Handout 4: Nonparametric Local Regression and Semiparametric estimation.

Master in Data Science for Decision Making
Barcelona School of Economics

Laura Mayoral

IAE and BSE

Barcelona, Winter 2026

1. Introduction

■ Goal: estimate the relationship between y and x without imposing a functional form.

→ the same, in more technical words:

Nonparametric estimation of the conditional expectation.

■ The conditional expectation of Y conditional on X (a univariate variable) at $X = x_0$:

$$E[Y|X = x_0] = m(x_0)$$

m(.) is not specified.

- In this lecture we will develop nonparametric regression techniques

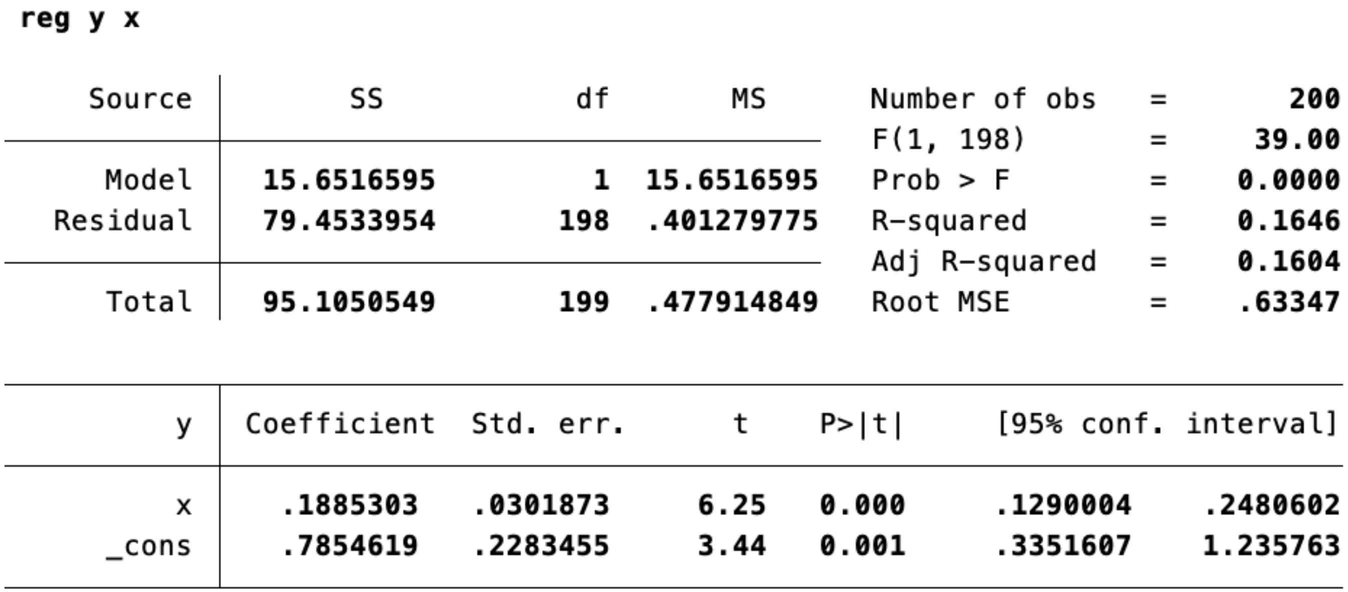
- We will start by considering that X is a scalar variable (recall the curse of dimensionality)

- These nonparametric methods are local averaging methods: estimates are obtained by cutting the data into ever smaller slices as $N \to \infty$ and estimating local behavior within each slice.

- In a nutshell: for each point x_0 those estimators are weighted (typically, kernel weights) local (=in a neighbor of x_0) averages of values of y

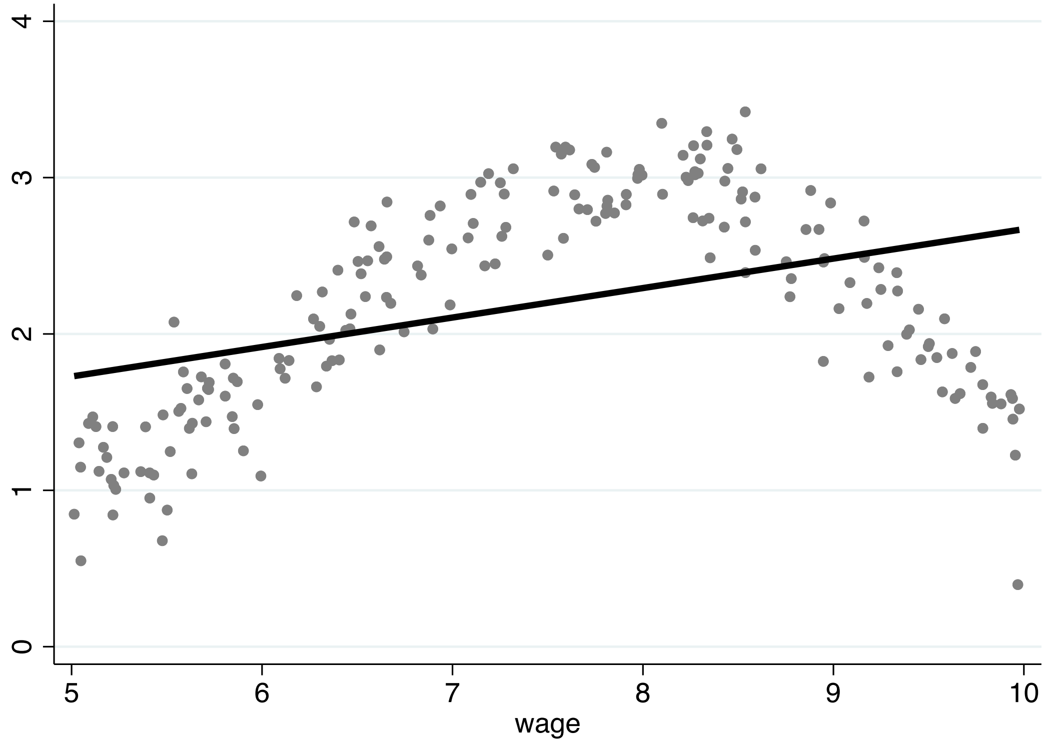
An introductory example

- Consider the relationship between hours worked per day and hourly wage (simulated data)
- We run an OLS regression and get a very positive relationship

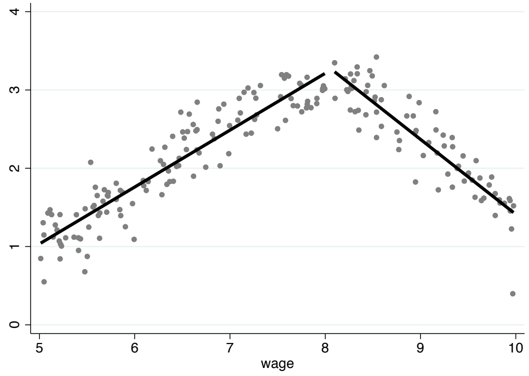


```
reg y x

      Source |       SS           df       MS      Number of obs   =       200
-------------+----------------------------------   F(1, 198)       =     39.00
       Model |  15.6516595         1  15.6516595   Prob > F        =    0.0000
    Residual |  79.4533954       198  .401279775   R-squared       =    0.1646
-------------+----------------------------------   Adj R-squared   =    0.1604
       Total |  95.1050549       199  .477914849   Root MSE        =    .63347

------------------------------------------------------------------------------
           y | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
           x |   .1885303   .0301873     6.25   0.000     .1290004    .2480602
       _cons |   .7854619   .2283455     3.44   0.001     .3351607    1.235763
------------------------------------------------------------------------------
```

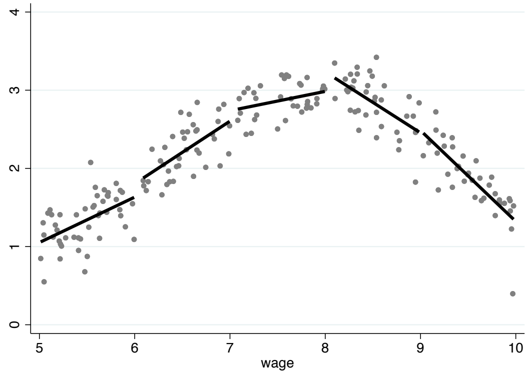
- However, plot the data: highly nonlinear relationship



- We could estimate two lines, one for the increasing part of the relationship and one for the decreasing one:



- Or we could even consider more regression lines, (i.e., a smaller "bandwidth")



- The previous methods will work if we know the breaking points (we're imposing them when running the OLS regressions).

- Nonparametric methods share a similar spirit: they are local averaging methods.

- No need to impose any breaking points as in this example!

- estimates are obtained by cutting the data into ever smaller slices as $N \to \infty$ and estimating local behavior within each slice.

- Parametric versus Nonparametric methods:

- Asymptotic properties are quite different

  - Lower convergence rates: because of local averages (less than less than N observations in estimating each slice)

  - In simplest cases still asymptotically normally distributed;

  - Due to lower convergence rates, biases appear

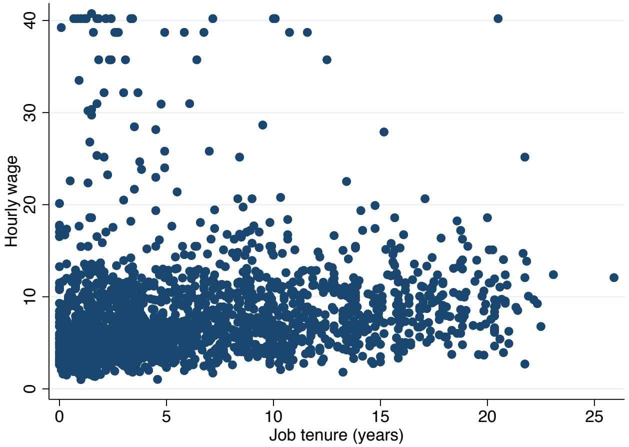
- Things become in general a bit “uglier” and properties are a bit ”less nice” than in parametric estimation, so be a bit patient!

1.2. Some simple visualization tools

Scatterplot

- Before we begin with the complicated stuff, let's always look at the data first!

- Consider this example: The relation between tenure on the job and hourly wage.
- DATA (example STATA: sysuse nlsw88)
- Simplest visualization tool: scatterplot



- What can you say about the relationship between wage and tenure by looking at this graph?

- Not much!

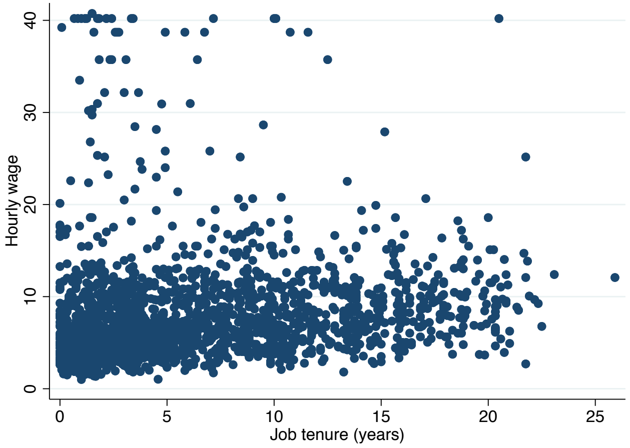
- scatterplots are not very useful for large data sets

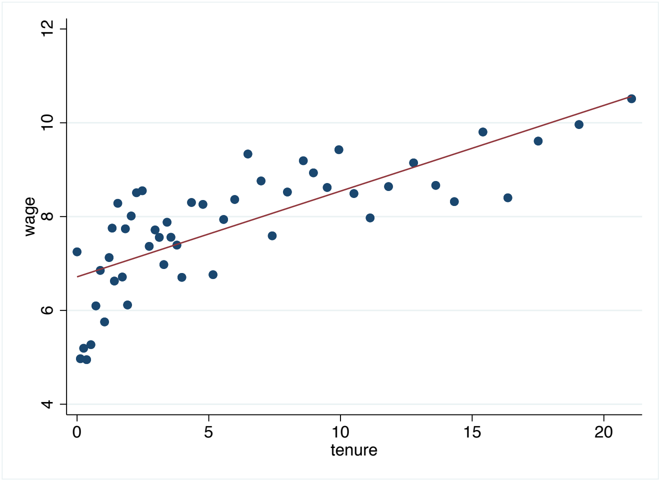
A better way of plotting the data: binned scatter plots

- If a lot of data points: scatter plots are not very useful (clouds of millions of points! impossible to see anything)

- Binned scatters: very useful visualization tools, particularly for large datasets

- Compare the scatter and the binned scatter plot (on same data)





- The second graph is much more informative that the first one about the shape of the conditional expectation.

- From the second graph, you can easily see that

  - There's a positive relationship between tenure and wage

  - This relationship seems to be pretty linear

- What's the magic?

- The second graph is much more informative that the first one about the shape of the conditional expectation.

- From the second graph, you can easily see that

  - There’s a positive relationship between tenure and wage

  - This relationship seems to be pretty linear

- What’s the magic?

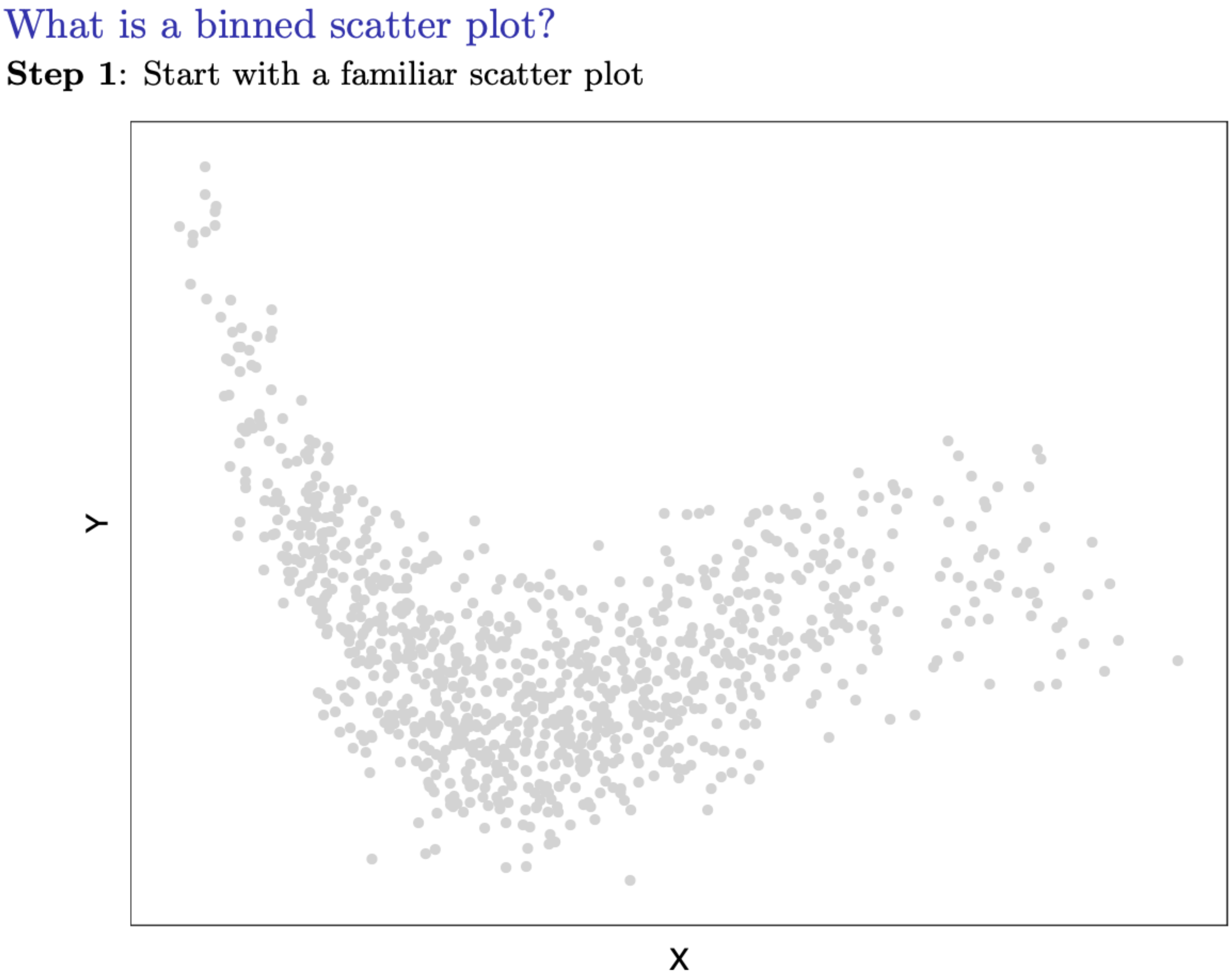
Binned scatter plots are visual, simple, nonparametric estimators of the conditional expectation.

- How they work (from STATA help)

STATA command: Binscatter

- groups the x-axis variable into equal-sized bins (number of bins to be determined by you, default empirical ventiles)
- computes the mean of the x-axis and y-axis variables within each bin (median is also an option)
- then creates a scatterplot of these data points.
- The result is a non-parametric visualization of the conditional expectation function.

