Lecture 2
OLS

1

OLS Estimation - Assumptions

• CLM Assumptions

(**A1**) DGP: $\mathbf{y} = \mathbf{X}\,\beta + \boldsymbol{\varepsilon}$ is correctly specified.

(**A2**) $E[\boldsymbol{\varepsilon}|\mathbf{X}] = 0$

(**A3**) $Var[\boldsymbol{\varepsilon}|\mathbf{X}] = \sigma^2\,\mathbf{I}_T$

(**A4**) $\mathbf{X}$ has full column rank – rank($\mathbf{X}$)=k-, where $T \geq k$.

• From assumptions (**A1**), (**A2**), and (**A4**)

$$\Rightarrow \mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'}\,\mathbf{y}$$

We define $\mathbf{e} = \mathbf{y\text{-}Xb}$ $\quad\Rightarrow \mathbf{X'e} = \mathbf{X'}\,(\mathbf{y\text{-}Xb})\ \mathbf{X'y} - \mathbf{X'X}(\mathbf{X'X})^{-1}\mathbf{X'y} = \mathbf{0}$

• Now, we will study the properties of **b**.

Small Sample Properties of OLS

• *Small sample* = For *all* sample sizes –i.e., for all values of T (or N).

• The OLS estimator of β is

$$\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'}\,\mathbf{y}$$

$$\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'}\,\mathbf{y} = (\mathbf{X'X})^{-1}\,\mathbf{X'}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}) = \boldsymbol{\beta} + (\mathbf{X'X})^{-1}\mathbf{X'}\boldsymbol{\varepsilon}$$

$$= \boldsymbol{\beta} + (\mathbf{X'X})^{-1}\,\Sigma_i\,\mathbf{x}_i'\varepsilon_i$$

$$= \boldsymbol{\beta} + \Sigma_i\,\mathbf{v}_i'\varepsilon_i$$

⇒ **b** is a vector of random variables.

• We condition on an **X**, then show that results do not depend on that particular **X**.

⇒ The results must be general –i.e., independent of **X**.

Small Sample Properties of OLS

• $\mathbf{b} = \boldsymbol{\beta} + (\mathbf{X'X})^{-1}\mathbf{X'}\boldsymbol{\varepsilon} = \boldsymbol{\beta} + \Sigma_i\,\mathbf{v}_i'\varepsilon_i$

• Properties

(1) $E[\mathbf{b}|\mathbf{X}] = \boldsymbol{\beta}$

(2) $Var[\mathbf{b}|\mathbf{X}] = E[(\mathbf{b}-\boldsymbol{\beta})\,(\mathbf{b}-\boldsymbol{\beta})'|\mathbf{X}] = \sigma^2\,(\mathbf{X'X})^{-1}$

(3) Gauss-Markov Theorem: **b** is BLUE (MVLUE).

(4) If (**A5**) $\boldsymbol{\varepsilon}|\mathbf{X} \sim N(\mathbf{0}, \sigma^2\mathbf{I}_T)$ ⇒ $\mathbf{b}|\mathbf{X} \sim N(\boldsymbol{\beta}, \sigma^2(\mathbf{X'X})^{-1})$

⇒ $b_k|\mathbf{X} \sim N(\beta_k, \sigma^2(\mathbf{X'X})_{kk}^{-1})$

(the marginals of a multivariate normal are also normal.)

Note: Under (**A5**), **b** is also the MLE. Thus, it has all the nice MLE properties: efficiency, consistency, sufficiency and invariance!

Sampling Distribution of b

- $\mathbf{b} = \boldsymbol{\beta} + (\mathbf{X'X})^{-1}\mathbf{X'}\boldsymbol{\varepsilon} = \boldsymbol{\beta} + \Sigma_i \mathbf{v}_i'\varepsilon_i$

Let's generate some y_i's. Set $\boldsymbol{\beta}$ = .4; then, the DGP is:

$$\mathbf{y} = (.4)\,\mathbf{X} + \boldsymbol{\varepsilon}$$

(1) Generate **X** (to be treated as numbers). Say **X** ~ N(2,4)

⇒ x_1=**3.22**, x_2=**2.18**, x_3=-0.37, ..., x_T =1.71

(2) Generate $\boldsymbol{\varepsilon}$ ~ N(0,1)

⇒ draws ε_1=**0.52**, ε_2=**-1.23**, ε_3=1.09, ..., ε_T=-0.09

(3) Generate $\mathbf{y}$ = .4 **X** + $\boldsymbol{\varepsilon}$

⇒ y_1 = .4 * 3.22 + **0.52** = 1.808

y_2 = .4 * **2.18** + (**-1.23**) = -0.358

y_3 = .4 * (-0.37) + 1.09 = 0.942

... y_T = .4 * 1.71 + (-0.09) = 0.594

(4) Generate $\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'y} = \Sigma_i (x_i - \bar{x})(y_i - \bar{y}) / \Sigma_i (x_i - \bar{x})^2$

Sampling Distribution of b

- We want to generate **b**'s. Steps
  - (1) Generate **X** (to be treated as numbers). Say **X** ~ N(2,4)
  - (2) Generate $\boldsymbol{\varepsilon}$ ~ N(0,1)
  - (3) Generate $\mathbf{y}$ = .4 **X** + $\boldsymbol{\varepsilon}$
  - (4) Generate $\mathbf{b} = (\mathbf{X'X})^{-1}\mathbf{X'y} = \Sigma_i (x_i - \bar{x})(x_i - \bar{y}) / \Sigma_i (x_i - \bar{x})^2$

Conditioning on step (1), we can repeat (2)-(4) B times, say 1,000 times. Then, we are able to generate a sampling distribution for **b**.

We can obviously play with T; say T=100; 1,000; 10,000.

We can check: $E[\mathbf{b}|\mathbf{X}] = (1/B)\,\Sigma_i\, b_i = \boldsymbol{\beta}$?

We can calculate the variance of Var[**b**|**X**].

Sampling Distribution of b – Code in R

• Steps (1)-(4) in R to generate **b**, with a sample of size T=100:

```
> T <- 100                                    # sample size
> x <- rnorm(T,2,2)                           # generate x
> ep <- rnorm(T,0,1)                          # generate errors
> y <- .4*x + ep
> b <- solve(t(x)%*% x)%*% t(x)%*%y           # OLS regression
```

Run these commands B times to get the sampling distribution of **b**.

Then, calculate means, variances, skewness, kurtosis coefficients, etc.

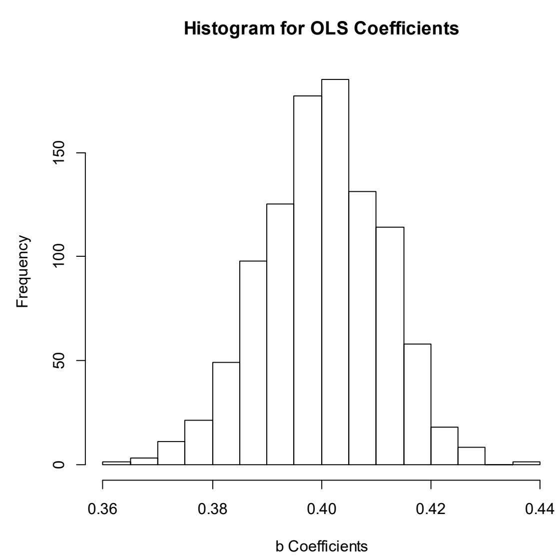
<u>Note</u>: You need to initialize a vector that collects the b's. For example:

```
Allbs = NULL                  # Initialize vector that collects the b
Allbs = rbind(Allbs,b)        # accumulate b as rows
```

Sampling Distribution of b – Code in R

For T=100
B = 1,000

Mean[b] = **0.39947**
SD[b] = 0.01154
Ex Kurt[b] = -0.0568



Estimating the Variance of b

• We want to estimate Var[**b**], the unconditional variance of **b.**

$$\text{Var}[\mathbf{b}] = E_{\mathbf{x}}[\text{Var}[\mathbf{b}|\mathbf{X}]] + \text{Var}_{\mathbf{x}}[E[\mathbf{b}|\mathbf{X}]] = \sigma^2 E[(\mathbf{X}'\mathbf{X})^{-1}].$$

But, the population parameter σ^2 is unknown.

