ECO 375–Homework 2

University of Toronto Mississauga

Due: 14 November, 2023, 5 PM

ANSWER KEY

Theoretical Problems

1. Consider data drawn from the following model:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + U \tag{1}$$

where X_1, X_2, and U are each iid. Assume that MLR.4 is satisfied. Your goal is to estimate β_1.

(a) *(points: 6)* Suppose you use OLS to estimate

$$Y = \beta_0 + \beta_1 X_1 + U \tag{2}$$

Write the asymptotic mean squared error (MSE) of the estimator $\hat{\beta}_1$ you would get from using OLS to estimate Equation 2. Write your answer in terms of β_1, β_2, and:

- $V_1 \equiv V[X_1] > 0$ and $V_2 \equiv V[X_2] > 0$
- $\sigma^2 \equiv V[U] > 0$
- $r \equiv corr[X_1, X_2] \neq 0$
- n is the sample size

Answer

- Recall that $MSE = bias^2 + var$.
- The bias is given by the omitted variable bias formula:

$$bias = \beta_2 \frac{cov(X_1, X_2)}{V_1} = \beta_2 r \sqrt{\frac{V_2}{V_1}}$$

- Asymptotic variance of the estimator is given by $var = \frac{V(\beta_2 X_2 + u)}{na^2}$, where $a^2 = plim(\frac{1}{n}SSR)$, where SSR is the sum of squared residuals in a regression of X_1 against all other variables (here, there aren't any), so $SSR = SST = nV_1$, where SST is the total sum of squares of X_1. Thus $var = \frac{\sigma^2}{nV_1}$.
- Putting this together:

$$MSE_{short} = \beta_2^2 r^2 \frac{V_2}{V_1} + \frac{\beta_2^2 V_2 + \sigma^2}{nV_1}.$$

(b) *(points: 6)* Write the asymptotic MSE of the estimator $\hat{\beta}_1$ you would get from using OLS to estimate Equation 1. Write your answer in terms of β_1, β_2, and the variables defined in the previous question.

Answer

- Now, there are no omitted variables, so there is no bias in the OLS estimators.
- Again, asymptotic variance of the estimator is given by $var = \frac{\sigma^2}{na^2}$, where $a^2 = plim(\frac{1}{n}SSR)$, where SSR is the sum of squared residuals in a regression of X_1 against all other variables.
- Recall that $R^2 = 1 - \frac{SSR}{SST}$ in any regression, and that in a simple regression, $R^2 = \hat{r}^2$, where $\hat{r}^2 \xrightarrow{p} r^2$. Thus $\frac{1}{n}SSR \xrightarrow{p} V_1(1 - r^2)$.

- Putting this together:
$$MSE_{long} = \frac{\sigma^2}{nV_1(1-r^2)}$$

(c) *(points: 8)* Suppose we have 10 observations, and $\beta_1 = \beta_2 = V_1 = V_2 = \sigma^2 = 1$. If the researcher wants to minimize MSE, for which values of r should the researcher use the short regression, for which the long regression, and for which would they have the same MSE?

Answer

The MSE for the short regression (Equation 2) minus the MSE for the long regression (Equation 1) is given by

$$\begin{aligned} MSE_{short} - MSE_{long} &= \beta_2^2 r^2 \frac{V_2}{V_1} + \frac{\beta_2^2 V_2 + \sigma^2}{nV_1} - \frac{\sigma^2}{nV_1(1-r^2)} \\ &= \beta_2^2 r^2 \frac{V_2}{V_1} + \frac{1}{nV_1(1-r^2)}((\beta_2^2 V_2 + \sigma^2)(1-r^2) - \sigma^2). \end{aligned}$$

Assuming $r^2 < 1$ (otherwise the long regression can't be run), this is positive if and only if:

$$\begin{aligned} 0 &< n\beta_2^2 r^2 V_2(1-r^2) + (\beta_2^2 V_2 + \sigma^2)(1-r^2) - \sigma^2 \\ &= (r^2)^2 \left(-n\beta_2^2 V_2\right) + r^2 \left(n\beta_2^2 V_2 - \beta_2^2 V_2 - \sigma^2\right) + \beta_2^2 V_2 \end{aligned}$$

Plugging in the parameter values, the researcher should use the long regression if $-10r^4 + 8r^2 + 1 > 0$; otherwise, the short regression should be used. Solving for r^2 with the quadratic formula, use the long regression if $-0.1099 < r^2 < 0.9099$. Since r^2 is always positive, we use the long regression if $-\sqrt{0.9099} = -0.9539 < r < 0.9539 = \sqrt{0.9099}$. When $r = 0.9539$, the two methods are equal in MSE. Intuitively: high very multicollinearity can cause the long regression to be imprecise, but in general, the long regression is better because it is unbiased.

