SCRIPT MOD1S3: SMALL-SAMPLE PROPERTIES OF OLS ESTIMATOR

INSTRUCTOR: KLAUS MOELTNER

Set basic R-options upfront and load all required R packages:

1. Data Generation

Set parameters for sampling.

```
R> N<-10000 #population size
R> n<-100 #sample size
R> r1<-500 #number of error vectors to draw
R> r2<-100 #number of samples to draw
```

Generate some simulated data for the entire "population".

```
R> x1<-rep(1,N)
R> x2<-rnorm(N,3,1.4)
R> x3<-rnorm(N,-2,2)
R> bvec<-c(1,0.5,1.2)
R> Xpop<-cbind(x1,x2,x3)
R> k<-ncol(Xpop)
R>
```

2. Conditional Unbiasedness

We want to show that, for the OLS estimator $\mathbf{b}$, we have $E(\mathbf{b}|\mathbf{X}) = \boldsymbol{\beta}$ for *a given* $\mathbf{X}$. To illustrate this notion of "conditional unbiasedness", we will proceed as follows:

(1) Draw a "sample" $\mathbf{X}$ of explanatory variables from the population $\mathbf{X}_{pop}$.
(2) Draw $r1$ sets of error terms, combine with the sample $\mathbf{X}$ (multiplied by some pre-defined vector of coefficients $\boldsymbol{\beta}$), and compute a corresponding "sample" of observations for the dependent variable. This mimics the notion that, in theory, there are different outcomes $\mathbf{y}$ possible for a given set of regressors $\mathbf{X}$ in our wider population.
(3) For each draw, compute the OLS solution $\mathbf{b}$, and examine the resulting *sampling distribution*.

In addition, we will repeat these steps for different variances of the regression error to examine its effect on the sampling distribution and unbiasedness properties of the OLS estimator.

```
R> bmat1<-matrix(0,k,r1) #pre-allocate collector matrix (for speed ); variance setting 1
R> bmat2<-matrix(0,k,r1) #pre-allocate collector matrix (for speed ); variance setting 2
R> bmat3<-matrix(0,k,r1) #pre-allocate collector matrix (for speed ); variance setting 3
R> bmat4<-matrix(0,k,r1) #pre-allocate collector matrix (for speed ); variance setting 4
R> sig1<-0.5
R> sig2<-1
```

```
R> sig3<-2
R> sig4<-4
R> int<-sample(1:N,n) #randomly select n units from your population
R> X<-Xpop[int,] #draw corresponding rows from Xpop
R> for (i in 1:r1){
+ eps1<-rnorm(n,0,sig1) #draw well-behaved OLS error, one for each variance setting
+ eps2<-rnorm(n,0,sig2)
+ eps3<-rnorm(n,0,sig3)
+ eps4<-rnorm(n,0,sig4)
+
+ y1<-X %*% bvec + eps1 #compute corresponding y's
+ y2<-X %*% bvec + eps2
+ y3<-X %*% bvec + eps3
+ y4<-X %*% bvec + eps4
+
+ b1<-solve((t(X)) %*% X) %*% (t(X) %*% y1)# compute corresponding OLS estimator
+ b2<-solve((t(X)) %*% X) %*% (t(X) %*% y2)
+ b3<-solve((t(X)) %*% X) %*% (t(X) %*% y3)
+ b4<-solve((t(X)) %*% X) %*% (t(X) %*% y4)
+
+ bmat1[,i]<-b1   #store draws
+ bmat2[,i]<-b2
+ bmat3[,i]<-b3
+ bmat4[,i]<-b4
+ }
```

Let's examine the results, first in tabular form, then graphically. As before, we use the mean over draws to simulate the expectation.

Table 1. Sampling distribution, sig=0.5

variable	true value	mean (samp.dist.)	std (samp.dist)
constant	1.000	1.001	0.129
x2	0.500	0.499	0.037
x3	1.200	1.199	0.025

Table 2. Sampling distribution, sig=1

variable	true value	mean (samp.dist.)	std (samp.dist)
constant	1.000	0.997	0.251
x2	0.500	0.502	0.074
x3	1.200	1.200	0.049

3. Unonditional Unbiasedness

Next, we want to show that $E(\mathbf{b}) = \boldsymbol{\beta}$ for *ANY* $\mathbf{X}$. To illustrate this notion of "unconditional unbiasedness", we will proceed as follows:

Table 3. Sampling distribution, sig=2

variable	true value	mean (samp.dist.)	std (samp.dist)
constant	1.000	1.017	0.480
x2	0.500	0.494	0.144
x3	1.200	1.198	0.106

Table 4. Sampling distribution, sig=4

variable	true value	mean (samp.dist.)	std (samp.dist)
constant	1.000	0.991	0.988
x2	0.500	0.504	0.275
x3	1.200	1.194	0.194