• We consider how to use the sample data to estimate this matrix.

• The ultimate goals are to estimate C.I. for $\boldsymbol{\beta}$ and to test hypotheses about $\boldsymbol{\beta}$. We need estimates of the variability of the distribution.

• We use the residuals instead of the disturbances:
Natural estimator: $\mathbf{e}'\mathbf{e}/T$ –sample counterpart for $\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon}/T$

• Imperfect observation of disturbances: $\varepsilon_i = e_i + (\boldsymbol{\beta} - \mathbf{b})'\mathbf{x}_i$

Estimating Var[b|X]

• We want to estimate $E[\mathbf{e}'\mathbf{e}|\mathbf{X}]$
Recall $\mathbf{e} = \mathbf{My} = \mathbf{M}\boldsymbol{\varepsilon} \Rightarrow \mathbf{e}'\mathbf{e} = \boldsymbol{\varepsilon}'\mathbf{M}'\mathbf{M}\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}'\mathbf{M}\boldsymbol{\varepsilon}$ (**M**: residual maker)

• We have a quadratic form. Recall Theorem 7.3, from Math Review:
Theorem 7.3. Let the $T\times 1$ vector $y \sim N(0, \sigma^2 I_T)$ and **M** be a symmetric idempotent matrix of rank m. Then,

$$y'\mathbf{M}y/\sigma^2 \sim \chi_{tr(M)}^2.$$

• We have already established: $tr(\mathbf{M}) = tr(\mathbf{I}_T) - tr(\mathbf{P}) = T - k$.
Recall trace property: $tr(ABC) = tr(CAB)$

$$\Rightarrow tr(\mathbf{P}) = tr(\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}') = tr(\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}) = tr(\mathbf{I}_k) = k$$

• Recall that if $Z \sim \chi_v^2 \Rightarrow E[Z] = v$

$$\Rightarrow E[\mathbf{e}'\mathbf{e}/\sigma^2|\mathbf{X}] = (T-k)$$

$$\Rightarrow E[\mathbf{e}'\mathbf{e}|\mathbf{X}] = (T-k)\sigma^2 \qquad \text{(Downward bias of } \mathbf{e}'\mathbf{e}/T.)$$

Estimating Var[b|X]

• $E[\mathbf{e}'\mathbf{e}|\mathbf{X}] = (T-k)\sigma^2$

$\Rightarrow s^2 = \mathbf{e}'\mathbf{e}/(T-k)$ unbiased estimator of σ^2

$(T-k)$ is referred as a *degrees of freedom* correction.

• True conditional covariance matrix: $\text{Var}[\mathbf{b}|\mathbf{X}] = \sigma^2 (\mathbf{X'X})^{-1}$

$\Rightarrow$ the natural estimator is $s^2(\mathbf{X'X})^{-1}$

This estimator gives us the *standard errors (SE)* of the individual coefficients. For example, for the b_k coefficient:

$$\text{SE}[b_k|\mathbf{X}] = \text{sqrt}[s^2(\mathbf{X'X})^{-1}]_{kk}$$

Q: How does the conditional covariance $\sigma^2(\mathbf{X'X})^{-1}$ differ from the unconditional one, $\sigma^2 E[(\mathbf{X'X})^{-1}]$?

Example (Greene): Gasoline Regression Results

```
----------------------------------------------------------------------
Ordinary     least squares regression ............
LHS=G        Mean                 =      226.09444
             Standard deviation   =       50.59182
             Number of observs.   =             36
Model size   Parameters           =              7
             Degrees of freedom   =             29
Residuals    Sum of squares       =      778.70227
             Standard error of e  =        5.18187  <*****  sqr[778.70227/(36 - 7)]
Fit          R-squared            =         .99131
             Adjusted R-squared   =         .98951
Model test   F[  6,    29] (prob) =    551.2(.0000)
--------+-------------------------------------------------------------
Variable| Coefficient    Standard Error  t-ratio  P[|T|>t]   Mean of X
--------+-------------------------------------------------------------
Constant|   -7.73975        49.95915        -.155   .8780
      PG|   -15.3008***      2.42171       -6.318   .0000      2.31661
       Y|     .02365***       .00779        3.037   .0050      9232.86
   TREND|    4.14359**       1.91513        2.164   .0389      17.5000
     PNC|    15.4387        15.21899        1.014   .3188      1.67078
     PUC|   -5.63438         5.02666       -1.121   .2715      2.34364
     PPT|   -12.4378**       5.20697       -2.389   .0236      2.74486
--------+-------------------------------------------------------------
Create  ; trend=year-1960$
Namelist; x=one,pg,y,trend,pnc,puc,ppt$
Regress ; lhs=g ; rhs=x$
```

Example (Greene): Gasoline Regression Results

Matrix - XX

[7, 7] Cell:

X'X

	1	2	3	4	5	6	7
1	36	83.398	332383	630	60.148	84.371	98.815
2	83.398	248.04	838669	1878.67	164.992	251.287	301.047
3	332383	838669	3.18054e+009	6.4692e+006	591999	859749	1.01845e+006
4	630	1878.67	6.4692e+006	14910	1277.71	1972.56	2384.18
5	60.148	164.992	591999	1277.71	114.542	171.935	205.811
6	84.371	251.287	859749	1972.56	171.935	267.306	322.011
7	98.815	301.047	1.01845e+006	2384.18	205.811	322.011	391.845

(X'X)$^{-1}$

	1	2	3	4	5	6	7
1	92.9516	-1.58239	-0.0142015	3.45656	-6.3863	2.85512	-5.3368
2	-1.58239	0.218408	0.000315846	-0.0830075	-0.665387	-0.02755	0.287509
3	-0.0142015	0.000315846	2.25808e-006	-0.000547423	0.000144609	-0.000330383	0.000995983
4	3.45656	-0.0830075	-0.000547423	0.136591	-0.061965	0.0821448	-0.251126
5	-6.3863	-0.665387	0.000144609	-0.061965	8.62577	-1.43238	-1.23058
6	2.85512	-0.02755	-0.000330383	0.0821448	-1.43238	0.940991	-0.360893
7	-5.3368	0.287509	0.000995983	-0.251126	-1.23058	-0.360893	1.00971

s^2(X'X)$^{-1}$

	1	2	3	4	5	6	7
1	2495.92	-42.49	-0.381335	92.8149	-171.484	76.6652	-143.303
2	-42.49	5.86466	0.00848103	-2.2289	-17.8668	-0.739767	7.72013
3	-0.381335	0.00848103	6.06335e-005	-0.0146993	0.003883	-0.00887138	0.026744
4	92.8149	-2.2289	-0.0146993	3.6677	-1.66387	2.20574	-6.74318
5	-171.484	-17.8668	0.003883	-1.66387	231.618	-38.4621	-33.0434
6	76.6652	-0.739767	-0.00887138	2.20574	-38.4621	25.2673	-9.69062
7	-143.303	7.72013	0.026744	-6.74318	-33.0434	-9.69062	27.1126

OLS Estimation – Example in R

- **Example:** 3 Factor Fama-French Model (continuation) for IBM:

```
Returns <- read.csv("http://www.bauer.uh.edu/rsusmel/phd/k-dis-ibm.csv", head=TRUE,
sep=",")

b <- solve(t(x)%*% x)%*% t(x)%*%y                      # b = (X'X)^-1 X' y  (OLS regression)
e <- y - x%*%b                                          # regression residuals, e
RSS <- as.numeric(t(e)%*%e)                             # RSS
R2 <- 1 - as.numeric(RSS)/as.numeric(t(y)%*%y)          # R-squared  (see Later)
Sigma2 <- as.numeric(RSS/(T-k))                         # Estimated σ² = s²
SE_reg <- sqrt(Sigma2)                                  # Estimated σ – Regression stand error
Var_b <- Sigma2*solve(t(x)%*% x)                        # Estimated Var[b|X] = s² (X'X)^-1
SE_b <- sqrt(diag(Var_b))                               # SE[b|X]
t_b <- b/SE_b                                           # t-stats (See Chapter 4)
```

OLS Estimation – Example in R

```
> R2
[1] 0.5679013
> SE_reg
[1] 0.1973441
> t(b)
                x1        x2         x3
[1,] -0.2258839  1.061934  0.1343667  -0.3574959
> SE_b
                x1        x2        x3
[1,] 0.01095196  0.26363344   0.35518792  0.37631714
> t(t_b)
                x1        x2        x3
[1,] -20.62498 4.028071 0.3782976 -0.9499857
```

Note: Again, you should get the same numbers using R's lm (linear model fit):

```
fit <- lm(y~x -1)
summary(fit)
```

Bootstrapping

• The *bootstrap* is a method for estimating the sampling distribution of a statistic, $\theta = \theta(x_1, x_2, x_3, ..., x_N)$, by resampling from the ED, where

$$x_1, x_2, x_3, ..., x_N \sim i.i.d.\ F \text{ (unknown)}$$

• We usually have one sample of size N. We do not have replicated samples to get a sampling distribution for θ.

