Nonparametrics and Semiparametrics

A. Colin Cameron
U.C.-Davis

CINCH Academy 2019
The Essen Summer School in Health Economics

April 6, 2019

1. Introduction

- Nonparametric methods place few restrictions on the data generating process
  - density estimation - use kernel density estimate
  - regression curve estimation - use kernel-weighted local constant or local linear regression
    - ★ but curse of dimensionality as # regressors increases
- Semiparametric regression places some structure
  - e.g. $E[y|\mathbf{x}] = g(\mathbf{x}'\beta)$ where $g(\cdot)$ is unspecified
  - reduces nonparametric component to one dimension.
- Bootstrap
  - most often used to get standard errors
  - more refined bootstraps can give better finite sample inference.

Summary

1. Introduction
2. Nonparametric (kernel) density estimation
3. Nonparametric (kernel) regression
4. npregress command (Stata 15)
5. Semiparametric regression
6. Stata commands

2. Nonparametric (kernel) density estimation

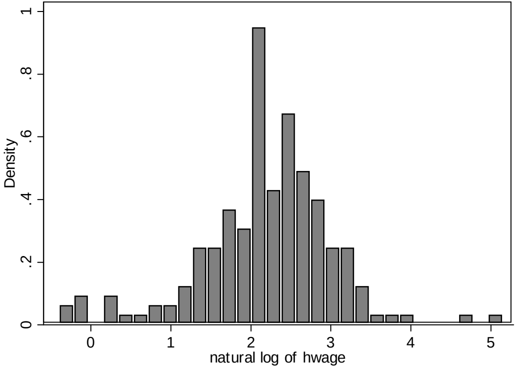
- Parametric density estimate
  - assume a density and use estimated parameters of this density
  - e.g. normal density estimate: assume $y_i \sim \mathcal{N}[\mu, \sigma^2]$ and use $\mathcal{N}[\bar{y}, s^2]$.
- Nonparametric density estimate: a histogram
  - break data into bins and use relative frequency within each bin
  - Problem: a histogram is a step function, even if data are continuous
- Smooth nonparametric density estimate: kernel density estimate.
  - smooths a histogram in two ways:
    - use overlapping bins so evaluate at many more points
    - use bins of greater width with most weight at the middle of the bin.

Histogram estimate

- A histogram is a nonparametric estimate of the density of y
  - break data into bins of width $2h$
  - form rectangles of area the relative frequency $= freq/N$
  - the height is $freq/2Nh$ (check: area $= (freq/2Nh) \times 2h = freq/N$).
- Use $freq = \sum_{i=1}^{N} \mathbf{1}(x_0 - h < x_i < x_0 + h)$
  - where indicator function $1(\mathbf{A})$ equals 1 if event $\mathbf{A}$ happens and equals 0 otherwise
- The histogram estimate of $f(x_0)$, the density of x evaluated at x_0, is

$$
\begin{aligned}
\widehat{f}_{HIST}(x_0) &= \tfrac{1}{2Nh} \sum_{i=1}^{N} \mathbf{1}(x_0 - h < x_i < x_0 + h) \\
&= \tfrac{1}{Nh} \sum_{i=1}^{N} \frac{1}{2} \times \mathbf{1}\left(\left| \tfrac{x_i - x_0}{h} \right| < 1 \right).
\end{aligned}
$$

- Data example: histogram of lnwage for $N = 175$ observations
  - Varies with the bin width (or equivalently the number of bins)
  - default is $\sqrt{N}$for $N \leq 861$ and $10\ln(N)/\ln(10)$ for $N > 861$
  - here specify 30 bins, each of width $2h \simeq 0.20$ so $h \simeq 0.10$
  - `histogram lnhwage, bin(30)` scale(1.1)



