fines |  .5631079    .924924


Local-linear regression                    Number of obs      =        500
Kernel   : epanechnikov                    E(Kernel obs)      =        282
Bandwidth: cross-validation                R-squared          =     0.4380
-------------+-----------------------------------------------------------------
   citations |   Estimate
-------------+-----------------------------------------------------------------
Mean         |
   citations |   22.33999
-------------+-----------------------------------------------------------------
Effect       |
       fines |  -7.692388
-------------------------------------------------------------------------------
Note: Effect estimates are averages of derivatives.
Note: You may compute standard errors using vce(bootstrap) or reps().
```

In Class Exercice

Using the data provided in the TA2 do file:

- Fit a nonparametric local-linear model for y with x1, x2 and a as covariates.
- Get the expected value of y when a=1, x1= 2 and x2 = 5.
- Get the effect of going from a=1 to a=2 when x1 = 2 and x2 =5.
- Get the expected value of y when x_1 is on the range from 1 to 4 for all possible values of a and $x_2 = 5$. Plot for all values of a.
- Repeat but for x_2 being now 2,5 or 8. Plot the results in the same figure.
- Compute the effect of moving from a=1 to a=3 for the previous range of x1 and x2. Plot the results.

Others

We can also use npregress for series estimation.