2. A retailer wants to sell laptop computers. To learn about the market, they gather data on 90 computers and estimate the following equation with OLS:

$$price = \beta_0 + \beta_1 screen + \beta_2 weight + \beta_3 screen \times weight + u$$

where *price* is the price of the computer in dollars, *screen* is the screen size in inches, and *weight* is the weight in pounds. Here are their estimates (with standard errors in parentheses):

$$\begin{aligned} \hat{\beta}_0 &= 600\ (100) \\ \hat{\beta}_1 &= 20\ (4) \\ \hat{\beta}_2 &= 50\ (20) \\ \hat{\beta}_3 &= -3\ (1) \end{aligned}$$

Further assume that we estimate $cov(\hat{\beta}_i, \hat{\beta}_j) = .1$ for all $i \neq j$. The R^2 from this regression is .3.

(a) *(points: 6)* What is the point estimate for the difference in price based on an additional inch of screen size for a 4-pound computer? What about a 5-pound computer?

Answer
$\frac{\partial\ price}{\partial\ screen} = \beta_1 + \beta_3 weight$. Thus for a 4-pound computer, an additional inch is associated with $20 + (-3) \times 4 = \$8$ higher price; for a 5-pound computer, it is associated with $20 + (-3) \times 5 = \$5$ higher price.

(b) *(points: 6)* Test the null hypothesis that the association between screen size and price is not related to a computer's weight at the 5% level. Do not use an asymptotic assumption here. Clearly state the test statistic and the critical value; you should look up critical values in the back of the textbook (which is available in MindTap). If the critical value you are interested in is not in the table, choose a critical value with degrees of freedom as close as possible to the one you are interested in, but note the correct degrees of freedom and the degrees of freedom you have chosen. (For example, you might say: "I wanted to find a critical value for a chi-squared distribution with 31 degrees of freedom, but I chose one for a chi-squared with 30 degrees of freedom.")

Answer
This is testing $\beta_3 = 0$. $t = \frac{-3-0}{1} = -3$. To test this, we need the two-sided 5% critical value of a $t_{n-k-1} = t_{90-3-1} = t_{86}$ distribution. That isn't in the textbook, but the critical value for t_{90} is: 1.987. $|-3| > 1.987$, so we reject the null hypothesis.

(c) *(points: 8)* The retailer believes that an additional inch of screen size for a 4-pound computer is associated with a \$5 higher price. State this null hypothesis in terms of the β's, then calculate the test statistic and 5% critical value (following the instructions for finding critical values above).

Answer
The null hypothesis is that $\beta_1 + 4\beta_3 = 5$. The test statistic is $t = \frac{20+4(-3)-5}{SE}$, where

$$\begin{aligned} SE &= \sqrt{V[\hat{\beta}_1 + 4\hat{\beta}_3]} \\ &= \sqrt{V[\hat{\beta}_1] + 16 \times V[\hat{\beta}_3] + 2cov(\hat{\beta}_1, 4\hat{\beta}_3)} \\ &= \sqrt{(4)^2 + 16 \times (1)^2 + 8 \times .1} \\ &= 5.727 \end{aligned}$$

Thus $t = \frac{20+4(-3)-5}{5.727} = .5238$. Again, we need the two-sided 5% critical value of a $t_{n-k-1} = t_{90-3-1} = t_{86}$ distribution. That isn't in the textbook, but the critical value for t_{90} is: 1.987. $|t| < 1.987$, so we fail to reject the null hypothesis.

Computer Based Problems

Notes:

- If you are using Citrix, remember to save your code and log file on your home computer, not the remote computer!
- Remember to include the Stata code as part of your writeup. (See "Notes" at the start of this assignment.)

