

## Econometrics I (ECON 6620)

## Week 1

Tonight you will learn to run a simple R program, both in the Windows GUI, and on our linux workstation (beast).

I'll show you how to use the Windows GUI in class. It's a very good interface for working with small datasets and for debugging portions of programs that you will use on beast. For most of your work in this class, the Windows GUI will be adequate, though your dissertation work will probably ultimately require beast.

You need a username and password to access beast, and those will be distributed tonight.

The best way to connect with beast is to use PuTTY, which produces a terminal window. The terminal window does not allow one to write graphical output unless one first activates Xming, which creates an X-windows environment. Therefore one should connect to beast by first clicking the Xming icon (nothing happens except that the Xming icon now becomes visible in the System Tray). Then one should click on the PuTTY icon, so that a dialog appears. In the dialog's "category" window (the long window to the left), first click "SSH", next click "X11", and finally click the box to enable X11 forwarding. Click "open" and a terminal will open to beast. You will be prompted for your username and password.

Once on beast, the commands that you type on the command line will be *linux* commands. First, perform the following steps (i.e., type each of these commands at the beast command-line prompt):

```
mkdir 662
cd 662
pico ssr
```

Now you are in an editor called *pico* editing a file called 'ssr', found in the subdirectory '662'. Type the following lines—be sure to differentiate the symbol for 'one' and the symbol for lower-case 'L'.

```
R CMD BATCH $1.R
```

## Introduction to R

```
less $1.Rout
pico $1.R
```

Now hit ' ^x' (control-x) and then ' y' to exit *pico*.

```
chmod 755 ssr
pwd
pico r01.R
```

Now you are in *pico* editing a file called 'r01.R'. Enter the program that begins on the next page, either by typing in the lines, or by pasting the program from the pdf file on my webpage to *pico*.

Once finished, hit ^x and then y to exit *pico*. At the beast command prompt type the following.

```
./ssr r01
```

You are using the ssr file you created earlier. This executes your R program and will show you first, your R output file (which will help you identify your errors); and second, your R program (this last is opened in *pico* so that you can immediately begin to edit).

Keep working with this until all errors are corrected.

## Getting help for R

There are two good online search engines for R:

1. <http://finzi.psych.upenn.edu/nmz.html>
2. <http://www.rseek.org/>

Using the Windows GUI, one can access help pages for a function in any of the *loaded* packages by typing a question mark before the name of the function:

```
?lm
```

To search for a keyword in the loaded packages' help pages, use two question marks:

```
??lm
```

## R program r01.R

```
#Hedonic home price model
#remove all objects from memory
rm(list=ls(all=TRUE))
#set the working directory. note the slashes are like UNIX, not like DOS
setwd("d:/class/662/R/")
#call in the routines from package foreign
library(foreign)
#read in the Williamson county house price data from the dbf file cnty187.dbf
gg<-read.dbf("s:/teff/662/R/cnty187.dbf")
names(gg)
#write data to your own computer (keep in flash drive)
write.dbf(gg,file="class1.dbf")
#several ways to look at your data
summary(gg)
str(gg)
names(gg)
head(gg)
tail(gg)
dim(gg)
NROW(gg)
NCOL(gg)

#make some new variables
gg$agesq<-gg$age^2
gg$spring<-(gg$month>=3 & gg$month<=6)*1
gg$mnf<-factor(gg$month)
gg$trend<-(gg$yr-1996)*12+month

#use only a subset of the observations
z<-which(gg$price>=100000 & gg$pracc==0)
gg<-gg[z,]
#check how many observations in this subset
length(z)
NROW(z)

#lm is the standard OLS function
ww<-lm(price~sqft+age+agesq+mnf,data=gg)
#take a look at the OLS output
ww
#summary gives much nicer output
vv<-summary(ww)
vv
names(vv)
#when you know the names, you can look at the objects
vv$coefficients
vv$coefficients[3,1]/(-2*vv$coefficients[4,1])

#some other syntax
summary(lm(log(price)~sqft*brick+age+I(age^2),data=gg))
summary(lm(log(price)~sqft:brick+age+I(age^2),data=gg))

#look at the coefficients and variance-covariance matrix
coef(ww)
vcov(ww)

#solve OLS using matrix manipulation
y<-as.matrix(gg$price)
intercept<-matrix(1,length(gg[,1]),1)
x<-as.matrix(cbind(intercept,gg[,c("sqft","age","sound","vacant","brick")]))
beta<-solve(t(x)%*%x)%*%t(x)%*%y
beta
#compare with coefficients from lm function
zz<-lm(price~sqft+age+sound+vacant+brick,data=gg)
coef(zz)
summary(zz)
```

```

#--test if we can drop the two insignificant independent variables--
#--this is an F-test, and the function is in the package "car"--
#--you might have to install car first with the command--
#  install.packages("car")

library(car)
dropt<-c("age","sound")
o<-linear.hypothesis(zz,dropt,white.adjust=FALSE)
o
names(o)
o$"Pr(>F)"[2]

#--the long way to do the F-test--

zz<-lm(price~sqft+age+sound+vacant+brick,data=gg)
ESSUR<-sum(zz$residuals^2)
dfUR<-zz$df
zz<-lm(price~sqft+vacant+brick,data=gg)
ESSR<-sum(zz$residuals^2)
nres<-2 #we are dropping 2 independent variables
fstat<-((ESSR-ESSUR)/nres)/(ESSUR/dfUR)
pval<-pf(fstat,nres,dfUR,lower.tail=FALSE)
cbind(fstat,pval)

#--write results to a csv file (to open in Excel)--
write.csv(summary(zz)$coefficients,file="zz.csv",append=FALSE)
write.csv(paste("R2=",round(summary(zz)$r.squared,4),sep=""),file="zz.csv",append=TRUE)
write.csv(paste("F for restrictions=",round(fstat,4),"; pvalue=",round(pval,4),sep=""),
file="zz.csv",append=TRUE)

```

---

## HYPOTHESIS TESTING

### Steps for conducting a hypothesis test:

- 1) Set up a null hypothesis (i.e., posit that the true value is equal to a specific number).
- 2) Create a test statistic.
- 3) Make a decision rule (i.e., reject the null hypothesis if the test statistic exceeds some cutoff value).

