Contributed Exercises for Vinod's book "HANDS-ON INTERMEDIATE ECONOMETRICS USING R"

Frank A. Canovatchel, Fordham University, Bronx, New York 10458
and
Brandon C. Vick, Fordham University, Bronx, New York 10458

Abstract

These are exercises to accompany H. D. Vinod's above-mentioned book. The book's URL is: (http://www.worldscibooks.com/economics/6895.html). The author names above are in alphabetical order. The suggested exercises, mostly along with abridged answers (hints) are in separate sections for each author.

1 Exercises suggested by Frank A. Canovatchel (FAC)

1.1 Exercise

FAC-1) Using the CigarettesB dataset of the AER package, run the regression of packs on price and income. Compute Studentized Breusch-Pagan test (relaxing the Normality assumption by using the studentized version) Nonconstant Variance Score test, and Goldfeld-Quant F test for heteroscedasticity and determine if heteroscedasticity is present.

```
library(AER)
data(CigarettesB); attach(CigarettesB)
```

```
reg1= lm(packs~price+income);su1=summary(reg1); su1
bptest(reg1)
ncv.test(reg1)
plot(resid(reg1)^2, typ="l")
gqtest(reg1, point=38)


R OUTPUT:
> bptest(reg1)

        studentized Breusch-Pagan test

data:  reg1
BP = 6.5936, df = 2, p-value = 0.037

> ncv.test(reg1)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.006027677    Df = 1     p = 0.938116
> plot(resid(reg1)^2, typ="l")
> gqtest(reg1, point=38)

        Goldfeld-Quandt test

data:  reg1
GQ = 0.9038, df1 = 5, df2 = 35, p-value = 0.4896
```

The Breusch-Pagan test fits a linear regression model to the residuals of a linear regression model using the same regressors and rejects if too much of the variance is explained by the additional explanatory variables. For above data, it rejects.

The ncv.test computes a score test of the hypothesis of constant error variance. The alternative hypothesis is that the error variance changes with the level of the dependent variable. Since p-value is large, we accept the null of constant variance.

Goldfeld-Quandt test tests for the null of constant variance against the alternative of increasing variances. For above data we use a plot to determine that the variance my have increased at round observation 38. Since p-value is large, the conclusion is to accept the null of constant variance.

1.2 Exercise

FAC-21) Copy the data below into Excel and save it as "caplab.csv" Load the file into R. Run a program that will create a new regressor named 'K' and set the values in this column to 0.5 of the values in 'L' column. Rename the columns 'Q', 'L' and 'K' as 'output', 'labor' and 'capital' respectively.

```
 Q     L
2350 2334
2470 2425
2110 2230
2560 2463
2650 2565
2240 2278
2430 2380
2530 2437
2550 2446
2450 2403
2290 2301
2160 2253
2400 2367
2490 2430
2590 2470
```

```
Answer:
rm(list=ls())
caplab <- read.table("c:\\data\\caplab.csv" , header=T, sep=",")
caplab
Q=caplab[1];L=caplab[2]
K=scale(L,center=100, scale=2/1) #scale to 1/2 of the values in "L"
K
caplabmat=cbind(Q,L,K) #create the X matrix by binding 3 columns
colnames(caplabmat)[1]="output"
colnames(caplabmat)[2]="labor"
colnames(caplabmat)[3]="capital"
caplabmat
```

```
output labor capital
1    2350  2334  1117.0
2    2470  2425  1162.5
3    2110  2230  1065.0
4    2560  2463  1181.5
5    2650  2565  1232.5
6    2240  2278  1089.0
7    2430  2380  1140.0
8    2530  2437  1168.5
9    2550  2446  1173.0
10   2450  2403  1151.5
11   2290  2301  1100.5
12   2160  2253  1076.5
13   2400  2367  1133.5
14   2490  2430  1165.0
15   2590  2470  1185.0
```

1.3 Exercise

FAC-22) Create a regression with the data from Question #22 above where output is the dependent variable and the independent variables are capital and labor. Explain how you know there is perfect collinearity in this model. Answer:

```
o=caplabmat[,1]
l=caplabmat[,2]
k=caplabmat[,3]
reg2=lm(o~l+k)
summary(reg2)
```

```
R-OUTPUT
Call:
lm(formula = o ~ l + k)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-73.632  -9.556  10.010  20.461  28.950

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.643e+03  2.122e+02  -7.744 3.19e-06 ***
l            1.702e+00  8.888e-02  19.154 6.55e-11 ***
k                   NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 30.66 on 13 degrees of freedom
Multiple R-squared: 0.9658,     Adjusted R-squared: 0.9631
F-statistic: 366.9 on 1 and 13 DF,  p-value: 6.549e-11
```

There is perfect collinearity because the results for "k" in the regression are all "NA" which means that the results for "k" are not available. This is how R handles the results of the second regressor when there is perfect collinearity.

