Regression II
Vanderbilt University
Prof. Sean P. Corcoran

Lecture 2 In-Class Exercise Solutions

---

1. This problem will estimate population regression functions using data from a known population that we define ourselves. Draw a $N = 100$ random sample of three indpendent $N(0, 1)$ variables: x_1, x_2, and u. The relevant command in Stata is `drawnorm`. From these, generate two outcome variables: $y_1 = 10+x_1+u$ and $y_2 = 10+x_1+2x_2+u$. Note: if you want to be able to replicate work done with randomly generated values in Stata, put the `set seed #` command at the beginning of your do-file. You will then get the same set of random numbers every time you run your program.

```
clear
set seed 626
// random draws of x1 x2 u (independent, standard normal variables)
drawnorm x1 x2 u, n(100)
corr

// DGP for y1 and y2
gen y1 = 10 + x1 + u
gen y2 = 10 + x1 + 2*x2 + u
```

   (a) What is the population mean of y_1, $E[y_1]$? What is the population variance of y_1, σ^2_{y1}? What is the conditional expectation function $E[y_1|x_1]$? Is it linear? What is the *conditional variance* of y_1 given x_1? Note: these questions can be answered without use of the data.

   See the handout with rules for expectation, variance, and covariance.

   $\mathbf{E[y_1] = E[10 + x_1 + u] = E[10] + E[x_1] + E[u] = 10 + 0 + 0 = 10}$

   $\boldsymbol{\sigma^2_{y1}} \mathbf{= Var[x_1] + Var[u] = 2}$**, since $\mathbf{x_1}$ and $\boldsymbol{u}$ are independent.**

   $\mathbf{E[y_1|x_1] = 10 + x_1}$**, a linear CEF.**

   $\mathbf{Var[y_1|x_1] = Var[u] = 1}$ **(homoskedasticity—variance is unrelated to $\mathbf{x_1}$)**

   (b) What is the population mean of y_2, $E[y_2]$? What is the population variance of y_2, σ^2_{y2}? What is the conditional expectation function $E[y_2|x_1]$? Is it linear? Note: these questions can be answered without use of the data.

   See the handout with rules for expectation, variance, and covariance.

   $\mathbf{E[y_2] = E[10 + x_1 + (2 * x_2) + u] = E[10] + E[x_1] + 2 * E[x_2] + E[u] =}$
   $\mathbf{10 + 0 + 2 * 0 + 0 = 10}$

$\sigma^2_{\mathbf{y2}} = \mathbf{1^2 Var[x_1] + 2^2 Var[x_2] + Var[u] + 2 * 1 * 2 * Cov[X1, X2] = 1 + 4 + 1 + 0 = 6}$, **since** $\mathbf{x_1}$ **and** $\mathbf{x_2}$ **are independent.**

$\mathbf{E[y_1|x_1] = 10 + x_1 + 2x_2}$**, a linear CEF. Note that this CEF depends on the value of** x_2

(c) Regress y_1 on x_1 (i.e., estimate the model $y_1 = \beta_0 + \beta_1 x_1$ using OLS). Note the slope coefficient and its standard error. Do the intercept and slope equal the known population intercept and slope? Why or why not?

```
. reg y1 x1

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(1, 98)        =    126.86
       Model |  129.848833         1  129.848833   Prob > F        =    0.0000
    Residual |  100.312116        98  1.02359302   R-squared       =    0.5642
-------------+----------------------------------   Adj R-squared   =    0.5597
       Total |   230.16095        99  2.32485808   Root MSE        =    1.0117

------------------------------------------------------------------------------
          y1 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   1.139554   .1011765    11.26   0.000     .9387729    1.340336
       _cons |   10.07918    .101691    99.12   0.000     9.877374    10.28098
------------------------------------------------------------------------------
```

Naturally the estimated intercept and slope $\hat{\beta}_0$ **and** $\hat{\beta}_1$ **differ from the known population values of 10 and 1 since they are estimated from a random sample.**