### P-Value, Critical Value, Size of Test

- The *size of test* is the probability that you are rejecting the null hypothesis when in fact it is true.
- The *critical value* of a t-statistic or f-statistic is constructed assuming a certain size of test (usually 0.05).
- The *p-value* gives the size of test at which the estimated t-statistic or f-statistic becomes the critical value.

### t-tests for one parameter

- 1)  $H_0: b = b_{\text{hypothesized}}$
- 2)  $t\text{-stat} = (b_{\text{estimated}} - b_{\text{hypothesized}}) / \text{standard error}(b_{\text{estimated}})$
- 3) Reject  $H_0$  if  $\text{abs}(t\text{-stat}) > t\text{-critical}$

### t-tests for linear combination of parameters

- 1)  $H_0: b_1 + b_2 = b_{\text{hypothesized}}$
- 2)  $t\text{-stat} = (b_{1,\text{estimated}} + b_{2,\text{estimated}} - b_{\text{hypothesized}}) / \text{standard error}(b_{1,\text{estimated}} + b_{2,\text{estimated}})$
- 3) Reject  $H_0$  if  $\text{abs}(t\text{-stat}) > t\text{-critical}$

### F-tests for group of parameters

- 1)  $H_0: b_1 = b_2 = 0$
- 2)  $F\text{-stat} = ((\text{error sum of squares in restricted regression} - \text{error sum of squares in unrestricted regression}) / \text{number of restrictions}) / (\text{error sum of squares in unrestricted regression} / \text{degrees of freedom in unrestricted regression})$
- 3) Reject  $H_0$  if  $F\text{-stat} > F\text{-critical}(\text{numerator degrees of freedom} = \text{number of parameters set equal to zero}; \text{denominator degrees of freedom set equal to degrees of freedom in the unrestricted regression})$

### F-test to Drop Irrelevant Independent variables

Create a model to explain variation in home values. You should follow these steps:

- 1) Run a regression in which you include all the independent variables that you think—*a priori*—are relevant. Call this the **unrestricted** regression.
- 2) Store the sum of squared residuals and the degrees of freedom from this regression.
- 3) Make a note of the variables which have a p-value above 0.10.
- 4) Set up a new regression, which omits all those independent variables with a high p-value
- 5) Store the sum of squared residuals and the degrees of freedom from this regression. Call this the **restricted** regression.
- 6) Use your stored values to carry out a hypothesis test
  - a) Null Hypothesis: the *omitted* variables do *not* belong in the model
  - b) Test Statistic: F-test
  - c) Decision rule: if the F-statistic is high enough, then *reject* the null hypothesis
    - i) How do we determine if the F-statistic is high enough? If it *exceeds* the critical value.
    - ii) How do we calculate the critical value? Set it equal to an F-statistic with numerator degrees of freedom equal to the number of omitted variables and denominator degrees of freedom equal to the degrees of freedom in the unrestricted regression. Set the size of test equal to .05.

### General modeling procedure:

- In building a model, one begins with theory, selecting independent variables that theory suggests explain the variation in the dependent variable.
- The first (unrestricted) regression will usually show that some coefficients are not different from zero. One identifies these insignificant coefficients by examining the t-test, given in the summary(lm(...)) output; the null hypothesis of the t-test is that the coefficient equals zero. If the p-value is less than 0.1 then you can reject the null hypothesis.
- Remove the insignificant variables from the model and perform an F-test to see if these independent variables jointly fail to explain the variation in the dependent variable. The null hypothesis is that these variables do *not* belong in the model. If the p-value is less than 0.1 then you can reject the null hypothesis.

**Homework assignment:** The homework is due by noon next Thursday (this gives me time to look at it before class).

- 1) Review any material you might have from previous courses on hypothesis testing and regression analysis. (nothing to be turned in)
- 2) Build a model explaining home prices in Williamson County. Your group may wish to analyze a submarket (e.g., all homes costing between \$140,000 and \$190,000)—just be sure to make the submarket broad enough to include at least 1,000 observations. Be creative and use as many independent variables as you think reasonable. Remove the insignificant variables following the procedure outlined above. Use the help files to find some new feature of R and include this in your program. Write a half page essay describing what your results mean. Write comments that explain what each step in your R program does. Turn in to me your edited output, your commented program, and your half page essay.
- 3) Experiment with linux, less, and pico commands. Next week, I'll check to see how far you've gotten. (nothing to be turned in)
- 4) Install R on your personal computer. Go to <http://cran.r-project.org/> and find the “binaries” for the operating system on your computer. Make a directory on your personal computer for your R work related to this class. Copy to this directory the data and R program used in this first class. Change the `setwd()` command in the R program to your directory name. Make sure that you can run the program on your own machine. (Nothing to turn in).