1.4 Exercise

FAC-23) The data below into Excel and save the file as "caplab2.csv". Load the file into R. Using your knowledge of matrix algebra, create a function using matrices to prove that there is perfect collinearity in the model shown in Question #22 above. Be sure to include a line in your code that will print the results of your function on the screen.

```
Q L  K
2350 2334 1117
2470 2425 1162.5
2110 2230 1065
2560 2463 1181.5
2650 2565 1232.5
2240 2278 1089
2430 2380 1140
2530 2437 1168.5
2550 2446 1173
2450 2403 1151.5
2290 2301 1100.5
```

```
2160 2253 1076.5
2400 2367 1133.5
2490 2430 1165
2590 2470 1185

Answer  R Code:
perfColl<-function(inv){
caplab2 <- read.table("c:\\data\\caplab2.csv" , header=T, sep=",")
x=caplab2
y=x["Q"]
l=x["L"]
k=x["K"]
X=as.matrix(y,l,k)
XTX=t(X) %*% X;det(XTX)
inv=solve(XTX)
inv
}
perfColl(inv)
```

1.5 Exercise

FAC-24) Given the results found in Question #24, how does it confirm there is perfect collinearity? Answer: The results provided show that the inverse matrix does not exist. The value 1.135616e-08 is near to zero to indicate that the inverse matrix equals zero which is an indicator of perfect collinearity.

1.6 Exercise

FAC-25) What is another simple test you can run in R that will confirm that there is perfect collinearity between 'Q' and 'K'? Answer: Copy lines 2 through 6 from the solution in Question #24. Perform a 'cor' test on 'l' and 'k'. The results of the test returns a 1 which indicates perfect correlation between labor and capital.

1.7 Exercise

FAC-26) What is "near collinearity" and how does it affect OLS estimation? What should you look for to detect its existence? Answer: It is a case whereby the inverse of the (X'X) matrix does exist even when collinearity exists. It generally is detected by very large estimated coefficients, large standard errors and the estimated coefficients are statistically insignificant.

1.8 Exercise

FAC-27) Using the file 'caplab2.csv' and your knowledge of R, perform a regression of output against labor and capital AFTER rounding the values for capital. How do the results of this regression differ from the one you ran in Question 22? How is "near collinearity" exemplified in this regression? Answer:

```
rm(list=ls())
caplab <- read.table("c:\\data\\caplab2.csv" , header=T, sep=",")
caplab=as.matrix(caplab)
y=caplab[,1]
l=caplab[,2]
k=round(caplab[,3])
k
reg1=lm(y~l+k)
summary(reg1)


Here are the results of the regression:

Call:
lm(formula = y ~ l + k)

Residuals:
    Min      1Q  Median      3Q     Max
-61.159  -9.512  -2.296  22.960  36.402

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -250.85    1102.03  -0.228    0.824
l             -12.14      10.76  -1.128    0.281
k              27.68      21.52   1.286    0.223
```

```
Residual standard error: 29.91 on 12 degrees of freedom
Multiple R-squared: 0.9699,     Adjusted R-squared: 0.9649
F-statistic: 193.5 on 2 and 12 DF,  p-value: 7.402e-10
```

In this example, we see that there are now values for capital which was not the case in Question #22. The incidence of collinearity has not disappeared yet there are output results for capital. We can see a very high R^2 of 0.9699 with an F-statistic of 193.5 yet the regressors show p-values that statistically insignificant. These are signs of near collinearity.

1.9 Exercise

FAC-28) Create a 3 by 2 matrix named 'x1' with values ranging from 1 through 6. Then create a 2 by 3 matrix named 'y1' with the values 1 through 6. Using your knowledge of R, perform a Kronecker Product of these matrices and explain why the result is a 6 by 6 matrix. Answer:

```
x=seq(1:6)
y=seq(1:6)
x1=matrix(x,ncol=2)
y1=matrix(y,ncol=3)
x1

R OUTPUT

     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6

>y1
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6


#R code
k=kronecker(x1,y1)
k
```

