Day 3B
Nonparametrics and Bootstrap

© A. Colin Cameron
Univ. of Calif.- Davis

Frontiers in Econometrics
Bavarian Graduate Program in Economics

Based on A. Colin Cameron and Pravin K. Trivedi (2009,2010), Microeconometrics using Stata (MUS), Stata Press. and A. Colin Cameron and Pravin K. Trivedi (2005), Microeconometrics: Methods and Applications (MMA), C.U.P.

March 21-25, 2011

1. Introduction

- Brief discussion of nonparametric and semiparametric methods and the bootstrap.

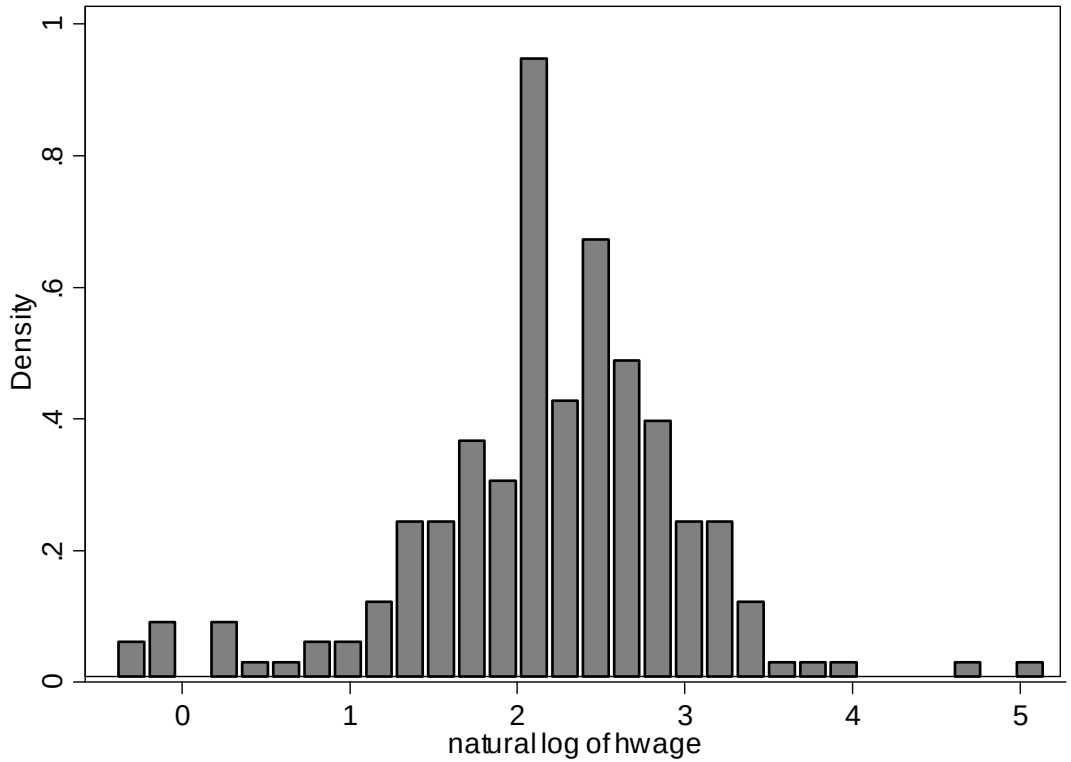
2. Nonparametric (kernel) density estimation

- 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.
- 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.

- Formula: $\widehat{f}_{HIST}(x_0) = \frac{1}{2Nh}\sum_{i=1}^{N}\mathbf{1}(x_0 - h < x_i < x_0 + h)$ or

