Assignment_01

Mohammed Al Hor

2022-10-08

Questions —-

Q1 - introduction and descriptives —-

a) Load the tidyverse-package.

```
library(tidyverse)
```

b) Read the data from 'oecd_data.csv'.

```
# Set our working directory
setwd("~/Data-Science-Business-Analytics/Data")
# Load our csv file with headers
oecd_data <- read.csv("oecd_data.csv", header = TRUE)
```

c) Get a first view on the data by getting the dimensions, show the first 5 rows of the data.frame and giving the summary.

To get the dimensions of a dataset we use the dim() function:

```
dim(oecd_data)

## [1] 15168     7

# Dimensions are 15168 rows, 7 columns
```

To show the first 5 rows we use the head() function with n = 5

```
head(oecd_data,n=5)

##   reg_id          region year country_code pc_real_ppp     per  real_ppp
## 1  ITG27        Cagliari 2000           IT       28821  203200  15650.50
## 2  KR031           Daegu 2000           KR       13764      NA  34806.70
## 3  ITG13         Messina 2000           IT       24273  207700  16050.20
## 4   US09     Connecticut 2000           US       61231 2118200 208907.00
## 5  UKF12 East Derbyshire 2000           UK       19919   95000   5318.88
```

To get summary of the data we use the summary() function

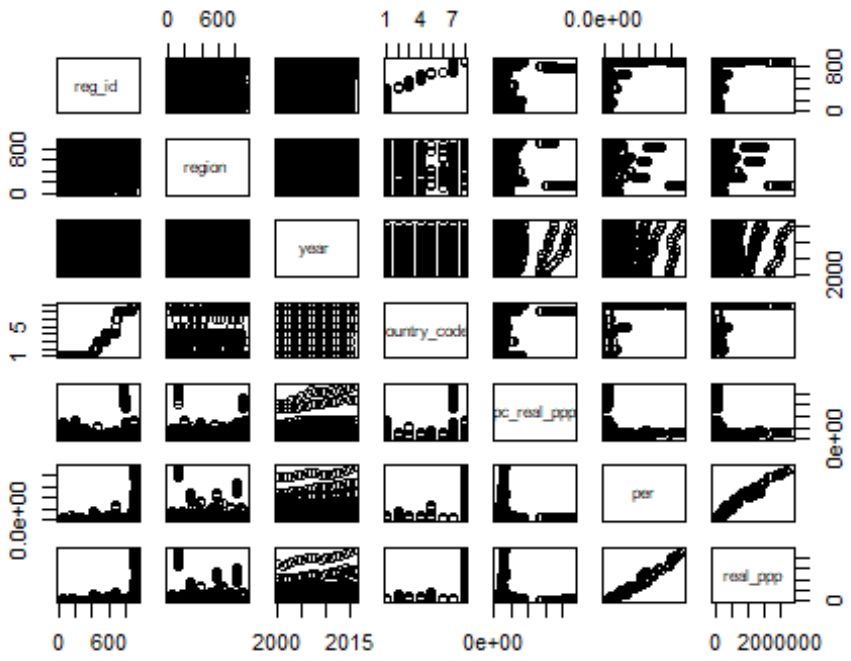
```
summary(oecd_data)

##     reg_id             region               year      country_code
##  Length:15168       Length:15168       Min.   :2000   Length:15168
##  Class :character   Class :character   1st Qu.:2004   Class :character
##  Mode  :character   Mode  :character   Median :2008   Mode  :character
##                                        Mean   :2008
```

```
##                                          3rd Qu.:2012
##                                          Max.   :2016
##
##   pc_real_ppp          per              real_ppp
##  Min.   : 11364   Min.   :    2600   Min.   :    175.7
##  1st Qu.: 26270   1st Qu.:   63388   1st Qu.:   4511.1
##  Median : 31530   Median :  117000   Median :   8551.9
##  Mean   : 35383   Mean   :  370223   Mean   :  30311.6
##  3rd Qu.: 38684   3rd Qu.:  221000   3rd Qu.:  17171.2
##  Max.   :462774   Max.   :23265300   Max.   :2382750.0
##  NA's   :27       NA's   :140
```