(d) Regress y_2 on x_1 (i.e., estimate the model $y_2 = \tilde{\gamma}_0 + \tilde{\gamma}_1 x_1$ using OLS). Note the slope coefficient and its standard error. If you are interested in an unbiased estimate of the slope on x_1 in the population regression function for y_1, will your slope estimator suffer from omitted variables bias? Why or why not?

```
. reg y2 x1

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(1, 98)        =     29.41
       Model |  127.188444         1  127.188444   Prob > F        =    0.0000
    Residual |  423.858659        98  4.32508836   R-squared       =    0.2308
-------------+----------------------------------   Adj R-squared   =    0.2230
       Total |  551.047103        99  5.56613236   Root MSE        =    2.0797

------------------------------------------------------------------------------
          y2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |    1.12782   .2079761     5.42   0.000     .7150983    1.540542
       _cons |   10.24087   .2090337    48.99   0.000     9.826046    10.65569
```

```
--------------------------------------------------------------------------------
```

We know the population model for $\mathbf{y_2}$ includes $\mathbf{x_2}$. A condition for omitted variables bias, however, is that $\mathbf{Cov(x_1, x_2) \neq 0}$. In this case, we know these two variables are independent and thus uncorrelated in the population.

(e) Now regress y_2 on x_1 and x_2 (i.e., estimate the model $y_2 = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2$ using OLS). Why does $\hat{\gamma}_1$ differ from $\hat{\hat{\gamma}}_1$, even though we know the population correlation between x_1 and x_2 is zero?

```
. reg y2 x1 x2

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(2, 97)        =    230.45
       Model |  455.239049         2  227.619525   Prob > F        =    0.0000
    Residual |  95.8080541        97  .987711898   R-squared       =    0.8261
-------------+----------------------------------   Adj R-squared   =    0.8225
       Total |  551.047103        99  5.56613236   Root MSE        =    .99384

------------------------------------------------------------------------------
          y2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   1.138324    .099389    11.45   0.000     .9410639    1.335583
          x2 |   1.790231   .0982322    18.22   0.000     1.595267    1.985195
       _cons |   10.09614    .100208   100.75   0.000      9.89725    10.29502
------------------------------------------------------------------------------
```

The estimated coefficient on $\mathbf{x_1}$ changes a bit when we add $\mathbf{x_2}$ as a covariate. In the population there is no OVB since $\mathbf{x_1}$ and $\mathbf{x_2}$ are uncorrelated. We are working with sample data, however, and there may be chance correlation between $\mathbf{x_1}$ and $\mathbf{x_2}$ in the sample.

(f) Compare the estimated standard errors on $\hat{\hat{\gamma}}_1$ from part (d) and $\hat{\gamma}_1$ from part (e). How and why did it change?

The standard error dropped considerably, from 0.208 to 0.099, because we reduced variation in the error term. In part (d), $\mathbf{x_2}$ remains in the error term and contributes to the variation in $\mathbf{y_2}$.

(g) Now modify x_2 to purge it of any sample correlation with x_1. Call this variable x_{2a}. Hint: you are looking for variation in x_2 that is orthogonal to ("not explained" by) x_1.

```
reg x2 x1
predict x2a, resid
```

By construction, the residuals from a regression of $\mathbf{x_2}$ on $\mathbf{x_1}$ are uncorrelated with $\mathbf{x_1}$. We know that $\mathbf{x_2}$ and $\mathbf{x_1}$ are not correlated in the population, but there is a small amount of correlation between them in the sample. This step "purges" the tiny amount of correlation between the two.

(h) Generate a new y_2 (call it y_{2a}) using x_{2a} in place of x_2. Repeat parts (d) and (e). What changed, and why? Why does the standard error on $\hat{\gamma}_1$ change with the inclusion of x_{2a}, when we know x_{2a} is uncorrelated (by construction) with x_1?

```
. gen y2a = 10 + x1 + 2*x2a + u

. reg y2a x1

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(1, 98)        =     30.02
       Model |  129.848831         1  129.848831   Prob > F        =    0.0000
    Residual |   423.85866        98  4.32508837   R-squared       =    0.2345
-------------+----------------------------------   Adj R-squared   =    0.2267
       Total |  553.707491        99  5.59300496   Root MSE        =    2.0797

------------------------------------------------------------------------------
         y2a |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   1.139554   .2079761     5.48   0.000     .7268325    1.552276
       _cons |   10.07918   .2090337    48.22   0.000     9.664356      10.494
------------------------------------------------------------------------------

. reg y2a x1 x2a

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(2, 97)        =    231.80
       Model |  457.899438         2  228.949719   Prob > F        =    0.0000
    Residual |  95.8080524        97  .987711881   R-squared       =    0.8270
-------------+----------------------------------   Adj R-squared   =    0.8234
       Total |  553.707491        99  5.59300496   Root MSE        =    .99384

------------------------------------------------------------------------------
         y2a |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   1.139554   .0993874    11.47   0.000      .942298    1.336811
         x2a |   1.790231   .0982322    18.22   0.000     1.595268    1.985195
       _cons |   10.07918   .0998928   100.90   0.000     9.880917    10.27744
------------------------------------------------------------------------------
```