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

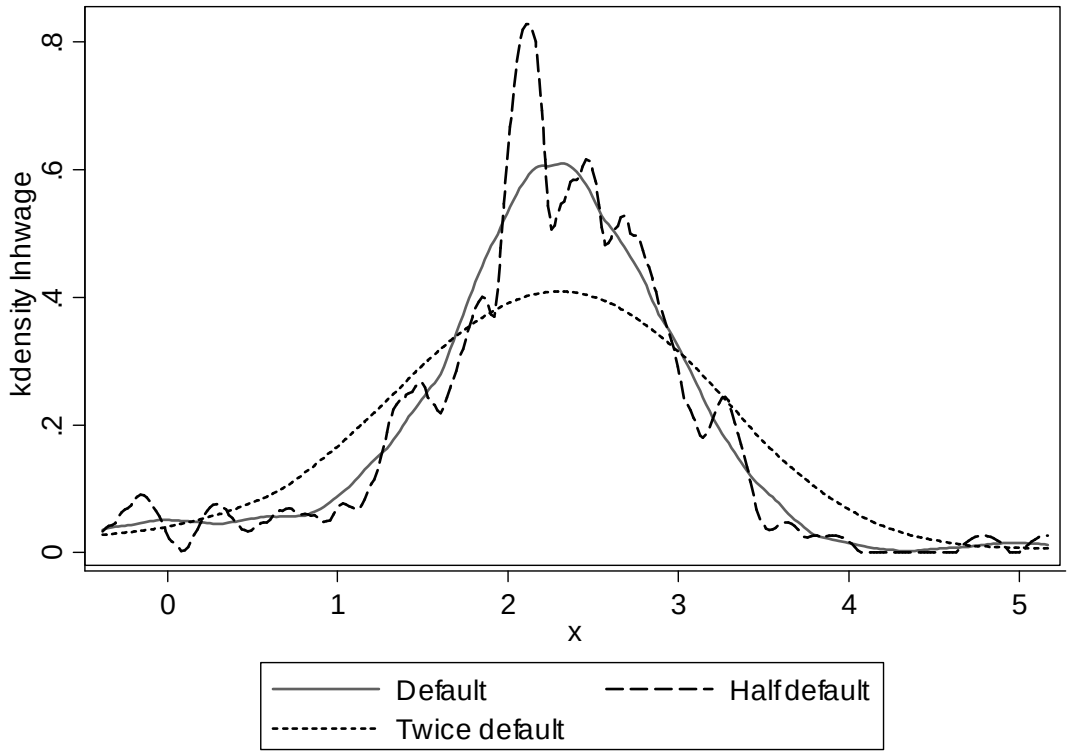
- Data example: histogram of lnwage for 175 observations
  - Varies with the bin width (or equivalently the number of bins)
  - Here 30 bins, each of width $2h \simeq 0.20$ so $h \simeq 0.10$.



- Kernel density estimate of $f(x_0)$ replaces $\mathbf{1}(A)$ by kernel $K(A)$:

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

- Data example: kernel of lnwage for 175 observations
  - Epanechnikov kernel $K(z) = 0.75(1 - z^2) \times \mathbf{1}(|z| < 1)$
  - $h = 0.07$ (oversmooths), 0.21 (default) or 0.63 (undersmooths)



Implementation

- Stata examples are
  - `kdensity y` uses defaults
  - `kdensity y, bw(0.2)` manually set bandwidth
  - `kdensity y, normal` overlays the $\mathcal{N}[\bar{y}, s^2]$ density
  - `hist y, kdensity` gives both histogram and kernel estimates
- Key is choice of bandwidth
  - The default can oversmooth: may need to decrease `bw()`
- Less important is choice of kernel: default is Epanechnikov.
- Other smooth estimators exist including k-nearest neighbors. But usually no reason to use anything but kernel.

3. Kernel regression: Local average estimator

- The regression model is $y_i = m(x_i) + u_i$, $u_i \sim$ i.i.d. $(0, \sigma(x_i)^2)$.
  - The functional form $m(\cdot)$ is not specified, so NLS not possible.
- If many obs have $x = x_0$ use the average of the $y_i's$ at $x_i = x_0$:

$$\begin{aligned} \widehat{m}(x_0) &= \left(\sum_{i:\ x_i = x_0} y_i\right) / \left(\sum_{i:\ x_i = x_0} 1\right) \\ &= \left(\sum_{i=1}^{N} \mathbf{1}\left(x_i = x_0\right) y_i\right) / \left(\sum_{i=1}^{N} \mathbf{1}\left(x_i = x_0\right)\right) \end{aligned}$$