3. **Data from the US**. For this problem, you will analyze data set `eco375hw2data2023` using Stata.[1] All questions below refer to this sample. In that data set, each observation describes data about a person who lived in the United States in 2023. Variables in this data set are:

Variable	Definition
age	Age in years
sex	Sex (1=male, 2=female)
region	Region of the US; 1=Northeast, 2=Midwest, 3=South, 4=West
indly	Industry worked in; see `https://cps.ipums.org/cps/codes/ind_2020_codes.shtml`
incwage	Labor income, in dollars
own	Whether household owns home (as opposed to renting); 1=yes, 0=no
uhrsworkly	Usual hours worked per week

Do NOT use heteroskedasticity-robust standard errors for this problem.

(a) *(points: 5)* Estimate the following model using OLS:

$$\ln(wage) = \beta_0 + \beta_1 age + u \tag{3}$$

State the estimated $\hat{\beta}_1$ using words a non-economist could understand; do not use causal language. Answer twice: once using an approximation, and once using an exact answer.

Answer

- Being one year older is associated with approximately 1.78% higher wages.
- The exact answer is $100 \times (\exp(.0177961) - 1) = 1.80$. Being one year older is associated with exactly 1.80% higher wages.

Code:

```
use eco375hw2data2023, clear
gen logwage = ln(incwage)
reg logwage age
```

[1] This data set is based on data is used with permission from Flood et al. (2023). You may not redistribute this data. However, if you wish, you may (and are encouraged to!) download data on your own from `https://cps.ipums.org/`.
Full citation:
Sarah Flood, Miriam King, Renae Rodgers, Steven Ruggles, J. Robert Warren, Daniel Backman, Annie Chen, Grace Cooper, Stephanie Richards, Megan Schouweiler and Michael Westberry. IPUMS CPS: Version 11.0 [dataset]. Minneapolis, MN: IPUMS, 2023. https://doi.org/10.18128/D030.V11.0.

(b) *(points: 4)* What is the t-statistic when testing whether β_1=.02 in Equation 3?

Answer

The t-stat is $\frac{.0177961-.02}{.0006356} = -3.4674323$.

(c) *(points: 5)* Now, estimate the following model using OLS:

$$\ln(wage) = \beta_0 + \beta_1 age + \beta_2 age^2 + \beta_3 age^3 + u \tag{4}$$

Using an approximation, based on these results, how much does income increase or decrease with age when someone is 25? What about when they are 65?

Answer

I estimate $\hat{\beta}_1 = .4139685$; $\hat{\beta}_2 = -.0075274$; and $\hat{\beta}_3 = .0000423$, so

$$\begin{aligned}\frac{\partial\ \ln(wage)}{\partial\ age} &= \beta_1 + 2\beta_2 age + 3\beta_3 age^2 \\ &= .4139685 + 2 \times -.0075274 \times age + 3 \times .0000423 \times age^2\end{aligned}$$

If *age*=25, then this derivative is .116911. If *age*=65, then this derivative is -.028441.

Code:

```
gen age2 = age^2
gen age3 = age^3
reg logwage age age2 age3
```

(d) *(points: 4)* Test whether log income varies non-linearly with age in Equation 4. Can you reject the null of linearity at the 5% level?

Answer

The p-value is $0.0000 < 0.05$, so we can reject the null.

Code:

```
test age2 = age3 = 0
```

(e) *(points: 6)* Now, estimate the following model using OLS:

$$\ln(wage) = \beta_0 + \beta_1 age + \beta_2 own + \beta_3 age \times own + u \tag{5}$$

Should estimated β's be interpreted causally? Why or why not? Refer to the assumption(s) we discussed in class. Then, state the estimated $\hat{\beta}_3$ using words a non-economist could understand; use an approximation, and use causal language if and only if the estimates should be interpreted causally.

Answer

The estimates should not be interpreted causally; the error term is likely correlated with several of the regressors, so MLR.4 is violated. For example, parental wealth might help people buy a home, and also help someone get a better job.

I estimate that wages go up with age 1.14% faster for those who own homes, as opposed to those who rent.

Code:

```
gen own_x_age = own * age
reg logwage own age own_x_age
```

(f) *(points: 5)* How much faster or slower do wages rise with age for people in the construction industry versus those in all other industries? (Use an approximation here.)

Answer

Wages rise 0.642% slower in construction than in other industries.

Code:

```
gen construction = (indly==770)
gen construction_x_age = construction * age
reg logwage age construction construction_x_age
```

(g) *(points: 5)* By how many standard deviations is log income higher for men than for women in this data? (You must use a standardized variable in Stata for this problem.)

Answer

Men earn 31.3% of a standard deviation more than women. (You could also say that men earn .313 of a standard deviation more than women.)

