Spring 2024 - Villas-Boas, *Problem set 2, ARE 212, Department of Agricultural and Resource Economics*

Part I Applied

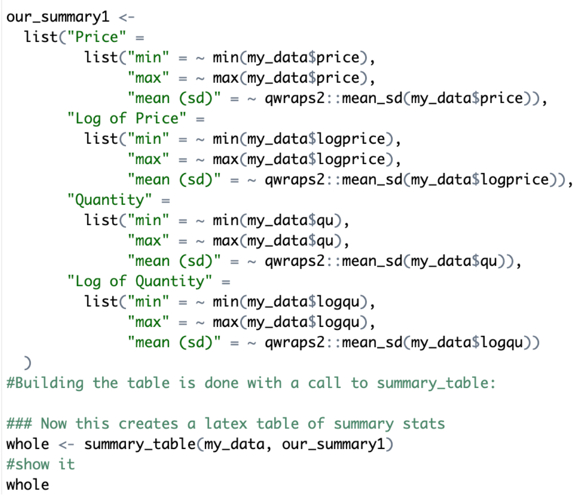
This problem set is to be solved using R, so that you get familiar with R and apply all the concepts learned up to the end of Greene Chapter 3. All is to be done with matrix algebra coded in R, you can always check if ok with canned lm() off course.

To produce a pdf with all code and output run, click File > Knit Document or Compile Report in R studio**.** This will produce a single pdf you can attach to handwritten answers for submission.

At the top of your R file

```
#when you run this for the first time, uncomment and install the packages below. For future runs, comment out #

#-------------------------------------------

# Install the 'pacman' package

install.packages("pacman")

#Now load it... the 'pacman' package

library(pacman)

#packages to use load them now using the pacman "manager"

p_load(dplyr, haven, readr, knitr, psych, ggplot2,stats4, stargazer, lmSupport, magrittr, qwraps2, Jmisc )

# the line above Installs all the packages named "haven" and "readr" and the rest using pacman.

#Type ?packaname  in the console prompt and enter, if you want to learn what each does

#set your working directory

setwd("/Users/sofiavillas-boas/Dropbox/ARE212/Spring2024/FirstHalf/PS2_Spring2024")
```

1. The dataset is in Stata and was created for the purpose of this problem set only. It is available on bcourses and is called pset2_2024.dta. Read the data into R `my_data <- read_dta("pset2_2024.dta")`
   Units for each data variable atre in the header of each column in the dataset
   Then Create three new variables: log price in euros, log of quantity (log(qu)), and number of registered cars per capita (carspc= qu/population)
2. Get the summary statistics for each of quantity, price in Euros, log quantity, and log price, with sample mean, standard deviation, minimum and maximum.
   `describe(my_data)`
   Then create a table ready for Latex of min, max, mean, and sd of the variables to be used in regressions in this problem set below Hint: use



```
our_summary1 <-
  list("Price" =
         list("min" = ~ min(my_data$price),
              "max" = ~ max(my_data$price),
              "mean (sd)" = ~ qwraps2::mean_sd(my_data$price)),
       "Log of Price" =
         list("min" = ~ min(my_data$logprice),
              "max" = ~ max(my_data$logprice),
              "mean (sd)" = ~ qwraps2::mean_sd(my_data$logprice)),
       "Quantity" =
         list("min" = ~ min(my_data$qu),
              "max" = ~ max(my_data$qu),
              "mean (sd)" = ~ qwraps2::mean_sd(my_data$qu)),
       "Log of Quantity" =
         list("min" = ~ min(my_data$logqu),
              "max" = ~ max(my_data$logqu),
              "mean (sd)" = ~ qwraps2::mean_sd(my_data$logqu))
  )
#Building the table is done with a call to summary_table:

### Now this creates a latex table of summary stats
whole <- summary_table(my_data, our_summary1)
#show it
whole
```

3. Create a histogram for the quantity **qu** (15 brackets) label everything, axis, add title

Hint: *histYvar<-ggplot(my_data, aes(x=my_data$Yvar)) + geom_histogram(bins = 15)*

#add labels etc and title

(histYvar <- histYvar + xlab("Name of Y variable") + ylab("Number of Observations") + ggtitle("Histogram of Y variable"))