• A large sample from a finite population should be well representative of the full population itself. Replicated samples (with replacement) from the original sample, which would just be an *i.i.d.* sample from the empirical CDF (ED), could be regarded as proxies for replicated samples from the population itself, provided N is large.

• Now, we have a *bootstrap distribution.*

Bootstrapping

• The DGP that generated the original data is unknown, and so it cannot be used to generate simulated data:

⇒ the *bootstrap DGP* estimates the unknown true DGP.

• Recall the *Fundamental Theorem of Statistics*: The empirical distribution (ED) of a set of independent drawings of a RV generated by some DGP converges to the true CDF of the RV under the DGP. This is just as true of simulated drawings.

• Then, an easy choice for an approximating distribution is the ED of the observed data. That is, the ED becomes a "fake population." John Fox (2005, UCLA):

"The population is to the sample as the sample is to the bootstrap samples."

Bootstrapping – Bootstrap Distribution

• Suppose we have a dataset with N *i.i.d.* observations drawn from $F(x)$:

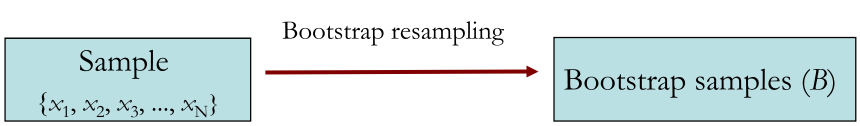
$$\{x_1, x_2, x_3, ..., x_N\}$$

From the ED, F^*, we sample with replacement N observations:

$$\{x_1^*, x_2^*, x_3^*, ..., x_N^*\}$$

This is an *empirical bootstrap sample*, which is a resample of the same size N as the original data, drawn from F^*.

• With the *i.i.d.* population assumption , we construct a number B of resamples from the ED. This is the bootstrap distribution.



Bootstrapping – Bootstrap Distribution

• Remark: Bootstrapping uses the ED –i.e., sample- as if it were the true CDF. Potentially we have N^N resamples. Unless N is small, we use a small number of resamples, B.

• For any statistic θ computed from the original sample, we compute a statistic θ^* by the same formula, but using the resampled data.

• θ^* is computed by resampling the original data; we can compute many θ^* by resampling many times from F^*. Say, we resample θ^* B times. This is the bootstrap distribution of θ defined as.

$$H_B(q) = P^*[\ \theta(x_1^*, x_2^*, x_3^*, ..., x_N^*; F^*) \leq q\ ]$$

P^* = probabilities under the bootstrap distribution.

• Since B is small, $H_B(q)$ is itself estimated by a Monte Carlo.

Bootstrapping – Consistency

• Two source of errors:

(i) Assuming $\{x_{b1}^*, x_{b2}^*, x_{b3}^*, ..., x_{bN}^*\}$ are resamples from F.

(ii) Estimating H_B by a Monte Carlo.

An adequately large B usually makes ignoring (ii) OK.

• Under suitable conditions, the bootstrap distribution, H_B, is asymptotically first-order equivalent to the asymptotic distribution of the statistic of interest, H. The bootstrap distribution, H_B, is *consistent*.

• Typical "suitable conditions" for the mean, under an L_2 metric: *i.i.d.* data with $E[X^2] < \infty$.

• Rule of thumb: If θ admits a CLT, a bootstrap is at least consistent.

Bootstrapping – Consistency: Delta Method

• Delta Theorem: If θ admits a CLT and g() is a smooth function, then, g(θ) also admits a CLT.

• Following the rule of thumb, the bootstrap should be consistent for g(θ) if it is consistent for θ.

• Many situations where the bootstrap fails are due to the lack of smoothness of g().

Bootstrapping – Consistency: Second-order

• Q: If θ admits a CLT, why use a bootstrap?
A: A theoretical results establishes that for certain types of statistics, the bootstrap approximation is more accurate than the approximation provided by the CLT.

The CLT (a normal is symmetric) cannot capture information about the skewness in the finite sample distribution of θ. The bootstrap does (think of an Edgeworth expansion).

This is called "*second-order accuracy of the bootstrap*."

• In practice, the accuracy of bootstraps is known through simulations.

Bootstrapping – Variations

• The bootstrap is simple and built around the ED and the *i.i.d.* assumption, but it is not limited to those situations.

• Variations:

- If the ED is used, the method is usually called the *nonparametric bootstrap*.
- If the *y*'s and the *x*'s are sampled together, this method is sometimes called the *paired bootstrap* –for example, in a regression.
- If blocks of data are sample together, the method is called *block bootstrap* –for example, in the presence of correlated data.
- If the data from the ED is smoothed before drawing from it, the method is called *smoothed bootstrap*.

Bootstrapping – In practice: Steps

• We have a collection of estimated θ^*:

$$\{\hat{\theta}_1^*, \hat{\theta}_2^*, \hat{\theta}_3^*, ..., \hat{\theta}_B^*\}.$$

From this collection of $\hat{\theta}^*$'s, we can compute the mean, the variance, skewness, draw a histogram, etc., and confidence intervals.

• Bootstrap Steps:

1. From the original sample, draw random sample with size N.
2. Compute statistic θ from the resample in 1: $\hat{\theta}_1^*$.
3. Repeat steps 1 & 2 B times ⇒ Get B statistics: $\{\hat{\theta}_1^*, \hat{\theta}_2^*, \hat{\theta}_3^*, ..., \hat{\theta}_B^*\}$
4. Compute moments, draw histograms, etc. for these B statistics.

• Recall that with a large enough B, the LLN allows us to use the $\hat{\theta}^*$'s to estimate the distribution of θ, $F(\theta)$. The variation in $\hat{\theta}$ is well approximated by the variation in $\hat{\theta}^*$.

Bootstrapping – Pros and Cons

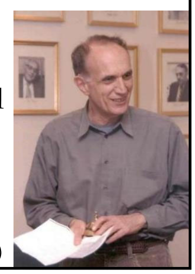
• Efron (1979) is the seminal paper. But, the related literature is older.

• It became popular in the 1980's due to the explosion of computer power.

• Disadvantage: Only *consistent* results, no finite sample results.

• Advantage: Simplicity.

• While it is a method for improving estimators, it is well known as a method for estimating standard errors, bias and constructing C.I. for parameters.



Bradley Efron (1938, USA)

Bootstrapping in Economics

• Bootstrapping provides a very general method to estimate a wide variety of statistics. It is most useful when:

(1) A "formula" is problematic because its assumptions are dubious.

(2) A formula holds only as $N \to \infty$, but N is not very big.

(3) A formula is complicated or it has not even been worked out yet.

• The most common econometric applications are situations where you have a consistent estimator of a parameter of interest, but it is hard or impossible to calculate its standard error or its C.I.

• Bootstrapping is easiest to implement if the estimator is "smooth," $\sqrt{N}$-consistent and based on an *i.i.d.* sample. In other situations, it is more complicated.

Bootstrapping: Simple example

• You are interested in the relation between CEO's education (**X**) and firm's long-term performance (**y**). You have 1,500 observations on both variables. You estimate the correlation coefficient, ρ, with its sample counterpart, *r*. You find the correlation to be very low.

• Q: How reliable is this result? The distribution of *r* is complicated. You decide to use a bootstrap to study the distribution of *r*.