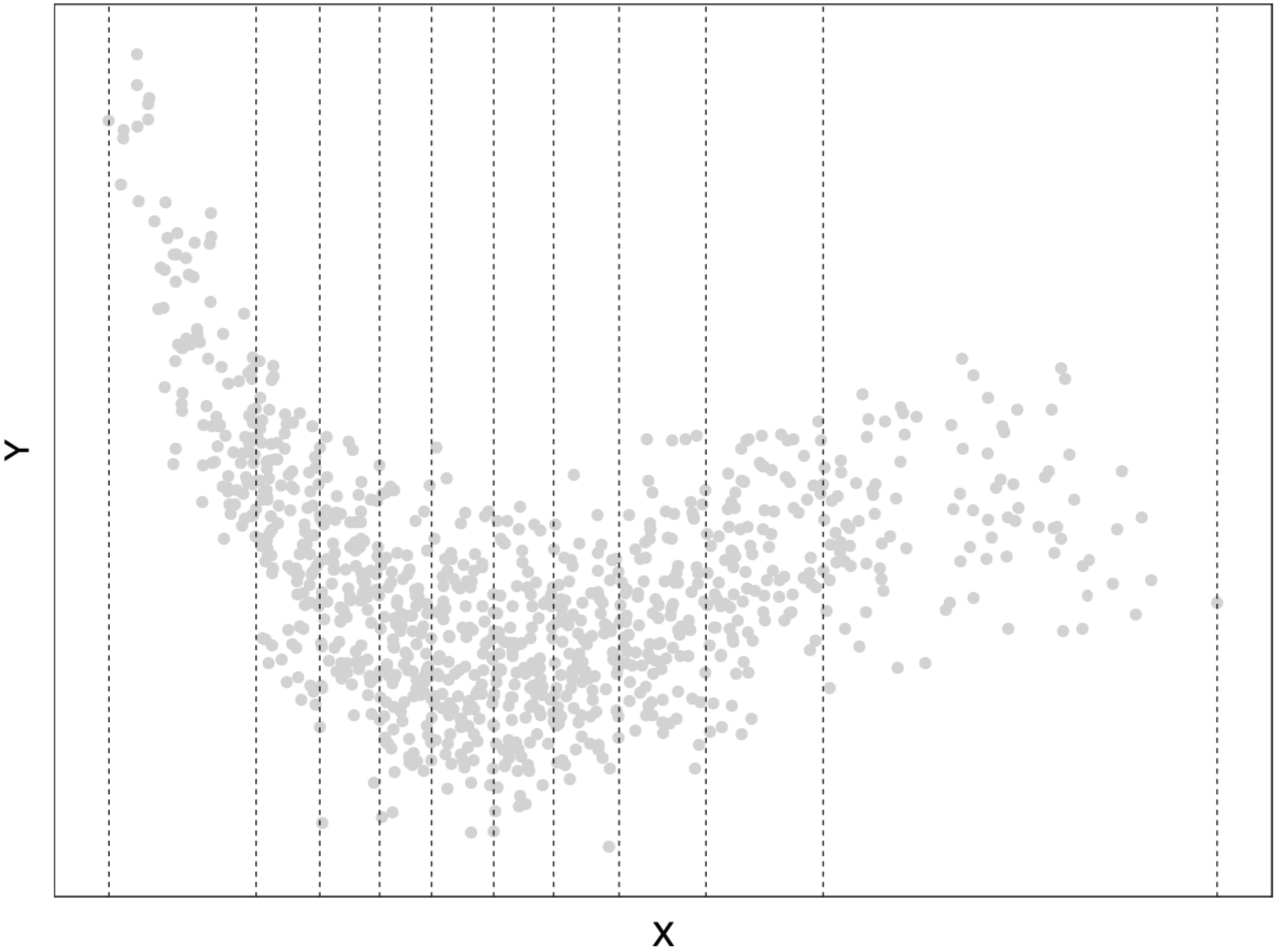
Let’s see this graphically



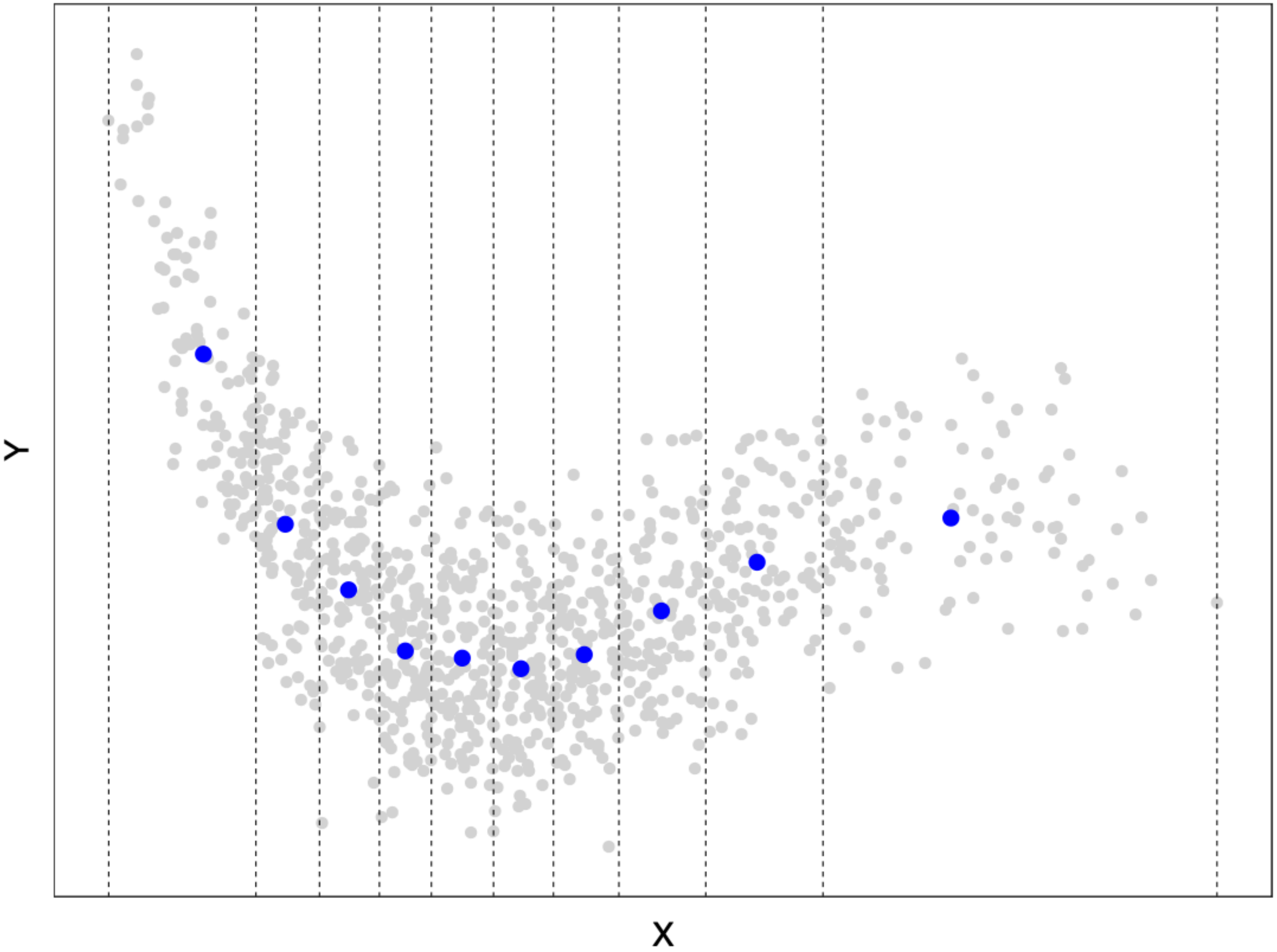
What is a binned scatter plot?

Step 1: Start with a familiar scatter plot

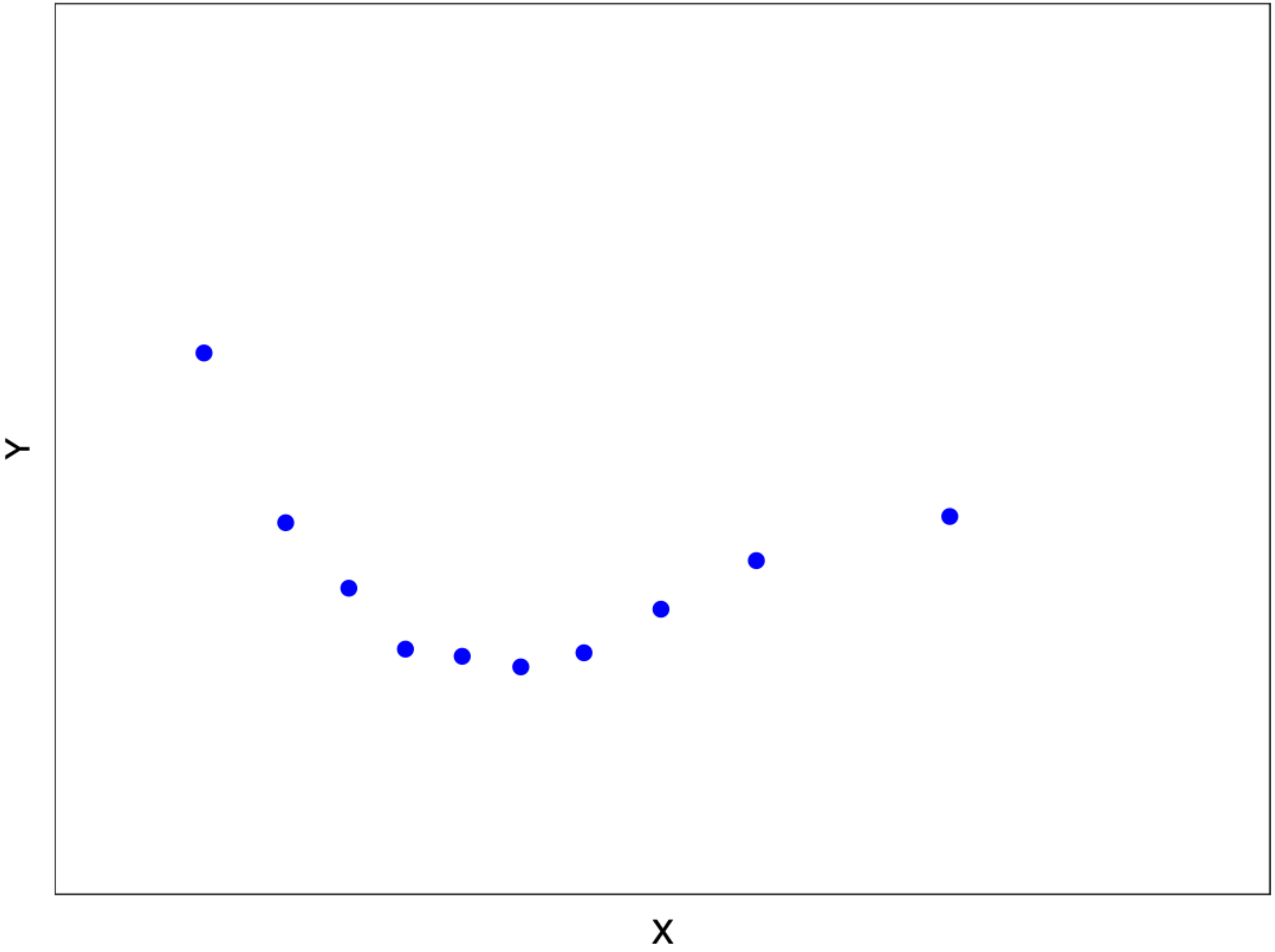
Step 2: Partition the support of X into bins



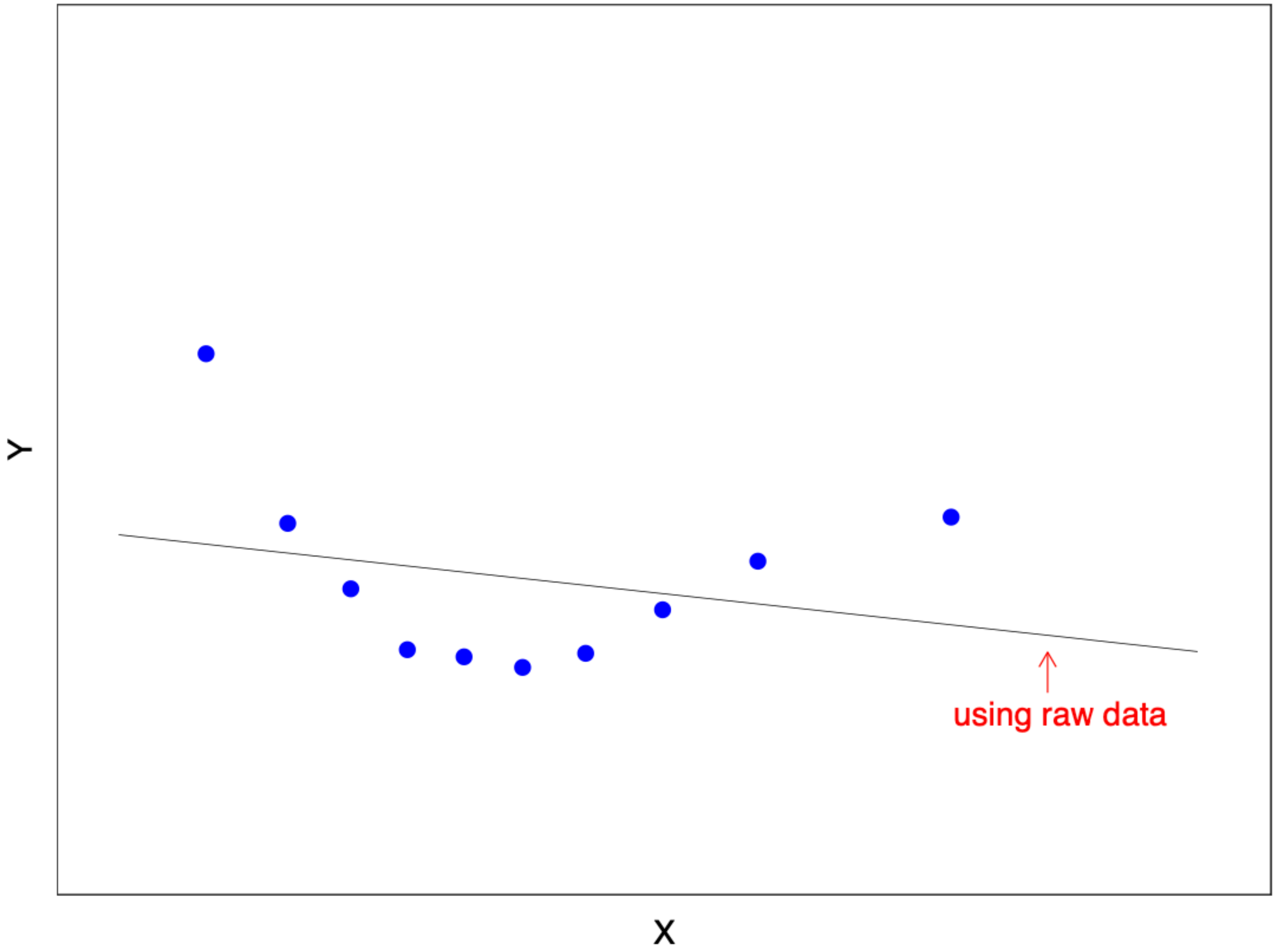
Step 3: Find the average Y in each bin



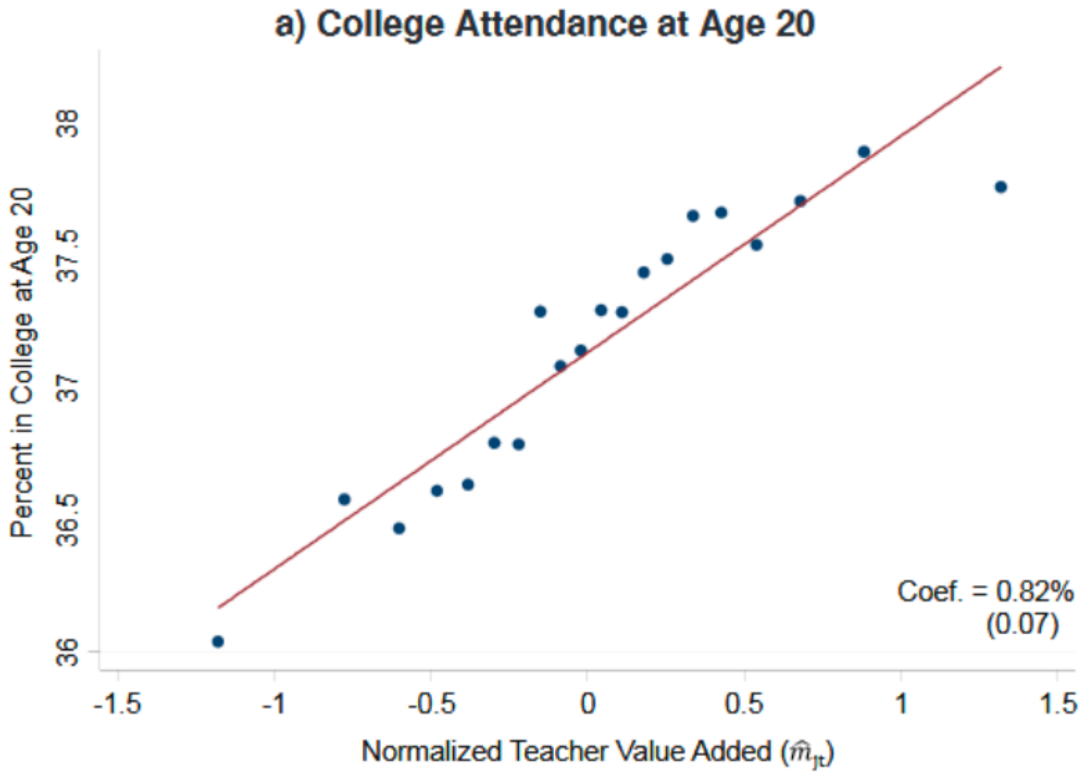
Step 4: Plot only bin means



Step 5: Add a polynomial fit to raw data



Typical Example: Chetty, Friedman and Rockoff (2014, AER)