d) Use the standard plot-function from R (not ggplot()) that get a first visual view on the data. To get a plot of the whole dataframe we use the plot function and input the dataframe

```
plot(oecd_data)
```



e) How many observations are there by country, by year? Show it in a table. To get the amount of observations by country and year we can use the table() function

```
table(oecd_data$country_code)

##
##   DE   ES   FR   IT   KR   SE   UK   US
## 6432  944 1616 1760  272  336 2941  867
```

```
table(oecd_data$year)
## 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015
##  933  933  933  933  933  933  933  933  933  933  933  933  933  933  933  933
## 2016
##  240
```

Q2 - dplyr-preparations + first visualizations —-

a) To get to our first visualizations, filter only the observations for UK.

We can do this with indexing and filter in the 'row index':

```
oecd_uk <- oecd_data[oecd_data$country_code=='UK',]
```

Or we can use the filter function from the dplyr package:

```
library(dplyr)
oecd_uk <- oecd_data %>%
  filter(oecd_data$country_code == 'UK')
```

b) Group the observations in the dataset from Q2a) by year and get the minimum and maximum of pc_real_ppp in the UK.

We use the group_by function to group the data by year and then we summarise using the summarise function (dplyr package)

```
uk_grouped <- oecd_uk %>%
              group_by(year) %>%
              summarise(minimum = min(pc_real_ppp, na.rm=TRUE), maximum =
max(pc_real_ppp, na.rm=TRUE))
```

c) Show in a time series plot the minimum and maximum of pc_real_ppp in the UK over time.

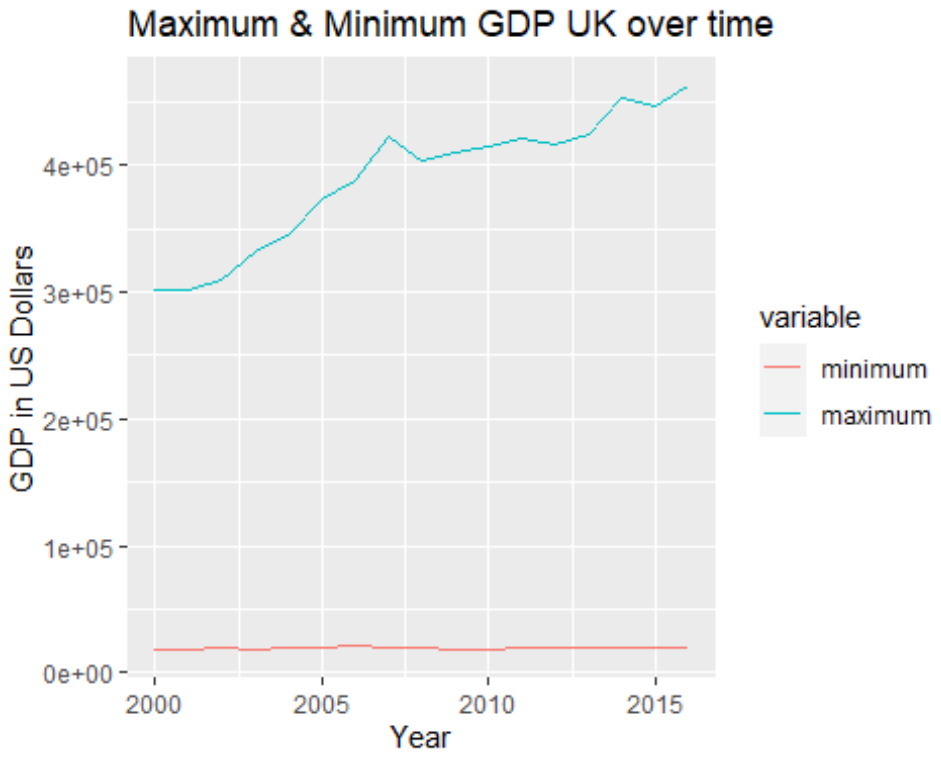
First, we reshape the data to long format, so that minimum and maximum can be used as categorical variables.

```
library(reshape2)

uk_grouped_m <- melt(uk_grouped,id.vars="year")
```