- Instead few values of y_i at $x = x_0$ so do local average estimator:

$$\widehat{m}(x_0) = \left(\sum_{i=1}^{N} w_{i0} y_i\right) / \left(\sum_{i=1}^{N} w_{i0}\right)$$

  - where weights $w_{i0} = w(x_i, x_0)$ are largest for x_i close to x_0.
- Evaluate at a variety of points x_0 gives regression curve.
- Different methods use different weight functions $w_{i0} = w(x_i, x_0)$

Common nonparametric regression estimators

- 1. Kernel estimate of $m(x_0)$ replaces $\mathbf{1}(x_i - x_0 = 0)$ by $K\left(\frac{x_i - x_0}{h}\right)$

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

- 2. Local linear estimate of $\widehat{m}(x_0)$ minimizes w.r.t. a_0 and b_0

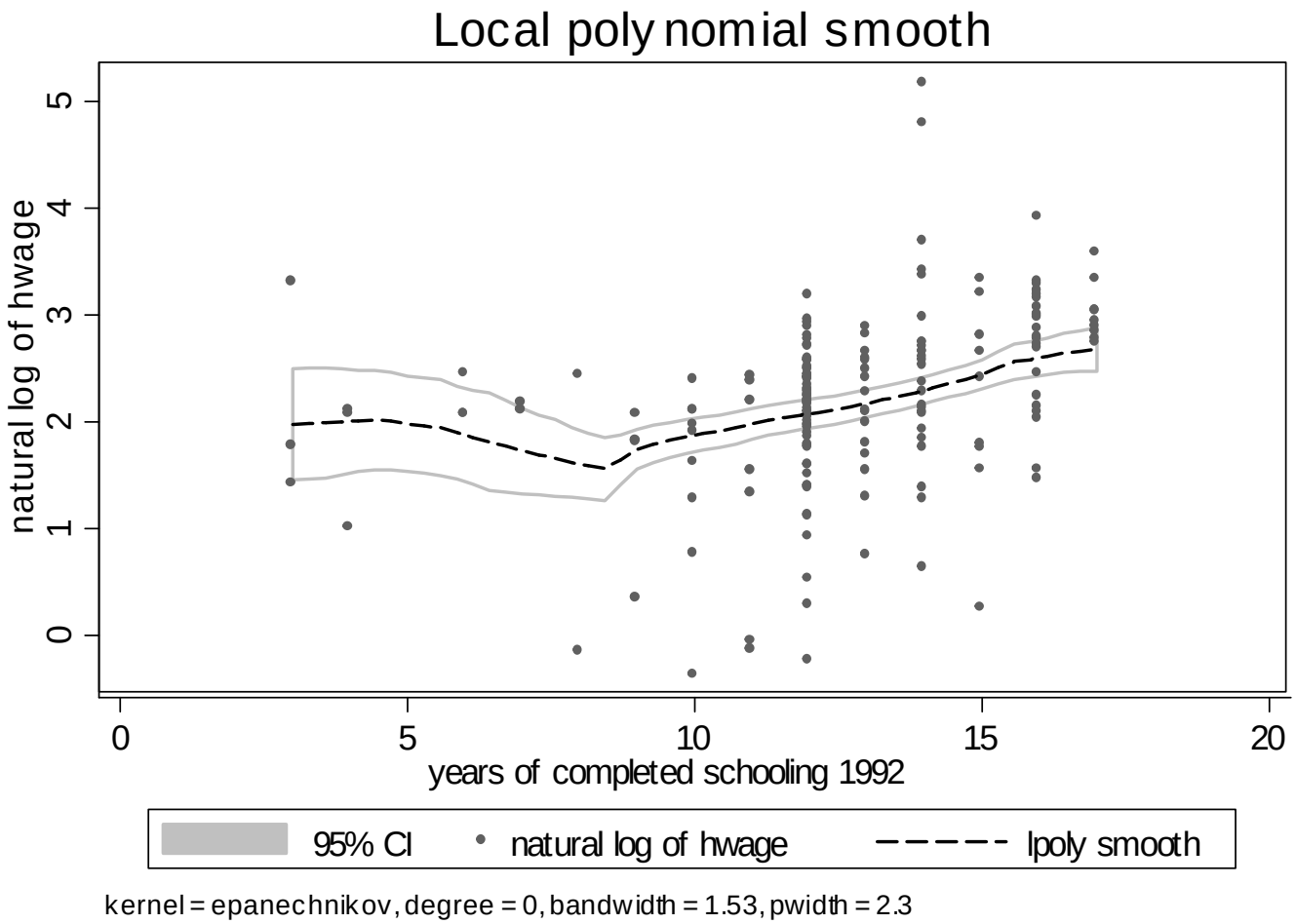
$$\frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right)(y_i - a_0 - b_0(x_i - x_0))^2$$