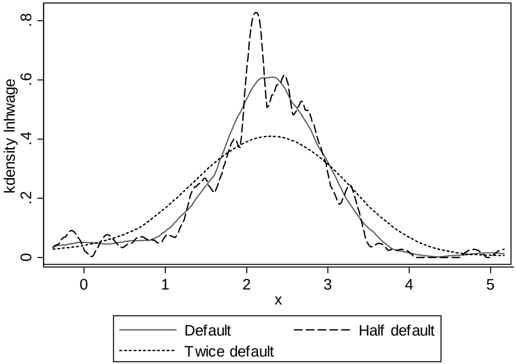
Kernel density estimate

- Recall $\widehat{f}_{HIST}(x_0) = \frac{1}{Nh}\sum_{i=1}^{N} \frac{1}{2} \times \mathbf{1}\left(\left|\frac{x_i - x_0}{h}\right| < 1\right)$
- Replace $\mathbf{1}(A)$ by a kernel function
- Kernel density estimate of $f(x_0)$, the density of x evaluated at x_0, is

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

  - $K(\cdot)$ is called a kernel function
  - h is called the bandwidth or window width or smoothing parameter h
- Example is Epanechnikov kernel
  - $K(z) = 0.75(1 - z^2) \times \mathbf{1}(|z| < 1)$ in Stata epan2 kernel
  - more weight on data at center, less weight at end
- More generally kernel function must satisfy conditions including
  - Continuous, $K(z) = K(-z)$, $\int K(z)dz = 1$, $\int zK(z)dz = 0$, tails go to zero.

- Data example: kernel of lnwage for 175 observations
  - Stata's **epanechnikov** kernel $K(z) = 0.75(1 - z^2)/\sqrt{5} \times \mathbf{1}(|z| < \sqrt{5})$
  - default $h = 0.9m/N^{0.2}$ where $m = \min(st.dev.(x),$ $\text{interquartilerange}_x/1.349)$ yields $h = 0.2093$.
  - $h = 0.07$ (oversmooths), 0.21 (default) or 0.63 (undersmooths)
  - e.g. `kdensity lnhwage, bw(0.21)`



Implementation

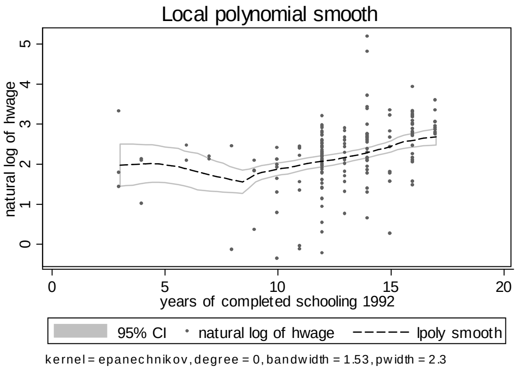
- Key is choice of bandwidth
  - The default can oversmooth: may need to decrease `bw()`
- For kernel choice
  - for given bandwidth get similar results across kernels if $K(z) > 0$ for $|z| < 1$ and $K(z) = 0$ for $|z| \geq 1$.
  - this is most kernels aside from `epanichnikov` and `gaussian`.
- Other smooth estimators exist
  - most notably k-nearest neighbors
  - but usually no reason to use anything but kernel.

3. Kernel regression: Local average estimator

- We want to estimate at various values x_0 the conditional mean function
$$m(x_0) = E[y|x = x_0]$$
- The functional form $m(\cdot)$ is not specified.
- A local average estimator is
$$\widehat{m}(x_0) = \sum_{i=1}^{N} w(x_i, x_0, h) y_i,$$
- The weights $w(x_i, x_0, h)$
  - sum over i to one
  - decrease as the distance between x_i and x_0 increases
  - place more weight on observations with x_i close to x_0 as bandwidth h decreases
  - most common: kernel weights, Lowess and k-nearest neighbors (average the $y_i's$ for the k $x_i's$ closest to x_0).
- Evaluate $\widehat{m}(x_0)$ at a variety of points x_0 gives a regression curve.