Now, we can start plotting using ggplot2. We plot the time variable (year) on the x axis and assign the y to 'value' (which is all the minimum and maximum values). Using the argument color, we're able to distinguish between minimum and maximum values. Furthermore, we add some labels and titles.

```
library(ggplot2)
ggplot(uk_grouped_m, aes(x=year, y=value, color=variable)) +
       geom_line() +
       labs(x = "Year",
       y = "GDP in US Dollars",
       title = "Maximum & Minimum GDP UK over time")
```



Q3 - data wrangling —-

a) Back to our original dataset, loaded in Q1b). Get for each country in 2015 the name of the region with the largest pc_real_ppp.

Using dplyr we use the group_by function to group the data by country and year. Using the filter and the max() function we only keep the largest values (per country and year) in the year 2015. Then, we arrange the data by pc_real_ppp desc and select only the variable in which we're interested.

```
oecd_region <- oecd_data %>%
               group_by(country_code, year) %>%
               filter(pc_real_ppp == max(pc_real_ppp), year==2015) %>%
               arrange(desc(pc_real_ppp)) %>%
               select(country_code, region, pc_real_ppp)

oecd_region
```

```
## # A tibble: 8 × 4
## # Groups:   country_code, year [8]
##    year country_code region                       pc_real_ppp
##   <int> <chr>        <chr>                               <int>
## 1  2015 UK           Camden & City of London          446495
## 2  2015 US           District of Columbia             166797
## 3  2015 DE           Ingolstadt Kreisfreie Stadt      151068
## 4  2015 FR           Hauts-de-Seine                   109111
## 5  2015 KR           Ulsan                             66795
## 6  2015 SE           Stockholm County                  62527
```

```
## 7  2015 IT           Milan                                61968
## 8  2015 ES           Araba/Álava                          48792
```

b) (Again use the dataset loaded in Q1b) We need to scale the data such that countries are comparable. Mutate pc_real_ppp such that it is relative to the countries average by year. You thus need to find the average over the observations of pc_real_ppp grouped by country_code and by year.

First, we group by country and year. Then, we use the mutate function to calculate an average over this group (removing NA's is necessary to make sure we always get a value), then we calculate relative pc_real_ppp by dividing by the average calculated previosuly. Lastly, we select the attributes in which we're interested.

```
oecd_scaled <- oecd_data %>%
               group_by(country_code, year) %>%
               mutate(average_by_country_year = mean(pc_real_ppp, na.rm = TRUE),
                      pc_real_ppp = pc_real_ppp/average_by_country_year) %>%
               select(country_code, year, region, pc_real_ppp)
oecd_scaled
```

```
## # A tibble: 15,168 × 4
## # Groups:   country_code, year [131]
##    country_code  year region                    pc_real_ppp
##    <chr>        <int> <chr>                           <dbl>
##  1 IT            2000 Cagliari                        0.880
##  2 KR            2000 Daegu                           0.660
##  3 IT            2000 Messina                         0.741
##  4 US            2000 Connecticut                     1.39
##  5 UK            2000 East Derbyshire                 0.608
##  6 UK            2000 Gwent Valleys                   0.576
##  7 UK            2000 Nottingham                      1.16
##  8 UK            2000 Cheshire West and Chester       0.996
##  9 FR            2000 Gard                            0.859
## 10 UK            2000 Suffolk                         0.920
## # … with 15,158 more rows
```

