BS2280 - Econometrics
R Workshop 4 - Multiple Regression Analysis and Dummy Variables

In this computer lab we will have to achieve the following tasks/learning outcomes:

- import .csv data and view the data
- create the bar plots
- run multiple regression model with dummy variables

Preparing your workspace

Before you do each task, you need to prepare your workspace first.

Step 1. Create a folder called RWorkshop4

Step 2. Go to Blackboard week 9 R Workshop 4 and download datafile: wages.csv, save it in the RWorkshop4 folder you created in step 1

Step 3. Open Rstudio and set working directory

Menu bar → Click Session → Set Working Directory → Choose Directory → Select RWorkshop4 folder you created in step 1

Step 4. Create an R script

!! If you forget how to prepare your workspace, please review R Workshop 0 first.

Task 1. Open the data set in R.

In this workshop we will use real world data to analyse if there is wage discrimination based on gender differences. We will use a new data set: wages.csv. This is a data set on earnings and other characteristics for 526 individuals:

wage	...	average hourly earnings ($)
educ	...	years of education
exper	...	years of potential experience
tenure	...	years with current employer
numdep	...	number of dependents
gender	...	gender of individual
sector	...	industry of occupation
location	...	region where the individual works

Before you start working on the wages dataset, ensure that you have prepared the workspace. Look at the name of the wages data set. For this task, you will have to open .csv file. Reading in a .csv file is easy and is part of read.table in the R utils package. The utils package, which is automatically loaded in your R session on startup, can import .csv files with the read.csv() function.

We import the data set with the read.csv() and store this dataset as an object with the name 'wages'.

```
wages <- XXXX("wages.csv")
```

Task 2. Describe the distribution of wages by plotting a histogram.

A histogram is a useful way to illustrate the distribution of variables graphically i.e. to identify if data is normally distributed or skewed. The histogram command will generate a distribution plot. We create a histogram for variable wage in data set wages.

```
hist(wages$wage, main = "Histogram of average hourly wages ($)", xlab = "Wages")
```

hist(): this function creates a histogram plot

wages$wage: pick variable wage from dataset wages and make use of it to create a histogram

main = : give this histogram a title "Histogram of average hourly wages ($)"

xlab = : name the x axis as "Wages" for this histogram

The histogram reveals that the variable wage is right skewed.

Task 3. Produce a bar chart that compares the average wages for men and women. Compare this chart with the bar chart that compared average level of education for men and women.

Bar charts are a very good tool to compare the descriptive statistics, like means, of different groups / categories within a data set.

```
XXXX(mean(wages$wage), main = "Bar plot of average hourly wages ($)",
     ylab = "Wages", XXXX = c(0,7))
```

barplot(): this function creates a bar chart

mean(wages$wage): pick variable wage from dataset wages and calculate the mean for variable wage

main = : give this bar plot a title "Bar plot of average hourly wages ($)"

ylab = : name the y axis as "Wages" for this bar plot

ylim = : giving the y coordinates ranges, here is from 0 to 7.

The bar chart showing the mean wage for the overall sample is not very useful, as the same information can be illustrated by just presenting the numerical mean value. However, if we want to compare the mean wage of men and women, then using a bar chart is an excellent way to present any differences.

Before we can produce the mean values for men and women, we have to save them first in a vector. The command tapply() allows us to generate statistics for variables of different groups, e.g. gender.

```
av.wage.gender <- XXXX(wages$wage, INDEX = wages$XXXX, FUN = XXXX)
```

av.wage.gender: We save the averages as av.wage.gender

tapply(wages$wage, INDEX = wages$gender, FUN = mean): split wage for different gender groups and calculate the mean wage for different gender groups

FUN = mean: a function to be applied, here we use function mean to calculate mean wage

```
barplot(XXXX,  main = "Average wage difference between men and women",
        ylab = "Wages", ylim = c(0,7))
```

barplot(): this function creates a bar chart

av.wage.gender: average wage for different gender groups

main = : give this bar plot a title "Average wage difference between men and women"

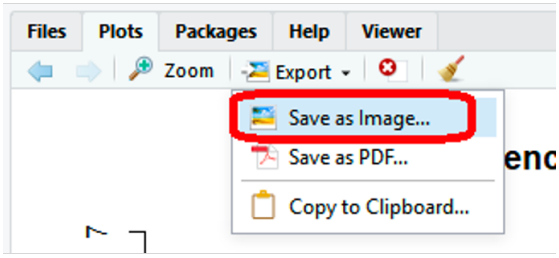
ylab = : name the y axis as "Wages" for this bar plot

ylim = : giving the y coordinates ranges, here is from 0 to 7.