Now the estimated coefficient on $\mathbf{x_1}$ is identical, whether one controls for $\mathbf{x_{2a}}$ or not. The reason is that $\mathbf{x_1}$ is now uncorrelated with $\mathbf{x_{2a}}$. The standard error drops, again because we have reduced variation in the error term with the inclusion of $\mathbf{x_{2a}}$.

(i) Return to part (c). Compare the reported standard error for $\hat{\beta}_1$ to the *population* standard error for $\hat{\beta}_1$. Hint: you know the population σ^2.

In a simple regression the population standard error for $\hat{\beta}_1$ is:

$$\mathbf{se}(\hat{\beta}_1) = \frac{\sigma_{\mathbf{u}}}{\sqrt{(\mathbf{n}-1)\mathbf{Var}(\mathbf{x})}}$$

Where $\sigma_{\mathbf{u}}$ is the square root of the error variance. The syntax below manually calculates the population standard error of $\hat{\beta}_1$ (0.10000369), which can be compared to the estimated standard error in the regression (0.1011765). These differ, since Stata is estimating σ using residuals. Note I used Var(x) from the sample data here ($1.005^2 = 1.01$) rather than using the known Var(x) of 1. This is taking the point of view that *x* is fixed from sample to sample and the only random variation is in *u*. This is how the usual statistical assumptions are stated. An alternative approach would use the known Var(x) = 1.

```
. summ x1

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
          x1 |        100   -.1013415    1.005001  -2.380183   2.706876

. local varx1 r(Var)

. local nobs  r(N)

. display sqrt(1/((`nobs'-1)*(`varx1')))
.10000369

.
. reg y1 x1

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(1, 98)        =    126.86
       Model |  129.848833         1  129.848833   Prob > F        =    0.0000
    Residual |  100.312116        98  1.02359302   R-squared       =    0.5642
-------------+----------------------------------   Adj R-squared   =    0.5597
       Total |   230.16095        99  2.32485808   Root MSE        =    1.0117

------------------------------------------------------------------------------
          y1 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   1.139554   .1011765    11.26   0.000     .9387729    1.340336
       _cons |   10.07918    .101691    99.12   0.000     9.877374    10.28098
------------------------------------------------------------------------------

. display _se[x1]
.10117651
```

(j) Start with an empty dataset and recreate your random variables x_1, u, and y_1, but this time draw a $N = 10,000$ random sample. Repeat part (i). Now how do your reported $\hat{\beta}_1$ and standard error for $\hat{\beta}_1$ compare to the population β_1 and standard error for $\hat{\beta}_1$?

```
. clear

. drawnorm x1 x2 u, n(10000)
(obs 10,000)

. gen y1 = 10 + x1 + u

. sum x1

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
          x1 |     10,000     .011974    1.005396  -3.667296   3.615652

. local varx1 r(Var)

. local nobs r(N)

. display sqrt(1/((`nobs'-1)*(`varx1')))
.00994683

. reg y x1

      Source |       SS           df       MS      Number of obs   =    10,000
-------------+----------------------------------   F(1, 9998)      =   9908.03
       Model |  10054.2942         1  10054.2942   Prob > F        =    0.0000
    Residual |  10145.5918     9,998  1.01476214   R-squared       =    0.4977
-------------+----------------------------------   Adj R-squared   =    0.4977
       Total |   20199.886     9,999  2.02019062   Root MSE        =    1.0074

------------------------------------------------------------------------------
          y1 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   .9973795     .01002    99.54   0.000     .9777384    1.017021
       _cons |   10.00787   .0100743   993.41   0.000     9.988126    10.02762
------------------------------------------------------------------------------

. display _se[x1]
.01001998
```

Proportionally speaking, the estimated standard error is closer to the population standard error with the larger sample size.