Code:

```
sum logwage
gen stdlogwage = (logwage - 10.61945) / 1.119511
gen male = (sex==1)
reg stdlogwage male
```

(h) *(points: 2)* Using the data, briefly present and describe one fact that you think is interesting. Your answer to this question must actually use Stata code and the data to find a fact that was not discussed in a previous question.

Answer

Answers will vary.

4. **Monte Carlo Simulation**. Consider the following model:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + U$$

where

$$\begin{aligned} \beta_0 &= 3, \\ \beta_1 &= 4, \\ \beta_2 &= 2, \\ X_1 &\sim N(0,1) \\ X_2 &= X_1 + V, \text{ where } V \sim N(0,4), \\ U &\sim Uniform(-6,6), \end{aligned}$$

and X_1, V, and U are independent. [Hint 1: a $N(0,4)$ random variable has standard deviation of 2. In Stata, you can generate a normal random variable with mean 0 and standard deviation of 2 using the command `rnormal(0,2)`.] [Hint 2: $Uniform(-6,6)$ indicates a uniform distribution between -6 and 6. In Stata, you can generate a uniform distribution between A and B using the command `runiform(A,B)`.]

Simulate this model 10,000 times in Stata. [Hint: get your code working with fewer simulations, so you can correct errors quickly. Once your code is ready, change it to simulate 10,000 times.] For each simulation, generate a data set with $n = 100$ observations. Then, for each sample, estimate β_0, β_1, and β_2 with OLS, calling your estimates β_0, β_1, and β_2. Create two sets of estimates for each simulation: one based on the first 5 observations in the data set, and one based on the full 100 observations.

Do NOT use heteroskedasticity-robust standard errors for this problem.

(a) *(points: 8)* Define a new variable $\gamma \equiv \beta_1 \times \beta_2$. Estimate γ using $\hat{\gamma} = \hat{\beta}_1 \times \hat{\beta}_2$ within each simulation, for each set of estimates. Create a table like the following with average estimates of $\hat{\beta}_1$, $\hat{\beta}_2$, and $\hat{\gamma}$ based on 5 observations or 100 observations.

	5 obs	**100 obs**
$\hat{\beta}_1$		
$\hat{\beta}_2$		
$\hat{\gamma}$		

Answer

	5 obs	**100 obs**
$\hat{\beta}_1$	4.002147	4.005451
$\hat{\beta}_2$	2.020967	2.001241
$\hat{\gamma}$	5.348006	7.98478

Code:

```
clear all
set seed 123
cap program drop hwprog
program hwprog, rclass
drop _all
```

```
set obs 100
gen x1 = rnormal(0,1)
gen x2 = x1 + rnormal(0,2)
gen u = runiform(-6,6)
gen y = 3 + 4* x1 + 2*x2 + u
foreach samplesize in 5 100 {
reg y x1 x2 if _n<=‘samplesize’
return scalar b1_‘samplesize’=_b[x1]
return scalar b2_‘samplesize’=_b[x2]
return scalar tstat_‘samplesize’ = (_b[x1]-4)/_se[x1]
}
end

simulate "hwprog" b1_5=r(b1_5) b2_5=r(b2_5) tstat_5=r(tstat_5) ///
b1_100=r(b1_100) b2_100=r(b2_100) tstat_100=r(tstat_100), reps(10000)
foreach samplesize in 5 100 {
gen g_‘samplesize’ = b1_‘samplesize’*b2_‘samplesize’
}
sum b* g_*
```

(b) *(points: 5)* Which of the values in this table are close to the true values of β_1, β_2, and γ? For each value (whether close or not), based on what we've learned in class, do we have a good reason to expect that the estimate would be close to the truth? If so, what is that reason?

Answer

All values are close to the true parameters except $\hat{\gamma}$ based on 5 observations. For $\hat{\beta}_i$, OLS estimates are unbiased regardless of the sample size, and therefore the average of all observations should be close to the true parameter. Further, because OLS estimates are consistent, $\hat{\beta}_i \xrightarrow{p} \beta_i$. By the continuous mapping theorem, $\hat{\gamma} = \hat{\beta}_1 \times \hat{\beta}_2 \xrightarrow{p} \beta_1 \times \beta_2$, so in large sample (e.g. 100 observations) we expect $\hat{\gamma} \approx \gamma$. However, we have no reason to believe that this relationship would hold in small sample since $E[\hat{\beta}_1 \times \hat{\beta}_2] \neq \beta_1 \times \beta_2$; that is the expected value of a product of random variables is not generally equal to the product of the expectation.