• Randomly construct a sequence of *B* samples (all with *N*=1,500). Say,

$B_1 = \{(x_1,y_1), (x_3,y_3), (x_6,y_6), (x_6,y_6), ..., (x_{1458},y_{1458})\} \quad \Rightarrow r_1$

$B_2 = \{(x_5,y_5), (x_7,y_7), (x_{11},y_{11}), (x_{12},y_{12}), ..., (x_{1486},y_{1486})\} \quad \Rightarrow r_2$

. . . .

$B_B = \{(x_2,y_2), (x_2,y_2), (x_2,y_2), (x_3,y_3), ..., (x_{1499},y_{1499})\} \quad \Rightarrow r_B$

Bootstrapping: Simple example – Remarks

• We rely on the observed data. We take it as our "fake population" and we sample from it *B* times.

• We have a collection of *bootstrap subsamples*.

• The sample size of each bootstrap subsample is the same, *N*. Thus, some elements are repeated.

• Now, we have a collection of estimators of ρ_i's: $\{r_1, r_2, r_3, ..., r_B\}$. We can do a histogram and get an approximation of the probability distribution. We can calculate its mean, variance, kurtosis, confidence intervals, etc.

Bootstrapping: Simple example in R

• We bootstrap the correlation between the returns of IBM & the S&P 500, using monthly data 1990-2018, with ***B* = 1,000** (with *r*= 0.341194).

```
dat_xy<-read.table("http://www.bauer.uh.edu/rsusmel/phd/k-dis-ibm.csv", sep=",", header=T)
sim_size = 1000

library(boot)
# function to obtain correlation from the data
cor_xy <- function(data, i) {
        d <- data[i,]
        return(cor(d$IBM,d$SP500))
}
# bootstrapping with sim_size replications
boot.samps <- boot(data=dat_xy, statistic=cor_xy, R=sim_size)

# view stored bootstrap samples and compute mean
boot.samps$t                # show 1,000 boostrapped correlation coeff
mean(boot.samps$t)          # calculates mean of 1,000 correlation coeff
```

Bootstrapping: Simple example in R

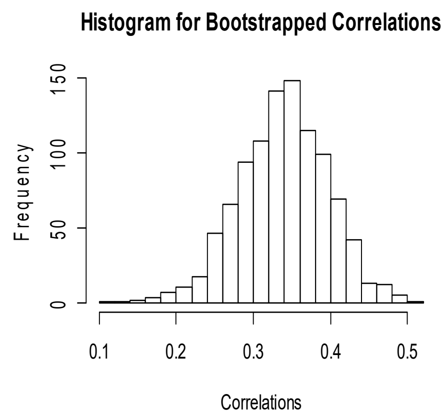
```
> boot.samps$t[1:20]                #show first 20 bootstrapped correlations coeff  -i.e., r
 [1] 0.2394950 0.3915076 0.2474868 0.3340402 0.2650797 0.2483522 0.3811410
 [8] 0.3713640 0.3648230 0.3375564 0.4060621 0.3203225 0.3025321 0.2993649
[15] 0.3293505 0.3500546 0.3153994 0.3028840 0.3809710 0.4645849

>mean(boot.samps$t)                 # calculates mean of boot.samps$t (size = 1,000)
[1] 0.3432176
>
> sd(boot.samps$t)                  # calculates SD of boot.samps$t (size = 1,000)
[1] 0.06141714
>
># Empirical 95% C.I. for bootstrapped r's
> quantile(boot.samps$t,.025)
    2.5%
0.2131369
> quantile(boot.samps$t,.975)
   97.5%
0.4602344
```

Bootstrapping: Simple example in R

```
> # Elegant histogram
> hist(boot.samps$t,main="Histogram for Bootstrapped Correlations",
+      xlab="Correlations", breaks=20)

>mean(boot.samps$t)
[1] 0.3432176
>
> sd(boot.samps$t)
[1] 0.06141714
>
> quantile(boot.samps$t,.025)
    2.5%
0.2131369
> quantile(boot.samps$t,.975)
   97.5%
0.4602344
```



Bootstrapping: How many bootstraps?

• It is not clear. There are many theorems on asymptotic convergence, but there are no clear rules regarding B. There are some suggestions.

Efron and Tibsharani's (1994) textbook recommends B=200 as enough. (Good results with B as low as 25!)

The purposed of the bootstrap plays a role in B. For example, in hypothesis testing, increasing B increases the power of test. In this context, Andrews and Buchinsky (2000, Econometrica) propose a 3-step process to select B. Davidson and Mackinnon (2001) attempt to improve on A & B's procedure –i.e., they get a lower B.

D&M suggest selecting B using a pretest procedure. In the D&M simulations, on average, B is between 300 and 2,400.

Bootstrapping: How many bootstraps?

• Wilcox's (2010) textbook recommends "599 [...] for general use."

Rule of thumb: Start with B=100, then, try B=1,000, and see if your answers have changed by much. Increase bootstraps until you get stability in your answers.

• But, be careful. Recall that we have N^N possible subsamples.

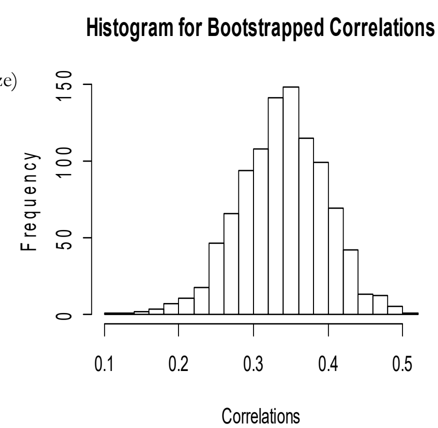
• Note: A *jack-knife* is a special kind of bootstrap. Each bootstrap subsample has all but one of the original elements of the list. For example, if original N=20, then there are 20 jack-knife subsamples.

Bootstrapping: How many bootstraps?

Example: We bootstrap the correlation between IBM returns and S&P 500 returns, using ***B* = 1,000**. (Sample r = 0.3411941.)

```
> # view bootstrap results
> sim_size = 1000
> boot.samps
Call:
boot(data = dat_xy, statistic = cor_xy, R = sim_size)


Bootstrap Statistics :
     original      bias     std. error
t1* 0.3411941 0.003224424  0.05860686
> mean(boot.samps$t)
[1] 0.3444186
> quantile(boot.samps$t,.025)
     2.5%
0.2283223
> quantile(boot.samps$t,.975)
    97.5%
0.4535492
```



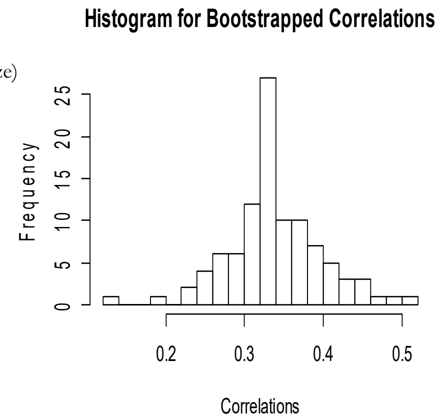
• Results do not change that much.

Bootstrapping: How many bootstraps?

Example: We bootstrap the correlation between IBM returns and S&P 500 returns, using ***B*** **= 100**. (Sample *r* = 0.3411941.)

```
> # view bootstrap results
> sim_size = 100
> boot.samps          #show results
Call:
boot(data = dat_xy, statistic = cor_xy, R = sim_size)

Bootstrap Statistics :
        original      bias          std. error
t1*     0.3411941     -0.003574917  0.06051004
> mean(boot.samps$t)
[1] 0.3376192
> quantile(boot.samps$t,.025)
   2.5%
0.2301927
> quantile(boot.samps$t,.975)
   97.5%
0.4609417
```



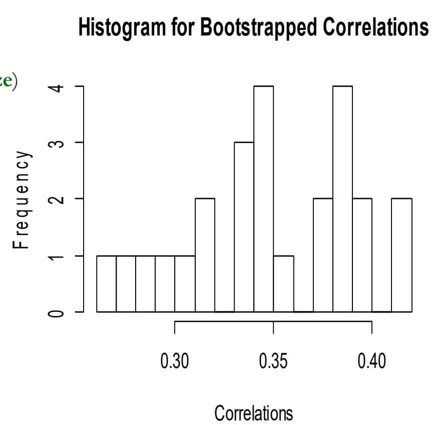
• Results do not change that much.