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    3    5    4   12   20
[2,]    2    4    6    8   16   24
[3,]    2    6   10    5   15   25
[4,]    4    8   12   10   20   30
[5,]    3    9   15    6   18   30
[6,]    6   12   18   12   24   36
```

x1 is a T by n matrix and y1 is a m by p matrix. The resulting Kronecker Product matrix, 'k', has dimensions of Tm by np which in this case is a 3*2 by 2*3 or a 6 x 6 matrix.

1.10 Exercise

FAC-29) What is the determinant of the matrix 'k' in Question 28 above and explain why this is the case. Answer:

```
det(k)
```

[1] 7.100615e-30

We see that the result is extremely close to zero that it is accepted that the determinant is zero. Since the matrices 'x1' and 'y1' are not square matrices, therefore you cannot calculate their determinants. The determinant of 'k' is $det(x1)^m \times det(y1)^n$. Since this assumes both matrices are square, the relationship will yield a zero determinant.

FAC-30) $y = x'Ax$ is a quadratic form. Find the FOC condition (partial of y w.r.t x) given the data below for A and x.

2 3 A= x= 1,2 1 4

Answer: According to Matrix Algebra, the partial of y w.r.t. to x is $x' * (A + A')$. To perform (view) this calculation in R, use the steps provided below:

```
rm(list=ls())
A=c(2,1,3,4) #creates a vector of values
A=matrix(A,ncol=2) #creates a 2x2 matrix of A as illustrated above
x=c(1,2) #creates a vector x
xt=t(x) #takes the transpose of x and saves it to the object xt
At=t(A) #takes the transpose of A and saves it to the object At
AA=A+At #this step now adds the matrix A with A'
partY=xt%*%AA #This multiplies the transpose of x with the addition of A and A' to
```

```
solve the FOC
partY #This prints the results to the screen which will always be a 1 by n vector.

R-OUTPUT
     [,1] [,2]
[1,]   12   20
```

1.11 Exercise

FAC-31) Given the results in Question 30 above, find the second order condition for the quadratic and explain how you found your answer.
Answer: The second order condition is found by taking the second partial of y w.r.t. the second partial of x. Since the FOC was $x'(A + A')$, the second order condition is $(A+A')$. To see the results in R, run the program provided in Question 30 and at the prompt, type 'AA' which will give you the results for the second order condition.

```
AA

R OUTPUT
     [,1] [,2]
[1,]    4    4
[2,]    4    8
```

1.12 Exercise

FAC-32) Given the results in Question 32. What type of matrix is generated with the second order condition? Why is it that the second order condition is NOT met by 2A?
Answer: The matrix produced is a symmetric matrix. In a situation where the matrix "A" is symmetric, the second order condition can be found by the product of 2A. But since this matrix is not symmetric, this formula cannot be used.

1.13 Exercise

FAC-33) $y = x'Ax$ is a quadratic form. find the FOC and 2nd order conditions (partial of y w.r.t x) given the data below for A and x. Note that

the second partial of y w.r.t. x is equal to 2A. Choose the two rows of A to be (2,3) and (3,3) respectively, and the column vector x to have (1,2). Now evaluate the answers for this special case.
Answer: The code and the results of the FOC and 2nd order condition as calculated by R is shown below:

```
rm(list=ls())
A=c(2,3,3,3)
A=matrix(A,ncol=2)
x=c(1,2)
xt=t(x)
At=t(A)
AA=A+At
partY=xt%*%AA
party

R-OUTPUT
     [,1] [,2]
[1,]   16   18

AA

R OUTPUT
     [,1] [,2]
[1,]    4    6
[2,]    6    6

twoA = 2*A
twoA

R OUTPUT
     [,1] [,2]
[1,]    4    6
[2,]    6    6
```

1.14 Exercise

FAC-34) Let $y = a'Xa$ where "a" is n by l vector of constants and X is an n by n square matrix. The FOC where the partial of y w.r.t X is 2aa' - diag(aa'). Create a program in R that illustrates this. Answer:

```
rm(list=ls())
x=c(2,4,6,8,10,12,14,16,18)
X=matrix(x,ncol=3)
X

R OUTPUT
     [,1] [,2] [,3]
[1,]    2    8   14
[2,]    4   10   16
[3,]    6   12   18

#R Input
a=c(5,6,7)
at=t(a)
aa=a*at
aa

R OUTPUT
     [,1] [,2] [,3]
[1,]   25   36   49

#R input
diaga=diag(aa)
diaga

R OUTPUT
[1] 25

#R input
partyX=2*aa-diaga
partyX

R OUTPUT
```

```
     [,1] [,2] [,3]
[1,]   25   47   73
```

1.15 Exercise

FAC-35) Given two 2 by 2 matrices 'A' and 'X' A having two rows with elements (3,1) and (5,2) and X having two rows with elements (1,2) and (2,3). Evaluate the partial of the trace of 'AX' w.r.t. 'X'.
Answer: The solution of this partial is the transpose of the matrix 'A'. Given the matrix 'A', the solution is shown below:

```
rm(list=ls())
A=c(3,5,1,2)
A=matrix(A,ncol=2)
A

R OUTPUT
     [,1] [,2]
[1,]    3    1
[2,]    5    2

#R input
At=t(A)
At

R OUTPUT
     [,1] [,2]
[1,]    3    5
[2,]    1    2
```

1.16 Exercise

FAC-36) Discuss three main contributors to the problem of autocorrelation in regression errors. Answer: There are several factors that can cause autocorrelation. One is misspecification of the least squares model. This occurs when a key regressor is left out of the model. This will cause what is known as false autocorrelation. Misspecification also occurs when the model has an incorrect functional form. For example, suppose in your model revenue is dependent on advertising: $R = b1 + b2 * advertising + e$, but the correct model should be $R = b1 + b2 * advertising + b3 * advertising^2 + e$. This type of misspecification will lead to autocorrelation. Another cause of autocorrelation is cyclical data such as GDP, price indexes, etc. The effect of momentum built into the data will cause autocorrelation to exist among regression errors.

1.17 Exercise

FAC-37) What three key problems for OLS occur due to autocorrelation?
Answer: One problem of autocorrelation is it causes R^2 to be overestimated. A second problem is that it underestimates the variance of the mean and it will also overestimate parameter t-statistics and F-statistic of the regression.

1.18 Exercise

FAC-38) What test should be used to test for autocorrelation and what is the proper way to interpret the results?
Answer: To test for autocorrelation, a Durbin-Watson statistic is run to check for the problem. The NULL Hypothesis is hat there is no autocorrelation and the Alternative Hypothesis, H1, is that autocorrelation exists. In order to properly interpret the value of the Durbin-Watson statistic, there are several factors to consider. These factors are the number of observations and the number of independent variables. A Durbin-Watson table is also needed to correctly interpret the test value.

For example, if there are 2 independent variables with 30 observations, the upper and lower bounds in the 95% confidence interval are 1.28 and 1.57 respectively. To correctly interpret the test value, if the Durbin-Watson test returns a value between 0 and the lower limit of 1.28, it indicates that there is positive autocorrelation between the residuals. A value that lies between 4 minus the lower limit and 4 indicates that there is negative autocorrelation. If it lies within the upper and lower range or within 4 minus the upper limit and 4 minus the lower limit, it indicates that the test is inconclusive and no decisions can be made. There is no autocorrelation if the test results falls between the upper limit of 1.57 and 4 minus the upper limit value of 1.57.

1.19 Exercise

FAC-39) Given the regression run in Question 27 above, create a function that will return the diagonal values of the hat matrix and the vector of the fitted values of the regression. Include summary statistics of the residuals. Interpret these results. Answer:

```
hatvalues(reg1)# diagonals of hat matrix H defined from
#the regression run in Question 27

R OUTPUT
         1          2          3          4          5          6          7
0.09158384 0.17870030 0.27325292 0.27902287 0.44266006 0.16680141 0.06925449
         8          9         10         11         12         13         14
0.18840871 0.09937841 0.23444769 0.22019477 0.30568447 0.23676519 0.08534819
        15
0.12849668

fitted(reg1)
       1        2        3        4        5        6        7        8
2332.367 2473.201 2155.599 2565.467 2711.159 2237.184 2410.553 2493.598
       9       10       11       12       13       14       15
2522.734 2463.485 2262.438 2180.853 2402.296 2495.538 2563.526

summary(fitted(reg1))

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2156    2297    2463    2418    2509    2711

resid(reg1)
         1          2          3          4          5          6          7
 17.632708  -3.201154 -45.598511  -5.467390 -61.159129   2.815897  19.446516
         8          9         10         11         12         13         14
 36.402448  27.266328 -13.485401  27.561623 -20.852786  -2.296207  -5.538475
        15
 26.473533

summary(resid(reg1))
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
```

```
-6.116e+01 -9.512e+00 -2.296e+00 -3.849e-16  2.296e+01  3.640e+01
```

1.20 Exercise

FAC-40) Rewrite the code shown in the solution to Question 39 so that the hat values are passed to an object named 'h' and the fitted values are passed to an object named 'f'. Plot both the hat values and fitted values and interpret the results.
Answer: the code to accomplish this is shown below:

```
h=hatvalues(reg1)
f=fitted(reg1)
qq.plot(h);qq.plot(f)
```

Each graph shows the outliers which is an indicator that there are potential issues such as heteroscedasticity. Examining these graphs shows that they are a reflection of each other.

1.21 Exercise

FAC-41) Load the data package named 'Airline' which is in the 'Ecdat' library into R. Convert the data into a matrix and then print the names of the fields contained in the data.
Answer:

```
data(Airline)
names(Airline)
air=matrix(Airline)
air
```

1.22 Exercise

FAC-42) Pass the total cost data and the fuel price data for all airlines to objects named 'tc' and 'fp' respectively. Then pass the data for airline #1 to an object named 'tc2' and 'fp2' respectively. Print the data for airline #1 to the screen.
Answer:

```
tc=air[,3]
fp=air[,4]
tc;fp
tc2=tc[1:15]
fp2=fp[1:15]
tc2;fp2
```

1.23 Exercise

FAC-43) Run a regression where you test to see the impact of fuel costs on the total costs for airline 1. Based on the results of your regression, what impact does the price of jet fuel have on the total costs for this airline?
Answer:

```
regcost=lm(tc~fp)
summary(regcost)
```

The beta coefficient is significant and the R^2 and $Adj.R^2$ are 0.8581 and 0.8565 respectively. At first glance, the conclusion would be that fuel costs explain better than 85 percent of the total costs of this airline.

1.24 Exercise

FAC-44) Given the conclusion of the regression run in Question #43 above, what should be done before making any definitive conclusions about the results of the regression?
Answer: A plot of the residuals should be done to see if there is any indication of heteroscedasticity or autocorrelation. If, after plotting residuals, there is an indication of one or both of these problems, then further diagnostic tests should be conducted.

1.25 Exercise

FAC-45) Plot the residuals of the regression in Question #43 above and provide an analysis of the plot. Are there ay outliers?
Answer: To plot the residuals, perform a qq plot and analyze the results. The code to do this is provided below. Note in the plot how the residuals how no outliers are indicated though the residuals seem to form a pattern that would indicate that heteroscedasticity exists.

```
qq.plot(regcost)
```

1.26 Exercise

FAC-46) Given the results of the qq plot performed in Question #44 above, what other plot would you do to perform more detailed analysis of the residuals? Create code that will generate these graphs in R.

```
plot(regcost)
```

R allows you to generate four detailed graphs that will identify outliers and residual patters. The first graph presents the residuals versus fitted values. The plot shows that there are 3 outliers and that the second plot is the Normal Q-Q graph showing the relationship between standardized residuals and theoretical quantiles. The third graph shows the relationship between the fitted values and the square root of the standardized residuals. What is interesting about this third graph is the patterns shown between the residuals and the fitted values. It indicates a problem since there should be no patterns. The last graph shows the leverage or the sensitivity of the regression to last individual observation. If the leverage is high, it is an indicator of heteroscedasticity. This graph indicates a high leverage and therefore the possibility of heteroscedasticity.

2 Exercises suggested by Brandon C. Vick (BCV)

2.1 Exercise BCV-1

Social scientists do not always have the luxury of having a well-developed theory when choosing appropriate set of regressors. Often, there are several candidate regressors and one wants to select the best subset.

1) Write a general R function that takes a y vector of dimension $(n \times 1)$ and a matrix X of possible regressors of dimension $(n \times p)$. The R function should regress y on all possible right hand side (RHS) combinations contained within the matrix X (i.e. $x1; x1 + x2, x2$, etc...) and return the coefficient estimates for only one specification yielding the highest Adjusted R^2 value.

2) Illustrate your function by using the Caschool dataset (in Ecdat package) to test with y=testscr and X = cbind (enrltot, teachers, calwpct, mealpct, computer, compstu, expnstu, str, avginc, elpct).
Answer:

```
##############################################################
# Function will perform  OLS regression y on all possible
#    RHS combinations contained
#     within X (i.e. x1; x1 + x2, x2, etc...)
#     and return the beta estimates for the
#     specification yielding the highest Adjusted-R-Square.
#
# By Brandon Vick
# November 11, 2008
# Inputs: y (vector nX1); X (matrix nXp)
# Outputs: Estimated b (vector pX1); R-Square
##############################################################