(c) *(points: 6)* Now, for each regression, test $H_0 : \beta_1 = 4$ against $H_1 : \beta_1 \neq 4$. Conduct this test in two ways: first with a small-sample t-test, and second with the asymptotic t-test. For each test, how often do we reject the null at a 5% significance level? Create tables like the following with the critical values for the tests and the fraction of observations in which we reject the null hypothesis.

Critical values

	5 obs	**100 obs**
Small sample		
Asymptotic		

Fraction of rejections

	5 obs	**100 obs**
Small sample		
Asymptotic		

Answer

Critical values

	5 obs	**100 obs**
Small sample	4.30	1.98
Asymptotic	1.96	1.96

Fraction of rejections

	5 obs	**100 obs**
Small sample	.0489	.0519
Asymptotic	.1872	.0541

Code:

```
* If you didn’t know these commands, you could look up the critical values in the textbook
di abs(invttail(2,.025))
di abs(invttail(97,.025))
di abs(invnormal(.025))

foreach samplesize in 5 100 {
gen reject_small_‘samplesize’ = abs(tstat_‘samplesize’) > abs(invttail(‘samplesize’-3,.025))
gen reject_asymp_‘samplesize’ = abs(tstat_‘samplesize’) > abs(invnormal(.025))
}
sum reject_*
```

(d) *(points: 5)* Which of the values in this table are close to the 5% that we hope for? (That is, do the tests have the correct size?) For each value (whether close or not), based on what we’ve learned in class, do we have a good reason to expect that the estimate would be close to that value? If so, what is that reason?

Answer

- 100 observations is a large number of observations, so we would expect the asymptotic test for 100 observations to have the correct size.
- As the degrees of freedom of a t distribution increases, it converges to a normal distribution. Thus for large sample sizes, the “small sample” test becomes close to the asymptotic test; you can see this in the fact that the critical values of the two tests for 100 observations are so close. Since the asymptotic test has the correct size for 100 observations, we therefore expect the small sample test to have approximately the correct size even though the error term is not normal, as required for the small sample test to be exactly valid.
- 5 observations is not close to ∞ observations, so we would not expect the asymptotic test to have the correct size.
- The error term is not normal, so there is no theoretical reason based on what we’ve learned why the small sample test with 5 observations should have the correct size–even though, in fact, it does. (This suggests that one should use t-tests rather than asymptotic tests when sample sizes are small unless there is a compelling reason not to.)

DO FILE

```
* Code for ECO375, homework 2

************************************

* Start the log
capture log close
capture noisily log using "code_hw2_2023.log", replace


************************************

clear all
set more off


************************************
* PROBLEM 3

use eco375hw2data2023, clear

* 3(a)
gen logwage = ln(incwage)
reg logwage age

* 3(c)
gen age2 = age^2
gen age3 = age^3
reg logwage age age2 age3

* 3(d)
test age2 = age3 = 0

* 3(e)
gen own_x_age = own * age
reg logwage own age own_x_age

* 3(f)
gen construction = (indly==770)
gen construction_x_age = construction * age
reg logwage age construction construction_x_age

* 3(g)
sum logwage
gen stdlogwage = (logwage - 10.61945) / 1.119511
gen male = (sex==1)
reg stdlogwage male
```

```
************************************
* PROBLEM 4

* 4(a)
clear all
set seed 123
cap program drop hwprog
program hwprog, rclass
drop _all
set obs 100
gen x1 = rnormal(0,1)
gen x2 = x1 + rnormal(0,2)
gen u = runiform(-6,6)
gen y = 3 + 4* x1 + 2*x2 + u
foreach samplesize in 5 100 {
reg y x1 x2 if _n<=‘samplesize’
return scalar b1_‘samplesize’=_b[x1]
return scalar b2_‘samplesize’=_b[x2]
return scalar tstat_‘samplesize’ = (_b[x1]-4)/_se[x1]
}
end

simulate "hwprog" b1_5=r(b1_5) b2_5=r(b2_5) tstat_5=r(tstat_5) ///
b1_100=r(b1_100) b2_100=r(b2_100) tstat_100=r(tstat_100), reps(10000)
foreach samplesize in 5 100 {
gen g_‘samplesize’ = b1_‘samplesize’*b2_‘samplesize’
}
sum b* g_*


* 4(c)

* If you didn’t know these commands, you could look up the critical values in the textbook
di abs(invttail(2,.025))
di abs(invttail(97,.025))
di abs(invnormal(.025))

foreach samplesize in 5 100 {
gen reject_small_‘samplesize’ = abs(tstat_‘samplesize’) > abs(invttail(‘samplesize’-3,.025))
gen reject_asymp_‘samplesize’ = abs(tstat_‘samplesize’) > abs(invnormal(.025))
}
sum reject_*


************************************
* End the log

capture log close
```