2. This problem is similar to #1, but we will assume x_1 and x_2 come from a *bivariate normal* distribution, so that we know x_1 and x_2 are correlated. The relevant command in Stata is `drawnorm`, but we need to specify a correlation matrix for the distribution (call this **C**). σ^2_{x1} and σ^2_{x2} will continue to be 1, but assume they have a correlation of 0.5. Continue to use $N = 100$. Create the outcome variable $y_2 = 10 + x_1 + 2x_2 + u$. See the syntax below for the `drawnorm` command and its correlation matrix.

```
clear
matrix C = (1, .5 , 0 \ .5, 1, 0 \ 0, 0, 1)
drawnorm x1 x2 u, n(100) corr(C)
corr
gen y2 = 10 + x1 + 2*x2 + u
```

(a) What is the population variance of y_2? How does this compare with your answer in question #1 part (b)?

The population variance of y_2 is:

$$\begin{aligned}\sigma^2_{y2} &= 1^2 Var[x_1] + 2^2 Var[x_2] + Var[u] + (2*1*2)Cov[x_1, x_2] \\ &= 1 + 4 + 1 + 4(0.5) \\ &= 8\end{aligned}$$

This uses the fact that $\mathbf{Corr(X, Y) = Cov(X, Y)/sd(X)sd(Y)}$, and that we know the correlation between $\mathbf{x_1}$ and $\mathbf{x_2}$ is 0.5 and their respective standard deviations are 1.

(b) For fun, use the user-written Stata command `tddens` to visualize the bivariate distribution of (x_1, x_2) as a "heat map".

```
ssc install tddens
tddens x1 x2
```

(c) Regress y_1 on x_1. Note the slope coefficient and its standard error. If you are interested in an unbiased estimate of β_1 (the slope coefficient on x_1 in the population), does this regression suffer from omitted variables bias? Why or why not? If so, in what direction is the bias?

The simple regression of $\mathbf{y_1}$ on $\mathbf{x_1}$ is shown below. Unlike in question 1, we now know that $\mathbf{x_1}$ and $\mathbf{x_2}$ are correlated. If our interest is in an unbiased estimate of β_1 in the full model, we have omitted variables bias. We can use the omitted variables bias formula to think about the direction of bias: $\beta_s = \beta_\ell + \pi\gamma$. Here we know $\gamma > 1$ (from the population model) and $\pi > 1$ (since we know $\mathbf{x_1}$ and $\mathbf{x_2}$ are positively correlated). So the short regression coefficient is biased upward. As expected, including $\mathbf{x_2}$ as a covariate reduces the estimated coefficient on $\mathbf{x_1}$:

```
. reg y2 x1

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(1, 98)        =    129.78
       Model |  605.877978         1  605.877978   Prob > F        =    0.0000
    Residual |  457.521779        98  4.66858958   R-squared       =    0.5698
-------------+----------------------------------   Adj R-squared   =    0.5654
       Total |  1063.39976        99  10.7414117   Root MSE        =    2.1607

------------------------------------------------------------------------------
          y2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   2.155543   .1892156    11.39   0.000     1.780051    2.531036
       _cons |   9.898806   .2165912    45.70   0.000     9.468988    10.32862
------------------------------------------------------------------------------
```

(d) Now regress y_2 on x_1 and x_2. What changed, and why?

As expected, including $\mathbf{x_2}$ as a covariate reduces the estimated coefficient on $\mathbf{x_1}$ (see part c):

```
. reg y2 x1 x2

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(2, 97)        =    556.49
       Model |  978.150139         2   489.07507   Prob > F        =    0.0000
    Residual |   85.249617        97  .878862031   R-squared       =    0.9198
-------------+----------------------------------   Adj R-squared   =    0.9182
       Total |  1063.39976        99  10.7414117   Root MSE        =    .93748

------------------------------------------------------------------------------
          y2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   .9126716   .1019149     8.96   0.000     .7103988    1.114944
          x2 |   2.136519   .1038094    20.58   0.000     1.930486    2.342552
       _cons |   10.06019   .0943007   106.68   0.000     9.873029    10.24735
------------------------------------------------------------------------------
```

(e) Apply the "regression anatomy" formula. That is, show that $\hat{\beta}_2$ is equal to the slope coefficient from a simple regression of y_2 on $\tilde{x}_2$, where $\tilde{x}_2$ is the residual from a regression of x_2 on x_1. Equivalently, $\hat{\beta}_2 = Cov(y_1, \tilde{x}_2)/Var(\tilde{x}_2)$.