Note that the ( ) are around the command line of the histogram above. So it will display and no need to type histYvar again. If you had written without parentheses around no(histYvar…) just like below

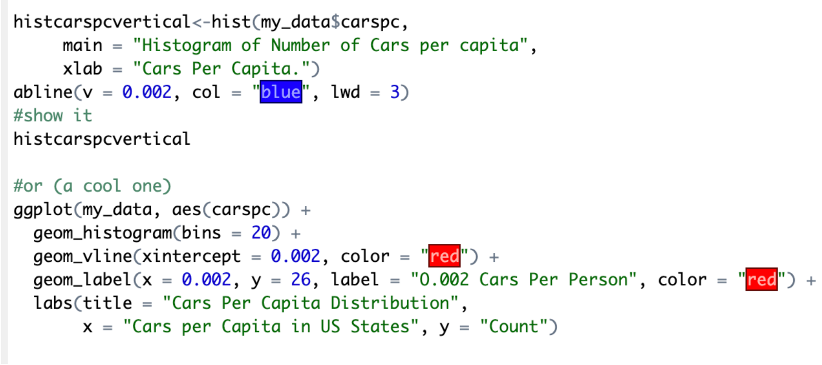
histYvar <- histYvar + xlab("Name of Y variable") + ylab("Number of Observations") + ggtitle("Histogram of Y variable")

then you would have to type also #show it

histYvar

4. In another histogram for quantity per capita (carspc), add a vertical red line at carspc=1

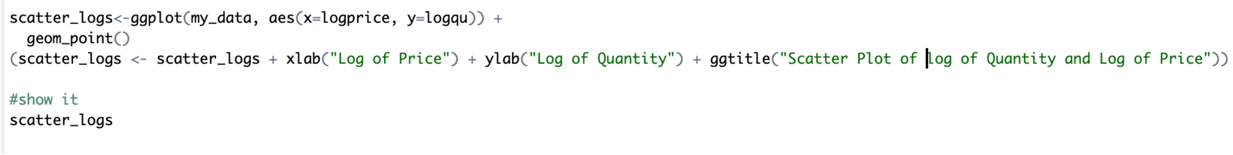
Hint: see below



```
histcarspcvertical<-hist(my_data$carspc,
     main = "Histogram of Number of Cars per capita",
     xlab = "Cars Per Capita.")
abline(v = 0.002, col = "blue", lwd = 3)
#show it
histcarspcvertical

#or (a cool one)
ggplot(my_data, aes(carspc)) +
  geom_histogram(bins = 20) +
  geom_vline(xintercept = 0.002, color = "red") +
  geom_label(x = 0.002, y = 26, label = "0.002 Cars Per Person", color = "red") +
  labs(title = "Cars Per Capita Distribution",
       x = "Cars per Capita in US States", y = "Count")
```

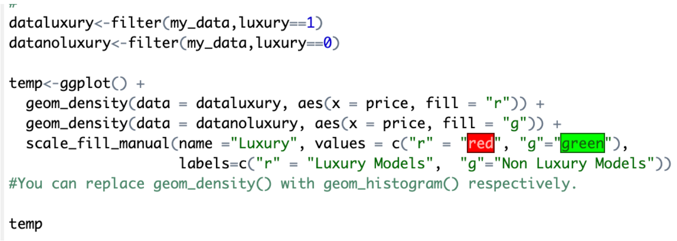
5. Create a graph that plots **qu** against **price** but uses series names, not variable names or abbreviations, so that the graph is very clear to anyone who sees it for the first time. Do another plot for **logqu** and **logprice**.

Hint:



```
scatter_logs<-ggplot(my_data, aes(x=logprice, y=logqu)) +
  geom_point()
(scatter_logs <- scatter_logs + xlab("Log of Price") + ylab("Log of Quantity") + ggtitle("Scatter Plot of Log of Quantity and Log of Price"))

#show it
scatter_logs
```

6. And produce two overlapping histograms for the price of luxury and nonluxury cars.



```
dataluxury<-filter(my_data,luxury==1)
datanoluxury<-filter(my_data,luxury==0)

temp<-ggplot() +
  geom_density(data = dataluxury, aes(x = price, fill = "r")) +
  geom_density(data = datanoluxury, aes(x = price, fill = "g")) +
  scale_fill_manual(name ="Luxury", values = c("r" = "red", "g"="green"),
                    labels=c("r" = "Luxury Models",  "g"="Non Luxury Models"))
#You can replace geom_density() with geom_histogram() respectively.

temp
```

#don't forget label everything and title, above is a possible way to do it.