c) Repeat Q2b) over the dataset created in Q3b), but now having the minimum and maximum for each year, for each country. First, we take the previous dataset (oecd_scaled) and group by year and country. Using the summary function we get the minimum and maximum, removing NA's (NA's will cause the value to be NA if 1 observation is NA). Lastly, we arrange the data by country and year.

```
oecd_scaled_grouped <- oecd_scaled %>%
                       group_by(year, country_code) %>%
                       summarise(minimum = min(pc_real_ppp, na.rm=TRUE),
maximum = max(pc_real_ppp, na.rm=TRUE)) %>%
                       arrange(country_code, year)

oecd_scaled_grouped
```

```
## # A tibble: 131 × 4
## # Groups:   year [17]
##     year country_code minimum maximum
##    <int> <chr>          <dbl>   <dbl>
##  1  2000 DE             0.463    3.33
##  2  2001 DE             0.459    3.57
##  3  2002 DE             0.461    3.39
##  4  2003 DE             0.457    3.33
##  5  2004 DE             0.449    3.29
##  6  2005 DE             0.457    3.35
##  7  2006 DE             0.453    3.20
##  8  2007 DE             0.454    3.28
##  9  2008 DE             0.453    3.19
## 10  2009 DE             0.455    3.22
## # … with 121 more rows
```

c) Read the data from 'oecd_names.csv'. We load the data using the read.csv function. Data has headers.

```
setwd("~/Data-Science-Business-Analytics/Data")
oecd_names <- read.csv("oecd_names.csv", header = TRUE)
```

d) Join the oecd_names and the data.frame from Q3c) making sure all observations of the latter data.frame are kept. We join the previous dataframe (oecd_scaled_grouped) with the oecd_names dataframe. To ensure we keep all observations from the first dataframe we use a left join with the join condition: country_code = oecd.imp.code

```
oecd_join <- oecd_scaled_grouped %>%
             left_join(oecd_names, by = c('country_code' = 'oecd.imp.code')) %>%
             select(country, year, minimum, maximum)
oecd_join
```

```
## # A tibble: 131 × 4
## # Groups:   year [17]
##    country  year minimum maximum
##    <chr>   <int>   <dbl>   <dbl>
##  1 Germany  2000   0.463    3.33
##  2 Germany  2001   0.459    3.57
##  3 Germany  2002   0.461    3.39
##  4 Germany  2003   0.457    3.33
##  5 Germany  2004   0.449    3.29
##  6 Germany  2005   0.457    3.35
##  7 Germany  2006   0.453    3.20
##  8 Germany  2007   0.454    3.28
##  9 Germany  2008   0.453    3.19
## 10 Germany  2009   0.455    3.22
## # … with 121 more rows
```

e) Repeat Q2c) and show the minimum and maximum by country (Use 'country' instead of 'country_code'). Give each country its own color (not the default colors). Let minimum and maximum have different line types. Update the

visualization such to make it look nicer using the tools at hand (given to you in the lectures)

We use the dataframe oecd_join with relative pc_real_ppp by country and year. First we reshape the data to long format, so that minimum, maximum and country can be used as categorical variables in upcoming plots