The code below shows this calculation. The first step is the "auxiliary" regression of $\mathbf{x_2}$ on $\mathbf{x_1}$ where the residuals are obtained.

```
. reg x2 x1

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(1, 98)        =     53.03
       Model |  44.1276395         1  44.1276395   Prob > F        =    0.0000
    Residual |  81.5543213        98  .832186952   R-squared       =    0.3511
```

```
-------------+------------------------------   Adj R-squared =   0.3445
       Total |  125.681961        99  1.26951476   Root MSE      =   .91224

------------------------------------------------------------------------------
          x2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   .5817274   .0798867     7.28   0.000     .4231948      .74026
       _cons |  -.0755356   .0914447    -0.83   0.411    -.2570046    .1059333
------------------------------------------------------------------------------

. predict uhat, resid

. reg y2 uhat

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(1, 98)        =     52.79
       Model |  372.272159         1  372.272159   Prob > F        =    0.0000
    Residual |  691.127597        98  7.05232242   R-squared       =    0.3501
-------------+----------------------------------   Adj R-squared   =    0.3434
       Total |  1063.39976        99  10.7414117   Root MSE        =    2.6556

------------------------------------------------------------------------------
          y2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        uhat |   2.136519   .2940645     7.27   0.000     1.552958    2.720081
       _cons |   10.07001   .2655621    37.92   0.000     9.543007    10.59701
------------------------------------------------------------------------------
```

Alternatively I show the Cov/Var version of this below (you get the same answer):

```
. corr y2 uhat, covar
(obs=100)

             |       y2     uhat
-------------+------------------
          y2 |  10.7414
        uhat |  1.76002  .823781


. local covyu `r(cov_12)'

. summ uhat

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
        uhat |        100    4.98e-10    .9076238  -2.480034   1.866416

. local varu `r(Var)'

. display `covyu' / `varu'
```

```
2.1365191
```

(f) Demonstrate the omitted variables bias formula by showing the coefficient in the "short" regression (part b) is equal to the coefficient on x_1 in the "long" regression (part c) + the product of β_2 (the coefficient on x_2 in the "long" regression) and π (the coefficient from a regression of the omitted on the included).

The code below shows this. Note the scalars `_b[ ]` **are one way of referencing estimated regression coefficients. These are temporary, so we store them as local macros.**

```
. reg y2 x1 x2

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(2, 97)        =    556.49
       Model |  978.150139         2   489.07507   Prob > F        =    0.0000
    Residual |   85.249617        97  .878862031   R-squared       =    0.9198
-------------+----------------------------------   Adj R-squared   =    0.9182
       Total |  1063.39976        99  10.7414117   Root MSE        =    .93748

------------------------------------------------------------------------------
          y2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   .9126716   .1019149     8.96   0.000     .7103988    1.114944
          x2 |   2.136519   .1038094    20.58   0.000     1.930486    2.342552
       _cons |   10.06019   .0943007   106.68   0.000     9.873029    10.24735
------------------------------------------------------------------------------

. local x1long = _b[x1]

. local x2long = _b[x2]

. reg y2 x1

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(1, 98)        =    129.78
       Model |  605.877978         1  605.877978   Prob > F        =    0.0000
    Residual |  457.521779        98  4.66858958   R-squared       =    0.5698
-------------+----------------------------------   Adj R-squared   =    0.5654
       Total |  1063.39976        99  10.7414117   Root MSE        =    2.1607

------------------------------------------------------------------------------
          y2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   2.155543   .1892156    11.39   0.000     1.780051    2.531036
       _cons |   9.898806   .2165912    45.70   0.000     9.468988    10.32862
------------------------------------------------------------------------------

. local x1short = _b[x1]
```

```
. reg x2 x1

      Source |       SS           df       MS      Number of obs   =       100
-------------+----------------------------------   F(1, 98)        =     53.03
       Model |  44.1276395         1  44.1276395   Prob > F        =    0.0000
    Residual |  81.5543213        98  .832186952   R-squared       =    0.3511
-------------+----------------------------------   Adj R-squared   =    0.3445
       Total |  125.681961        99  1.26951476   Root MSE        =    .91224

------------------------------------------------------------------------------
          x2 |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   .5817274   .0798867     7.28   0.000     .4231948      .74026
       _cons |  -.0755356   .0914447    -0.83   0.411    -.2570046    .1059333
------------------------------------------------------------------------------

. local pi = _b[x1]

. display `x1long'
.91267159

. display `x2long'
2.1365192

. display `pi'
.5817274

. display `x1long' + (`x2long'*`pi')
2.1555433

. // compare to:
. display `x1short'
2.1555433
```