Bootstrapping: How many bootstraps?

Example: We bootstrap the correlation between IBM returns and S&P 500 returns, using ***B*** **= 25**. (Sample *r* = 0.3411941.)

```
> # view bootstrap results
> sim_size = 25
> boot.samps$t
Call:
boot(data = dat_xy, statistic = cor_xy, R = sim_size)

Bootstrap Statistics :
        original      bias          std. error
t1*     0.3411941     0.006412186   0.04230658
mean(boot.samps$t)
[1] 0.3476063
> quantile(boot.samps$t,.025)
   2.5%
0.2706706
> quantile(boot.samps$t,.975)
   97.5%
0.4110658
```



• 97.5% C.I., a bit different. But, overall, within range.

Bootstrapping: Bias

• You can estimate the bias of the estimator $\hat{\theta}$:

$$Bias(\hat{\theta}) = (1/B)\Sigma_r \hat{\theta}(r) - \hat{\theta}$$

Note: In the OLS case, $\hat{\theta}$ = **b** is an unbiased estimator, but as an estimate, the bias can be non-zero. This estimate must be analyzed along the SE's.

Example: In the previous bootstrapping correlations exercise (B=25), R (boots.samps) displays the bias (relative to sample r = **0.3411941**, "original" below):

Bootstrap Statistics :

	original	bias	std. error
t1*	**0.3411941**	**0.006412186**	0.04230658

bias = 0.3476063 – **0.3411941** = **0.0064122**

Bootstrapping: Var[b]

• Some assumptions in the CLM are not reasonable –for example, normality. Note that by assuming normality, we also assume the sampling distribution of **b**.

• We can use a bootstrap to estimate the sampling distribution of **b**. Then, we can estimate the Var[**b**].

• Monte Carlo (MC=repeated sampling) method:

1. Estimate model using full sample (of size N) ⇒ we get **b**
2. Repeat B times:

- Draw T observations from the sample, *with replacement*
- Estimate **β** with **b**(r).

3. Estimate variance with

$$\mathbf{V}_{boot} = (1/B)\ [\mathbf{b}(r) - \mathbf{b}][\mathbf{b}(r) - \mathbf{b}]'$$

Bootstrapping: Var[b]

• In the case of one parameter, say $\mathbf{b}_1$: Estimate variance with

$$\text{Var}_{\mathbf{boot}}[\mathbf{b}_1] = (1/B)\Sigma_r\,[\mathbf{b_1}(r) - \mathbf{b_1}\,]^2$$

• You can also estimate Var[$\mathbf{b}_1$] as the variance of $\mathbf{b}_1$ in the bootstrap

$$\text{Var}_{\mathbf{boot}}[\mathbf{b}_1] = (1/B)\Sigma_r\,[\mathbf{b_1}(r) - \text{mean}(\mathbf{b_{1\text{-}r}})\,]^2;$$

$$\text{mean}(\mathbf{b_{1\text{-}r}}) = (1/B)\Sigma_r\,\mathbf{b_1}$$

Note: Obviously, this method for obtaining standard errors of parameters is most useful when no formula has been worked out for the standard error (SE), or the formula is complicated –for example, in some 2-step estimation procedures.

Bootstrapping: Var[b] in R

Example: 3 Factor Fama-French Model for IBM (continuation):

```
> fit <- lm(y ~ x -1)
> summary(fit)
Call:
lm(formula = y ~ x - 1)

Residuals:
    Min       1Q   Median       3Q      Max
-0.48076 -0.16205  0.02532  0.18265  0.36144

Coefficients:
       Estimate   Std. Error   t value    Pr(>|t|)
x      -0.22588   0.01095     -20.625     < 2e-16 ***
xx1     1.06193   0.26363      4.028      6.98e-05 ***
xx2     0.13437   0.35519      0.378       0.705
xx3    -0.35750   0.37632     -0.950      0.343
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1973 on 332 degrees of freedom
Multiple R-squared:  0.5679,    Adjusted R-squared:  0.5627
F-statistic: 109.1 on 4 and 332 DF,  p-value: < 2.2e-16
```

Bootstrapping: Var[b] in R

Example: 3 Factor Fama-French Model for IBM (continuation):

```
> sim_size <- 1000
> library(boot)
> # function to obtain b from the data
> bols_xy <- function(data, i) {
+ d <-data[i,]
+ y1 <- d$IBM; rf <- d$Rf; y <- y1 - rf
+ x1 <- d$Rm_Rf; x2 <- d$SMB; x3 <- d$HML
+ T <- length(x1)
+ x0 <- matrix(1,T,1)
+ x <- cbind(x0,x1,x2,x3)
+ b_i <- solve(t(x)%*% x)%*% t(x)%*%y
+   return(b_i)
+ }
> # bootstrapping with sim_size replications
> boot.samps <- boot(data=Returns, statistic=bols_xy, R=sim_size)
>
> # view stored bootstrap samples and compute mean
```

Bootstrapping: Var[b] in R

Example: 3 Factor Fama-French Model (continuation):

```
> boot.samps              # original: OLS b;  bias: [OLS b - mean(boot.samps$t)];  std. error: sqrt{Var(b)}

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data = Returns, statistic = bols_xy, R = sim_size)

Bootstrap Statistics :
          original          bias            std. error
t1*       -0.2258839        -0.0001378402   0.01087016
t2*       1.0619343         0.0129093074    0.27145374
t3*        0.1343667        0.0431736591    0.41107154
t4*       -0.3574959         0.0121591493   0.41944286

> apply(boot.samps$t,2,mean)
[1] -0.2260218  1.0748436  0.1775404 -0.3453368
> apply(boot.samps$t,2,sd)
[1] 0.01087016 0.27145374 0.41107154 0.41944286
```

Bootstrapping: Var[b] in R

• **Example**: 3 Factor Fama-French Model (continuation):

```
Bootstrap Statistics :
        original        bias            std. error
t1*     -0.2258839      -0.0001378402   0.01087016
t2*     1.0619343       0.0129093074    0.27145374
t3*      0.1343667      0.0431736591    0.41107154
t4*     -0.3574959       0.0121591493   0.41944286

> bs_m <- apply(boot.samps$t,2,mean)
> bols_m <- fit$coefficients
> bols_m - bs_m                         #Bootstrap b bias
        x               xx1             xx2             xx3
-0.0001378402   0.0129093074    0.0431736591    0.0121591493

> apply(boot.samps$t,2,sd)
[1] 0.01087016  0.27145374      0.41107154      0.41944286
> bols_sd <- coef(summary(fit))[, "Std. Error"]
> bols_sd
        x               xx1             xx2             xx3
0.01095196      0.26363344      0.35518792      0.3763 1714
```

Bootstrapping: Var[b] in R

Example: 3 Factor Fama-French Model for IBM (continuation):

• Comparison of SE(b): OLS vs Bootstrapped (***B* = 1,000**)

	b (OLS)	b (Boot)	SE (OLS)	SE (Boot)
Constant	-0.2259	**-0.2259**	0.0110	**0.0110**
Market	1.0619	**1.0848**	0.2636	**0.2738**
SMB	0.1344	**0.1629**	0.3552	**0.4111**
HML	-0.3575	**-0.3482**	0.3763	**0.4280**

Note: Mean of bootstrapped **b's**:
-0.22586 1.08483 0.16294 -0.34820

Bootstrapping: Estimating Var[b]

> # print the first 10 of B=1,000 bootstrap samples

```
            x        xMkt_RF        xSMB         xHML
 [1,] -6.109007e-03 0.9186830 -0.1299534100 -0.163421636
 [2,] -1.757503e-03 0.8333006 -0.2067565390 -0.147604991
 [3,] -3.907573e-03 0.9746878 -0.2870744815 -0.169189619
 [4,]  1.596103e-03 0.9185157 -0.2937731120 -0.296972497
 [5,] -8.409239e-03 0.7309406 -0.0681714313 -0.149883639
 [6,] -1.998929e-03 0.9133751 -0.3001713380 -0.315913280
 [7,] -6.289286e-03 0.9441856 -0.2276894034 -0.058924929
 [8,] -5.533354e-03 0.8210057 -0.2221866298 -0.078512341
 [9,] -6.152301e-03 1.0389917 -0.2592958758 -0.237930809
[10,] -3.778058e-03 0.9544829 -0.1859554067 -0.217702583
```

- From the B samples, we compute variances and SD as usual.