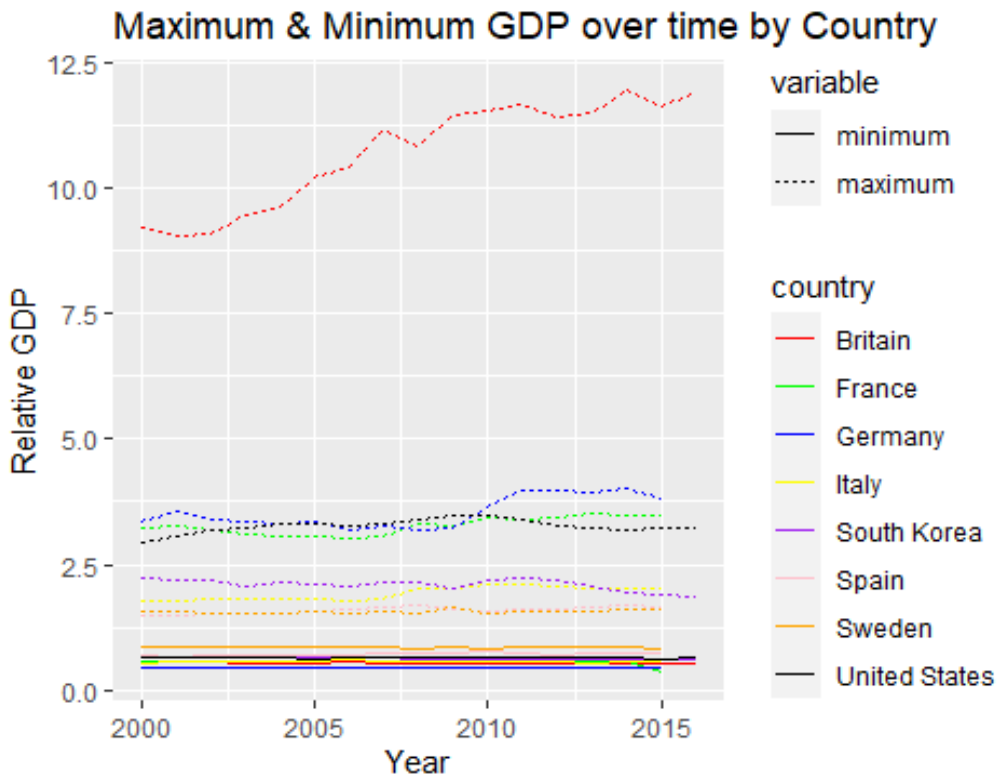
```
oecd_3e_m <- melt(oecd_join, id.vars = c('country','year'))
```

Secondly, we manually make up some colors and put these in the vector 'colors'. These are to be used in the plot itself.

```
colors <- c('red', 'green', 'blue', 'yellow', 'purple', 'pink', 'orange',
'black')
```

We plot the time variable (year) on the x axis, we assign y to 'value' (previously minimum or maximum pc_real_ppp). Now its time to do some finetuning, adding arguments for color (which we want to be determined by country) and linetype (which must be determined by minimum or maximum value of pc_real_ppp). Next, we add our manually selected list of colors to be used in the plot. Last, we add some labels and titles to make the plot more easily understood.

```
ggplot(oecd_3e_m, aes(x=year, y=value, color = country, linetype = variable))
+ geom_line() +
  scale_color_manual(values = colors) +
  labs(x = "Year",
       y = "Relative GDP",
       title = "Maximum & Minimum GDP over time by Country")
```



Q4 - wrap-up —-

Reproduce the plot from Q3e) with 1. real_ppp per worker (which can be created using real_ppp and per from the data.frame 2. 5% en 95% quantiles instead of minimum and maximum Still remember to do the scaling as in Q3b) Can you do this without any intermediate results? (using Pipes %>%) Based on the first graph, people reacted: dispersion between regions grows! What can you conclude (differently)?