Kernel (local constant) regression

- Let
$$w(x_i, x_0, h) = K\left(\frac{x_i - x_0}{h}\right) / \left(\sum_{j=1}^{N} K\left(\frac{x_j - x_0}{h}\right)\right).$$
- Kernel regression with 95% confidence bands, default kernel (Epanechnikov) and default bandwidth
  - lpoly lnhwage educatn, ci msize(small)



Local linear regression

- A sample mean of y = OLS of y on an intercept.
- A weighted sample mean of y = weighted OLS of y on an intercept.
- So the kernel (local constant) estimator $\widehat{m}(x_0) = \widehat{\alpha}_0$ that minimizes

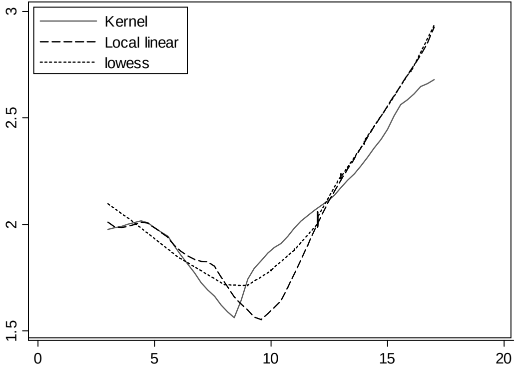
$$\sum_{i=1}^{N} w(x_i, x_0, h)(y_i - \alpha_0)^2.$$

- The local linear estimator generalizes to $\widehat{m}(x_0) = \widehat{\alpha}_0$ that minimizes

$$\sum_{i=1}^{N} w(x_i, x_0, h)\{y_i - \alpha_o - \beta_0(x_i - x_0)\}^2.$$

  - furthermore $\widehat{\beta}_0 = \widehat{m}'(x_0)$, an estimate of $\partial E[y|x]/\partial x|_{x_0}$.
- Advantage - better estimates at endpoints of the data.
- In Stata `lpoly lnhwage educatn,degree(1)`.
- And can extend to higher order polynomials.

- Lowess (locally weighted scatterplot smoothing) is a variation of local linear with variable bandwidth, tricubic kernel and downweighting of outliers.
- Kernel, local linear and lowess with default bandwidths
  - `graph twoway lpoly y x || lpoly y x, deg(1) || lowess y x`
  - kernel erroneously underestimates $m(x)$ at the endpoint $x = 17$.



Implementation

- Different methods work differently
  - Local linear and local polynomial handle endpoints better than kernel.
- $\widehat{m}(x_0)$ is asymptotically normal
  - this gives confidence bands that allow for heteroskedasticity
- Bandwidth choice is crucial
  - optimal bandwidth trades off bias (minimized with small bandwidth) and variance (minimized with large bandwidth)
  - theory just says optimal bandwidth for kernel regression is $O(N^{-0.2})$
  - "plug-in" or default bandwidth estimates are often not the best
  - so also try e.g. half and two times the default.
  - cross validation minimizes the empirical mean square error $\sum_i (y_i - \widehat{m}_{-i}(x_i))^2$, where $\widehat{m}_{-i}(x_i)$ is the "leave-one-out" estimate of $\widehat{m}(x_i)$ formed with y_i excluded
    - ★ empirical estimate of $\text{MSE}[\widehat{m}(x_i)] = \text{Variance} + \text{Bias}^2$.

4. npregress command

- Stata 15 has new `npregress` command.
- Does local constant and local linear regression.
- 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 have more than one regressor.

- `npregress` with defaults
  - LOOCV separate for bandwidth for $\widehat{m}(x_0)$ and $\widehat{m}'(x_0)$

```
. * npregress command - local linear
. npregress kernel lnhwage educatn

Computing mean function

Minimizing cross-validation function:

Iteration 0:   Cross-validation criterion = -.54003013
Iteration 1:   Cross-validation criterion = -.55652254
Iteration 2:   Cross-validation criterion = -.55725573
Iteration 3:   Cross-validation criterion = -.55764199
Iteration 4:   Cross-validation criterion = -.55764199
Iteration 5:   Cross-validation criterion =  -.5577778
Iteration 6:   Cross-validation criterion =  -.5578764
Iteration 7:   Cross-validation criterion =  -.5578764
Iteration 8:   Cross-validation criterion =  -.5578764

Computing optimal derivative bandwidth

Iteration 0:   Cross-validation criterion =  .00293233
Iteration 1:   Cross-validation criterion =  .00293233
Iteration 2:   Cross-validation criterion =  .00293233
Iteration 3:   Cross-validation criterion =  .00291228
Iteration 4:   Cross-validation criterion =  .00291228
```