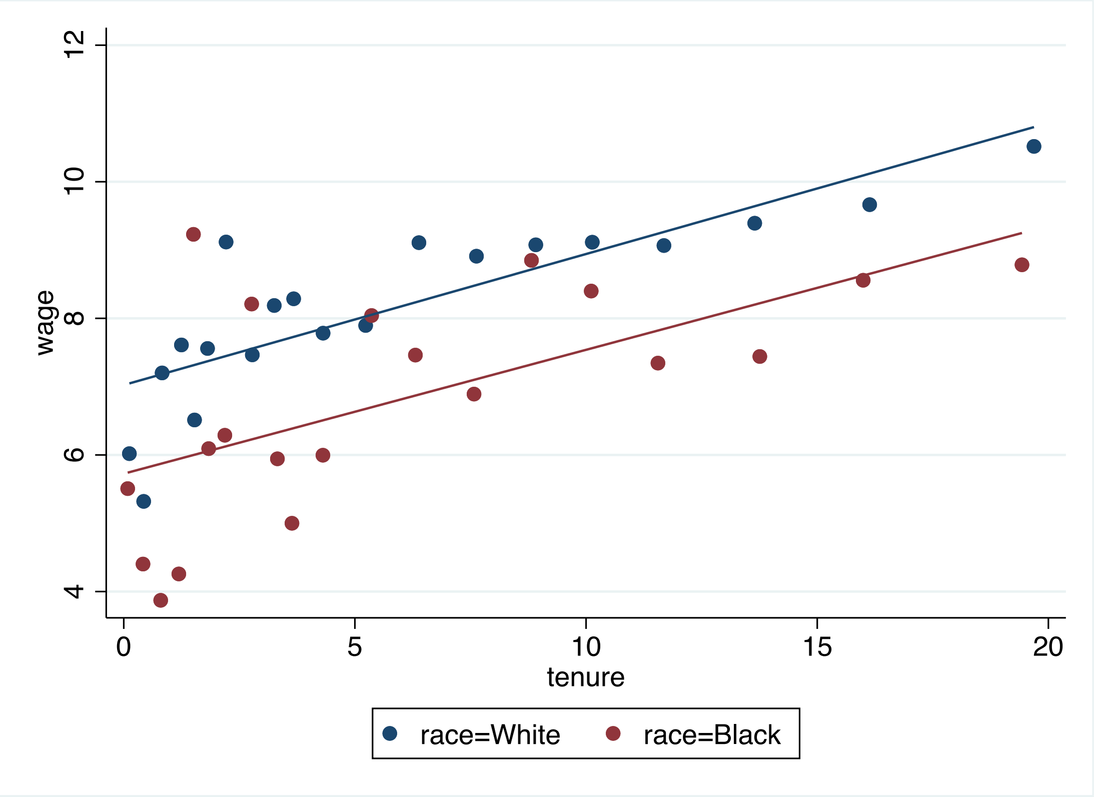
Note: $n = 4,170,905$ with # of bins $J = 20$

Additional options of the binscatter command Plot the data conditioning by different values of other variable

1. You can plot the data for values of other variable.

For instance, by race

```
binscatter wage tenure, by(race) nq(50)
```

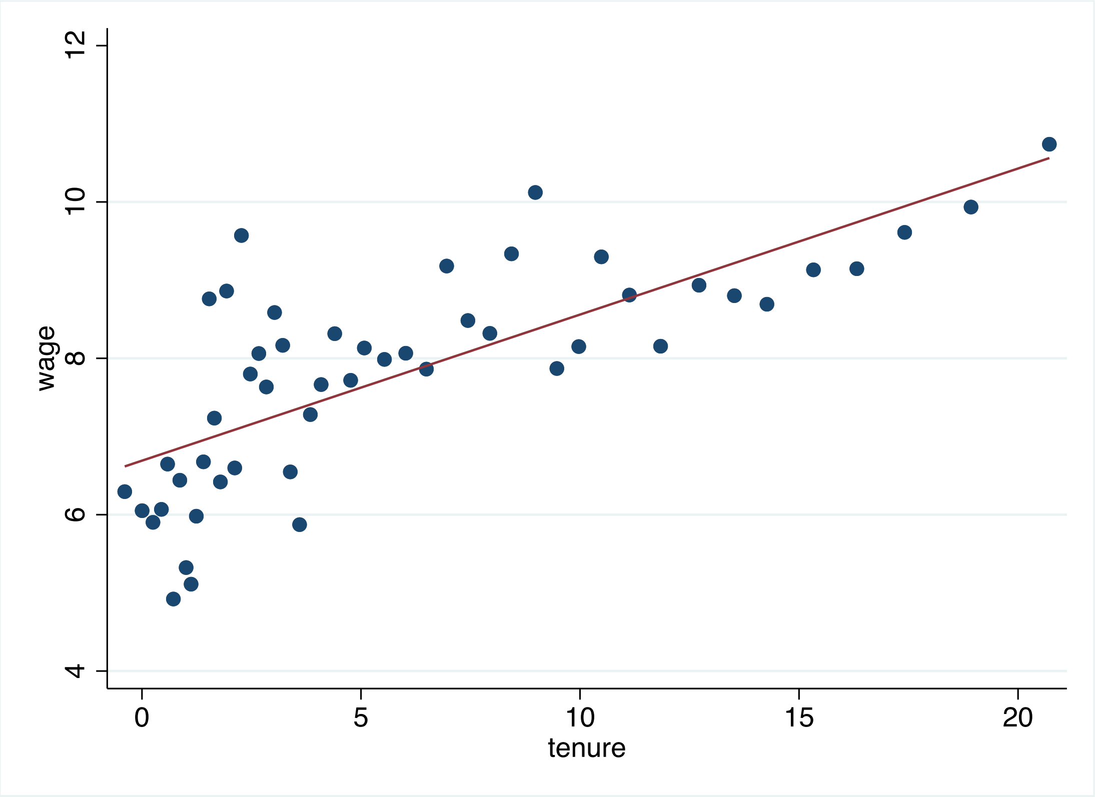


Control for other variables

2. You can control for other variables that you might think are relevant.

Control for age.

binscatter wage tenure, control(age) nq(50)



How does binscatter deal with control variables?

- Method inspired by a famous theorem in regression analysis: The Frisch-Waugh-Lowell Theorem

- Binscatter residualizes the x-variable and y-variables on the specified controls before binning and plotting.

- That is,

  - regress y/z, save residuals, e_1, add mean of y to e_1, obtain e_1'
  - regress x/z, sav residuals, e_2, add mean of x to e_2, obtain e_2'
  - Plot e_1' on e_2'

- This is in fact trickier than it looks and not so "safe": only valid if conditional expectation is linear

An improved approach to binned scatterplots: Cattaneo et al., 2024

- A very recent paper improves on traditional methodology (published May 2024!): Cattaneo et al (2024)

- The "traditional" approach of residualizing first the data only justified when the conditional expectation is linear.

- Otherwise: don't do it!

- This paper also provides:

  - Ways of doing inference

  - Optimal binning selection

- We'll go back to this paper when we study semiparametric methods (partially linear model)

- But you can install the stata package to play with it in the meantime:

STATA package: binsreg

STATA code to generate this example:

- Load data in stata memory:

```
sysuse nlsw88
keep if inrange(age,35,44) & inrange(race,1,2)
keep if inrange(age,35,44) & inrange(race,1,2)
scatter wage tenure, graphregion(color(white)) lwidth(thick)
binscatter wage tenure, nq(50)
binscatter wage tenure, control(age) nq(50)
binscatter wage tenure,by(race) nq(50)
```

Takeaways

- Always start by plotting your data

- Binned scatterplots are very useful tools, particularly when there are a lot of data points

  - Visual and quick estimator of the conditional expectation

  - Quite flexible stata command, allows to eliminate impact of other variables (linearly) (but notice the limitations of this, only valid under linearity!)

- But binscatter is not enough! (no inference, a bit too crude...)

- New binned scatter technique: Cattaneo et al, (2024) –to be reviewed soon.

Overview of the handout

- The remaining of this handout: Different approaches to carry out nonparametric regression.

- Different methods: Kernel local (constant) regression; Local linear/polynomial regression; Lowess, . . .

- Intuition is simple, technical stuff becomes complicated

- We will look at

1) Intuition;

2) implementation;

3) differences across the methods; etc

4) stata tips (more on this in the TA session);

Roadmap of this handout

1. Introduction: Nonparametric Local Regression;

    1.2. Some simple visualization tools

2. Local Weighted Averages

3. Kernel Local Regression: implementation, properties

4. Local Linear Regression

5. K-Nearest Neighbor

6. Lowess

7. Semiparametric regression

2. A bif of intuition: Local Weighted Averages

- Model:

$$y_i = m(x_i) + \epsilon_i, \quad i = 1, \dots, N, \quad \epsilon_i \overset{iid}{\sim} (0, \sigma_\epsilon^2). \tag{1}$$

and $E(\epsilon|x) = 0$.

- Under these assumptions, the conditional expectation is

$$m(x_0) = E(y|x = x_0). \tag{2}$$

- Problem:

m(.) unspecified → use nonparametric methods to estimate it at a point x_0

A bit of intuition about local average estimators

- Suppose that at x_0, there are multiple observations on y, say N_0 observations.

- A simple estimator for $m(x_0)$ is the sample average of these N_0 values of y.

$$\hat{m}(x_0) = \sum_{i=1}^{N_0} w_i y_i$$

where $w_i = 1/N_0$ if $x = x_0$ and 0 otherwise.

- Notice that (for fixed x_0):

$$\bar{m}(x_0) \sim \left(m(x_0), \frac{\sigma^2}{N_0} \right), \tag{3}$$

- Why? it is the average of N_0 observations that are i.i.d with mean $m(x_0)$ and variance σ_ϵ^2.

- The estimator $\bar{m}(x_0)$ is unbiased but not consistent (in general)

  - Why? Consistency requires $N_0 \to \infty$ as $N \to \infty$, so that $V[\bar{m}(x_0)] \to 0$.

  - But N_0 can be really small, particularly for continuous variables! (most likely, just one observation of y)

Then:

- The Problem of this approach: not enough observations to average (N_0 can be too small, it can even be 1 for continuous variables even with a huge sample!)

- A Solution: consider averages of y when x is close to x_0, (in addition to when x exactly equals x_0).

Local weighted average estimator:

- a weighted average of the dependent variable in a neighborhood of x_0.

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

where the weights $w(x_i, x_0, h)$ sum to 1 and vary with :

  - the sample values of the regressors, x_i
  - the evaluation point x_0
  - the value of h, i.e., the length of the window around x_0

Note: The OLS estimator has a “similar” structure

- This estimator is not “that different” from those you’ve used in the past!

- Recall that the OLS estimator is also a weighted average of y_i, since some algebra yields

$$\hat{m}_{OLS}(x_0) = \sum_{i=1}^{N} \frac{1}{N} + \frac{(x_0 - \bar{x})(x_i - \bar{x})}{\sum_j (x_j - \bar{x})^2} y_i$$

- The OLS weights are different though:

  - Local regression uses weights that are decreasing as x_i gets far away from x_0 (if, for example, $x_i > x_0 > \bar{x}$)

  - OLS weights don’t verify this, in fact, weights can even increase with increasing distance from x_0

Back to the local weighted average estimator

- h: bandwidth parameter. Smaller values of h → smaller window → more weight being placed on those observations with x_i close to x_0.
- 2h: window width

- The most common weight functions are:

- 1. Kernel weights
- 2. Lowess
- 3. k-nearest neighbors

- Modus operandi: compute $\widehat{m(x_0)}$ at a variety of points of x_0 to obtain a regression curve.

3. Kernel regression: Nadaraya-Watson (NW) estimator

- Recall the Model:

$$y_i = m(x_i) + \epsilon_i, \quad i = 1, \ldots, N, \tag{4}$$

$$E(\epsilon|x) = 0,$$
$$E(\epsilon^2|x) = \sigma^2(x)$$

.

- Recall the Goal: Estimate of $m(x_0)$,

$$m(x_0) = E(y|x = x_0). \tag{5}$$

- Let's now analyze the case where we use Kernel weights

- Kernel regression is a weighted average estimator using kernel weights.

- Consider again the local weighted average estimator, where we compute the average of the y's in an interval of length $2h$ around x_0

$$\widehat{m(x_0)} = \frac{\sum_{i=1}^{N} 1(|\frac{x_i - x_0}{h}| < 1) y_i}{\sum_{i=1}^{N} 1(|\frac{x_i - x_0}{h}| < 1)}$$

- The numerator: sums the y's in the interval $(x_0 \pm h)$
- The denominator: gives the total number of y's that have been summed in the numerator

- Thus: the previous expression is an average of the y's with equal weights (weights are relative frequency of y in the window)

- Consider instead Kernel weights

- Why?

- non-constant weights

- give more weight to observations close to x_0

- Kernel Regression Estimator

$$\widehat{m(x_0)} = \frac{\sum_{i=1}^{N} K(\frac{x_i - x_0}{h}) y_i}{\sum_{i=1}^{N} K(\frac{x_i - x_0}{h})}$$

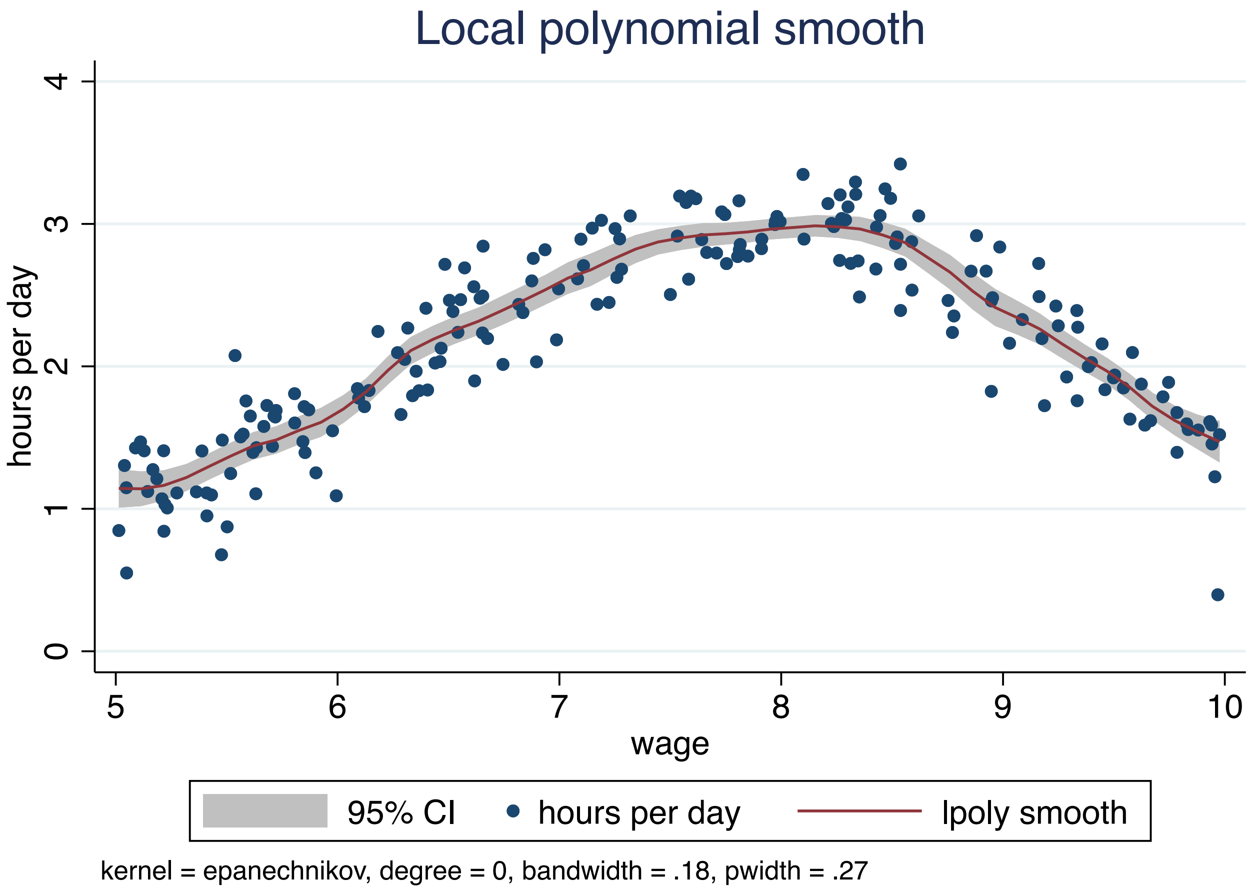
(also called Nadaraya-Watson estimator)

- similar Kernels as before: Gaussian, Epanechnikov, etc.