There is a big difference between averages wages of men and women. We can use regression analysis to get further information on why there is such a big wage gap.

Attempt now to create a bar chart to illustrate differences in educational attainments between men and women. You will see that there are hardly any differences. Therefore differences in levels of education are very unlikely to cause gender differences in wages.

Save your graph in .png format from the plot window in RStudio.



Task 4. Consider the following model:

$$wage_i = \beta_1 + \beta_2 educ_i + \beta_3 exper_i + \beta_4 tenure_i + u_i$$

a. Run the above Model 1.

```
model1 <- lm(wage~educ+exper+tenure,data=wages)
summary(model1)
```

model1: name this regression model as model1

lm(): lm() activates the regression function for linear model

wage: the dependent variable

educ+exper+tenure: the independent variable

data=wages: specify variable wage, educ, exper and tenure are from dataset wages

b. Add the **gender** dummy variable to the above model and rerun the model. Interpret the regression coefficients and compare your regression results with the results from Model 1.

Gender is a string (text) variable, which can take in our dataset two values: male or female. R can easily deal with character strings: It coerces it into a factor variable, that takes the value 0 or 1 tor each respective category. Our regression output shows that R named the dummy variable 'gendermale', illustrating that R assumes the dummy is one if the observation is male.

```
model2 <- lm(wage~educ+exper+tenure+XXXX,data=wages)
summary(model2)
```

model2: name this regression model as model2

lm(): lm() activates the regression function for linear model

wage: the dependent variable

educ+exper+tenure+gender: the independent variable (gender is a dummy)

data=wages: specify variable wage, educ, exper, tenure and gender are from dataset wages

The coefficient of gender dummy is 1.81, which is positive and highly statistically significant (p_value $<$ 1%).

A man with the same years of education, the same work experience, the same number of years of work tenure will earn USD 1.8 more an hour than women.

Task 5. Show the frequency for how many people work in a specific sector and location. (Note that sector and location are categorical variables with more than two outcomes.)

Before we add dummy variables for sector and location to our models, it is a good idea to look at the frequency of each location and sector before. We use the table() command.

```
table(wages$XXXX)
table(wages$XXXX)
```

table(wages$sector): pick variable sector from data set wages and show frequency

table(wages$location): pick variable location from data set wages and show frequency

Task 6. Construct bar charts showing differences in wage in different sectors and the differences in wages in different locations.

See Task 3 of this Workshop, use tapply() and barplot()

```
av.wage.sector <- tapply(wages$wage, INDEX = wages$XXXX, FUN = mean)
barplot(av.wage.sector,  main = "Average wage difference in different sector",
        ylab = "Wages", ylim = c(0,7))
```

```
av.wage.location <- tapply(wages$wage, INDEX = wages$XXXX, FUN = mean)
barplot(av.wage.location,  main = "Average wage difference in different location",
        ylab = "Wages", ylim = c(0,7))
```

Task 7. Use the regression Model 1 from task 4 and add location and sector dummy variables and re-estimate the model. Beware of the Dummy Variable trap!

Before we add our dummies we have created in the previous exercises, we have to remember the dummy variable trap.

We cannot compute estimates if we add, e.g., dummy variables for each location due to perfect multicollinearity.

R will therefore drop one location and sector dummy from our regression. The excluded category will be used as base category/comparison group for all remaining locations and sectors.

Location: eastern, northern, south, west

Sector: construction, manufacturing, other, services, wholesale

Which dummy categories did R drop? Let's run the regression again including 'sector' and 'location'.

```
model3 <- lm(wage~educ+exper+tenure+location+sector,data=wages)
summary(model3)
```

For variable location, eastern is dropped so eastern is base category/comparison group for all remaining locations.

For variable sector, construction is dropped so eastern is base category/comparison group for all remaining sectors.

R drops the first alphabetically sorted category by default.

Task 8. Change the base category/comparison group for dummy variables.

It often makes sense for interpretation to select a suitable category as a base category/comparison group.

If we have no a priori judgement, we usually use the most frequent category as the base category. From task 5 we know:

```
construction  manufacturing  other  services  wholesale
24            60             102    189       151

> table(wages$location)

eastern  northcen  south  west
118      132       187    89
```

The most frequent category of location is "south" and The most frequent category of sector is "service". So we prefer to use "south" and "service" as base category for location and sector dummies, respectively.