- `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 $\widehat{\beta} = 0.10$

- Get bootstrap standard errors

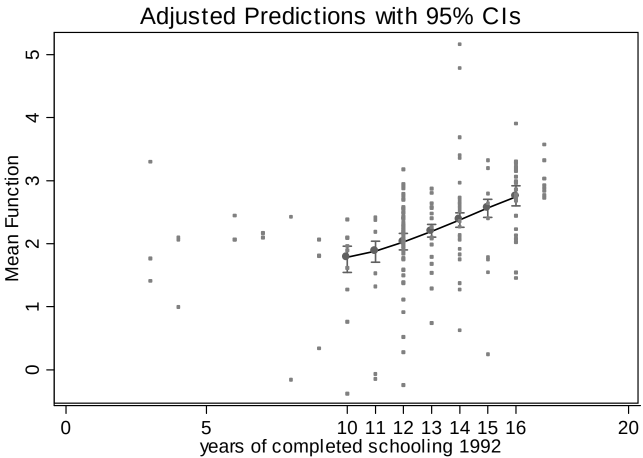
```
. * npregress with bootstrap standard errors
. npregress kernel lnhwage educatn, vce(bootstrap, seed(10101) reps(50))
(running npregress on estimation sample)

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 $\text{se}(\widehat{\alpha}) = 0.302$ and $\text{se}(\widehat{\beta}) = 0.023$.

- Predict at selected values of education

```
. margins, at(educatn = (10(1)16)) vce(bootstrap, seed(10101) reps(50))
(running margins on estimation sample)

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

Adjusted predictions                            Number of obs     =        177
                                                Replications      =         50

Expression   : mean function, predict()

1._at        : educatn         =          10

2._at        : educatn         =          11

3._at        : educatn         =          12

4._at        : educatn         =          13

5._at        : educatn         =          14

6._at        : educatn         =          15

7._at        : educatn         =          16

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Percentile
             |     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         _at |
          1  |   1.784381   .1152519    15.48   0.000     1.545979    1.961678
          2  |   1.881796   .0917833    20.50   0.000     1.708159     2.03875
          3  |   2.025275   .0719339    28.15   0.000     1.901929    2.165223
          4  |   2.195183   .0627936    34.96   0.000     2.104903    2.309129
          5  |   2.381722   .0663851    35.88   0.000     2.261005    2.492229
          6  |   2.566578   .0796751    32.21   0.000     2.420242    2.702775
          7  |   2.744897   .0975604    28.14   0.000     2.597464    2.920562
------------------------------------------------------------------------------
```

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



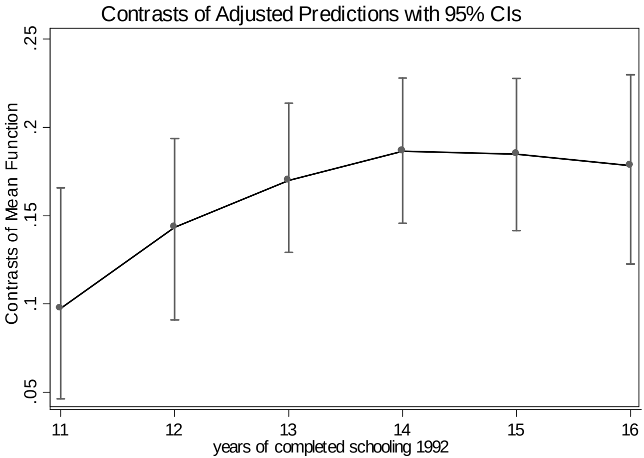
- Now consider partial effects at selected values of education
- * Partial effects of changing education

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

  vce(bootstrap, seed(10101) reps(50))
- Output includes

	Observed Contrast	Bootstrap Std. Err.	Percentile [95% Conf.	Interval]
_at				
(2 vs 1)	.0974155	.034265	.0462881	.1657016
(3 vs 2)	.1434789	.0346023	.0910292	.1937705
(4 vs 3)	.1699081	.0303118	.1290779	.2136698
(5 vs 4)	.1865389	.028668	.145619	.2280139
(6 vs 5)	.1848565	.0276149	.1415323	.2275936
(7 vs 6)	.1783189	.0297354	.1226577	.2296861

- marginsplot, legend(off)



5. Semiparametric estimation

- Nonparametric regression is problematic when more than one regressor
  - in theory can do multivariate kernel regression
  - in practice the local averages are over sparse cells
  - called the "curse of dimensionality"
- Semiparametric methods place some structure on the problem
  - parametric component for part of the model
  - nonparametric component that is often one dimensional
- Ideally $\sqrt{N}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \stackrel{d}{\rightarrow} \mathcal{N}[\mathbf{0}, \mathbf{V}]$ despite the nonparametric component.
- Three leading examples
  - partial linear
  - single-index
  - generalized additive model.

OLS estimates

- Consider log hourly wage regressed on years of education and annual hours worked