  - Motivation: Kernel estimate is equivalent to $\widehat{m}(x_0)$ minimizes $\frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x_i - x_0}{h}\right)(y_i - m_0)^2$ with respect to m_0.
  - better on endpoints

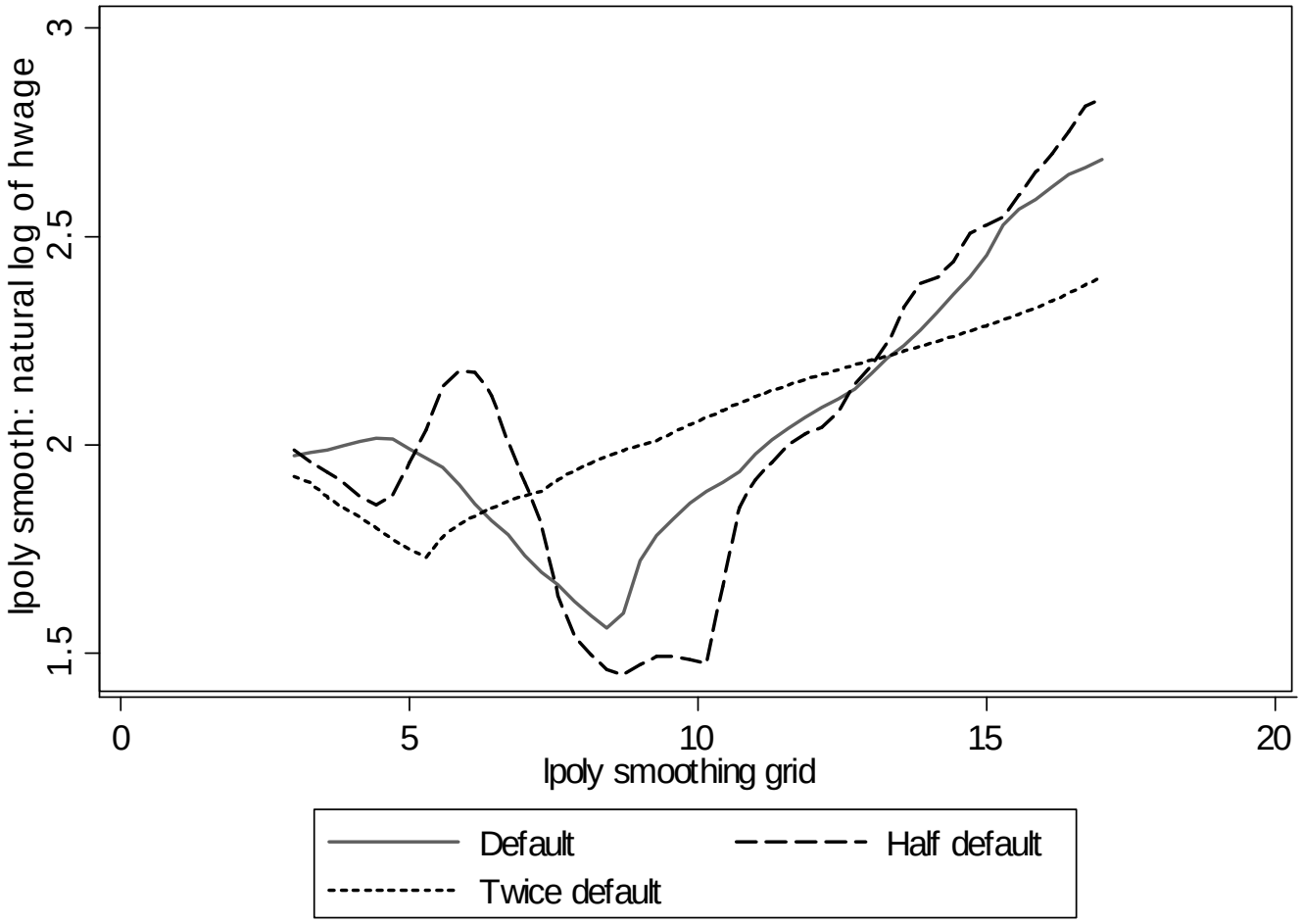
- 3. Lowess (locally weighted scatterplot smoothing)
  - variation of local linear with variable bandwidth, tricubic kernel and downweighting of outliers.