LOG FILE

```
.
.
. ************************************
.
. clear all

. set more off


.
.
. ************************************
. * PROBLEM 3
.
. use eco375hw2data2023, clear


.
. * 3(a)
. gen logwage = ln(incwage)
(799 missing values generated)

. reg logwage age

      Source |       SS           df       MS      Number of obs   =    13,695
-------------+----------------------------------   F(1, 13693)     =    783.93
       Model |  929.369227         1  929.369227   Prob > F        =    0.0000
    Residual |  16233.3922    13,693  1.18552488   R-squared       =    0.0542
-------------+----------------------------------   Adj R-squared   =    0.0541
       Total |  17162.7614    13,694   1.2533052   Root MSE        =    1.0888

------------------------------------------------------------------------------
     logwage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .0177961   .0006356    28.00   0.000     .0165503     .019042
       _cons |   9.866734   .0284485   346.83   0.000     9.810972    9.922497
------------------------------------------------------------------------------


.
. * 3(c)
. gen age2 = age^2

. gen age3 = age^3

. reg logwage age age2 age3
```

```
      Source |       SS           df       MS      Number of obs   =    13,695
-------------+----------------------------------   F(3, 13691)     =   1303.52
       Model |  3813.07366         3  1271.02455   Prob > F        =    0.0000
    Residual |  13349.6878    13,691   .97507032   R-squared       =    0.2222
-------------+----------------------------------   Adj R-squared   =    0.2220
       Total |  17162.7614    13,694   1.2533052   Root MSE        =    .98746

------------------------------------------------------------------------------
     logwage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
         age |   .4139685   .0116326    35.59   0.000      .391167      .43677
        age2 |  -.0075274   .0002693   -27.95   0.000    -.0080553   -.0069996
        age3 |   .0000423   1.95e-06    21.64   0.000     .0000384    .0000461
       _cons |    3.81264   .1555861    24.51   0.000      3.50767     4.11761
------------------------------------------------------------------------------


.
. * 3(d)
. test age2 = age3 = 0

 ( 1)  age2 - age3 = 0
 ( 2)  age2 = 0

       F(  2, 13691) = 1478.72
            Prob > F =    0.0000


.
. * 3(e)
. gen own_x_age = own * age

. reg logwage own age own_x_age

      Source |       SS           df       MS      Number of obs   =    13,695
-------------+----------------------------------   F(3, 13691)     =    306.83
       Model |  1081.22048         3  360.406826   Prob > F        =    0.0000
    Residual |  16081.5409    13,691  1.17460674   R-squared       =    0.0630
-------------+----------------------------------   Adj R-squared   =    0.0628
       Total |  17162.7614    13,694   1.2533052   Root MSE        =    1.0838

------------------------------------------------------------------------------
     logwage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
         own |  -.2832331   .0610683    -4.64   0.000    -.4029354   -.1635307
         age |   .0083589   .0012427     6.73   0.000     .0059231    .0107947
   own_x_age |   .0114346   .0014552     7.86   0.000     .0085822    .0142871
       _cons |   10.11099   .0497975   203.04   0.000     10.01338     10.2086
------------------------------------------------------------------------------
```

```
.
. * 3(f)
. gen construction = (indly==770)

. gen construction_x_age = construction * age

. reg logwage age construction construction_x_age

      Source |       SS           df       MS      Number of obs   =    13,695
-------------+----------------------------------   F(3, 13691)     =    266.00
       Model |  945.277011         3  315.092337   Prob > F        =    0.0000
    Residual |  16217.4844    13,691  1.18453615   R-squared       =    0.0551
-------------+----------------------------------   Adj R-squared   =    0.0549
       Total |  17162.7614    13,694   1.2533052   Root MSE        =    1.0884

------------------------------------------------------------------------------------
           logwage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------------+----------------------------------------------------------------