Example: Nadaraya-Watson estimator for the hours worked /wage problem

(stata defaults for h, kernel... –we'll learn about them)

lpoly y x , ci msize(small) graphregion(color(white))



Implementation of the NW estimator

1. Kernel choice

- Kernel choice: $MISE(h^*)$ is minimized by the Epanichnikov Kernel (as before)

- but small differences across kernels for optimal h^*

- Choice of bandwidth is much more important than choice of kernel

Implementation of the NW estimator, II

2. Bandwidth choice

- Optimal bandwidth: recall the tradeoff between bias&variance in the choice of h.

- Optimal bandwidth: trades off bias (minimized with small bandwidth) and variance (minimized with large bandwidth)
- Recall the trade-off:

  - 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.

- The optimal bandwidth balances the trade-off between increased bias and decreased variance, using squared error loss.

- Variance=$O((Nh)^{-1})$; bias=$O(h^2)$

- Theory just says that the optimal bandwidth (=the one that minimizes MISE) for kernel regression is $O(N^{-0.2})$ (but this is useless for choosing h in applications). Why this value: makes the squared bias and the variance of the same order of magnitude.

- In practice: plug-in estimator of the optimal h using MISE(h) is complicated now (estimation of the plug-in estimation requires estimation of $m''(x)$, second derivative of conditional expectation which is difficult to estimate).

- Alternative: Cross-validation, computationally intensive, but easier to implement

Choosing the bandwidth: Cross-validation

- Cross-validation is a popular technique for many prediction problems

- Cross-validation, in general:

- Construct prediction models that perform well out of sample

- Simple idea:

  - we split the data in two sets: training set and validation set

  - Use the data in the training set to construct the estimator.

  - Using this estimator, predict the "out of sample" observations, i.e., the obs. in the "validation set", calculate the error.

  - Choose the estimator with best out of sample performance

Why leaving some observations out?

- Avoid overfitting:
  - an estimator that is very good for the in-sample data but can perform badly for non-seen observations
  - why is that? because in a dataset there's always noise. If we perfectly fit that data, we fit both the "signal" (what really matters in the data) AND the noise, something that is pure random variation.
  - Since the noise changes in every realization of the data, a model that fits very well a dataset can perform badly out of sample

- Cross validation, in particular:

- Goal: use cross-validation to choose a value of h that yields a good estimate $m(x)$

- Idea

  - For each observation i, compute an estimator m_i using cross-validation i.e., using a “training sample” to compute the estimator

  - ...and a validation sample only used to compute out of sample prediction error

  - Then, choose h that yields smallest MSE.

- How it works (a bit simplified):

- 1. For each i, define the training sample as all the observations except obs. i; validation sample: observation i

- 2. The estimator leaving i out is given by

$$\hat{m}_{-i}(h, x_i) = \sum_{j \neq i} w_{j,h} y_j / \sum_{j \neq i} w_{j,h}$$

- 3. Compute CV(h) (very similar to the MSE(h))

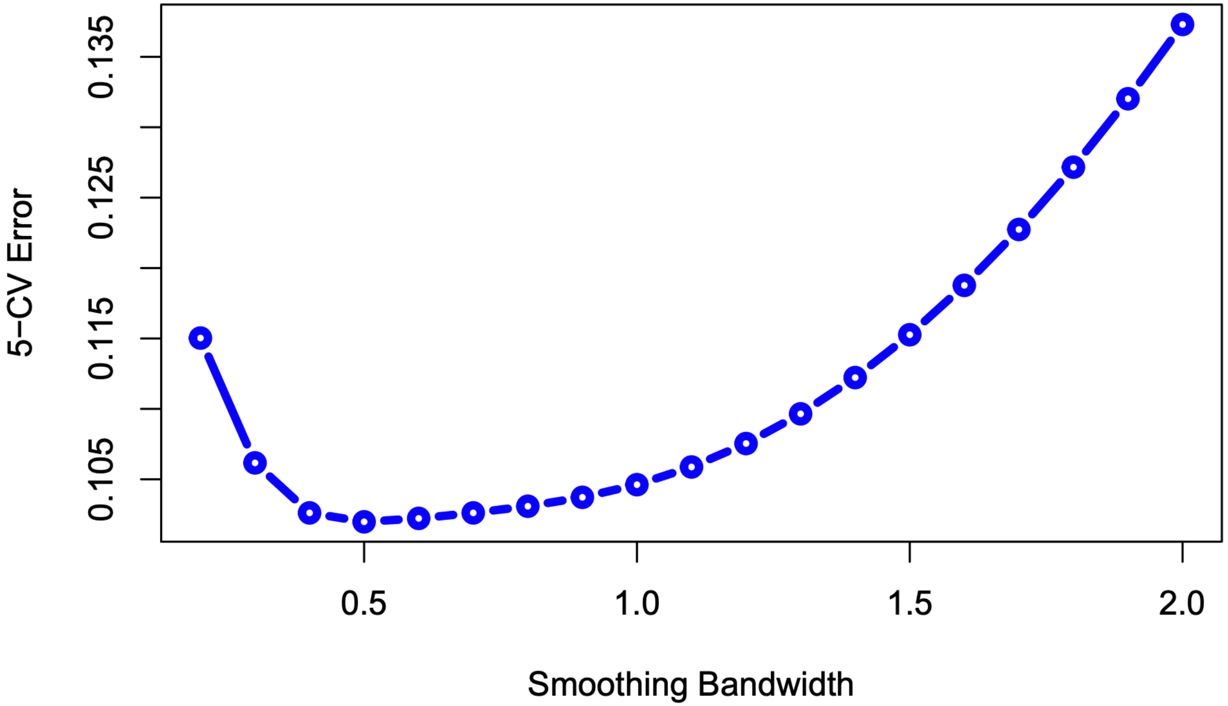
$$CV(h) = \sum_{i=1}^{n} \left(y_i - \hat{m}_{-i}(x_i)\right)^2 \pi(x_i), \tag{6}$$

- $\pi(x_i)$: weights introduced to potentially downweight the end points, to prevent those points to receive too much attention (local weighted estimates can be quite highly biased at the end points)

- 4. h^*_{cv} is chosen as the value that minimizes the CV(h)

- 5. In practice CV(h) is computed over a range of values of h. Choose the value of h that makes it smallest.

- Properties of $\hat{h}_{cv}$: converges to h^* (optimal h), but slowly ($\approx$ low convergence rate)



Takeaways

- In Kernel regression, cross-validation tends to perform better than the plug-in estimator

- Logic of Cross-validation: choose the h that minimizes the (out of sample) mean prediction error

- Why leaving one observation out at a time?

  - nonparametric methods are very flexible, and if we consider the whole sample, we can get an almost “perfect fit”

  - $\Rightarrow$ Overfitting!

Statistical Properties of Kernel regression estimators

1. The Kernel regression estimator is consistent

- $\widehat{m_0}$ is consistent if some conditions on h and Nh hold

- Recall: these conditions are needed for developing the theory; not informative to choose the value of h in practice

- The estimator is consistent provided:

  - $h \to 0$: i.e., substantial weight is given only to x_i very close to x_0.

AND

$Nh \to \infty$: i.e., there's "many" x_i close to x_0 as $n \to \infty$, so that many observations are used in forming the weighted average.

2. The Kernel regression estimator is biased in finite samples

- It can be shown that

$$\widehat{m(x_0)} = m(x_0) + O(h^2)$$

- Asymptotically, the bias tends to zero under the assumptions above (i.e. if h tends to zero)

- However, the bias can be substantial in finite samples

- Particularly, at the end points (where few observations exist)

- When considering confidence intervals, the estimate is centered in the true value of m plus the bias!

3. The Kernel regression estimator is asymptotically normal

- Rate of converge: $\sqrt(Nh)$: smaller than the usual $\sqrt(N)$

- Asymptotic distribution (notice the bias!)

$$\sqrt{Nh}(\hat{m}(x_0) - m(x_0) - b(x_0)) \to N\left(0, \frac{\sigma_\epsilon^2}{f(x_0)} \int K(z)^2 dz\right) \tag{7}$$

- Notice that $f(x_0)$ appears in the denominator

- This implies that the variance term in is larger for small $f(x_0)$, i.e., when there're few 'x's in the neighborhood of x_0, which makes sense

Constructing Confidence Intervals

- Estimates of $m(x_0)$ typically are provided with CI

- How can we compute them?

1. Use the asymptotic distribution above ignoring the bias. Then:

$$m(x_0) \in \hat{m}(x_0) \pm 1.96 \sqrt{\frac{1}{Nh} \frac{\hat{\sigma}_\epsilon^2}{\hat{f}(x_0)} \int K(z)^2 dz}$$

- But two problems

Problem 1 Convergence to the normal distribution is slow (recall the lower convergence rates)

Problem 2 Forgetting the bias means that the CI are not centered correctly!

- Solutions.

Problem 1. Don't use the asymptotic distribution, instead use bootstrap (i.e., a method that approximates the finite sample distribution)

Problem 2. Reduce the bias:

a) Undersmoothing

b) using higher order Kernels (Fourth-order, Gaussian Fourth-order quartic): the bias when these kernels are employed are $O(h^4)$

c) Use alternative methods that are less biased: Local polynomial regression, Lowess . . . (smaller bias)

For instance, you can use bootstrad AND undersmoothing

Trimming

- Recall the definition of the NW estimator:

$$\widehat{m(x_0)} = \frac{\sum_{i=1}^{N} K(\frac{x_i - x_0}{h}) y_i}{\sum_{i=1}^{N} K(\frac{x_i - x_0}{h})}$$

- Notice that the denominator is $\hat{f}(x_0)$, the kernel density estimator.

- Problem : For some x_i, $f(x_i)$ can be very small (i.e., values that are unlikely). Since the estimate of the density appears in the denominator, this can lead to a very large value (in abs. value) of $\hat{m}(x_i)$.

- Such problems are most likely to occur in the tails of the distribution.

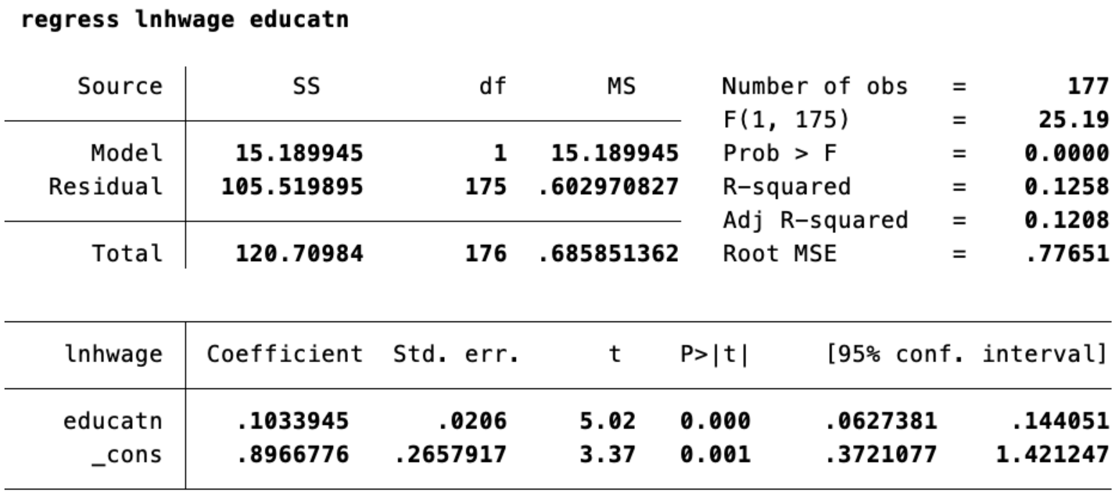
- Trimming: eliminates or greatly downweights all points with $f(x_i) < b$, say, where $b \to 0$ as $N \to \infty$.
  - For nonparametric estimation one can just focus on estimation of $m(x_i)$ for more central values of x_i,
  - However, the semiparametric methods of Section 9.7 can entail computation of $m(x_i)$ at all values of x_i, in which case trimming is typically employed.

Example

- DATA: PSID Individual Level Final Release 1993 data, (www.isr.umich then choose Data Center)

- Relation between years of completed education and (log of) wages
- Females in their 30's
- Data from Cameron and Trivedi

- OLS regression:
- highly significant role of education;

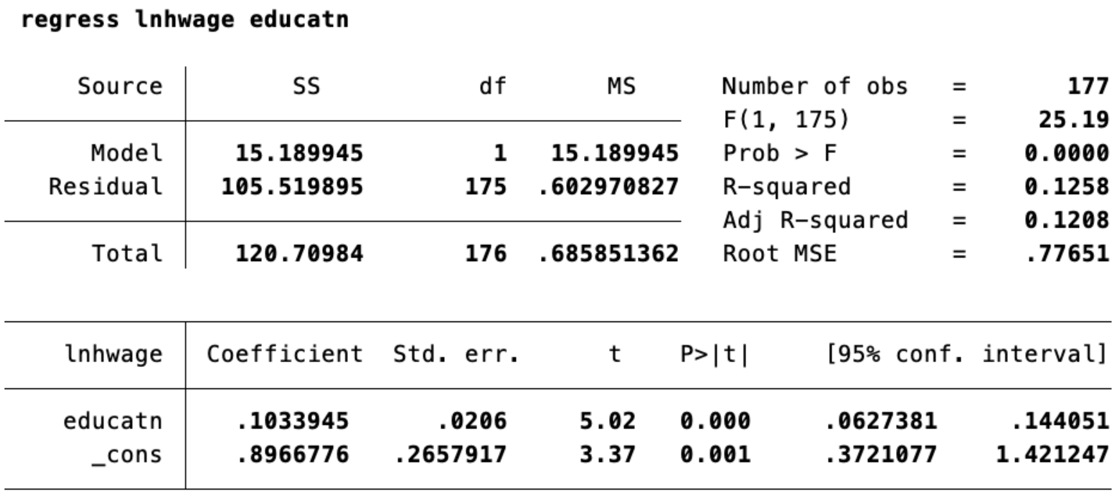


```
regress lnhwage educatn

      Source |       SS           df       MS      Number of obs   =       177
-------------+----------------------------------   F(1, 175)       =     25.19
       Model |  15.189945         1   15.189945    Prob > F        =    0.0000
    Residual | 105.519895       175  .602970827    R-squared       =    0.1258
-------------+----------------------------------   Adj R-squared   =    0.1208
       Total |  120.70984       176  .685851362    Root MSE        =    .77651

------------------------------------------------------------------------------
     lnhwage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
     educatn |   .1033945      .0206     5.02   0.000     .0627381     .144051
       _cons |   .8966776   .2657917     3.37   0.001     .3721077    1.421247
------------------------------------------------------------------------------
```

- interpretation : marginal effect

- OLS regression:

- highly significant role of education;



```
regress lnhwage educatn

      Source |       SS           df       MS      Number of obs   =       177
-------------+----------------------------------   F(1, 175)       =     25.19
       Model |  15.189945         1   15.189945   Prob > F        =    0.0000
    Residual | 105.519895       175  .602970827   R-squared       =    0.1258
-------------+----------------------------------   Adj R-squared   =    0.1208
       Total |  120.70984       176  .685851362   Root MSE        =    .77651

------------------------------------------------------------------------------
     lnhwage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
     educatn |   .1033945      .0206     5.02   0.000     .0627381     .144051
       _cons |   .8966776   .2657917     3.37   0.001     .3721077    1.421247
------------------------------------------------------------------------------
```

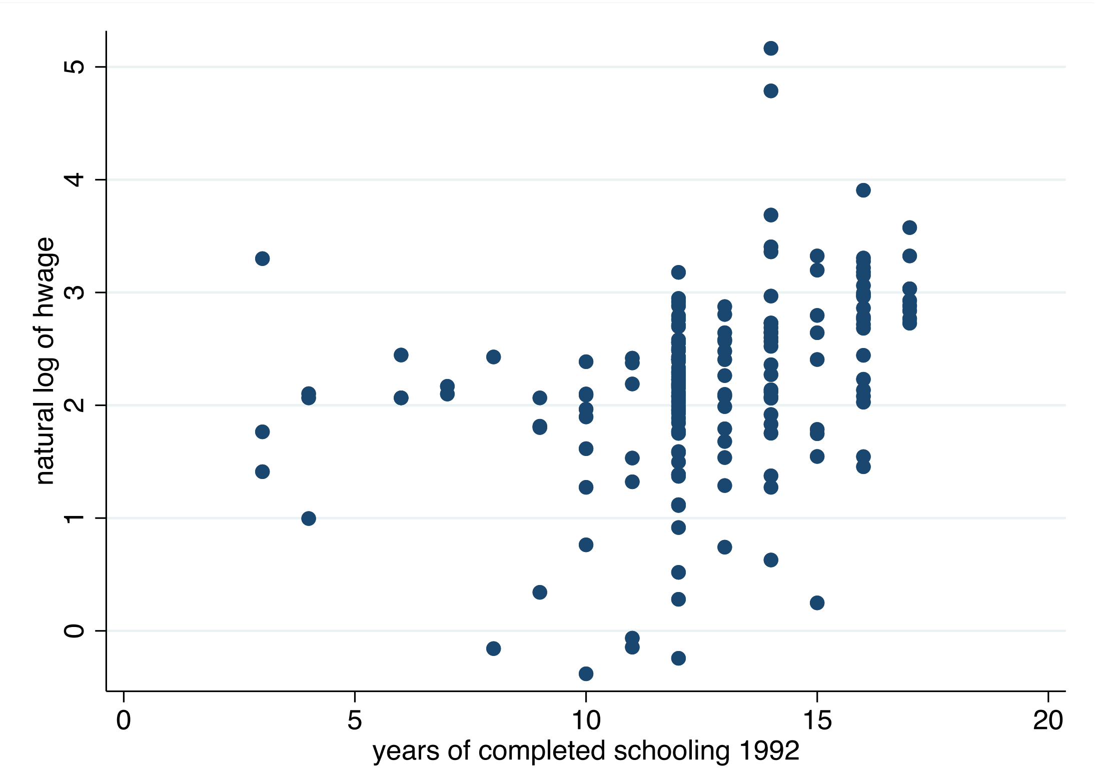
- interpretation : marginal effect

an increase in one year of education increases by 10% hourly wage.