```
. regress lnhwage educatn hours, vce(robust)

Linear regression                               Number of obs     =        177
                                                F(2, 174)         =      10.12
                                                Prob > F          =     0.0001
                                                R-squared         =     0.1389
                                                Root MSE          =     .77289
```

lnhwage	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
educatn	.1071543	.0239147	4.48	0.000	.0599542	.1543545
hours	.0001365	.0001023	1.33	0.184	-.0000655	.0003384
_cons	.6437424	.3946326	1.63	0.105	-.1351406	1.422626

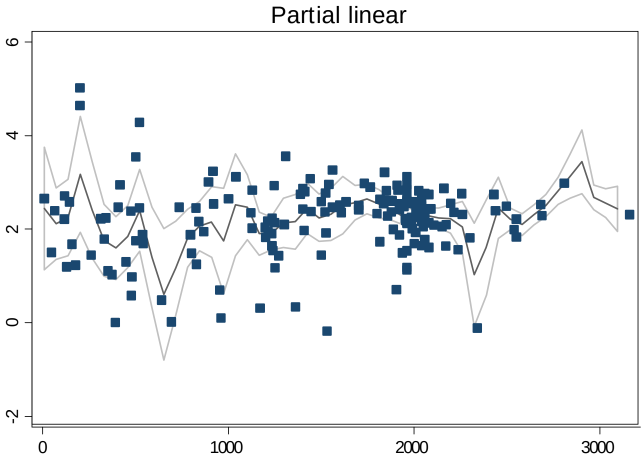
Partial linear model

- Model: $\mathrm{E}[y_i|\mathbf{x}_i, \mathbf{z}_i] = \mathbf{x}_i'\boldsymbol{\beta} + \lambda(\mathbf{z}_i)$ where $\lambda(\cdot)$ not specified.
- Robinson differencing estimator
  - kernel regress y on $\mathbf{z}$ and get residual $y - \widehat{y}$
  - kernel regress $\mathbf{x}$ on $\mathbf{z}$ and get residual $\mathbf{x} - \widehat{\mathbf{x}}$
  - OLS regress $y - \widehat{y}$ on $\mathbf{x} - \widehat{\mathbf{x}}$

```
. * Partial linear model - Robinson differencing estimator
. semipar lnhwage educatn, nonpar(hours) robust ci title("Partial linear")