7. Export your data as a comma delimited ascii file.
   Do you need a new package?
   Hint: write.csv(my_data, file = "my_data2024.csv")

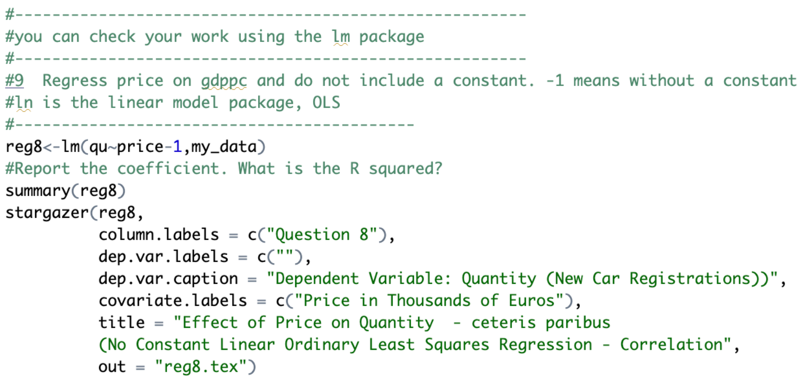
8. Regress **qu** on **price** and do not include a constant. Report the coefficient. What is the R squared? Now regress **carspc** on **price**. Report the coefficient on price and compare it to the previous coefficient. Check if they are different in R using all.equal() . Explain your findings.

Hint : *Do this using matrix multiplications b=inv(X'X) X'y.. To check whether your matrix created coefficients and R squared are correct, #use lm() command and no constant, x1 and x2 RHS variables, Y is LHS variable. See below hint. Then summary shows output of linear regression estimates. Part -1 means no constant in the lm() command (lm stands for the linear model (lm)) below to check is you did the matrix algebra correctly in R*

Reg6 <- lm(Y~x1+x2-1,my_data)

summary(Reg6)

#Hint on nice tables



```
#--------------------------------------------------------
#you can check your work using the lm package
#--------------------------------------------------------
#9  Regress price on gdppc and do not include a constant. -1 means without a constant
#ln is the linear model package, OLS
#--------------------------------------------
reg8<-lm(qu~price-1,my_data)
#Report the coefficient. What is the R squared?
summary(reg8)
stargazer(reg8,
          column.labels = c("Question 8"),
          dep.var.labels = c(""),
          dep.var.caption = "Dependent Variable: Quantity (New Car Registrations))",
          covariate.labels = c("Price in Thousands of Euros"),
          title = "Effect of Price on Quantity  - ceteris paribus
          (No Constant Linear Ordinary Least Squares Regression - Correlation",
          out = "reg8.tex")
```

9. Get degrees of freedom (n-k), b (the coefficient), n from the regression of quantity on price and no constant. Generate a series of the predicted values of quantity and plot those against the quantity data series. Compute the residuals series and plot the residuals against price. What do you see in terms of fit and whether the constant variance assumption for the residuals is valid or not?

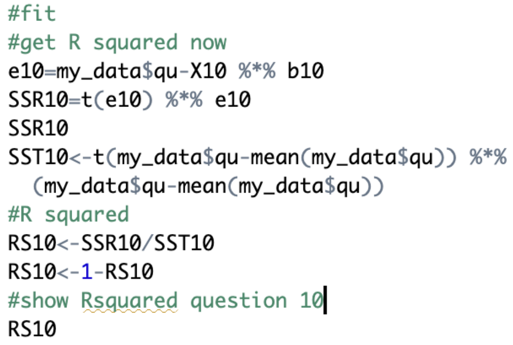
Hint: Do it with matrix again . Then check betas and all measures asked for using lm() get Reg8, then N<-nobs(Reg8) there are saved variables Yhat<-Reg8$fitted.values etc… lm also has saved output with that information, you can check if your matrix calculations are correct by looking at lm output from Reg8.