But...is the linearity assumption reasonable?

Let's plot the data (scatter plot)

```
twoway scatter lnhwage educatn, graphregion(color(white))
```



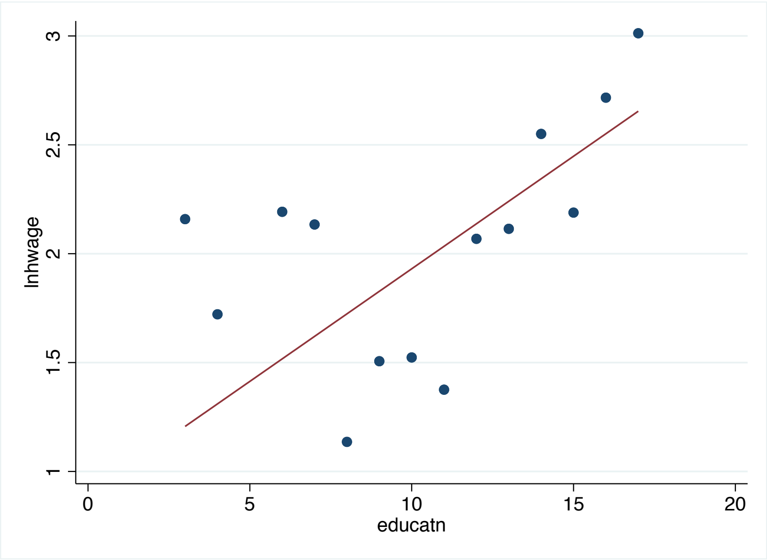
Binned scatter plot

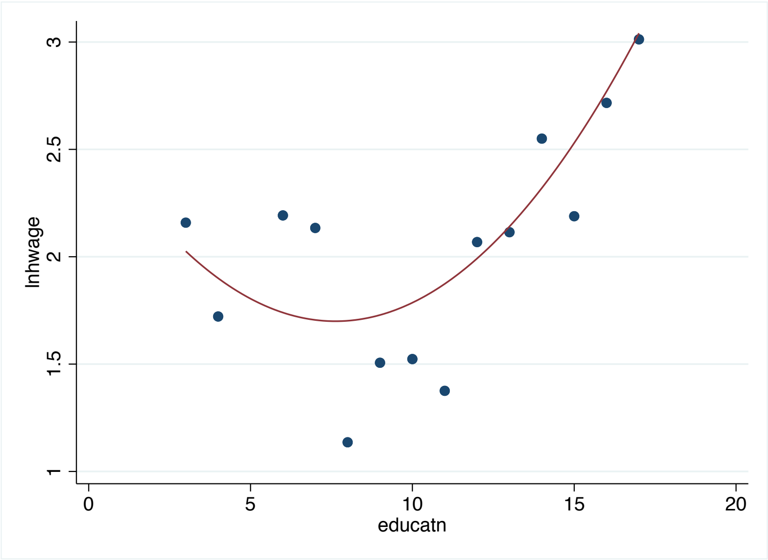
- STATA: binscatter command

```
binscatter lnhwage educatn, nq(20)
```

```
binscatter lnhwage educatn, nq(20) line(qfit)
```

(first graph imposes a linear fit on the data, second is more flexible, allows for a quadratic one)





Nonparametric regression in STATA

The lpoly and npregress commands

- STATA has several commands to do nonparametric regression: lpoly, npregress (the latter has more options)

- lpoly: Kernel-weighted local or polynomial smoothing
- Less options than npregress
- Very easy to use

- npregress kernel:

- From Stata 15 onwards: a new command, npregress

- Determines bandwidth by cross-validation whereas lpoly uses plug-in value

- Evaluates at each x_i value (whereas lpoly default is to evaluate at 50 equally spaced values)

- For local linear, computes partial effects.

- Can use margins and marginsplot for plots and average partial effects.

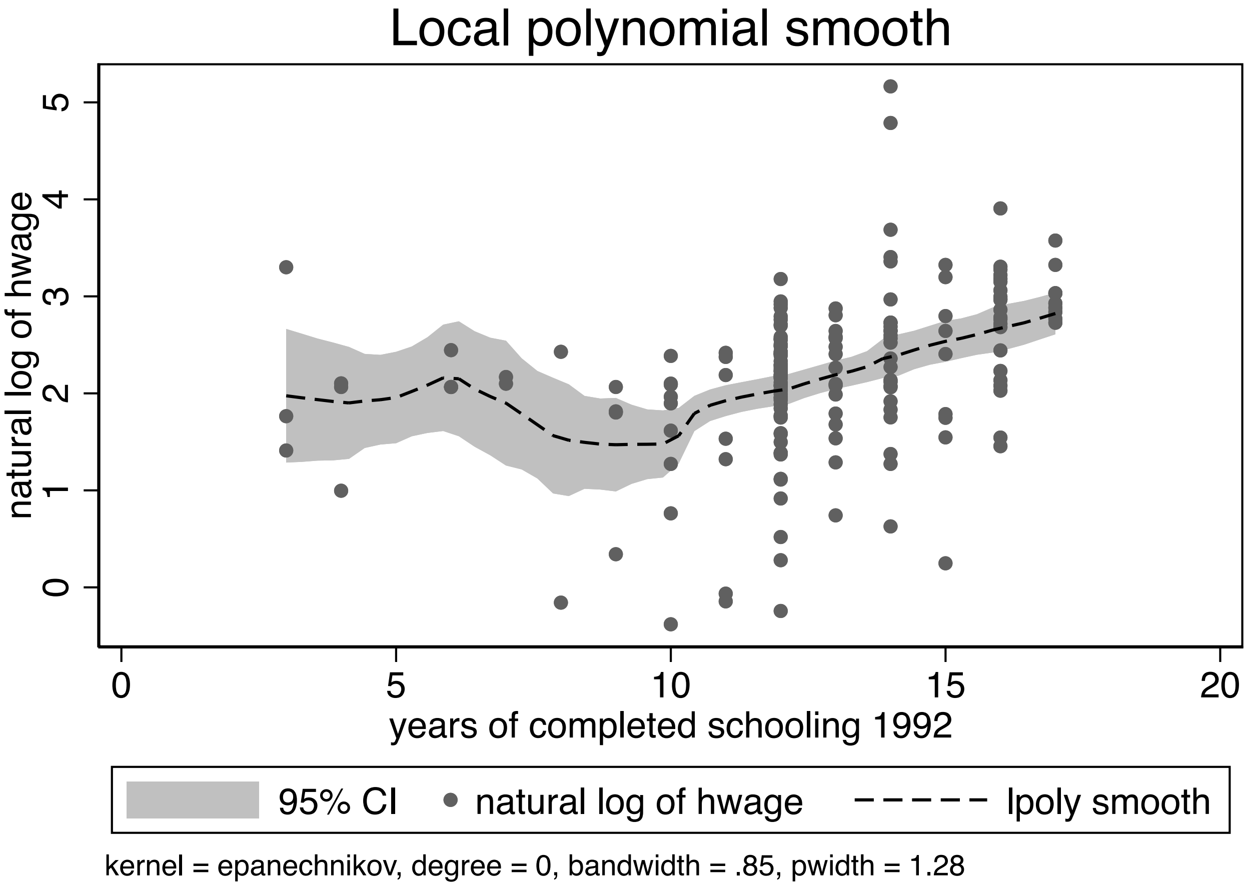
- Can deal with more than one regressor.

- we'll see an example in a few slides

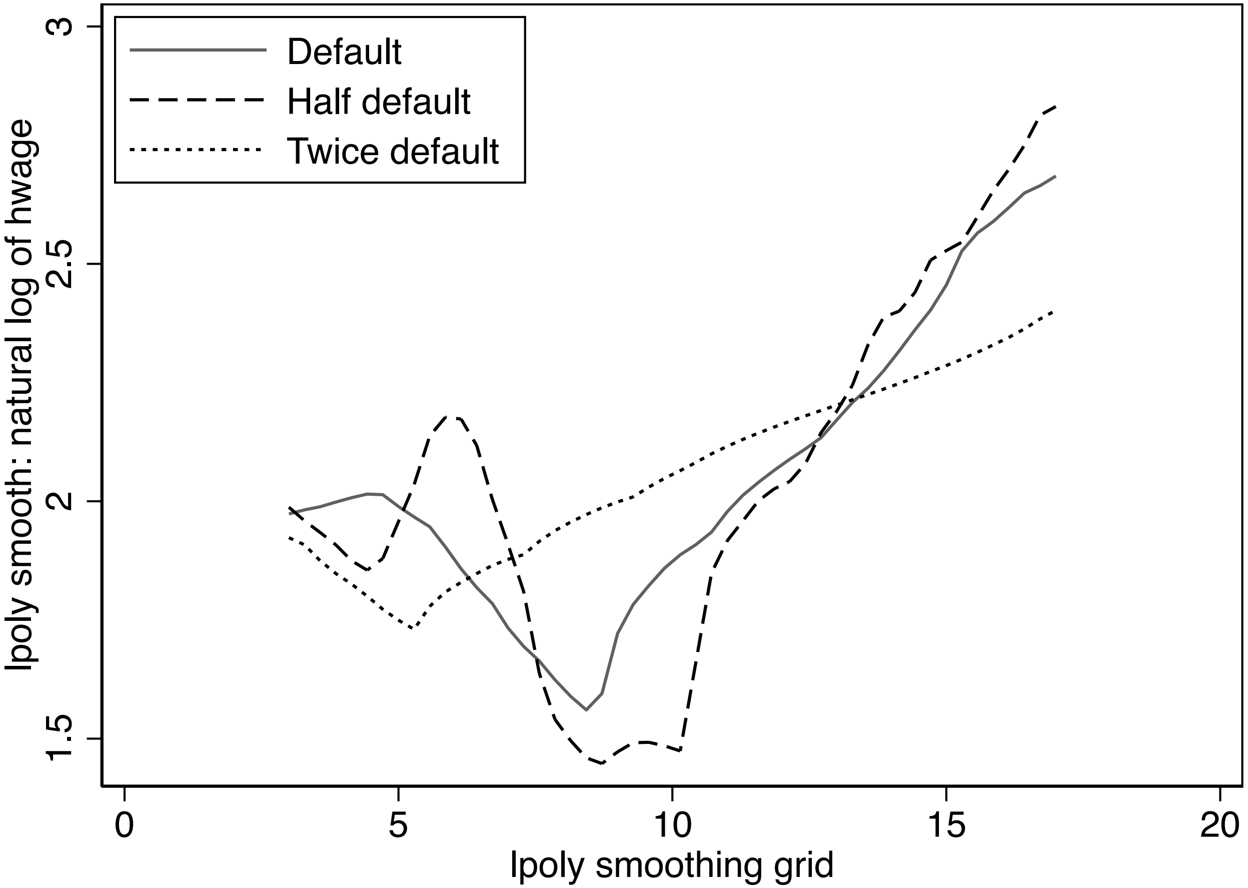
Example

lpoly lnhwage educatn, ci

(STATA default values, default is degree 0 –constant–; plug in estimator)



- Try different values for the bandwidth



Takeaways

- First nonparametric regression method: Kernel local regression
- In a nutshell: local averages of the dependent variable, y
- Choice of bandwidth is key
- Use cross-validation to select h
- Choice of kernel is less important, optimal kernel: Epanechnikov
- STATA commands: npregress, lpoly
- Asymptotic properties: consistent, asymptotically normal
- Lower convergence rates

- A few problems to be aware about:

- When computing confidence intervals: take into account bias reduction techniques

  - If asymptotic distribution is employed: undersmoothing, higher order kernels

  - Use bootstrap

- Open problem: How to compute marginal effects?

4. Other methods: Local Linear Regression

- The Nadaraya–Watson estimator can be seen as a particular case of a wider class of nonparametric estimators, the so-called local polynomial estimators.

- The Nadaraya–Watson estimator is a local constant estimator because it assumes that $m(x)$ equals a constant in the local neighborhood of x_0.

- Now: let $m(x)$ be linear in the neighborhood of x_0,

$$m(x) = a_0 + b_0(x - x_0) \text{ in the neighborhood of } x_0$$

.

Implementation of this idea

1) Notice that the kernel regression estimator (previous estimator) $m(x_0)$ can be obtained as

$$\widehat{m(x_0)} = argmin_{m_0} \sum_i W(\frac{x_i - x_0}{h})(y_i - m_0)^2$$

where the weights are the NW weights:

$$W(\frac{x_i - x_0}{h}) = K(\frac{x_i - x_0}{h}) / \sum_{j=1}^{N} K(\frac{x_i - x_0}{h})$$

■ Why? remember that $m(x_0)$ is a constant and $e_i = y_i - m_0$. Then, this is similar as weighted least squares, $e_i = y_i - m_0$.

2) Consider now $m_0 = a_0 + a_1(x_i - x_0)$. Obtain the local linear estimator as:

$$\widehat{m(x_0)} = argmin_{a_0,a_1} \sum_i W(\frac{x_i - x_0}{h})(y_i - a_0 - a_1(x_i - x_0))^2$$

Then, the estimate of m is a neighborhood of x_0 is given by

$$\hat{m}(x) = \hat{a}_0 + \hat{a}_1(x - x_0)$$

- Same idea: this is (local) weighted least squares regression, where the weights are kernel weights

- Interpretation:

- The constant a_0 is the conditional mean at x_0.
- The slope parameter, a_1: is the derivative of the mean function with respect to x.

3) More generally, we can consider a local polynomial estimator of degree p

$$argmin_{a_0,a_1} \sum_i W(\frac{x_i - x_0}{h})(y_i - a_0 - a_1(x_i - x_0) \cdots - a_p(x_i - x_0)^p)^2$$

Some advantages over NW

- Higher accuracy: Local linear regression estimators use a more flexible model that allows for a more accurate fit to the data, especially in regions where the data may be changing rapidly. Better behavior at end points (always problematic because of low density of data points).

- Easy computation of derivatives: (very useful for interpreting results)

- Cons: A bit more costly computationally than NW

Example

- Consider again the education/wage example: we will estimate a local linear regression

- STATA: Can be estimated using lpoly or npregress

lpoly lnhwage educatn,degree(1).

Or npregress kernel lnhwage educatn –several options available!–

- Let's look at the latter

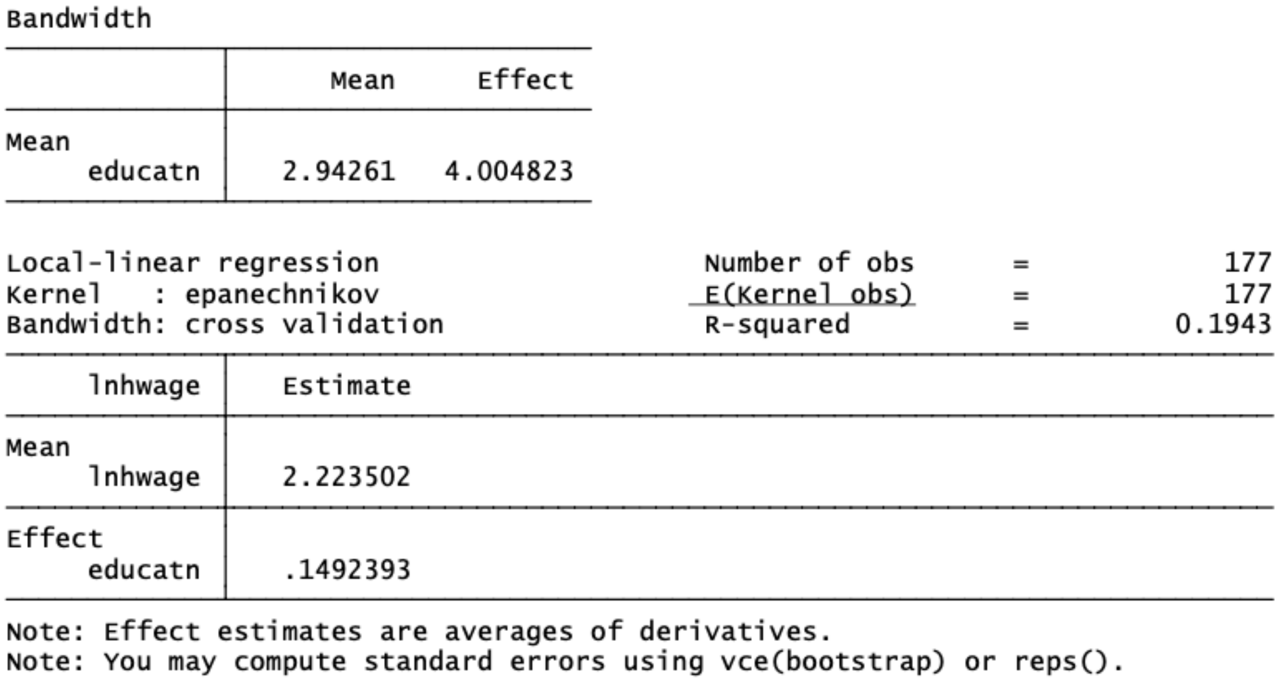
- npregress command - default is local linear

- The output reports averages of the mean function and the effects of the mean function.