AllRegressCombinations=function(y, X)
{
nx = ncol(X) #Maximum number of regressors
nr = nrow(X)
regcount = 1
SST = sum((y-mean(y))^2) # Total Sum of Squares
r2ij = matrix(NA, 2^nx-1, 3) # creates empty matrix holding adjusted R-Squared
                             # values for all regression combinations
for (i in 1:nx) #sets number of regressors (pair of 1, 2, 3, 4,...)
{
comb = combn(nx, i) #Returns regressor combinations

for (j in 1:ncol(comb)) #looks at all combinations for given number
{
xij = X[,comb[,j]]
regij = lm(y ~ xij)
#Find AdjR2
su=summary(regij)
            AdjR2 = su$adj  #extract adjusted R square by $
r2ij[regcount, 1] = i # number of regressors
```

```
r2ij[regcount, 2] = j # specific combination of regressors
r2ij[regcount, 3] = AdjR2
regcount = regcount+1
} #end j-loop
} #end i-loop

maxR2row = which.max(r2ij[,3]) # yields row # of maximum Adjusted-R2
imax = r2ij[maxR2row,1]; jmax = r2ij[maxR2row,2];
R2max= r2ij[maxR2row,3]

####MUST RE-RUN THIS REGRESSION COMBINATION AND RETURN VALUES######
combmaxR2 = combn(nx, imax) # gives number of regressors
xijmaxR2 = X[,combmaxR2[,jmax]]
regijmax = lm(y ~ xijmaxR2)
list(regijmax = regijmax, R2max = R2max)
} # End AllRegressCombinations Function

##############################################################
# To Run AllRegressCombinations Function:
rm(list=ls());  # to clean out the memory
#
#  ENTER THE ABOVE FUNCTION HERE
# source("PLACE DIRECTORY AND FILENAME HERE")
#
library(Ecdat); data(Caschool);
attach(Caschool); y=testscr;
X=cbind(enrltot, teachers, calwpct, mealpct,
computer, compstu, expnstu, str, avginc, elpct)
AllRegressCombinations(y,X)
##############################################################
```

2.2 Exercise BCV-2

1) Using the Cigar dataset (Ecdat package), perform a linear regression to find estimates of price and income elasticity of demand using the variables sales, price, ndi, and cpi (for real values). Perform this for the $state = 14$. What should the sign of each coefficient be? What do these coefficients stand for?

2) Check for serial autocorrelation and report the Durbin-Watson statistic (Hint: Use car library).

Answer:

To find elasticities, regress the following double-log form:

$$ln(sales) = \beta_0 + \beta_1 ln(\frac{price}{cpi}) + \beta_2(\frac{ndi}{cpi}) + \epsilon \tag{1}$$

```
##########Load DATASET Cigar##############
rm(list=ls()); library(Ecdat); data(Cigar);
Cigar14=subset(Cigar, Cigar[1]==14); attach(Cigar14)

##########Compute logs and Run Regression##############
lnsales = log(sales); lnRealprice = log(price/cpi);
lnRealIncome = log(ndi/cpi);
reg1=lm(lnsales ~ lnRealprice+lnRealIncome)
summary(reg1)

##########Check for Autocorrelation##############
library(car)
durbin.watson(reg1, max.lag=4)
##########Use HAC estimate of standard errors ###########
library(sandwich)
coef(reg1)/sqrt(diag(vcovHAC(reg1))) #refined t stats
qt(0.025, df=27) #-2.051831 is the critical value
# the refined t stats remain statistically significantly negative
```

For state 14, the coefficient for $lnRealprice = -.427$, indicating negative price elasticity (as cigarette prices rise, then quantities bought decrease). The coefficient for $lnRealIncome = -0.709$, indicating that cigarettes are inferior. The Durbin-Watson statistic yields a p-value less than 0.05 for lags 1 and 2, indicating autocorrelation at these lags, i.e., evidence rejecting the null hypothesis of no autocorrelation. This rejection means that our conclusions about negative price elasticity and inferiority of cigerettes is subject to some doubt. Hence we use the R package ‘sandwich’ to compute heteroscedascity and autocorrelation consistent (HAC) estimate of the standard errors and also to compute the refined t statistics. We use the 'qt' function in R to find the critical value for the t test. Since observed refined t statitics are more negative than the critical value -2.051831 at the 2.5% level in one tail, the doubts regarding conclusion of negative price elasticity and inferiority can be removed.

2.3 Exercise BCV-3

Use the plm package to format the Cigar data for panel estimation using state and year as index. For the panel dataset, perform the same regression as in BCV-2 using fixed effects. Report any significant coefficients, the multiple R-Squared, the F-Statistics, and p-value.

Answer:

```
##########Load DATASET Cigar##############
rm(list=ls()); library(Ecdat); data(Cigar); attach(Cigar);
library(plm); # loads library for panel data

##########Prepare Panel Data and Run PLM Regression#############
cigarpanel <- plm.data(Cigar, index = c("state", "year"))
panelreg1=plm(log(sales) ~ log(price/cpi) + log(ndi/cpi),
data = Cigar, model = "within");
summary(panelreg1)
fixef(panelreg1) #extract fixed effects for 51 states
```

The coefficient representing price elasticity equals $-.0702$ and is significant at 0.001 percent. Multiple $R^2 = 0.5445$; F-stat = 796.453; and p-value = 0.0013.

2.4 Exercise BCV-4

For both Cobb-Douglas and MNH form, estimate the marginal elasticity of capital and labor, as well as the elasticity of scale and elasticity of substitutions for transportation equipment manufacturing using the TranspEq dataset (Ecdat package). Do these yield the same value?
Answer:

EOS for Cobb-Douglass should always equal 1. The following equation must be regressed to find $ME_K = a_1$ and $ME_L = a_2$:

$$ln(va) = a_0 + a_1 ln(capital) + a_2 ln(labor) \tag{2}$$

EOS for MNH need not equal 1. The following equation must be regressed to find $ME_K = b_1$ and $ME_L = b_2$:

$$ln(va) = b_0 + b_1 ln(capital) + b_2 ln(labor) + b_3 ln(capital) * ln(labor) \tag{3}$$

$$EOS_{MNH} = \frac{ME_K + ME_L}{ME_K + ME_L + b_3} \tag{4}$$

```
##########Load DATASET TranspEq##############
rm(list=ls()); library(Ecdat); data(TranspEq); attach(TranspEq);
regCD = lm(log(va) ~ log(capital) + log(labor))
regMNH = lm(log(va) ~ log(capital) + log(labor) + log(capital)*log(labor))
summary(regCD)
summary(regMNH)

##########Extract Coefficients and Calculate SCE and EOS##############
a1=regCD$coef[2]; a2=regCD$coef[3]; SCE_CD=a0+a1; SCE_CD
b1=regMNH$coef[2];b2=regMNH$coef[3]; b3=regMNH$coef[4];
SCE_MNH=b1+b2; SCE_MNH;
EOS_MNH=(b1+b2)/(b1+b2+b3); EOS_MNH
```

For the Cobb-Douglass estimation, $ME_K = 0.245$ and $ME_L = 0.805$. The scale elasticity measured is 1.051, suggesting increasing returns to scale.

For the MNH estimation: $ME_K = 0.404$ and $ME_L = 0.811$. The scale elasticity measured is 1.214, suggesting increasing returns to scale. The elasticity of scale equals 1.0214.

2.5 Exercise BCV-5

Use the Consumption dataset (Ecdata package) to compare the following regressions (See Davidson and MacKinnon (2004) Econometric Theory and Methods. New York, Oxford Univ. Press, p. 41, 121, and 176):

i) The simplest regression of consumption on income
ii) The change of consumption on the change in income and the past four lags of change of income.
iii) Consumption regressed on lagged consumption, income, and lagged income

Answer:

```
##########Load DATASET Consumption#############
rm(list=ls()); library(Ecdat); data(Consumption);
I=Consumption[,1]; C=Consumption[,2]

##########Find Values for Lags and Changes in C, I##############
lagC=lag(C, k=-1); chgC=C-lagC;
lagI=lag(I, k=-1); lagI2=lag(I, k=-2);lagI3=lag(I, k=-3);
lagI4=lag(I, k=-4); lagI5=lag(I, k=-5);
chgI=I-lagI; chgI2=lagI-lagI2; chgI3=lagI2-lagI3
chgI4=lagI3-lagI4; chgI5=lagI4-lagI5
###NOTE: must use k=-1 for lags in the paste###

###NOTE: NA values must be removed before regression###
chgmatrix = cbind(C, chgC, I, chgI, chgI2, chgI3, chgI4, chgI5, lagC, lagI)
chgmatrix <- chgmatrix[complete.cases(chgmatrix), ]  # removes NA
C=chgmatrix[,1];  chgC=chgmatrix[,2];
I=chgmatrix[,3] ; chgI=chgmatrix[,4];  #renames columns
chgI2=chgmatrix[,5]; chgI3=chgmatrix[,6]; chgI4=chgmatrix[,7];
chgI5=chgmatrix[,8]; lagC=chgmatrix[,9]; lagI=chgmatrix[,10]