               age |   .0181821   .0006551    27.75   0.000     .0168979    .0194662
      construction |    .375349   .1193664     3.14   0.002     .1413745    .6093235
construction_x_age |   -.006419   .0026853    -2.39   0.017    -.0116826   -.0011554
             _cons |   9.843579   .0293376   335.53   0.000     9.786074    9.901085
------------------------------------------------------------------------------------

.
. * 3(g)
. sum logwage

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
     logwage |     13,695    10.61945    1.119511          0   13.91082

. gen stdlogwage = (logwage - 10.61945) / 1.119511
(799 missing values generated)

. gen male = (sex==1)

. reg stdlogwage male

      Source |       SS           df       MS      Number of obs   =    13,695
-------------+----------------------------------   F(1, 13693)     =    343.43
       Model |  335.048762         1  335.048762   Prob > F        =    0.0000
    Residual |  13358.9548    13,693  .975604675   R-squared       =    0.0245
-------------+----------------------------------   Adj R-squared   =    0.0244
       Total |  13694.0036    13,694  1.00000026   Root MSE        =    .98773

------------------------------------------------------------------------------
  stdlogwage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
```

```
-------------+----------------------------------------------------------------
        male |   .3130927   .0168949    18.53   0.000     .2799763     .346209
       _cons |  -.1630024   .0121904   -13.37   0.000    -.1868973   -.1391074
------------------------------------------------------------------------------

.
.
. ************************************
. * PROBLEM 4
.
. * 4(a)
. clear all

. set seed 123

. cap program drop hwprog

. program hwprog, rclass
  1.         drop _all
  2.         set obs 100
  3.         gen x1 = rnormal(0,1)
  4.         gen x2 = x1 + rnormal(0,2)
  5.         gen u = runiform(-6,6)
  6.         gen y = 3 + 4* x1 + 2*x2 + u
  7.         foreach samplesize in 5 100 {
  8.                 reg y x1 x2 if _n<=‘samplesize’
  9.                 return scalar b1_‘samplesize’=_b[x1]
 10.                 return scalar b2_‘samplesize’=_b[x2]
 11.                 return scalar tstat_‘samplesize’ = (_b[x1]-4)/_se[x1]
 12.         }
 13. end

.
. simulate "hwprog" b1_5=r(b1_5) b2_5=r(b2_5) tstat_5=r(tstat_5) ///
>         b1_100=r(b1_100) b2_100=r(b2_100) tstat_100=r(tstat_100), reps(10000)

Command:      hwprog
Statistics:   b1_5       = r(b1_5)
              b2_5       = r(b2_5)
              tstat_5    = r(tstat_5)
              b1_100     = r(b1_100)
              b2_100     = r(b2_100)
              tstat_100  = r(tstat_100)

. foreach samplesize in 5 100 {
  2.         gen g_‘samplesize’ = b1_‘samplesize’*b2_‘samplesize’
  3. }
```

```
. sum b* g_*

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
        b1_5 |     10,000    4.002147    3.776259  -54.01699   85.35341
        b2_5 |     10,000    2.020967    1.657286   -22.9453   29.40149
      b1_100 |     10,000    4.005451    .3938178   2.507533   5.589094
      b2_100 |     10,000    2.001241    .1780954   1.335437   2.681705
         g_5 |     10,000    5.348006    24.22123  -936.5603   655.3232
-------------+---------------------------------------------------------
       g_100 |     10,000     7.98478     .797926   4.857135   11.29286

.
.
. * 4(c)
.
. * If you didn't know these commands, you could look up the critical values in the textbook
. di abs(invttail(2,.025))
4.3026527

. di abs(invttail(97,.025))
1.9847232

. di abs(invnormal(.025))
1.959964

.
. foreach samplesize in 5 100 {
  2.          gen reject_small_`samplesize' = abs(tstat_`samplesize') > abs(invttail(`samplesize
> )
  3.          gen reject_asymp_`samplesize' = abs(tstat_`samplesize') > abs(invnormal(.025))
  4. }

. sum reject_*

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
reject_sma~5 |     10,000       .0489    .2156698          0          1
reject_asy~5 |     10,000       .1872    .3900915          0          1
reject_s~100 |     10,000       .0519    .2218362          0          1
reject_a~100 |     10,000       .0541    .2262262          0          1

.
.
. ************************************
. * End the log
.
. capture log close
```