- 4. K-Nearest neighbors
  - Average the $y_i's$ for the k $x_i's$ that are closest to x_0

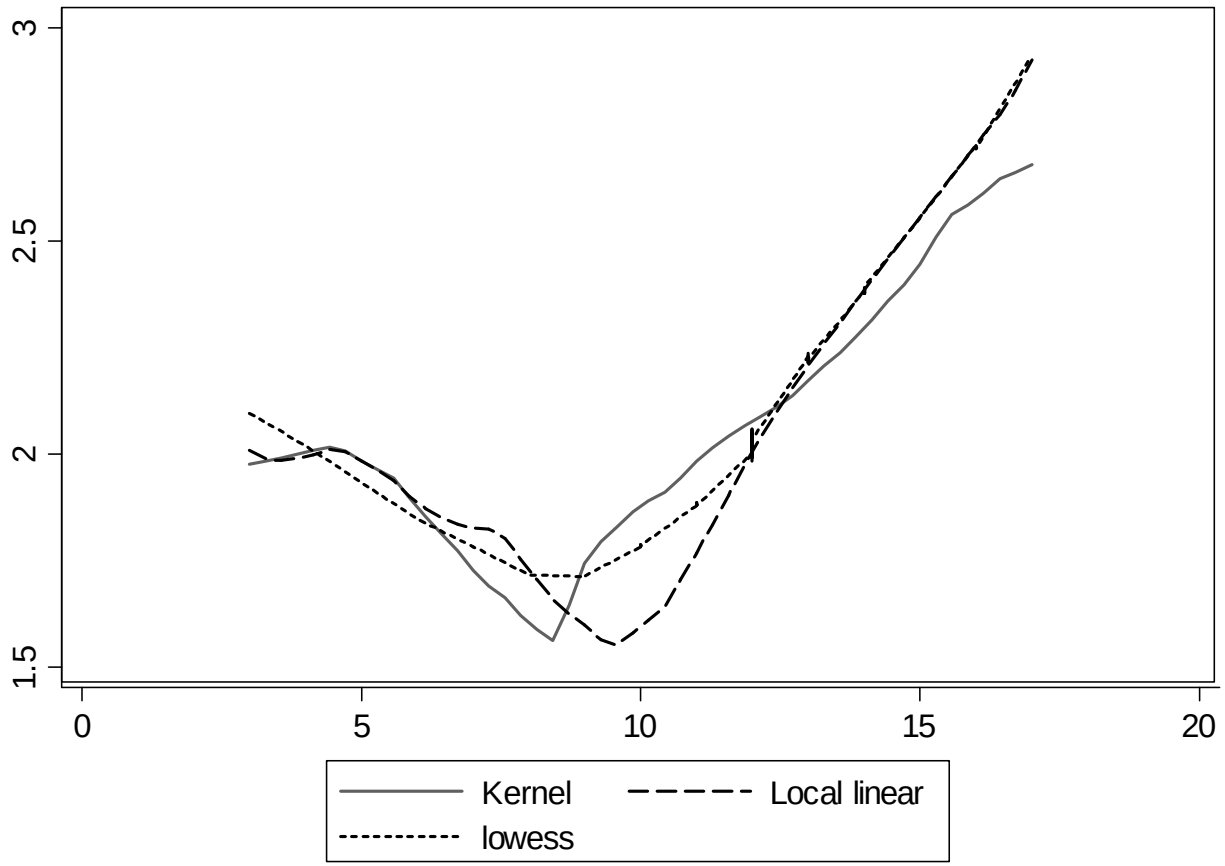
- Kernel regression with 95% confidence bands, default Kernel (Epanechnikov) and default bandwidths
  - `lpoly lnhwage educatn, ci msize(medsmall)`



- Kernel regression with three bandwidths: default, half and double.
  - smoother with larger bandwidth



- 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.

