```
#  EASY  R  FOR  REGRESSIONS

# Notes on now to use the  free statistical package R in Windows.
#I will show how to load data into it and run regressions.

# The hash market means this is a comment line.
# February 26, 2010.
# Professor  Eric Rasmusen, erasmuse@Indiana.edu.
# See http://www.rasmusen.org/a/r.htm for more files.
#Some of this draft is untested.

#R is a wonderfully crafted statistics packages with
#amazingly bad documentation.
#These notes show how to make it do the basics of regressions.

#######################################

# To download R go to: The Comprehensive R Archive Network
# at   http://cran.case.edu/
#Download it and install it on your computer, which is easy to do.

#######################################

# GETTING STARTED

# Click on the R icon that the installation program created on
#your desktop. A window will open up.

#To find out what directory R is writing files to, reset it to the D:
# root directory, and check again to see if it worked,  type:

 getwd()
 setwd("D:\\")
 getwd()

#Notice how slashes need to be doubled for R to read them.

#A subdirectory would look like
# "D:\\_Take-to-office\\rfiles\\

#Now let's fix it so R doesn't show us lots
# of meaningless numbers.
 #  Scientific notation discouraged at intensity 6.
#4 digits suggested (there is no way to require only 4)
  options(scipen = 6)
  options(digits = 4)

#Only  uncomment the next line if you know why.
  #options(defaultPackages=c("car", "hmisc") )


#######################################

#READING IN DATA

#A data file is just plain text. The format is like this
#file called r-data.txt:

#   price floor area rooms age cent.heat state
#1         52.00 111.0 830      5 6.2           no Illinois
# 2 60    128.0 710 5 7.5 no Illinois
```

```
#3 35  101.0 1000 5 4.2 no Illinois
#4 50 131.0 690 6 8.8 no Indiana
#5 20 93.0 900 5 1  yes Ohio
#6 57  101.0        1000 5 4  no Illinois
#7 80 100  690 6 8              no Indiana
#8 30  90   800 5 2  yes Ohio

#Notice the lack of an observation number on the first line.
# Capitalization matters. price and Price could
# be different variables.
#Spacing and tab symbols  don't matter, but
#linebreaks do, I think.
#Notice how binary yes/no variables and string variables
# are easily accommodated.




#The first read-in command creates an object called "data2":

data2 <- read.table("r-data.txt")

#That uses the file D:/r-data.txt, since we set D:/ as
#the working directory.


# In case we want to use  lags or time series functions,   transform
#the dataset thus:

 tdata2 <- ts(data2)


#Then  create some lags if you want--- one  and four  periods
#here:

temp1= lag(tdata2,-1)
temp4= lag(tdata2,-4)
 tdata3 <- ts.union(tdata2, temp1, temp4)



# Now convert it back to a normal dataset:

data3 <-  data.frame(tdata3)

#Unfortunately, R has given stupid names to many of the
#variables.  To see the stupid names, type

 colnames(data3)

#Thus, let's convert at least some of them back.
#R is very bad at renaming.
#First, let's get R to put  our original names in quotes:

colnames(data2)

#Then let's cut-and-paste those  into a command to fix up data3:

colnames(data3)[1:7] <- c ( "price",     "floor",     "area",
"rooms",     "age",       "cent.heat", "state")
```

```
#Cut and paste from that last command and insert 1's and 4's:

 colnames(data3)[8:14] <- c ( "price1",     "floor1",     "area1",
"rooms1",     "age1",       "cent.heat1", "state1")

  colnames(data3)[15:21] <- c ( "price4",     "floor4",     "area4",
"rooms4",     "age4",       "cent.heat4", "state4")

# Now look at your dataset to see if the names and  lags turned
#out right.  Notice that the cent.heat and state variables have
#been made numerical.

data3

#Make this dataset the default dataset to use for this session:

attach(data3)

#####################################

# SUMMARY STATISTICS AND PLOTTING

 #We can get some summary statistics.
#Use your original dataset without the lags, thus:

summary(data2)

#To get a correlation matrix, use your original dataset without the
#lags, thus:

cor(data2)

#The cor() command ignores our request to only have 2 digits.
#To get it down to 2 digits, read it into Excel, and change it there.


#To plot two variables   type

plot(price,floor, main="Figure 2: Price and Floor")

#To save a figure as a jpg file, go to FILE, SAVE-AS, jpg on the
#top menu.




#####################################

# RUNNING A REGRESSION

#To do a regression, create a regression output object called