Bootstrapping: Var[b] in R

- **Example**: 3 Factor Fama-French Model (continuation):

```
> sim_size <- 100
> boot.samps <- boot(data=Returns, statistic=bols_xy, R=sim_size)
> boot.samps

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:
boot(data = Returns, statistic = bols_xy, R = sim_size)

Bootstrap Statistics :
       original       bias           std. error
t1*    -0.2258839     -0.001875955   0.01079366
t2*    1.0619343 -    0.048363387    0.27853542
t3*     0.1343667     0.029859101    0.38133799
t4*    -0.3574959     0.061470947    0.43286410
> apply(boot.samps$t,2,mean)
[1] -0.2277599  1.0135709  0.1642258 -0.2960250
> apply(boot.samps$t,2,sd)
[1] 0.01079366 0.27853542 0.38133799 0.43286410
```

Bootstrapping: Some Remarks

• Q: How reliable is bootstrapping?

- There is still no consensus on how far it can be applied, but for now nobody is going to dismiss your results for using it.
- There is a general agreement that for normal (or close to normal) distributions it works well.
- Bootstrapping is more problematic for skewed distributions.
- It can be unreliable for situations where there are not a lot of observations. Typical example in finance: estimation of quantiles in the tails of returns distributions.

Note: We presented two simple examples. There are many variations that have not been discussed.

Bootstrapping: Some Remarks

• Always keep in mind: Convergence in distribution of a random sequence does not imply convergence of moments – see Billingsley's textbook (1968).

• Suppose you are interested in estimating a moment, say the variance (but, it can be skewness or kurtosis), through a bootstrap. The consistency of the bootstrap distribution, however, does not guarantee the consistency of the variance of the bootstrap distribution (the "*bootstrap variance*") as an estimator of the asymptotic variance.

In the usual situations we find in econometrics, the bootstrap variance tends to work well and is consistent; but for other statistics (skewness or kurtosis) the consistency of the bootstrap estimator is difficult to establish.

Data Problems

"*If the data were perfect, collected from well-designed randomized experiments, there would hardly be room for a separate field of econometrics.*" Zvi Griliches (1986, **Handbook of Econometrics**)

• Three important data problems:
(1) Missing Data – very common, especially in cross sections and long panels.
(2) Outliers – unusually high/low observations.
(3) Multicollinearity – there is perfect or high correlation in the explanatory variables.

• In general, data problems are exogenous to the researcher. We cannot change the data or collect more data.

Missing Data

• *General Setup*
We have an indicator variable, s_i. If $s_i = 1$, we observe Y_i, and if $s_i = 0$ we do not observe Y_i.

Note: We always observe the missing data indicator s_i.

• Suppose we are interested in the population mean $\theta = E[Y_i]$.

• With a lot of information – large T–, we can learn $p = E[s_i]$ and $\mu_1 = E[Y_i | s_i = 1]$, but nothing about $\mu_0 = E[Y_i | s_i = 0]$.

• We can write: $\theta = p \cdot \mu_1 + (1 - p) \cdot \mu_0$.

Problem: Since even in large samples we learn nothing about μ_0, it follows that without additional information/assumptions there is no limit on the range of possible values for θ.

Missing Data

- *General Setup*
- Now, suppose the variable of interest is binary: $Y_i \in \{0, 1\}$. We also have an explanatory variable of Y_i, say W_i.

- Then, the natural (not data-informed) lower and upper bounds for μ_0 are 0 and 1 respectively.

- This implies bounds on θ:

$$\theta \in [\theta_{LB}, \theta_{UB}] = [p \cdot \mu_1, p \cdot \mu_1 + (1 - p)].$$

These bounds are *sharp*, in the sense that without additional information we can not improve on them.

Formally, for all values $\theta \in [\theta_{LB}, \theta_{UB}]$, we can find a joint distribution of (Y_i, W_i) that is consistent with the joint distribution of the observed data and with θ.

Missing Data

- Now, suppose we have the CLM: $y_i = \mathbf{x}_i\beta + \varepsilon_i$
- We use the selection indicator, s_i , where $s_i = 1$ if we can use observation *i*. Then,

$$\mathbf{b} = \boldsymbol{\beta} + (\Sigma_i\, s_i\, \mathbf{x}_i'\mathbf{x}_i\, / T)^{-1} (\Sigma_i\, s_i\, \mathbf{x}_i'\varepsilon_i\, / T)$$

- For unbiased (and consistent) results, we need $E[s_i\, \mathbf{x}_i'\varepsilon_i]=0$, implied by $E[\varepsilon_i | s_i\, \mathbf{x}_{ii}]=0$ (*)

A sufficient condition for (*) is $E[\varepsilon_i | \mathbf{x}_i]=0$, $s_i = h(\mathbf{x}_i)$.

<u>Note</u>:· Zero covariance assumption in the population, $E[\mathbf{x}_i'\varepsilon_i]= 0$, is not sufficient for consistency when $s_i = h(\mathbf{x}_i)$

⇒ selection is a function of $\mathbf{x}_i$ (*selection bias*).

Missing Data

Example of Selection Bias: Determinants of Hedging.

A researcher only observes companies that hedge. Estimating the determinants of hedging from this population will bias the results!

• If missing observations are randomly (exogenously) "selected," it is likely safe to ignore problem. Rubin (1976) calls this assumption "*missing completely at random*" (or MCAR).

In general, MCAR is rare. In general, it is more common to see "*missing at random,"* where missing data depends on observables (say, education, sex) but one item for individual *i* is NA (Not Available).

If in the regression we "control" for the observables that influence missing data, it is OK to delete the whole observation for *i*.

Missing Data

Otherwise, we can:

a. Fill in the blanks –i.e., *impute* values to the missing data– with averages, interpolations, or values derived from a model. For example, use the Data-augmentation methods in Bayesian analysis.

b. Use (inverse) probability weighted estimation. Here, we inflate or "over-weight" unrepresented subjects or observations.

c. Heckman selection correction. We build a model for the $h(x_i)$.

• Little and Rubin (2002) provide an overview of methods for analysis with missing data.

Outliers

• Many definitions: Atypical observations, extreme values, conditional unusual values, observations outside the expected relation, etc.

• In general, we call an *outlier* an observation that is numerically different from the data. But, is this observation a "mistake," say a result of measurement error, or part of the (heavy-tailed) distribution?

• In the case of normally distributed data, roughly 1 in 370 data points will deviate from the mean by 3xSD. Suppose T=1000. Then, 9 data points deviating from the mean by more than 3xSD indicates outliers. But, which of the 9 observations can be classified as an outliers?

Problem with outliers: They can affect estimates. For example, with small data sets, one big outlier can seriously affect OLS estimates.

Outliers: Identification

• Several identifications methods:

- *Eyeball*: Look at the observations away from a scatter plot.
- *Standardized residual*: Check for errors that are two or more standard deviations away from the expected value.
- *Leverage statistics*: It measures the difference of an independent data point from its mean. High leverage observations can be potential outliers. Leverage is measured by the diagonal values of the **P** matrix:

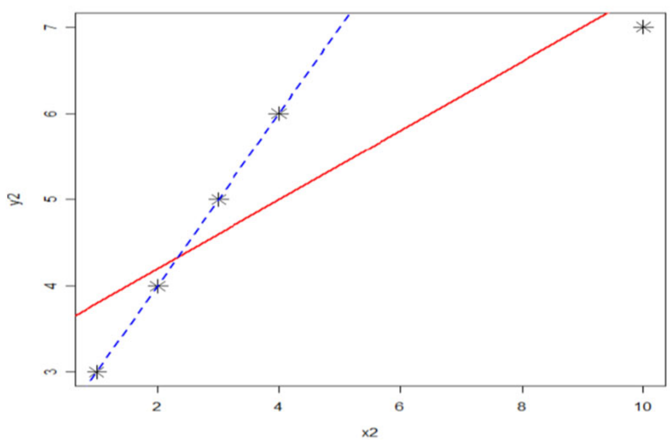
$$h_t = 1/T + (x_t - \bar{x})/[(T-1)s_x^2].$$

But, an observation can have high leverage, but no *influence*.