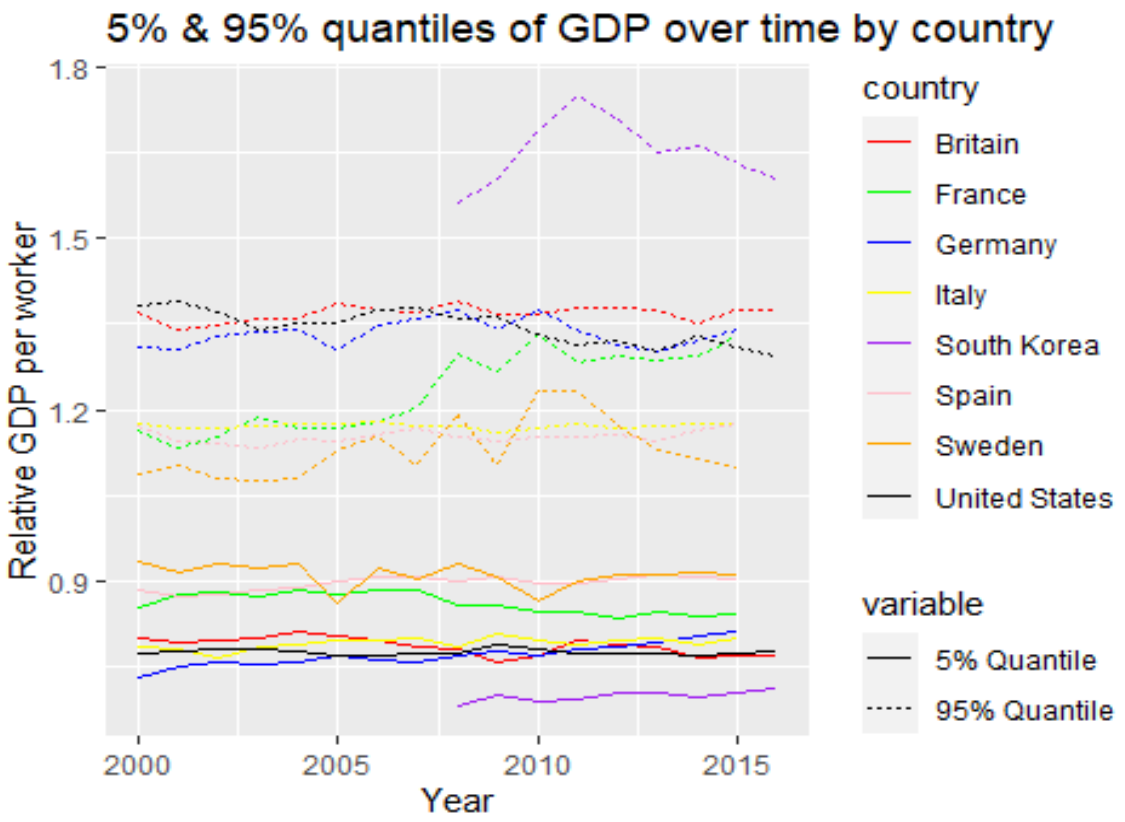
The variable 'per' has some missing values (KR en FR) we need to filter these out in our dplyr statement before calculations. 'Real_ppp' has no missing values.

```
summary(oecd_data$per)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##     2600    63388   117000   370223   221000 23265300      140
```

Step by step what we do in the following dplyr statement. First we take the oecd_data dataframe and filter out the rows with missing values for 'per'. Thereafter, we join oecd_names dataframe to get the full country names. Then, we group by country and year to calculate real_ppp per worker (real_ppp_per), the average ppp per worker per country by year. We use this to calculate the relative GDP per worker. We use summarise to calculate 5% and 95% quantiles of this variable. Now we have to get our data ready to plot. We reshape the data to long format, using the melt function. Last, we plot our new variable over time (year) by country and for the different quantiles. We do this in 1 dplyr statement.

```
  oecd_data %>%
  filter(!is.na(per)) %>%
  left_join(oecd_names, by = c('country_code' = 'oecd.imp.code')) %>%
  group_by(country, year) %>%
  mutate(real_ppp_per = real_ppp/per,
         average_by_country_year = mean(real_ppp_per, na.rm = TRUE),
         real_ppp_per = real_ppp_per/average_by_country_year) %>%
  summarise('5% Quantile' = quantile(real_ppp_per, 0.05,na.rm=TRUE),'95% Quantile' =
quantile(real_ppp_per, 0.95, na.rm=TRUE)) %>%
  melt(id.vars = c('country','year')) %>%
  ggplot(aes(x=year, y=value, color = country, linetype = variable)) +
    geom_line() +
    scale_color_manual(values = colors) +
    labs(x = "Year",
         y = "Relative GDP per worker",
         title = "5% & 95% quantiles of GDP over time by country")
```



Compared to the first plot, we can conclude the dispersion between regions has not been growing over the years. In some cases(South Korea) we can even see a decline in the dispersion. We had to drop missing values for South Korea, that's why the data for this country starts from 2007.