#"output4":

output4 <- lm(price~ price1 + floor +area+  floor*area +
cent.heat )

#Then display the output:

summary(output4)
```

```
#If you want the residuals, variance-covariance matrix, or
#predicted values, type

residuals5  <- output4$resid
residuals5

vcov5 <-vcov(output4)
vcov5

pred5  <- output4$fitted.values
pred5

#Thesummary(), residuals(), and vcov() commands partly ignore
#our request to only have  2 digits.
#To get them down to 2 digits, cut-and-paste into Excel.

#####################################

#ODDS AND ENDS

#We can plot a data histogram.

#First get rid of all the observations with missing data.

data3a<-na.omit(data3)
attach(data3a)

#Then create and plot the histogram:

var234<-hist(price)
plot(var234, main="Figure 1: A histogram  of price")

# We can also do kernel densities, for variables that
#are close to continuous. Not true here, but try it anyway:

var123 <- density (price)
plot(var123, main="Figure 3: A kernel density estimate of price")

#Finally, let's go back to our original dataset:

attach(data3)

##################

#If you don't want an intercept in a regression, type

output6 <- lm(price~ 0+  floor +area+  cent.heat)
summary(output6)

##################

#To plot two variables and the regression line for a
#TWO-VARIABLE regression:

output5 <- lm(price~ floor   )
 plot( floor,price,abline(output5$coef))

##################

#I don't have fixed state effects working yet.
```

```
#What is below is mistaken.
# To put in fixed state effects,   type

output7 <- lm(price~    floor +area+  age + factor(state))
summary(output7)

 #The dummy variables for "state" will be created automatically.


##################

# To run a logit,  convert the y-variable to have only
#values of 0 or 1 thus:

cent.heat = cent.heat -1
cent.heat

#Then run the regression

 output8 <- glm(cent.heat~   floor +area, family =binomial(link =
"logit") )
summary(output8)

##################

# If you have your data in a spreadsheet form, save
# it as a *.csv file,  which   is comma separated.  Then
 #your R  data input command  would be

data2a <- read.csv("D:\\_Take-to-office\\r-data.csv")

 #I haven't tested this one.

##################


#We can write the data  out to a file called data3.csv,
#  no variable column names,  comma-separated:

write.table(data3, col.names = FALSE,  sep = ",",
file="data3.csv")

#We can write the data  out to a file called data3.txt,
#  with variable column names,  space-separated:

write.table(data3, col.names =TRUE,  sep = " ",
file="data3.txt")

#####################################

#USING PACKAGES FOR SPECIAL COMMANDS


##################

# To do an F-Test for b2 + b3 =1, first download
# the "car" "package".

#On the top menu, pick
# Packages, Select Repository, (pick any site), Install Package, car
```

```
 #Then, whenever you want to use "car" pick
# Packages, Load Package, car.

#Finally, type (BUT THIS DOESN'T WORK YET)

output6 <- lm(price~ floor +area+  cent.heat)
rhs <- c(1)
restricted6  <- rbind(c(0,0,1,1))
linear.hypothesis(output6, restricted6, rhs)

##################

#To run a Durbin-Watson test for autocorrelation, load the car
#package  as described above and then  type

output9 <-  lm(price ~ floor+ area)
durbin.watson(output7, max.lag=2)

##################

#To save regression output in a Latex file, download and load the
#"hmisc" package in the same way as the "car" package described
#above and type

output10 <-  lm(price~ floor +area+  floor*area +  cent.heat)
latex(summary(output10)$coef)

#You'll get something pretty close to latex, with
# standard errors, t-stats, and p-values.
#It will be in a file called "summary.tex" in your
# working directory.


#####################################

#REFERENCES

# Mark Gardener, "Using R for statistical analyses" is at:

#
http://www.gardenersown.co.uk/Education/Lectures/R/regressi
on.htm#what_is_R"

 #Gardener  is the best documentation, giving full step-by-step
#procedures.

# Grant Farnsworth's "Econometrics in R"  is at:

#
http://cran.r-project.org/doc/contrib/Farnsworth-
EconometricsInR.pdf

#Farnsworth is the best longer  reference for economists

# MAT 356 R Tutorial, Spring 2004 is at:

#  http://math.illinoisstate.edu/dhkim/rstuff/rtutor.html

#I learned some things from it.
```

#"An Introduction to R", by Venables, Smith, et al. is bad.

#B. D. Ripley and D. J. Murdoch "R for Windows FAQ" is at:

http://cran.r-project.org/bin/windows/base/rw-FAQ.html

#It probably won't be useful.

##