10. Regress **quantity** on **price and a constant**, and get b (the coefficient). Generate a series of the predicted price values and plot those against the price data series. Compute the residuals series and plot the residuals against gdppc. What do you see in terms of fit and whether the constant variance assumption for the residuals is valid or not? Has the fit improved or not relative to the question 8 analysis?

Hint: To add a constant to a matrix of X's do this #create a column of ones and add to X

X<-my_data$var1

Xn<-cbind(1, X)

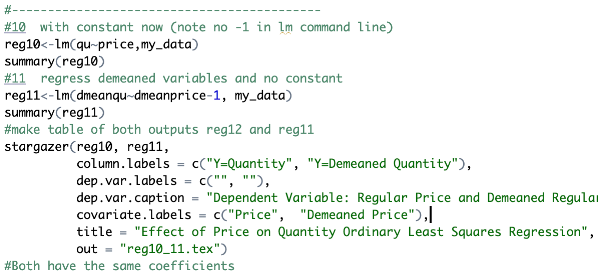


```
#fit
#get R squared now
e10=my_data$qu-X10 %*% b10
SSR10=t(e10) %*% e10
SSR10
SST10<-t(my_data$qu-mean(my_data$qu)) %*%
  (my_data$qu-mean(my_data$qu))
#R squared
RS10<-SSR10/SST10
RS10<-1-RS10
#show Rsquared question 10
RS10
```

11. Demean regular quantity, call it **dmeanqu** and demean price and call it **dmeanprice**. Regress the demeaned quantity on demeaned price variable and no constant, and compare to analysis in question 10. Why do you get this? Explain the theorem behind this briefly.

Hint: demean variables first create a new Y and new X my_data$dmeanx<-my_data$x-mean(my_data$x9) and then do OLS matrix version formula for b. Check also with lm() to make sure all are coded correctly with matrix algebra

Hint to make a nice ready for latex table and have side by side estimates reg 10 and reg 11



```
#-------------------------------------------
#10  with constant now (note no -1 in lm command line)
reg10<-lm(qu~price,my_data)
summary(reg10)
#11  regress demeaned variables and no constant
reg11<-lm(dmeanqu~dmeanprice-1, my_data)
summary(reg11)
#make table of both outputs reg12 and reg11
stargazer(reg10, reg11,
          column.labels = c("Y=Quantity", "Y=Demeaned Quantity"),
          dep.var.labels = c("", ""),
          dep.var.caption = "Dependent Variable: Regular Price and Demeaned Regula
          covariate.labels = c("Price",  "Demeaned Price"),
          title = "Effect of Price on Quantity Ordinary Least Squares Regression",
          out = "reg10_11.tex")
#Both have the same coefficients
```

12. Regress quantity on a constant, on price, on a luxury indicator, on weight, and on fuel efficiency. Generate a series of the predicted quantity values and plot those against the quantity data series: What do you see in terms of fit? Compute the residuals series and plot the residuals against the fuel efficiency: is the constant variance assumption for the residuals valid or not? *Hint: do this with ols matrix formulas and then check then with lm() package. Also, you may make nice latex tables if you like see above hint*
13. Regress **Quantity** on **a constant, price, weight, luxury indicator..** Save residuals as **qres or Y13**. Now regress **fuel** on **a constant, price, weight, and luxury indicator.** Save these residuals as **fuelRres or X13.** Now regress **qres** on **fuelRres (or Y13 on X13)** and no constant. Report your findings. We wanted to get the effect of fuel consumption on quantity, all else constant. To which coefficient of a previous question is the coefficient of fuel**Rres (or X13)** equal to, and why?
Hint: do this with ols matrix formulas and check then with lm() package. Make nice latex tables

14. Repeat regression 12 but now use **log qu** and **log price** and the other variables. Call this the Regression in logs. Is the estimated car demand elastic with respect to price?

15. The midterm will feature some of these style questions to be done in R. Set seed equal to 12345. Generate two random variables, x and e, of dimension n = 100 such that x, e N(0, 1). Generate a random variable **y** according to the data-generating process $y_i = x_i + e_i$. Show that if you regress **y** on **x** and a constant, then you will get an estimate of the intercept β_0 and the coefficient on **x**, β_1. Increase the sample to 1000, then 10000, and repeat the estimation. What do you see as you increase the sample? [later in the class, When we talk about large samples & what happens to the std error of the betas, come back to this, code the std error of betas, and see what happens as N increases.]