- *Influence statistics: Dif beta*. It measures how much an observation influences a parameter estimate, say b_j. Dif beta is calculated by removing an observation, say i, recalculating b_j, say $b_j(-i)$, taking the difference in betas and standardizing it. Then,

$$Dif\ beta_{j(-i)} = [b_j - b_j(-i)]/SE[b_j].$$

Outliers: Leverage & Influence



• Deleting the observation in the upper right corner has a clear effect on the regression line. This observation has *leverage* and *influence*.

Outliers: Influence

• A related popular influence statistic is *Distance D (as in Cook's D)*. It measures the effect of deleting an observation on the fitted values, say $\hat{y}_j$.

$$D_j = \Sigma_j \, [\hat{y}_j - \hat{y}_j(-i)]/[K \, \text{MSE}]$$

where K is the number of parameters in the model and MSE is mean square error of the regression model.

• The influence statistics are usually compared to some ad-hoc cut-off values used for identifying highly influential points, say $D_i > 4/T$.

• The analysis can also be carried out for groups of observations. In this case, we would be looking for blocks of highly influential observations.

Outliers: Summary of Rules of Thumb

• General rules of thumb used to identify outliers:

Measure	**Value**
abs(stand resid)	> 2
leverage	>(2k+2)/T
abs(*Dif Beta*)	> 2/sqrt(T)
Cook's D	> 4/T

Outliers: Example

Example: Cook's D for IBM returns using the 3 FF Factor Model

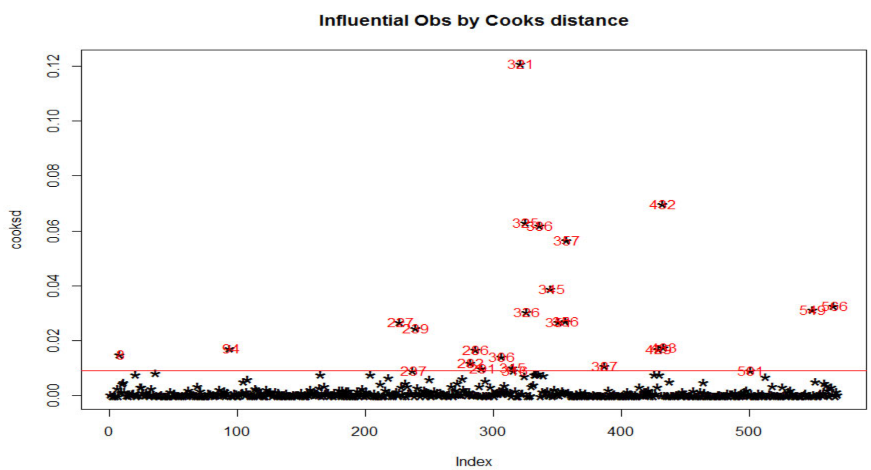
```
mod <- lm(y ~ x-1)
cooksd <- cooks.distance(mod)
# plot cook's distance
plot(cooksd, pch="*", cex=2, main="Influential Obs by Cooks distance")
# add cutoff line
abline(h = 4*mean(cooksd, na.rm=T), col="red")  # add cutoff line
# add labels
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4*mean(cooksd,
na.rm=T),names(cooksd),""), col="red")  # add labels

 # influential row numbers
influential <- as.numeric(names(cooksd)[(cooksd > 4*mean(cooksd, na.rm=T))])
 # print first 10 influential observations.
head(dat_xy[influential, ],n=10L)
```

Note: There are easier ways to plot Cook's D and identify the suspect outliers. The package *olsrr* can be used for this purpose too.

Outliers: Example

Example (continuation): Cook's D for IBM (3 Factor-Model)



Outliers: Example

Example (continuation): Cook's D for IBM (3 Factor-Model)

> \# print first 10 influential observations.
>head(dat_xy[influential,],n=10L)

	y	V1	Mkt_RF	SMB	HML
8	-0.16095068	1	0.0475	0.0294	0.0219
94	0.01266444	1	0.0959	-0.0345	-0.0835
227	-0.04237227	1	0.1084	-0.0224	-0.0403
237	-0.19083575	1	0.0102	0.0205	-0.0210
239	-0.30648638	1	0.0153	0.0164	0.0252
282	0.07787100	1	-0.0597	-0.0383	0.0445
286	0.20734626	1	0.0625	-0.0389	0.0117
291	0.15218986	1	0.0404	-0.0565	-0.0006
306	0.13928315	1	-0.0246	-0.0512	-0.0096
315	0.16196934	1	0.0433	0.0400	0.0253

Outliers: Example

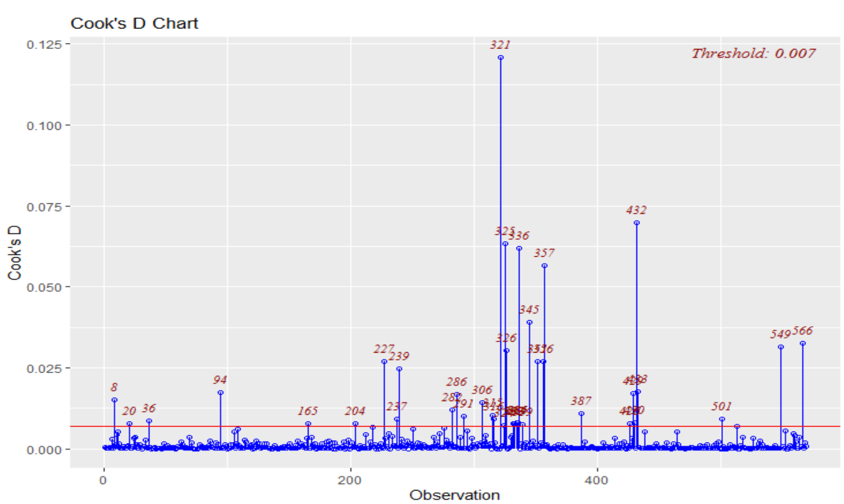
Example: Different tools to check for outliers for IBM returns
We will use the package *olsrr*.

```
library(olsrr)                          # need to install package olsrr
x_resid <- residuals(mod)
x_stand_resid <- x_resid/sd(x_resid)    # standardized residuals
sum(x_stand_resid > 2)                  # Rule of thumb count (5% count is OK)
x_lev <- ols_leverage(mod)              # leverage residuals
sum(x_lev >(2*k+2)/T)                   # Rule of thumb count (5% count is OK
ols_plot_resid_stand(mod)               # Plot standardized residuals
ols_plot_cooksd_bar(mod)                # Plot Cook's D measure
ols_plot_dffits(mod)                    # Plot Difference in fits
ols_plot_dfbetas(mod)                   # Plot Difference in betas

> sum(x_lev >(2*k+2)/T)
[1] 32                                   # 5%? = 32/569 = 0.0562
> sum(x_stand_resid > 2)
[1] 13                                   # 5%? = 13/569 = 0.0228
```

Outliers: Example

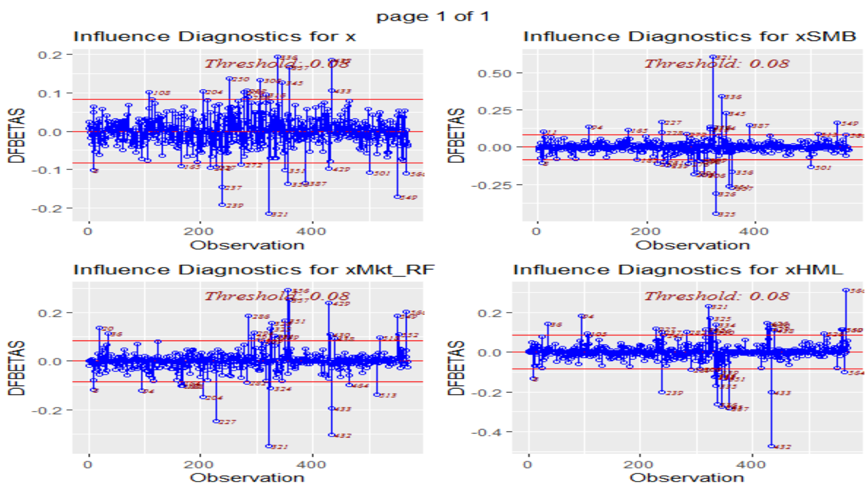
Example (continuation):
>ols_plot_cooksd_bar(mod)   # Plot Cook's D measure



Outliers: Example

Example (continuation):

>ols_plot_dfbetas(mod)



Outliers: Application - Rules of Thumb

• The histogram, Boxplot, and quantiles helps us see some potential outliers, but we cannot see which observations are potential outliers. For these, we can use Cook's D, Diffbeta's, standardized residuals and leverage statistics, which are estimated for each i .