- An average effect may be either 1) an average marginal effect, for continuous covariates or 2) the mean of contrasts for discrete covariates.

npregress kernel lnhwage educatn

- `npregress` reports averages $\widehat{\alpha} = \frac{1}{N}\sum_{i=1}^{N}\widehat{\alpha(x_i)}$ and $\widehat{\beta} = \frac{1}{N}\sum_{i=1}^{N}\widehat{\beta(x_i)}$



```
Bandwidth
─────────────────────────────────────
             │      Mean     Effect
─────────────┼───────────────────────
Mean         │
     educatn │   2.94261   4.004823
─────────────────────────────────────

Local-linear regression                     Number of obs      =          177
Kernel   : epanechnikov                     E(Kernel obs)      =          177
Bandwidth: cross validation                 R-squared          =       0.1943
──────────────────────────────────────────────────────────────────────────────
     lnhwage │   Estimate
─────────────┼────────────────────────────────────────────────────────────────
Mean         │
     lnhwage │   2.223502
─────────────┼────────────────────────────────────────────────────────────────
Effect       │
     educatn │   .1492393
──────────────────────────────────────────────────────────────────────────────
Note: Effect estimates are averages of derivatives.
Note: You may compute standard errors using vce(bootstrap) or reps().
```

- Versus OLS $\widehat{\alpha} = 0.897$ and $\widehat{\beta} = 0.10$

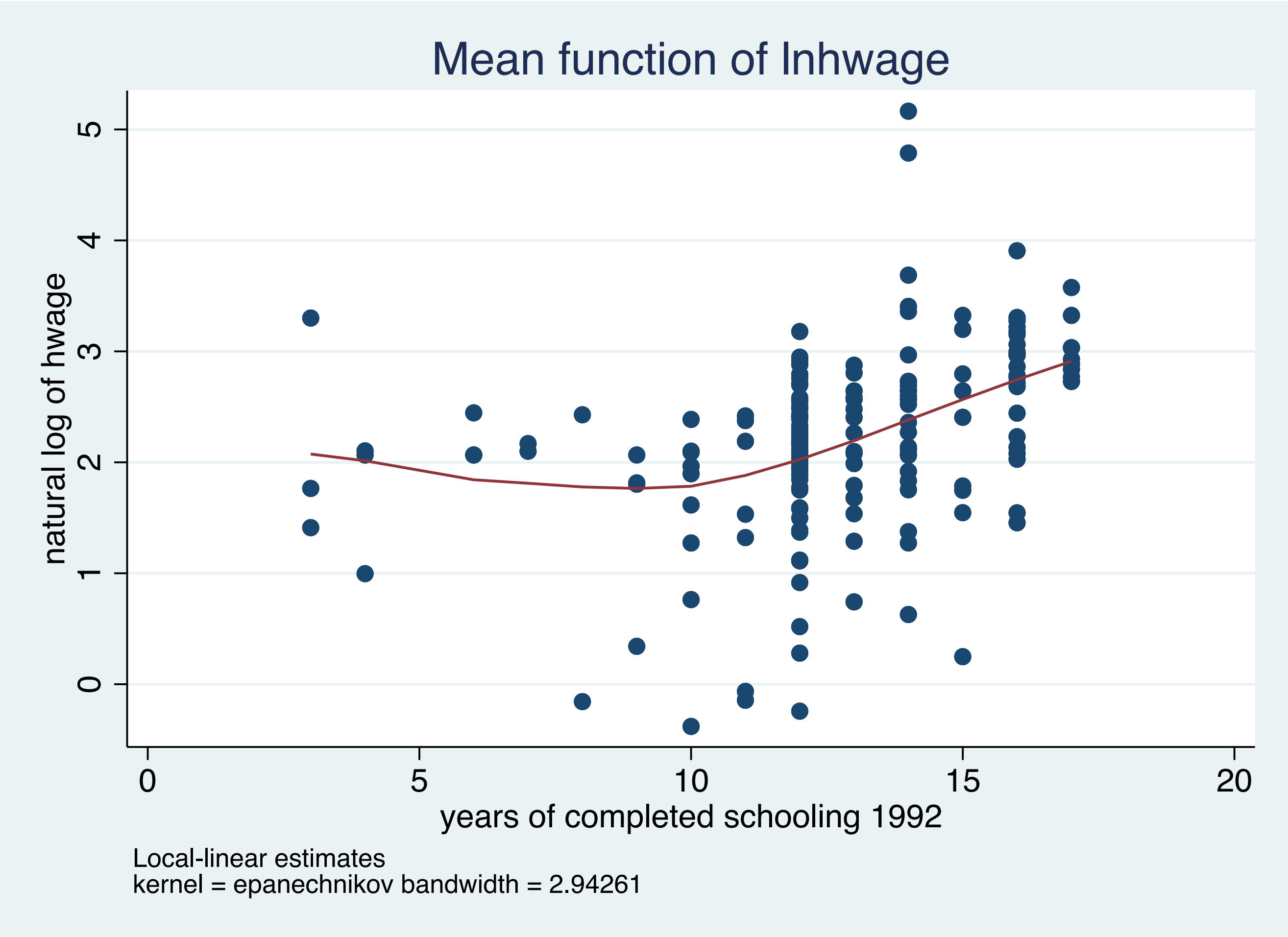
- First table: bandwidth employed

- Notice that different bandwidths are employed for mean effect and for the derivative effect

- Second table: averages of the means point and for the effects (derivative)

- Notice that by default standard errors do not appear, you can get them though by explicitly asking for them.

npgraph:



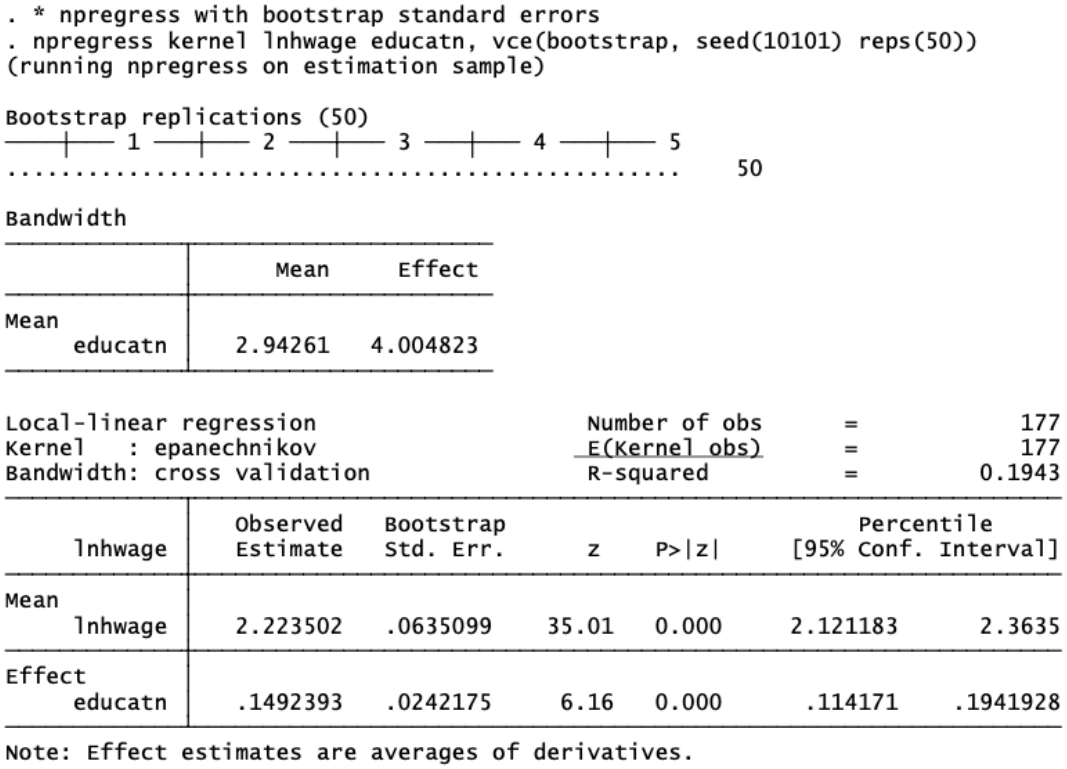
Mean function of lnhwage

Local-linear estimates
kernel = epanechnikov bandwidth = 2.94261

Obtain bootstrap standard errors and confidence intervals for these values

npregress kernel lnhwage educatn, vce(bootstrap, seed(10101) reps(50))

- Get bootstrap standard errors

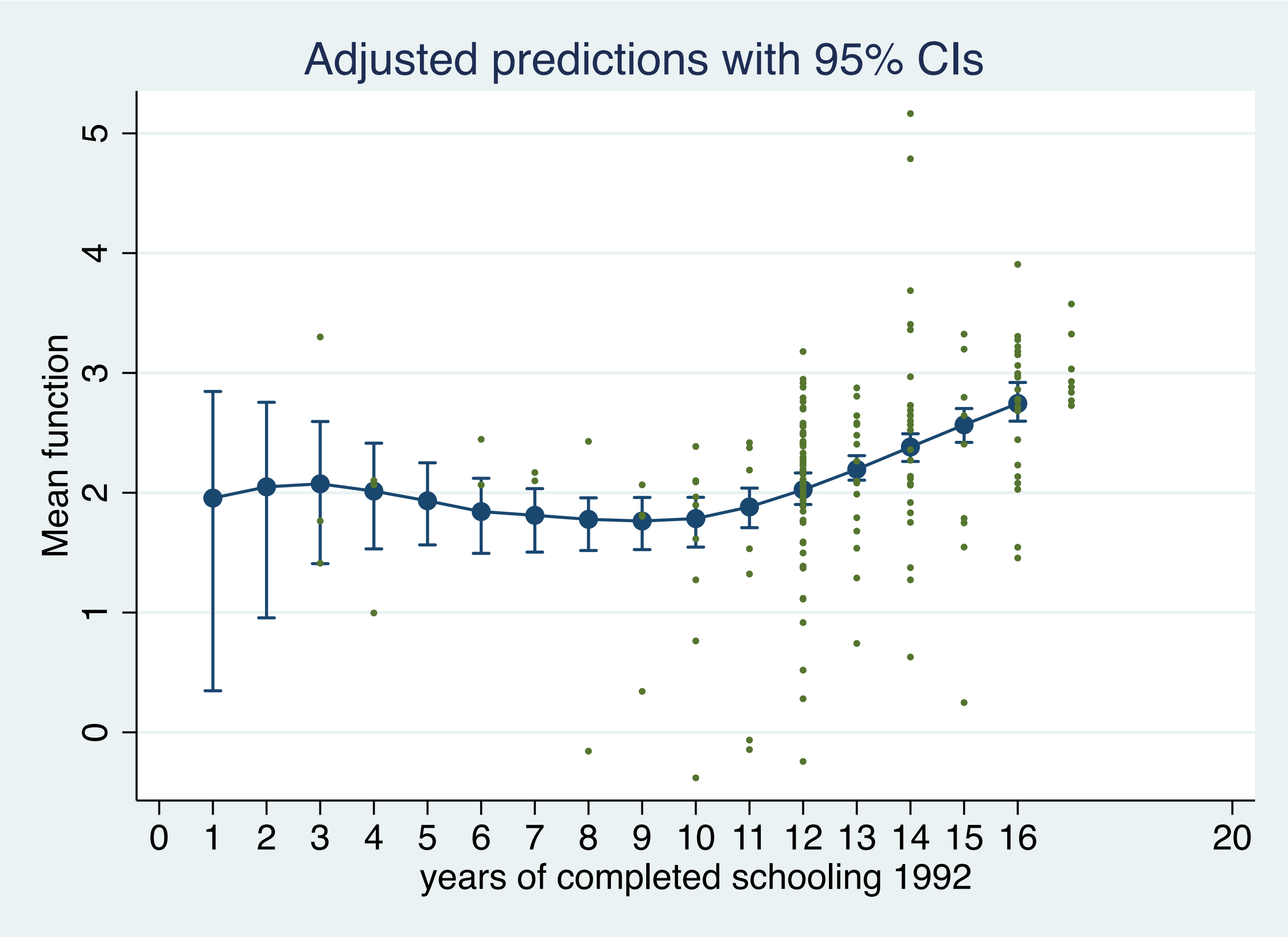


```
. * npregress with bootstrap standard errors
. npregress kernel lnhwage educatn, vce(bootstrap, seed(10101) reps(50))
(running npregress on estimation sample)

Bootstrap replications (50)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50

Bandwidth
------------------------------------
             |      Mean     Effect
-------------+----------------------
Mean         |
     educatn |   2.94261   4.004823
------------------------------------

Local-linear regression                    Number of obs      =        177
Kernel   : epanechnikov                    E(Kernel obs)      =        177
Bandwidth: cross validation                R-squared          =     0.1943
------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Percentile
     lnhwage |   Estimate   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Mean         |
     lnhwage |   2.223502   .0635099    35.01   0.000     2.121183      2.3635
-------------+----------------------------------------------------------------
Effect       |
     educatn |   .1492393   .0242175     6.16   0.000      .114171    .1941928
------------------------------------------------------------------------------
Note: Effect estimates are averages of derivatives.
```

- Versus OLS se$(\widehat{\alpha}) = 0.302$ and se$(\widehat{\beta}) = 0.023$.

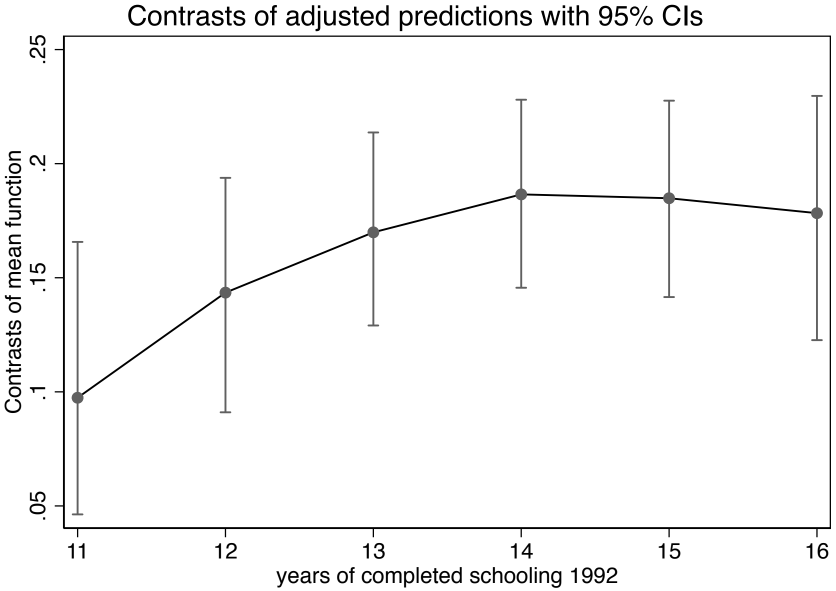
Plot the graph: Estimated value of $m(x_0)$, the conditional mean of log wage at x_0

```
margins, at(educatn = (1(1)16)) vce(bootstrap, seed(10101) reps(50))
marginsplot, legend(off) scale(1.1) /// addplot(scatter lnhwage
educatn if lnhwage<50000, msize(tiny))
```



- Partial effects of changing education

```
margins, at(educatn = (10(1)16)) contrast(atcontrast(ar)) ///
vce(bootstrap, seed(10101) reps(50))
marginsplot, legend(off)
```



- Stata code

```
npregress kernel lnhwage educatn

npregress kernel lnhwage educatn, vce(bootstrap, seed(10101) reps(50))

margins, at(educatn = (10(1)16)) vce(bootstrap, seed(10101) reps(50))

marginsplot, legend(off) scale(1.1) /// addplot(scatter lnhwage educatn if lnhwage¡50000, msize(tiny))

graph export nonparametricfig11.wmf, replace

margins, at(educatn = (10(1)16)) contrast(atcontrast(ar)) ///

vce(bootstrap, seed(10101) reps(50))

marginsplot, legend(off)

graph export nonparametricfig13.wmf, replace
```

5. Nearest Neighbor Estimator

- Simple idea: The k-nearest neighbor estimator is the weighted average of the y values for the k observations of x_i closest to x_0.