End of Problem Set 2 -

Part II Theory – No need to turn in this one; it is meant as a review of lectures.

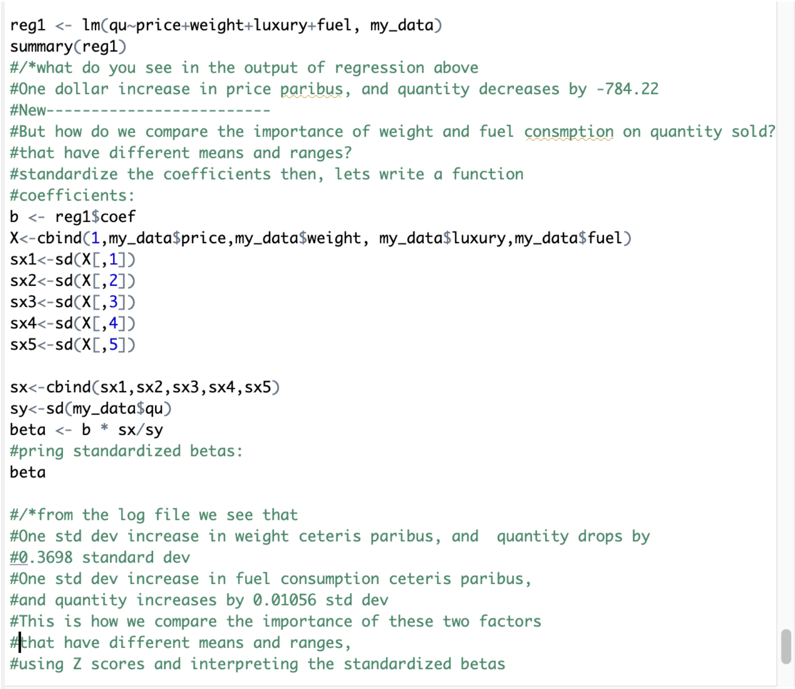
1. Show that PM=MP=0. Where P and M are the projection and residual maker matrices defined in the lecture.
2. Show that PX=X
3. Show that y=Py+My
4. Show that a regression of y on the transformed variable Z=XA, where A is non-singular and defined in lecture $A=(X'X)^{-1}X'$, has the same R squared as a regression of Y on X. (this was in one of the above-applied exercises).

Part III- Optional – FYI below

- How to choose between levels and logs? #compare correlation of predicted and actual quantities and choose the model with highest one

```
Yhatlog<-X14 %*% b14

Yhatlevel<-X12 %*% b12

Y<-my_data$qu

cor(Yhatlog,Y)   and   cor(Yhatlevel,Y)
```

- How to interpret coefficients in regressions when the regressors are measured in different units and have different ranges of possible values.



```
reg1 <- lm(qu~price+weight+luxury+fuel, my_data)
summary(reg1)
#/*what do you see in the output of regression above
#One dollar increase in price paribus, and quantity decreases by -784.22
#New-------------------------
#But how do we compare the importance of weight and fuel consmption on quantity sold?
#that have different means and ranges?
#standardize the coefficients then, lets write a function
#coefficients:
b <- reg1$coef
X<-cbind(1,my_data$price,my_data$weight, my_data$luxury,my_data$fuel)
sx1<-sd(X[,1])
sx2<-sd(X[,2])
sx3<-sd(X[,3])
sx4<-sd(X[,4])
sx5<-sd(X[,5])

sx<-cbind(sx1,sx2,sx3,sx4,sx5)
sy<-sd(my_data$qu)
beta <- b * sx/sy
#pring standardized betas:
beta

#/*from the log file we see that
#One std dev increase in weight ceteris paribus, and  quantity drops by
#0.3698 standard dev
#One std dev increase in fuel consumption ceteris paribus,
#and quantity increases by 0.01056 std dev
#This is how we compare the importance of these two factors
#that have different means and ranges,
#using Z scores and interpreting the standardized betas
```

3. make a table of all regressions

```
# nice tables for Linear Model output

stargazer(reg8, reg10, reg11, reg12, digits = 2,

        keep.stat = c("n", "rsq", "adj.rsq"))
```