##########Run Different Regressions#############
reg1=lm(C ~ I)
reg2=lm(chgC ~ chgI+chgI2+chgI3+chgI4+chgI5)
reg3=lm(C ~ lagC+I+lagI
```

For the first regression, $MPC = 0.86$ and $AdjustedR - Square = 0.99$.

The second regression yields the following results:

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 1080.37997  200.44901   5.390 2.09e-07 ***
chgI           0.23605    0.04383   5.386 2.13e-07 ***
chgI2          0.09507    0.04338   2.191  0.02964 *
chgI3          0.01687    0.04306   0.392  0.69574
chgI4          0.09173    0.04361   2.103  0.03675 *
chgI5         -0.11626    0.04396  -2.645  0.00887 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The third regression yields $MPC = 0.22$ and $AdjustedR{-}Square = 0.99$. A marginal propensity to consume of 0.22 is perhaps more realistic after controlling for lagged consumption and income.

2.6 Exercise BCV-6

Use the 'Hstarts' dataset of Housing data ('Ecdat' package) to do the following (see Davidson and MacKinnon (2004), Econometric Theory and Methods. New York, Oxford Univ. Press, p. 602):

i) Create dummy variables for Q1, Q2, and Q3
ii) Create a trend vector for quarterly data t=[1 1 1 1 2 2 2 2 3 3 3 3...]
iii) Regress hs on the lag of hs, the seasonal dummies, dummies interacted with trend, and dummies interacted with trend squared (NOTE: use unadjusted data)
iv) Check Durbin Watson statistic for first four lags using the 'car' package

Answer:

```
##########Load DATASET Hstarts ##############
rm(list=ls()); library(Ecdat); data(Hstarts);
hs=Hstarts[,1]

########## Create Matrix of Dummy Variables##############
```

```
########## NOTE: Data is already sorted here ##############
qdum=nrow(Hstarts)/4
dq1=rep(c(1,0,0,0), qdum)
dq2=rep(c(0,1,0,0), qdum)
dq3=rep(c(0,0,1,0), qdum)

########## Create Trend Vector##############
trend=rep(1:qdum, each=4)

########## Create Interaction with Trend-squared Vector##############
int1=dq1*trend; int2=dq2*trend; int3=dq3*trend
int1sq=dq1*trend^2; int2sq=dq2*trend^2; int3sq=dq3*trend^2

########## Create lag of hs ##############
laghs=lag(hs, k=-1)

########## Run regression##############
reg1=lm(hs ~ laghs+dq1+dq2+dq3+int1+int2+int3+int1sq+int2sq+int3sq)

##########Check Durbin Watson##############
library(car)
durbin.watson(reg1, max.lag=5)
```

The following shows results from the regression and Durbin-Watson test.

```
Coefficients:
              Estimate Std. Error    t value Pr(>|t|)
(Intercept)  0.000e+00  1.911e-16  0.000e+00 1.000000
laghs        1.000e+00  2.022e-17  4.945e+16  < 2e-16 ***
dq1         -2.935e-17  4.004e-17 -7.330e-01 0.464718
dq2          1.721e-16  3.351e-17  5.135e+00 8.26e-07 ***
dq3         -1.046e-17  3.301e-17 -3.170e-01 0.751835
int1         2.091e-18  3.668e-18  5.700e-01 0.569508
int2        -1.595e-17  3.515e-18 -4.538e+00 1.12e-05 ***
int3         4.307e-19  3.430e-18  1.260e-01 0.900245
int1sq      -5.386e-20  8.195e-20 -6.570e-01 0.511973
int2sq       3.087e-19  7.898e-20  3.908e+00 0.000138 ***
```

```
int3sq      -8.383e-21  7.766e-20 -1.080e-01 0.914185
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 6.409e-17 on 157 degrees of freedom
Multiple R-squared:     1,      Adjusted R-squared:     1
F-statistic: 5.665e+32 on 10 and 157 DF,  p-value: < 2.2e-16

> durbin.watson(reg1, max.lag=4)
 lag Autocorrelation D-W Statistic p-value
   1    -0.102428140      2.197927       0
   2    -0.026200796      1.311102       0
   3    -0.074904796      1.404388       0
   4    -0.002351982      1.257582       0
   5    -0.008574595      1.266951       0
 Alternative hypothesis: rho[lag] != 0
```