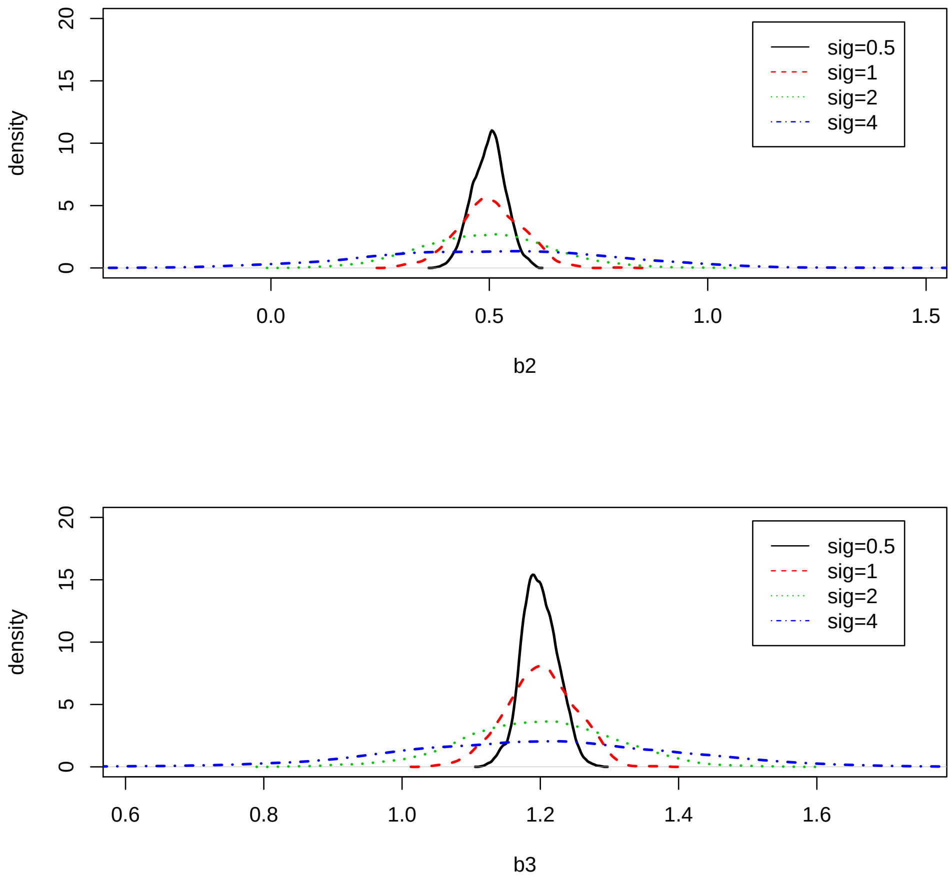

Figure 1. CONDITIONAL Sampling distribution for OLS estimates and at different settings for the error variance

(1) Draw a "sample" **X** of explanatory variables from the population $\mathbf{X}_{pop}$.
(2) Draw $r1$ sets of error terms, combine with the sample **X**, and compute a corresponding "sample" of dependent variables. This mimics the notion that, in theory, there are different outcomes **y** possible for a given set of regressors **X** in our wider population.
(3) For each draw, compute the OLS solution **b**
(4) Repeat all steps above $r2$ times. This mimics the notion that different outcomes **y** can occur both due to different **X**'s AND different unobservables for a given **X**. Examine the resulting *sampling distribution*.

We can do this again for different variance settings.

```
R> Bmat1=matrix(0,k,1) #just a starter column which we'll drop later
R> Bmat2=matrix(0,k,1)
R> Bmat3=matrix(0,k,1)
R> Bmat4=matrix(0,k,1)
R> for (i in 1:r2)   {
+
+ bmat1<-matrix(0,k,r1) #pre-allocate collector matrix (for speed ); variance setting 1
+ bmat2<-matrix(0,k,r1) #pre-allocate collector matrix (for speed ); variance setting 2
+ bmat3<-matrix(0,k,r1) #pre-allocate collector matrix (for speed ); variance setting 3
+ bmat4<-matrix(0,k,r1) #pre-allocate collector matrix (for speed ); variance setting 4
+
+ int<-sample(1:N,n) #randomly select n units from your population
+ X<-Xpop[int,] #draw corresponding rows from Xpop
+
+     for (i in 1:r1){
+         eps1<-rnorm(n,0,sig1) #draw well-behaved OLS error, one for each variance setting
+         eps2<-rnorm(n,0,sig2)
+         eps3<-rnorm(n,0,sig3)
+         eps4<-rnorm(n,0,sig4)
+
+         y1<-X %*% bvec + eps1 #compute corresponding y's
+         y2<-X %*% bvec + eps2
+         y3<-X %*% bvec + eps3
+         y4<-X %*% bvec + eps4
+
+         b1<-solve((t(X)) %*% X) %*% (t(X) %*% y1)# compute corresponding OLS estimator
+         b2<-solve((t(X)) %*% X) %*% (t(X) %*% y2)
+         b3<-solve((t(X)) %*% X) %*% (t(X) %*% y3)
+         b4<-solve((t(X)) %*% X) %*% (t(X) %*% y4)
+
+         bmat1[,i]<-b1   #store draws
+         bmat2[,i]<-b2
+         bmat3[,i]<-b3
+         bmat4[,i]<-b4
+         }  #end inner loop
+
+ Bmat1=cbind(Bmat1,bmat1)  #this will grow steadily
+ Bmat2=cbind(Bmat2,bmat2)
+ Bmat3=cbind(Bmat3,bmat3)
```

```
+ Bmat4=cbind(Bmat4,bmat4)
+ } #end outer loop
R> #drop starter columns
R> Bmat1<-Bmat1[,-1]
R> Bmat2<-Bmat2[,-1]
R> Bmat3<-Bmat3[,-1]
R> Bmat4<-Bmat4[,-1]
R>
```