                                                  Number of obs =      176
                                                  R-squared     =   0.1298
                                                  Adj R-squared =   0.1248
                                                  Root MSE      =   0.6365
```

lnhwage	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
educatn	.1023295	.0256881	3.98	0.000	.0516312	.1530278

- Plot of $\lambda(z)$ against z where z is annual hours worked.



Single-index model

- Model: $\mathrm{E}[y_i|\mathbf{x}_i] = g(\mathbf{x}_i'\boldsymbol{\beta})$ where $g(\cdot)$ not specified
- Ichimura semiparametric least squares $\widehat{\boldsymbol{\beta}}$ and $\widehat{g}$ minimize

$$\sum_{i=1}^{N} w(\mathbf{x}_i)\{y_i - \widehat{g}(\mathbf{x}_i'\boldsymbol{\beta})\}^2$$

  - where $w(\mathbf{x}_i)$is a trimming function that drops outlying $\mathbf{x}$ values.
- Can only estimate $\boldsymbol{\beta}$ up to scale in this model
  - Still useful as ratio of coefficients equals ratio of marginal effects in a single-index models
- From next slide one more year of education has same effect on log hourly wage as working 1,048 more hours
  - versus OLS $0.1071453/0.0001365 = 785$.

```
. * Single index model - Ichimura semiparametric least squares
. 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) =  98.007825
SLS 3:   SSq(b) =  98.007526
SLS 4:   SSq(b) =  98.007526
  pilot bandwidth
  1052.001876
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
```

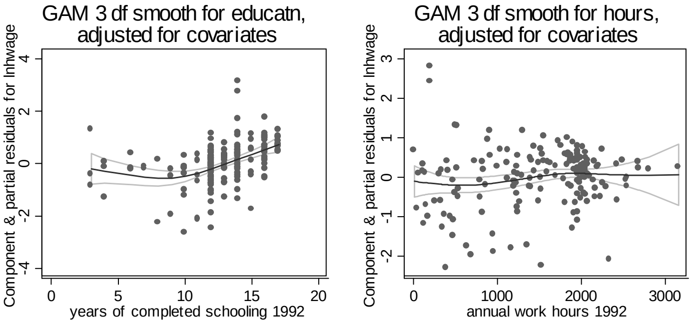
lnhwage	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Index						
educatn	1048.102	276.0092	3.80	0.000	507.1341	1589.07
hours	1	(offset)				

Generalized additive model

- Model: $E[y_i|\mathbf{x}_i] = g_1(x_{1i}) + \cdots + g_K(x_{Ki})$ where $g_j(\cdot)$ are unspecified.
- Estimate by backfitting and here by smoothing spline for each $g_j(\cdot)$

```
. * Generalized additive model
. gam lnhwage educatn hours, df(3)

177 records merged.

Generalized Additive Model with family gauss, link ident.

Model df     =    7.003                          No. of obs =       177
Deviance     =  93.1255                          Dispersion =   .547807

-------------------------------------------------------------------------
     lnhwage |    df    Lin. Coef.  Std. Err.        z        Gain   P>Gain
-------------+-----------------------------------------------------------
     educatn |   3.001   .1032296   .0197596     5.224     16.384   0.0003
       hours |   3.002    .000146   .0000804     1.816      3.228   0.1994
       _cons |       1    2.19816   .0556323    39.512          .        .
-------------------------------------------------------------------------
Total gain (nonlinearity chisquare) =    19.612 (4.003 df), P = 0.0006
```

- Plot each $g_j(\cdot)$ function
  - looks like education linear or quadratic; hours linear



6. Stata commands

- Command `kernel` does kernel density estimate.
- Command `lpoly` does several nonparametric regressions
  - kernel is default
  - local linear is option `degree(1)`
  - local polynomial of degree p is option `degree(p)`
- Command `lowess` does Lowess.
- Stata 15 command `npregress` does local constant and local linear for one or more regressors with bandwidth chosen by leave-on-out cross validation.
- For semiparametric use add-ons `semipar`, `sls`, `gam`
  - `gam` requires MS Windows.

6. References

- A. Colin Cameron and Pravin K. Trivedi (2005), *Microeconometrics: Methods and Applications (MMA)*, chapter 9, Cambridge Univ. Press.
- A. Colin Cameron and Pravin K. Trivedi (2009), *Microeconometrics using Stata (MUS)*, chapter 2.6, Stata Press.