- Define $N_k(x_0)$: the set of k observations of x_i closest to x_0. Then:

$$m_{KNN}(x_0) = \frac{1}{k}\sum_{i}^{N} 1(x_i \in N_k(x_0))y_i$$

- This estimator is
  - a kernel estimator with uniform weights
  - except that the bandwidth is variable.

- Here the bandwidth h_0 at x_0 equals the distance between x_0 and the furthest of the k nearest neighbors, and more formally $h_0 = k/(2Nf(x_0))$.

- Pros: a simple rule for variable bandwidth selection.

- It is computationally faster to use a symmetrized version that uses the k/2 nearest neighbors to the left and a similar number to the right

6. Lowess

- Lowess: locally weighted scatterplot smoothing estimator

- A variant of local polynomial estimation (kernel)

- Computational Differences:

  - uses a variable bandwidth $h_{0,k}$ determined by the distance from x_0 to its kth nearest neighbor;

  - tricubic kernel

  - Robust against outliers: downweights observations with large residuals $e_i = y_i - m(x_i)$, which requires passing through the data N times.

- Lowess has some advantages with respect to local lineal regression:

- More robust against outliers

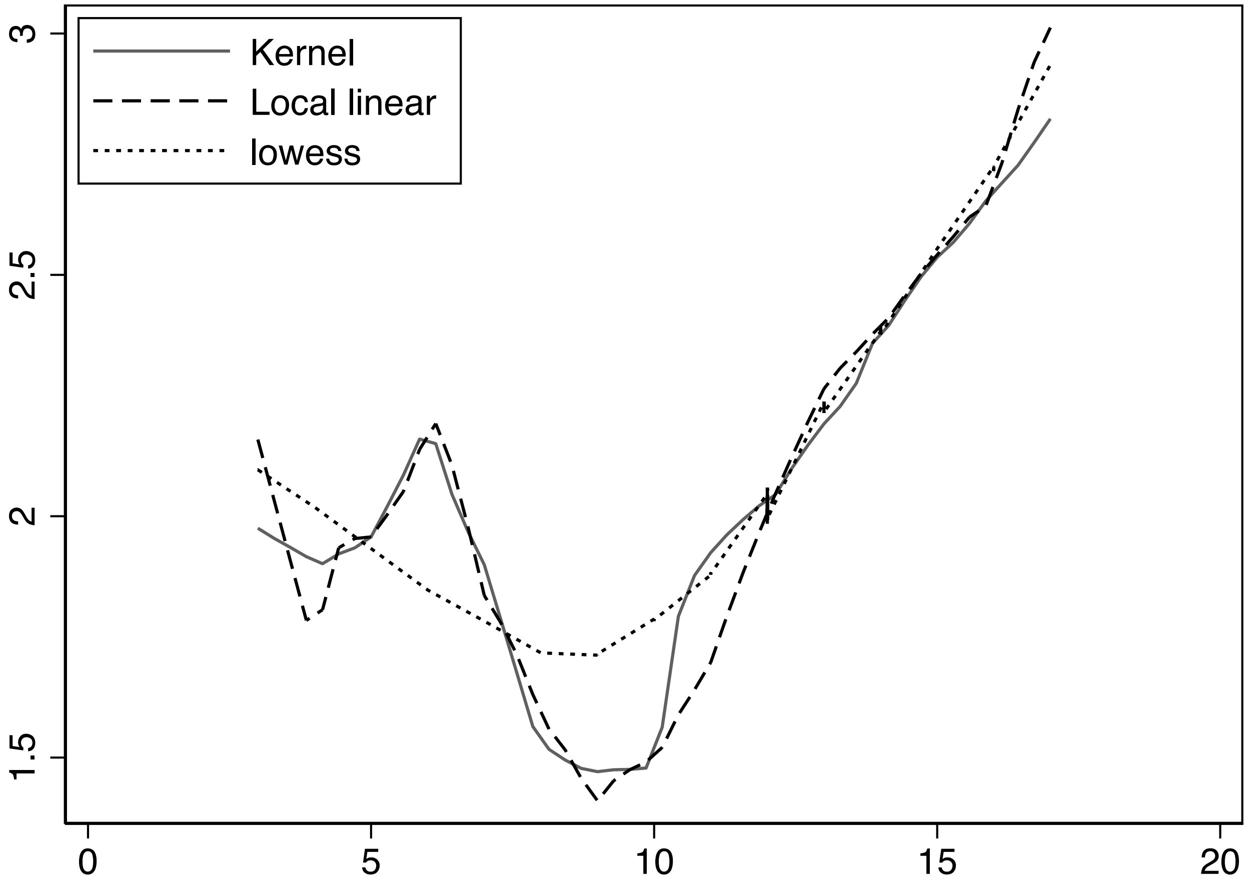
- But computationally more expensive

- See Fan and Gijbels (1996, p. 24). for additional details Lowess is attractive compared to kernel regression as it uses a variable

Example

Comparison of local constant, local linear and lowess: wage and years of education

To compute lowess: (lowess lnhwage educ, clstyle(p3)), scale(1.1) ///



Multivariate Kernel Regression:

- Conceptually, multivariate kernel regression is identical to univariate one

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

where x is a $k \times 1$ vector, $W(x_i, x_0, h) = K((x_i - x_0)/h) / \sum_i K((x_i - x_0)/h)$ and K(.) is a multivariate kernel

- Often, the multivariate kernel is just the product of univariate kernels
- If this is the case, divide by standard deviation so that all variables have similar scale
- Use cross validation to choose a common bandwidth h^*

- Important: convergence rates decreases (curse of dimensionality)
- Before: $\sqrt{Nh}$,
- Now: $\sqrt{Nh^k}$, where k is the number of covariates

Takeaways

- So far: Kernel-based methods to visualize/estimate conditional expectation in a flexible way

- Methods based on local averages of the dependent variable

- Several methods: local Kernel, local polynomial, Lowess, nearest neighbor . . .

- Methods differ in bandwidth used, weights used, etc.

- Not huge differences, but Lowess and local polynomial behave better at end points.

- These methods can handle multivariate regression, but rates of convergence decrease, so performance deteriorates as the number of regressors increases.

Semiparametric Methods

1. Introduction

- Previous slides: regression models without any structure.

- This gives a lot of flexibility but it also has some limitations :

  - Sometimes, theory may place some structure on the data. We might want to incorporate this information in the model

  - We can only include in the analysis a relative small set of variables (curse of dimensionality)

  - ...but incorporating many variables might be needed to avoid endogeneity of regressors

- This lecture: Semiparametric methods

Semiparametric models: examples (from Cameron&Trivedi)

- Many semiparametric models and many methods to estimate them. This is only a short intro to these methods.

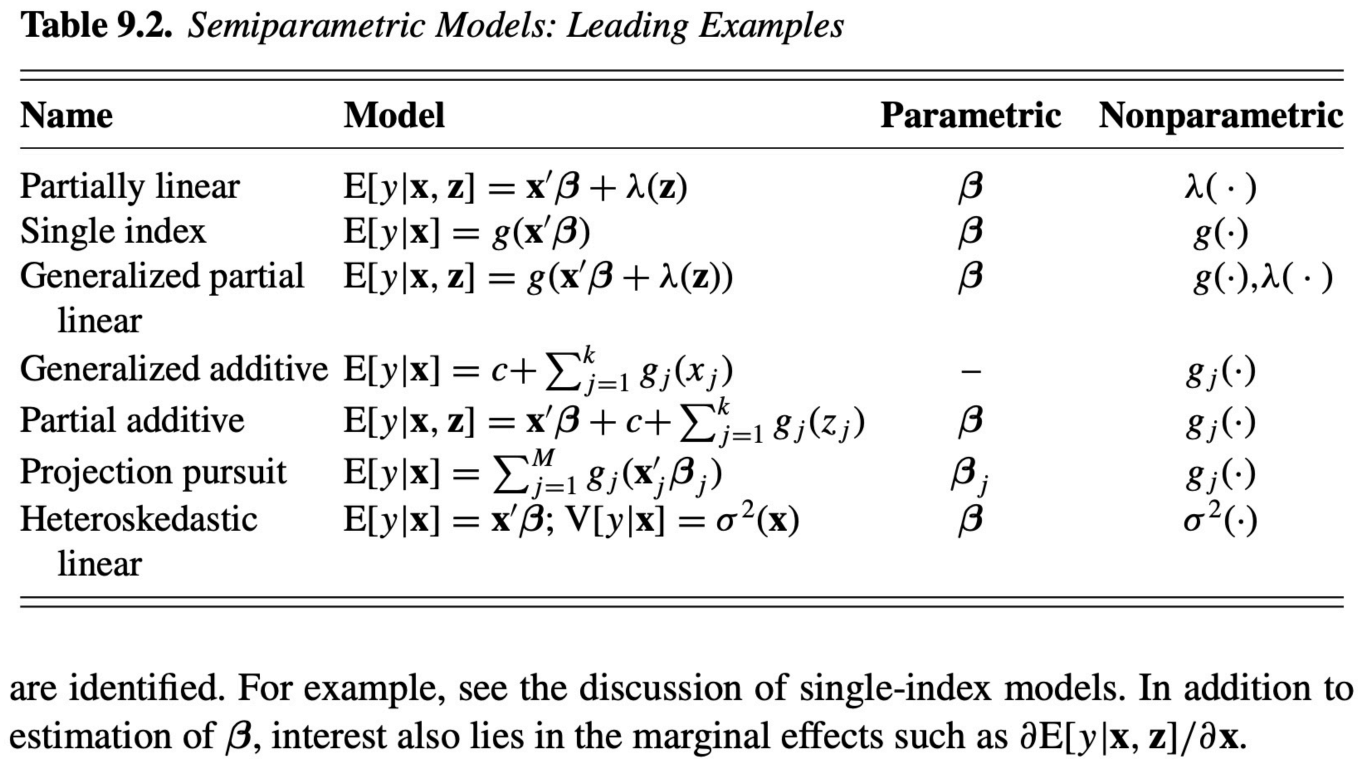


Table 9.2. *Semiparametric Models: Leading Examples*

Name	Model	Parametric	Nonparametric
Partially linear	$\mathrm{E}[y\vert\mathbf{x},\mathbf{z}] = \mathbf{x}'\boldsymbol{\beta} + \lambda(\mathbf{z})$	$\boldsymbol{\beta}$	$\lambda(\cdot)$
Single index	$\mathrm{E}[y\vert\mathbf{x}] = g(\mathbf{x}'\boldsymbol{\beta})$	$\boldsymbol{\beta}$	$g(\cdot)$
Generalized partial linear	$\mathrm{E}[y\vert\mathbf{x},\mathbf{z}] = g(\mathbf{x}'\boldsymbol{\beta} + \lambda(\mathbf{z}))$	$\boldsymbol{\beta}$	$g(\cdot),\lambda(\cdot)$
Generalized additive	$\mathrm{E}[y\vert\mathbf{x}] = c + \sum_{j=1}^{k} g_j(x_j)$	–	$g_j(\cdot)$
Partial additive	$\mathrm{E}[y\vert\mathbf{x},\mathbf{z}] = \mathbf{x}'\boldsymbol{\beta} + c + \sum_{j=1}^{k} g_j(z_j)$	$\boldsymbol{\beta}$	$g_j(\cdot)$
Projection pursuit	$\mathrm{E}[y\vert\mathbf{x}] = \sum_{j=1}^{M} g_j(\mathbf{x}_j'\boldsymbol{\beta}_j)$	$\boldsymbol{\beta}_j$	$g_j(\cdot)$
Heteroskedastic linear	$\mathrm{E}[y\vert\mathbf{x}] = \mathbf{x}'\boldsymbol{\beta}$; $\mathrm{V}[y\vert\mathbf{x}] = \sigma^2(\mathbf{x})$	$\boldsymbol{\beta}$	$\sigma^2(\cdot)$

are identified. For example, see the discussion of single-index models. In addition to estimation of $\boldsymbol{\beta}$, interest also lies in the marginal effects such as $\partial\mathrm{E}[y|\mathbf{x},\mathbf{z}]/\partial\mathbf{x}$.

Roadmap

1. Partially Linear Models

2. Single Index Models

3. Summary, other models exist . . .

Partially Linear Model

- Partially Linear Model: conditional mean is a linear regression function plus an unspecified nonlinear component.

$$E[y|x,z] = x\beta + \lambda(z) \quad \lambda(.) \text{ unspecified.}$$

- Model to be estimated:

$$y = x\beta + \lambda(z) + u, \quad E(u|x,z) = 0. \tag{8}$$

- Estimation Method: Robinson Difference Estimator

We will obtain estimates for β and for λ in two steps

- Step 1: get rid of $\lambda(z)$ and estimate β (only)
- Step 2: Use the estimates of β in a model that will allow us to obtain an estimate for $\lambda(.)$

Robinson Difference Estimator

- Step 1: A) get rid of $\lambda(.)$, B) estimate β

A) get rid of lambda:

- Take conditional expectations (conditioning by z) on both sides of Model 8 (and notice that $E(u|z) = 0$):

$$E[y|z] = E[x|z]'\beta + \lambda(z) \tag{9}$$

- Subtract the two equations –eq (8) and eq (9)– and obtain the model:

$$y - E[y|z] = (x - E[x|z])'\beta + u \tag{10}$$

- The conditional moments $E[y|z]$ and $E[x|z]$ are unknown $\Rightarrow$ Replace them by non parametric estimators $\hat{m}_{y_i}$ and $\hat{m}_{x_i}$

- Robinson's difference estimator:

- Step 1: B) estimate β in the model

$$y - \hat{m}_{y_i} = (x - \hat{m}_{x_i})'\beta + u \tag{11}$$

- The resulting estimator of β is consistent and A.N (assuming u is $i.i.d.$):

$$\sqrt{N}(\hat{\beta}_{PL} - \beta) \xrightarrow{d} N(0, \sigma^2 \Bigg(\text{ plim } \tfrac{1}{N} \sum_{i=1}^{N} (x_i - E[x_i|z_i])(x_i - E(x_i|z_i)' \Bigg)^{-1}$$

- Notes:

1. Cost of non-specifying λ? higher variance (efficiency loss) [But no loss if $E(x|z)$ is linear!]

2. The distribution assumes homokedasticity (u is $i.i.d$). Use Eicker-White standard errors to make it robust to heteroskedasticity

3. To estimate the variance: replace $(x_i - E[x_i|z_i])$ by $(x_i - \hat{m}_{x_i})$

4. How to compute $\hat{m}_{x_i}$ and $\hat{m}_{y_i}$?

- Robinson: Kernel estimates with convergence no slower than $N^{-1/4}$.

Step 2: Estimate λ

- Recall that $\lambda(z) = E(y|z) - E(x|z)'\beta$.
- Estimate $\lambda(z)$ as

$$\hat{\lambda}(z) = \hat{m}_{y_i} - \hat{m}'_{x_i}\beta$$

Summarizing: Robinson difference estimator

- Model: $E[y_i|x_i, z_i] = x_i'\beta + \lambda(z_i)$, unspeficied $\lambda(\cdot)$

- Steps:

1. Kernel regress y on z and get residual $y - \hat{y}$.

2. Kernel regress x on z and get residual $x - \hat{x}$.

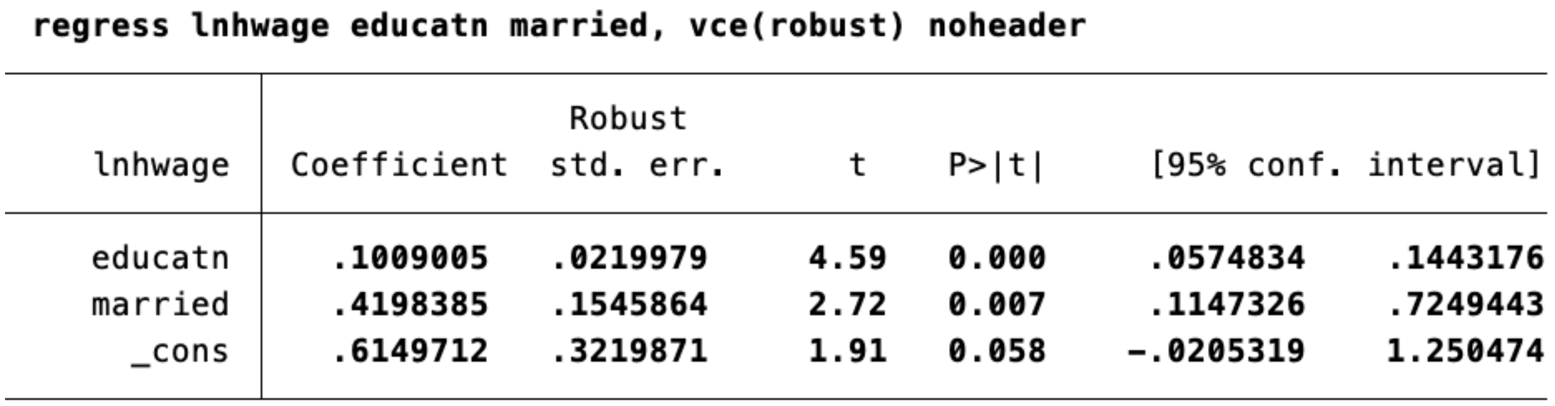
3. OLS regress $y - \hat{y}$ on $x - \hat{x}$, get $\hat{\beta}$

4. Combine the estimates in 1) 2) and 3) to get $\hat{\lambda}(z) = \hat{m}_{y_i} - \hat{m}'_{x_i}\hat{\beta}$

An example

- Same data as before: now, wage on marital status and education.