We follow four steps.

Step 1: Convert a character variable into a factor variable

As we know, R cannot run a regression with character string (words). You have to transform the character variable into factor variable[1] . After convert to a factor variable, we can change the ordering of categories of a factor variable, so that the first category will be the category that will be base category.

as.factor(): convert character variable to factor variable

```
class(wages$sector)                  #check the type of variable sector in data set wages
class(wages$location)                #check the type of variable location in data set wages

wages$sectorfactor <- XXXX(wages$sector)
#convert character variable sector to a factor variable sectorfactor

wages$locationfactor <- XXXX(wages$location)
#convert character variable location to a factor variable locationfactor
```

View data set wages, especially the last two columns.

Step 2: Reveals the different categories of a factor variable

levels(): show the categories of a factor variable

```
XXXX(wages$sectorfactor)         #show the categories of factor variable sectorfactor
XXXX(wages$locationfactor)       #show the categories of factor variable locationfactor
```

Step 3: Change the base category

relevel(): change the categories of a factor variable

```
wages$sectorneworder <- XXXX(wages$sectorfactor, ref="services")
#Change the base category as service and give it to a new variable sectorneworder

wages$locationneworder <- XXXX(wages$locationfactor, ref="south")
#Change the base category as south and give it to a new variable locationneworder
```

View data set wages, especially the last two columns.

Step 4: Check the base category

```
levels(wages$sectorneworder)          #show the categories of new factor variable sectorneworder
levels(wages$locationneworder)        #show the categories of new factor variable locationneworder
```

Task 9. Run the regression Model 4 with changed base category dummy variables location and sector

[1]For example, a calculator cannot calculate Fred × 10, but it can solve 1 × 10. R therefore changes the characters into a factor variable, that allocates a numerical value to each character string of the variable, e.g. 'construction' is equal to 0, 'manufacturing' is equal to 1, etc. This process is referred to as 'coercion'.

```
model4 <- lm(wage~educ+exper+tenure+XXXX+XXXX,data=wages)
summary(model4)
```

Task 10. Use stargazer() to get the regression output

stargazer() is an R package that creates LATEX code, HTML code and ASCII text for well-formatted regression tables, with multiple models side-by-side, as well as for summary statistics tables, data frames, vectors and matrices.

A quick reproducible example shows just how easy stargazer is to use. You can install stargazer from CRAN in the usual way:

```
install.packages("stargazer")
library(stargazer)
```

You can also use starfazer() to summarise the output for single model and multiple models.

```
stargazer(model1, type = "text")
```

```
stargazer(model1, model2, model3, type = "text")
```

More applications of stargazer() can be accessed on Blackboard under Week 9 in the R Workshop 4 section.

Further Exercise

For this section, we return to the crime dataset (crime.Rdata). Complete the following tasks:

1. Open the crime.Rdata dataset.
2. Use the variable pcinc to construct a new variable with the name pcincd which is '1' for cities that have per capita income higher than 8000, and '0' otherwise. Advice: You can use a new function ifelse to speed up the process of creating a dummy. Access R's help function by typing ?ifelse into the Terminal window to get more information.
3. Produce a bar chart showing the average number of police officers in each of the two categories of cities you have created in step 1.
4. Estimate the following model and interpret its coefficients and the coefficients' statistical significance.

$$officers_i = \beta_1 + \beta_2 crimes_i + \delta pcincd_i + u_i$$

5. Use the variable pcinc to construct a new variable with the name pcinc3 which is 'Very poor' for cities that have per capita income of [0, 6442), 'Poor' for per capita income of [6442, 7182), 'Rich' if per capita income is [7182, 8010), and 'Very rich' if per capita income is [8010, 10204]. Advice: Use the new function cut to speed up the process of creating this categorical variable. Access R's help function by typing ?cut into the Terminal window to get more information.
6. Produce a bar chart showing the average number of police officers in each of the four categories of cities you have created in step 5.

7. Estimate the following model and interpret its coefficients and the coefficients' statistical significance.

$$officers_i = \beta_1 + \beta_2 crimes_i + \delta_1 pcincd3_i^{Poor} + \delta_2 pcincd3_i^{Rich} + \delta_3 pcincd3_i^{VeryRich} + u_i$$