Table 5. Sampling distribution, sig=0.5

variable	true value	mean (samp.dist.)	std (samp.dist)
constant	1.000	0.999	0.129
x2	0.500	0.500	0.036
x3	1.200	1.200	0.026

Table 6. Sampling distribution, sig=1

variable	true value	mean (samp.dist.)	std (samp.dist)
constant	1.000	1.001	0.259
x2	0.500	0.500	0.072
x3	1.200	1.200	0.052

Table 7. Sampling distribution, sig=2

variable	true value	mean (samp.dist.)	std (samp.dist)
constant	1.000	0.993	0.520
x2	0.500	0.502	0.144
x3	1.200	1.200	0.102

Table 8. Sampling distribution, sig=4

variable	true value	mean (samp.dist.)	std (samp.dist)
constant	1.000	1.003	1.031
x2	0.500	0.500	0.285
x3	1.200	1.200	0.207

```
R> proc.time()-tic
   user  system elapsed
  22.01    1.45   23.63
```

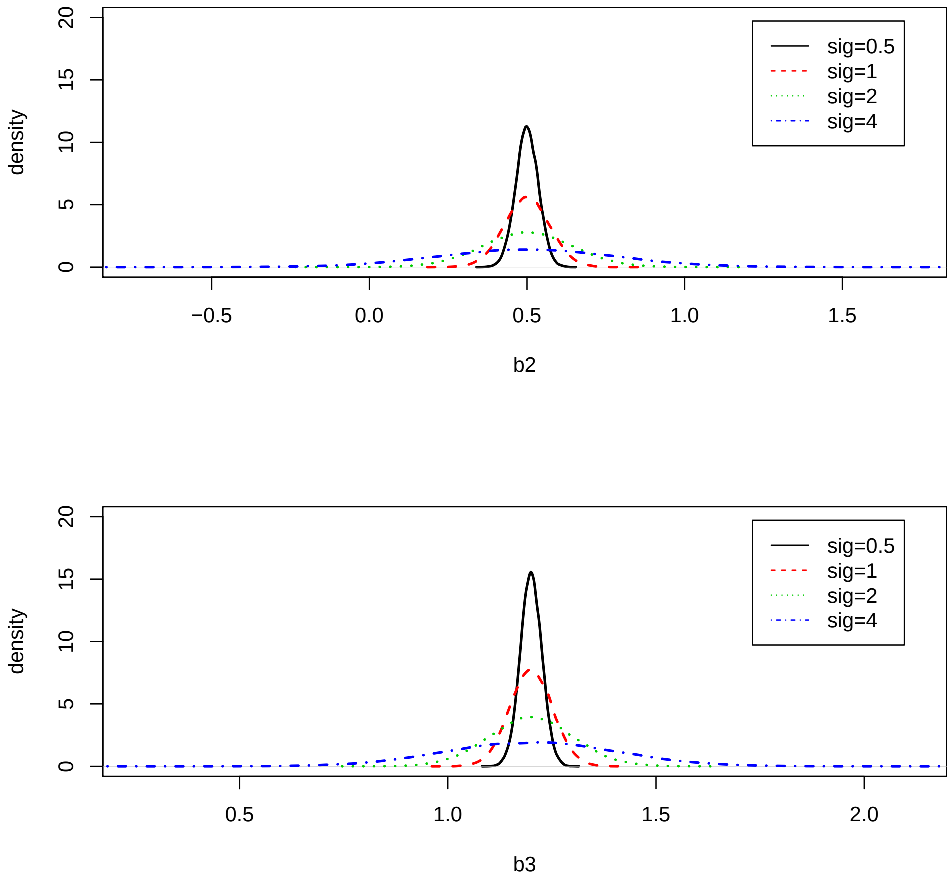

FIGURE 2. UNCONDITIONAL sampling distribution for OLS estimates and at different settings for the error variance