4. 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

Leading semiparametric examples

- Partially linear model

$$\mathsf{E}[y_i|\mathbf{x}_i, \mathbf{z}_i] = \mathbf{x}_i'\boldsymbol{\beta} + \lambda(\mathbf{z}_i)$$

  - Estimate $\lambda(\cdot)$ nonparametrically and ideally $\sqrt{N}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{d} \mathcal{N}[\mathbf{0}, \mathbf{V}]$

- Single-index model

$$\mathsf{E}[y_i|\mathbf{x}_i] = g(\mathbf{x}_i'\boldsymbol{\beta})$$

  - Estimate $g(\cdot)$ nonparametrically and ideally $\sqrt{N}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{d} \mathcal{N}[\mathbf{0}, \mathbf{V}]$
  - 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

- Generalized additive model

$$\mathsf{E}[y_i|\mathbf{x}_i] = g_1(x_{1i}) + \cdots + g_K(x_{Ki})$$

5. Bootstrap estimate of standard error

- Basic idea is view $\{(y_1, \mathbf{x}_1), ..., (y_N, \mathbf{x}_N)\}$ as the population.
- Then obtain B random samples from this population
  - Get B estimates $\widehat{\theta}_1, ..., \widehat{\theta}_S$.
  - Then estimate $\text{Var}[\widehat{\theta}]$ using the usual standard deviation of the B estimates

$$\widehat{\text{V}}[\widehat{\theta}] = \frac{1}{B-1}\sum_{b=1}^{B}(\widehat{\theta}_s - \overline{\widehat{\theta}})^2, \quad \text{where } \overline{\widehat{\theta}} = \frac{1}{B}\sum_{b=1}^{B}\widehat{\theta}_b.$$

  - Square root of this is called a bootstrap standard error.
- To get B different samples of size N we resample with replacement from $\{(y_1, \mathbf{x}_1), ..., (y_N, \mathbf{x}_N)\}$
  - In each bootstrap sample some original data points appear more than once while others not appear at all.

Regression application

- Data: Doctor visits (count) and chronic conditions. $N = 50$.

```
Contains data from musbootdata.dta
  obs:            50
 vars:             3                          16 Apr 2010 10:32
 size:           750 (99.9% of memory free)
------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
------------------------------------------------------------------------------
docvis          int    %8.0g                  number of doctor visits
age             float  %9.0g                  Age in years / 10
chronic         byte   %8.0g                  = 1 if a chronic condition
------------------------------------------------------------------------------
Sorted by:

. summarize
```

Variable	Obs	Mean	Std. Dev.	Min	Max
docvis	50	4.12	7.82106	0	43
age	50	4.162	1.160382	2.6	6.2
chronic	50	.28	.4535574	0	1

Bootstrap standard errors after Poisson regression

- Use option `vce(boot)`
  - Set the seed!
  - Set the number of bootstrap repetitions!

```
. * Compute bootstrap standard errors using option vce(bootstrap) to
. poisson docvis chronic, vce(boot, reps(400) seed(10101) nodots)

Poisson regression                              Number of obs      =         50
                                                Replications       =        400
                                                Wald chi2(1)       =       3.50
                                                Prob > chi2        =     0.0612
Log likelihood = -238.75384                     Pseudo R2          =     0.0917
```

docvis	Observed Coef.	Bootstrap Std. Err.	z	P>|z|	Normal-based [95% Conf.	Interval]
chronic	.9833014	.5253149	1.87	0.061	-.0462968	2.0129
_cons	1.031602	.3497212	2.95	0.003	.3461607	1.717042

- Bootstrap se = 0.525 versus White robust se = 0.515.