Let’s look first at the OLS:



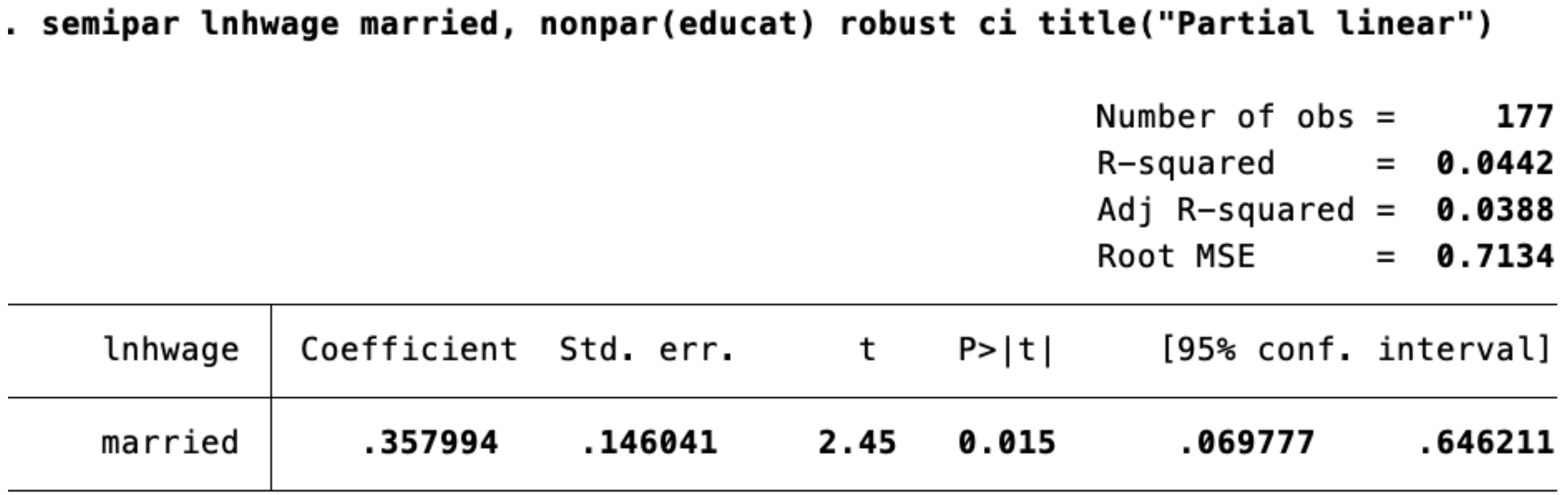
```
 regress lnhwage educatn married, vce(robust) noheader

------------------------------------------------------------------------------
             |               Robust
     lnhwage | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
     educatn |   .1009005   .0219979     4.59   0.000     .0574834    .1443176
     married |   .4198385   .1545864     2.72   0.007     .1147326    .7249443
       _cons |   .6149712   .3219871     1.91   0.058    -.0205319    1.250474
------------------------------------------------------------------------------
```

- Robison’s estimator:
- STATA command: semipar
- married enters linearly, we allow education to enter nonparametrically

semipar lnhwage married, nonpar(educatn) robust ci title(”Partial linear”)

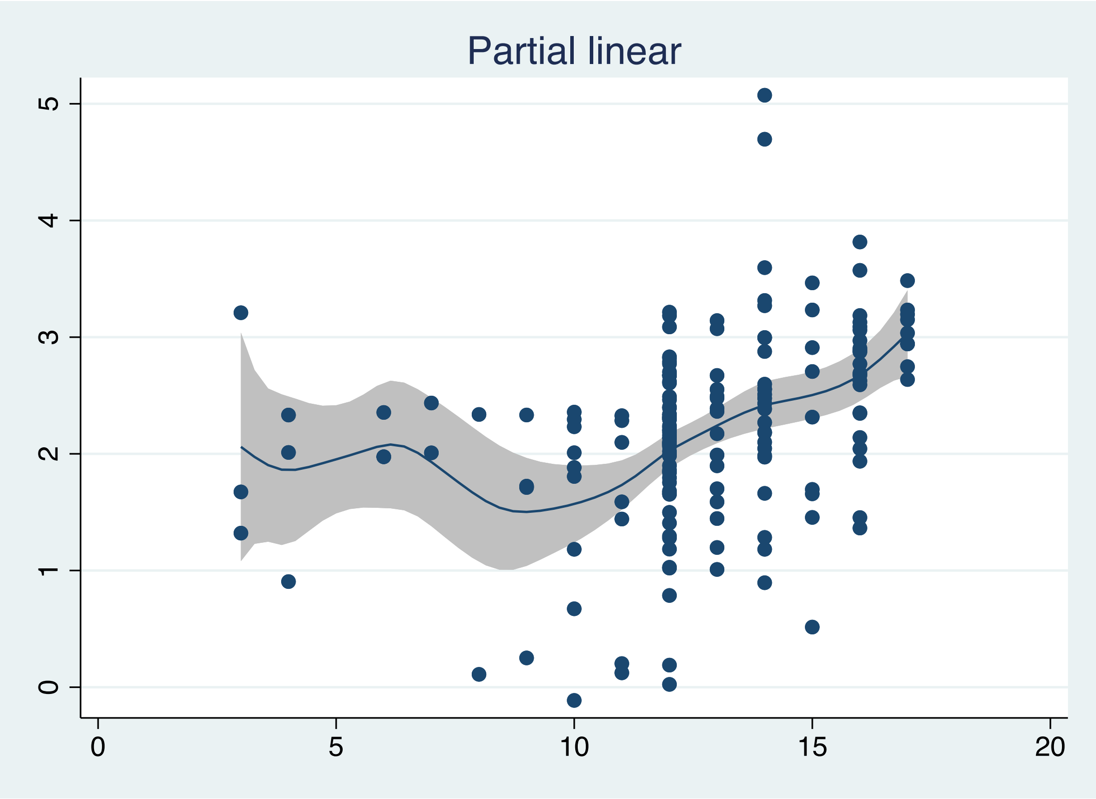
- with robust standard errors, confidence intervals...

Output has two parts: 1) the parametric component



```
. semipar lnhwage married, nonpar(educat) robust ci title("Partial linear")

                                                  Number of obs =      177
                                                  R-squared     =   0.0442
                                                  Adj R-squared =   0.0388
                                                  Root MSE      =   0.7134

------------------------------------------------------------------------------
     lnhwage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
     married |    .357994    .146041     2.45   0.015      .069777    .646211
------------------------------------------------------------------------------
```

- . . . and the non-parametric component: Plot of $\lambda(z)$ against z where z is education



Trimming

- Trimming: we can apply trimming to estimate the nonparametric component.

- In kernel estimation: estimates are not good in areas of the support of z with low density values

- Why? low density=few values to compute the local average

- Trimming consists of excluding data points for which the density $f(z) < b$, for some positive value b

- the command semipar allows for the introduction of trimming (default is no trimming)

Summarizing

- Partially linear model: additive model with a parametric part and a non-parametric one

- Estimation: Robinson's two step estimator

- Stata: semipar

- Advantages of this method:

- The model allows "any" form of the unknown

- $\hat{\beta}$ is $\sqrt{n}$-consistent

Summarizing II

- These methods have been recently revisited in the machine learning literature (really cutting edge at the moment)

- Double machine learning, see Chernozhukov et al (AER, 2017)

- In a nutshell: same idea but estimate the conditional expectations needed in the procedure above using machine learning, instead of kernel regression

- Several advantages, in particular, avoid the curse of dimensionality

- If interested, check out this link for an easy introduction to the topic.

Back to binned scatter plots

- Recall that binned scatter plots are very popular and very useful visualization tools of the conditional expectation

- STATA Binscatter command (see handout 3), very popular but...

- problematic as well (for instance, controlling for additional variables).

- A recent paper improves considerably on this: On binscatter, Cataneo et al. (2024). STATA: binsreg

- Main features of the new binned scatter plots:

- Framework: partially linear model.

$$y_i = \mu(x_i) + w_i'\gamma + e_i$$

- we're insterested on the shape of the relationship between x and y, controlling for additional variables w.

  - it provides ways of controlling (correctly!) for additional variables
  - Optimal choice of the number of bins (i.e., number of quantiles of x plotted)
  - Uncertainty quantification: confidence bands on the binscatter!

- Bottom line: in your applications, use the new binsreg command!

Single Index Models

- Model:

$$E[y_i|x_i] = g(x_i'\beta)$$

where $g(\cdot)$ is not specified.

- Many standard nonlinear (parametric) models such as logit, probit, and Tobit are of single-index form. (In these cases g(.) is known)

- But we can also estimate this model leaving $g(.)$ unspecified and estimate it non-parametrically.

- Advantages:

- Advantage 1: generalizes the linear regression model (which assumes g(.) is the identity function)

- Advantage 2: the curse of dimensionality is avoided as there is only one nonparametric dimension

Interpretation: marginal effects

- For single-index models the effect on the conditional mean of a change in the jth regressor using calculus methods is

$$\frac{\partial E[y|x]}{\partial x_j} = g'(x'\beta)\beta_j, \tag{12}$$

where $g'(z) = \frac{\partial g(z)}{\partial z}$.

- Then, relative effects of changes in regressors are given by the ratio of the coefficients since

$$\frac{\partial E[y|x]/\partial x_j}{\partial E[y|x]/\partial x_k} = \frac{\beta_j}{\beta_k}, \tag{13}$$

because the common factor $g'(x'\beta)$ cancels.

- Thus, if β_j is two times β_k, then a one-unit change in x_j has twice the effect as a one-unit change in x_k.

Estimation with unspecified g(.)

- Identification: β can only be identified up to location and scale
- That is, we estimate $a + b\beta_i$
- This is still useful to compute relative marginal effects!
- Ichimura semiparametric least squares: choose β and $g(\cdot)$ that minimizes

$$argmin_\beta \sum_{i=1}^{N} w(x_i) f(y_i - \hat{g}(x_i'\beta))^2,$$

where $\hat{g}(.)$ is the leave-one-out NW estimator and $w(\cdot)$ is a trimming function that drops outlying x values.

- β can only be estimated up to scale in this model,
- but still useful as ratio of coefficients equals ratio of marginal effects in a single-index models.

Example

- Same data as before. Now: wages on hours worked and education

- STATA command: sls (Semiparametric Least Squares from Ichimura, 1993)

- Install it first:

```
ssc install sls
```

- Run both ols and sls and compare results

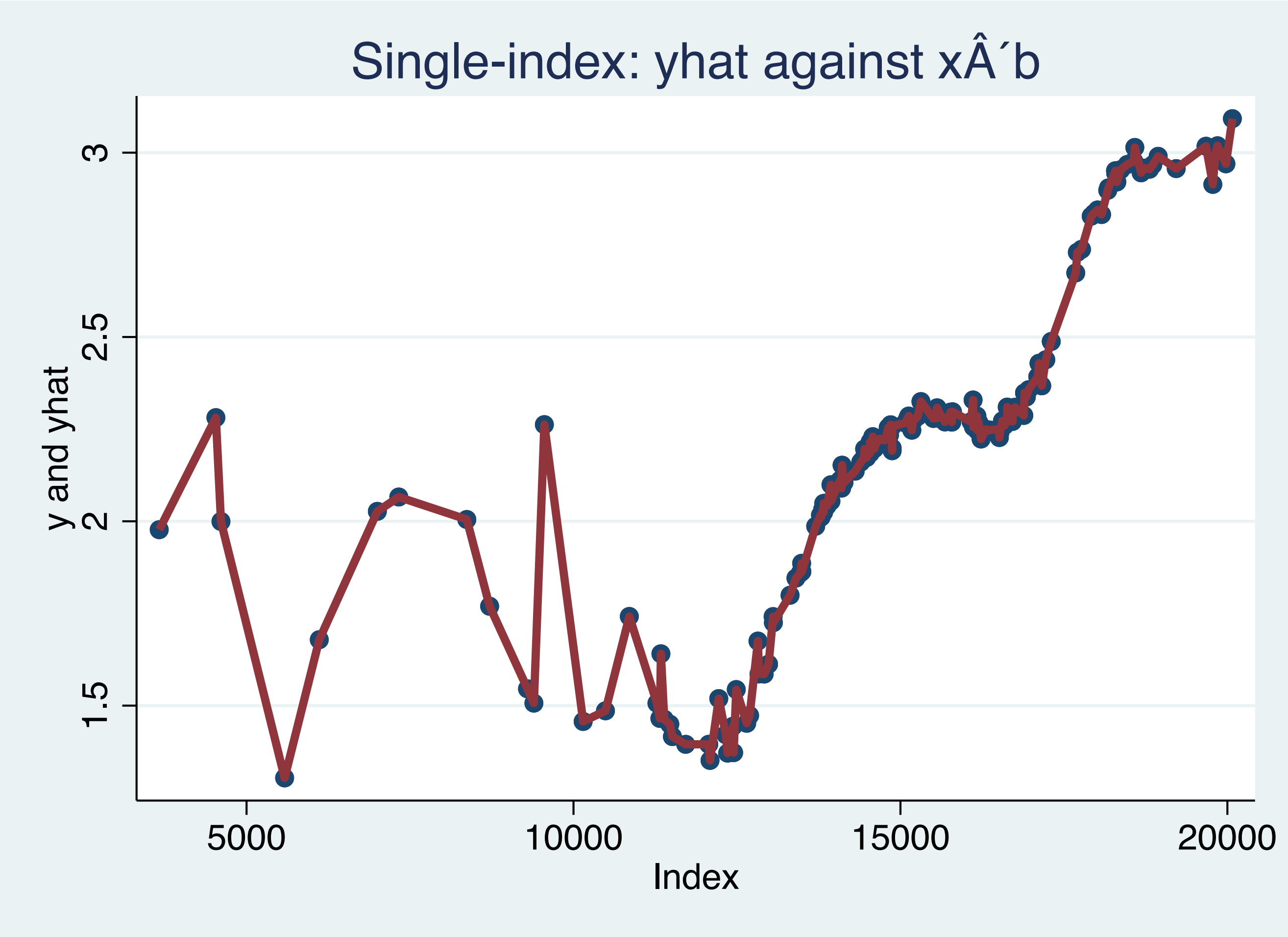
```
------------------------------------------------------------------------------
     lnhwage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       hours |   .0001365   .0000839     1.63   0.106    -.0000292    .0003022
     educatn |   .1071543   .0206339     5.19   0.000     .0664293    .1478793
       _cons |   .6437424   .3068995     2.10   0.037     .0380175    1.249467
------------------------------------------------------------------------------

. sls lnhwage hours educatn, trim(1,99)
initial:       SSq(b) =  120.10723
alternative:   SSq(b) =   120.1062
rescale:       SSq(b) =  98.292016
SLS 0:   SSq(b) =  98.292016
SLS 1:   SSq(b) =  98.195246
SLS 2:   SSq(b) =  98.007811
SLS 3:   SSq(b) =  98.007526
SLS 4:   SSq(b) =  98.007526
  pilot bandwidth
  1052.001873
SLS 0:   SSq(b) =  99.252078  (not concave)
SLS 1:   SSq(b) =  97.285143
SLS 2:   SSq(b) =  97.202952
SLS 3:   SSq(b) =  97.201992
SLS 4:   SSq(b) =  97.201988
                                                      Number of obs =     177
                                                      root MSE      =  .741056

------------------------------------------------------------------------------
     lnhwage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Index        |
     educatn |   1048.102   275.9987     3.80   0.000     507.1545     1589.05
       hours |          1  (offset)
------------------------------------------------------------------------------
```

- Interpretation:

- one more year of education has the same effect on log hourly wage as working 1048 more hours a year!

- Compared to OLS, $0.1071453/0.0001365 = 785$.

- This graph plots the predicted conditional expectation versus $x'\beta$ (highly nonlinear)



Index models, summary

- Generalizes the linear regression model (which assumes g(.) constant)

- Gain over parametric models: more flexible

- Gain over fully non-parametric: only one nonparametric dimension (avoid curse of dimensionality).

- We only identified the parameters up to location and scale but

- The ratio of the coefficients provides the relative marginal effects

Takeaways

- Semiparametric methods aim to overcome some of the limitations of fully parametric and fully nonparametric methods

- Flexible, yet tractable (many variables can be included)

- Literature is very large

- Here, we've only presented a short introduction.

- Partial linear model, single index models . . . but many more models are available

- See this document to learn about other semiparametric methods STATA can handle