Observation Type	Proportion	Cutoff	
Outlier	0.0356	2.0000	(abs(standardized residuals) > 2)
Outlier	0.1474	2/sqrt(T)	(diffit > 2/sqrt(1038)=0.0621)
Outlier	0.0501	4/T	(cookd > 4/1038=0.00385)
Leverage	0.0723	(2k+2)/T	(h=leverage > .00771)

Outliers: What to do?

• Typical solutions:

- Use a non-linear formulation or apply a transformation (log, square root, etc.) to the data.
- Remove suspected observations. (Sometimes, there are theoretical reasons to remove suspect observations. Typical procedure in finance: remove public utilities or financial firms from the analysis.)
- Winsorization of the data.
- Use dummy variables.
- Use LAD (quantile) regressions, which are less sensitive to outliers.
- Weight observations by size of residuals or variance (robust estimation).

• General rule: Present results with or without outliers.

Multicollinearity

• The **X** matrix is *singular* (perfect collinearity) or *near singular (multicollinearity)*.

• Perfect collinearity. Not much we can do. OLS will not work => **X'X** cannot be inverted. The model needs to be reformulated.

• Multicollinearity. OLS will work. $\boldsymbol{\beta}$ is unbiased. The problem is in $(\mathbf{X'X})^{-1}$. Consider the OLS estimator of β_k ⇒ $E[b_k] = \beta_k$

⇒ The variance of b_k is the kth diagonal element of $\sigma^2(\mathbf{X'X})^{-1}$.

We can show that the estimated variance of b_k is

$$\text{Var}[b_k|\mathbf{X}] = \frac{s^2}{[(1-R_{k.}^2)\sum_{i=1}^{n}(x_{ik}-\bar{x}_k)^2]}.$$

⇒ the higher $R_{k.}^2$ –i.e., the fit between $\boldsymbol{x}_k$ and the rest of the regressors–, the higher $\text{Var}[b_k|\mathbf{X}]$.

Multicollinearity

• The ratio $\frac{1}{(1-R_{k.}^2)}$ is called the Variance Inflation Factor of regressor *k*, or VIF_k.

It should be equal to 1 when $\boldsymbol{x_k}$ is unrelated to the rest of the regressors (including a constant). The higher it is, the higher the linear correlation between $\boldsymbol{x_k}$ and the rest of the regressors.

Multicollinearity: Signs

• Signs of Multicollinearity:
- Small changes in **X** produce wild swings in **b**.
- High R^2, but **b** has low t-stats –i.e., high standard errors
- "Wrong signs" or difficult to believe magnitudes in **b**.

• There is no *cure* for collinearity. Estimating something else is not helpful (transforming regressors, principal components, etc).

• There are "measures" of multicollinearity, such as the
- *Condition number* of **X** = max(singular value)/min(singular value)
- *Variance inflation factor* = $\text{VIF}_k = \frac{1}{(1-R_{k.}^2)}$

Multicollinearity: Condition Number

• Singular value decomposition (SVD) of any matrix **X**

$$\mathbf{X} = \mathbf{H}\,\mathbf{\Sigma}\,\mathbf{G}^T.$$

- <u>The first matrix in SVD</u>: **H** is (Txk) like **X**. It has sample principal coordinates of **X** standardized in the sense that $\mathbf{H}^T\mathbf{H}=\mathbf{I}_k$. Note, however, that $\mathbf{H}\mathbf{H}^T$ does not equal identity. **H** is not an orthogonal matrix.

- <u>The middle matrix in SVD</u>: **Σ** has k non-negative elements. It is a diagonal matrix. It contains the *singular values* of **X**, in general in descending order.

- <u>The last matrix of SVD</u>: **G′** is (kxk) orthogonal in the sense that its inverse equals its transpose –i.e., $\mathbf{G}'\mathbf{G} = \mathbf{I}_k$

The matrix **G** is (kxk) containing columns g_1 to g_k. The g_i are columns of **G** and are direction cosine vectors which orient the *i-th* principal axis of **X** with respect to the given original axes of the **X** data.

Multicollinearity: Condition Number

• Singular value decomposition (SVD) of matrix **X**

$$\mathbf{X} = \mathbf{H}\,\mathbf{\Sigma}\,\mathbf{G}^T.$$

• <u>Computation</u>: The columns of **H** are orthonormal eigenvectors of $\mathbf{X}\mathbf{X}^T$, the columns of **G** are orthonormal eigenvectors of $\mathbf{X}^T\mathbf{X}$, and **Σ** is a diagonal matrix containing the square roots of eigenvalues from **H** or **G** in descending order.

• The condition number $K\#$ = max(singular value)/ min(singular value) assesses the condition of the matrix.

• <u>Rule of thumb in numerical math</u>: If $K\#>30$ such matrix cannot be inverted reliably. Thus, **X** shows severe multicollinearity.

<u>Note</u>: You can find the SVD written as **H** as a (TxT), **Σ** as (Txk), and **G** as (kxk). In both systems, **H** and **G** have orthogonal columns –i.e., $\mathbf{H}^T\mathbf{H}=\mathbf{I}_T$. and $\mathbf{G}^T\mathbf{G}=\mathbf{I}_k$.

Multicollinearity: VIF and Condition Index

• Belsley (1991) proposes to calculate the VIF and the condition index, using R_X, the correlation matrix of the standardized regressors:

$$VIF_k = diag(R_X^{-1})_k$$
$$\text{Condition Index} = \varkappa_k = sqrt(\lambda_1 / \lambda_k)$$

where $\lambda_1 > \lambda_2 > ... > \lambda_p > ...$ are the ordered eigenvalues of R_X.

• Belsley's (1991) rules of thumb for $\varkappa_k$:
- below 10 ⇒ good
- from 10 to 30 ⇒ concern
- greater than 30 ⇒ trouble
- greater than 100 ⇒ disaster.

• Another common rule of thumb: If $VIF_k > 5$, concern.

Multicollinearity

• Best approach: Recognize the problem and understand its implications for estimation.

Note: Unless we are very lucky, some degree of multicollinearity will always exist in the data. The issue is: when does it become a problem?

Multicollinearity: Example

Example: Different tools to check for outliers for IBM returns

```
library(olsrr)
ols_vif_tol(mod)
ols_eigen_cindex(mod)

> ols_vif_tol(mod)
  Variables   Tolerance     VIF
1  xMkt_RF    0.8901229 1.123440
2  xSMB       0.9147320 1.093216
3  xHML       0.9349904 1.069530
> ols_eigen_cindex(mod)
  Eigenvalue Condition Index  intercept     xMkt_RF      xSMB         xHML
1  1.4506645     1.000000     0.01557614  0.24313961  0.212001760  0.1518949
2  1.0692689     1.164770     0.66799183  0.01432250  0.001789253  0.2129328
3  0.7967889     1.349310     0.16184731  0.01239755  0.576432492  0.4107435
4  0.6832777     1.457085     0.15458473  0.73014033  0.209776495  0.2244287
```

Note: Multicollinearity does not seem to be a problem.

Multicollinearity in Other Models

• Looking ahead to nonlinear models: The preceding results may not extend beyond the linear regression model:

In a nonlinear model, lack of multicollinearity among the variables is no guarantee that a similar phenomenon related to certain other functions of the **x'**s might not still reappear.