Results vary with seed and number of reps

```
. * Bootstrap standard errors for different reps and seeds
. quietly poisson docvis chronic, vce(boot, reps(50) seed(10101))

. estimates store boot50

. quietly poisson docvis chronic, vce(boot, reps(50) seed(20202))

. estimates store boot50diff

. quietly poisson docvis chronic, vce(boot, reps(2000) seed(10101))

. estimates store boot2000

. quietly poisson docvis chronic, vce(robust)

. estimates store robust

. estimates table boot50 boot50diff boot2000 robust, b(%8.5f) se(%8.5f)
```

Variable	boot50	boot50~f	boot2000	robust
chronic	0.98330	0.98330	0.98330	0.98330
	0.47010	0.50673	0.53479	0.51549
_cons	1.03160	1.03160	1.03160	1.03160
	0.39545	0.32575	0.34885	0.34467

legend: b/se

Leading uses of bootstrap standard errors

- Sequential two-step m-estimator
  - First step gives $\widehat{\alpha}$ used to create a regressor $z(\widehat{\alpha})$
  - Second step regresses y on x and $z(\widehat{\alpha})$
  - Do a paired bootstrap resampling (x, y, z)
  - e.g. Heckman two-step estimator.
- 2SLS estimator with heteroskedastic errors (if no White option)
  - Paired bootstrap gives heteroskedastic robust standard errors.
- Functions of other estimates e.g. $\widehat{\theta} = \widehat{\alpha} \times \widehat{\beta}$
  - replaces delta method
  - Clustered data with many small clusters, such as short panels.
    - ★ Then resample the clusters.
    - ★ But be careful if model includes cluster-specific fixed effects.

For these in Stata need to use prefix command `bootstrap:`

The bootstrap: general algorithm

- A general bootstrap algorithm is as follows:
  - **1.** Given data $\mathbf{w}_1, ..., \mathbf{w}_N$
    - ★ draw a bootstrap sample of size N (see below)
    - ★ denote this new sample $\mathbf{w}_1^*, ..., \mathbf{w}_N^*$.
  - **2.** Calculate an appropriate statistic using the bootstrap sample. Examples include:
    - ★ (a) estimate $\widehat{\theta}^*$ of θ;
    - ★ (b) standard error $s_{\widehat{\theta}^*}$ of estimate $\widehat{\theta}^*$
    - ★ (c) $t-$statistic $t^* = (\widehat{\theta}^* - \widehat{\theta})/s_{\widehat{\theta}^*}$ centered at $\widehat{\theta}$.
  - **3.** Repeat steps 1-2 B independent times.
    - ★ Gives B bootstrap replications of $\widehat{\theta}_1^*, ..., \widehat{\theta}_B^*$ or $t_1^*, \ldots, t_B^*$ or .....
  - **4.** Use these B bootstrap replications to obtain a bootstrapped version of the statistic (see below).

Implementation

- Number of bootstraps: B high is best but increases computer time.
  - CT use 400 for se's and 999 for tests and confidence intervals.
  - Defaults are often too low. And set the seed!
- Various resampling methods
  - 1. Paired (or nonparametric or empirical dist. func.) is most common
    - ★ $\mathbf{w}_1^*, ..., \mathbf{w}_N^*$ obtained by sampling with replacement from $\mathbf{w}_1, ..., \mathbf{w}_N$.
  - 2. Parametric bootstrap for fully parametric models.
    - ★ Suppose $y|\mathbf{x} \sim F(\mathbf{x}, \boldsymbol{\theta}_0)$ and generate y_i^* by draws from $F(\mathbf{x}_i, \widehat{\boldsymbol{\theta}})$
  - 3. Residual bootstrap for regression with additive errors
    - ★ Resample fitted residuals $\widehat{u}_1, ..., \widehat{u}_N$ to get $(\widehat{u}_1^*, ..., \widehat{u}_N^*)$ and form new $(y_1^*, \mathbf{x}_1), ..., (y_N^*, \mathbf{x}_N)$.
- Need to resample over i.i.d. observations
  - resample over clusters if data are clustered
    - ★ But be careful if model includes cluster-specific fixed effects.
  - resample over moving blocks if data are serially correlated.

Asymptotic refinement

- The simplest bootstraps are no better than usual asymptotic theory
  - advantage is easy to implement, e.g. standard errors.
- More complicated bootstraps provide asymptotic refinement
  - this may provide a better finite-sample approximation.
- Conventional asymptotic tests (such as Wald test).
  - α = nominal size for a test, e.g. $\alpha = 0.05$.
  - Actual size= $\alpha + O(N^{-1/2})$.
- Tests with asymptotic refinement
  - Actual size= $\alpha + O(N^{-1})$.
  - asymptotic bias of size $O(N^{-1}) < O(N^{-1/2})$ is smaller asymptotically.
  - But need simulation studies to confirm finite sample gains.
    - ★ e.g. if $N = 100$ then $100/N = O(N^{-1}) > 5/\sqrt{N} = O(N^{-1/2})$.

Asymptotically pivotal statistic

- Asymptotic refinement bootstraps an asymptotically pivotal statistic
  - this means limit distribution does not depend on unknown parameters.
- An estimator $\widehat{\theta} - \theta_0 \stackrel{a}{\sim} \mathcal{N}[0, \sigma^2_{\widehat{\theta}}]$ is not asymptotically pivotal
  - since $\sigma^2_{\widehat{\theta}}$ is an unknown parameter.
- But the studentized $t-$statistic is asymptotically pivotal
  - since $t = (\widehat{\theta} - \theta_0)/s_{\widehat{\theta}} \stackrel{a}{\sim} \mathcal{N}[0, 1]$ has no unknown parameters.
- So bootstrap Wald test statistic to get tests and confidence intervals with asymptotically refinement.
- For confidence intervals can also use BC (bias-corrected) and BCa methods.
- Econometricians rarely use asymptotic refinement.

```
. * Bootstrap confidence intervals: normal-based, percentile, BC, and BCa
. quietly poisson docvis chronic, vce(boot, reps(999) seed(10101) bca)

. estat bootstrap, all

Poisson regression                              Number of obs      =         50
                                                Replications       =        999
```

docvis	Observed Coef.	Bias	Bootstrap Std. Err.	[95% Conf.	Interval]	
chronic	.98330144	-.0244473	.54040762	-.075878	2.042481	(N)
				-.1316499	2.076792	(P)
				-.0820317	2.100361	(BC)
				-.0215526	2.181476	(BCa)
_cons	1.0316016	-.0503223	.35257252	.3405721	1.722631	(N)
				.2177235	1.598568	(P)
				.2578293	1.649789	(BC)
				.3794897	1.781907	(BCa)

```
(N)     normal confidence interval
(P)     percentile confidence interval
(BC)    bias-corrected confidence interval
(BCa)   bias-corrected and accelerated confidence interval
```

- (N) is observed coefficient $\pm$ 1.96 $\times$ bootstrap s.e.
- (P) is 2.5 to 97.5 percentile of the bootstrap estimates $\widehat{\beta}^*_1, ..., \widehat{\beta}^*_B$.
- (BC) and (BCa) have asymptotic refinement.

Bootstrap failure

- The following are cases where standard bootstraps fail
  - so need to adjust standard bootstraps.
- GMM (and empirical likelihood) in over-identified models
  - For overidentified models need to recenter or use empirical likelihood.
- Nonparametric Regression:
  - Nonparametric density and regression estimators converge at rate less than root-N and are asymptotically biased.
  - This complicates inference such as confidence intervals.
- Non-Smooth Estimators: e.g. LAD.

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 has no built-in commands for the semiparametric estimators
  - These methods are not easy to automate as no easy way to automate bandwidth choice and treatment of outliers.
- For bootstrap use option `,vce(boot)` or command `bootstrap:`
  - set the seed!!

7. Appendix: 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 $freq/2Nh$ (then 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

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

Appendix: 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
  - $K(z) = 0.75(1 - z^2) \times \mathbf{1}(|z| < 1)$
  - more weight on data at center. less weight at end
- More generally kernel function must satisfy conditions including
  - Continuous, $K(z) = K(-z)$, $\int K(z)dz = 1$, $\int K(z)dz = 1$, tails go to zero.