Assignment\_01

Mohammed Al Hor

2022-10-08

### Questions —-

#### Q1 - introduction and descriptives —-

1. Load the tidyverse-package.

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.1   
## ✔ readr 2.1.2 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

1. 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)

1. 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

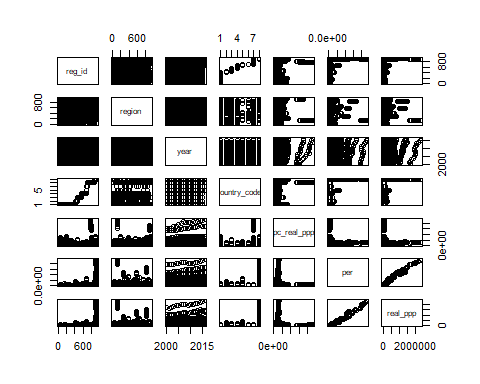
To get summary of the data we use the summary() function

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

1. 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)



1. 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 —-

1. 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')

1. 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))

1. 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)

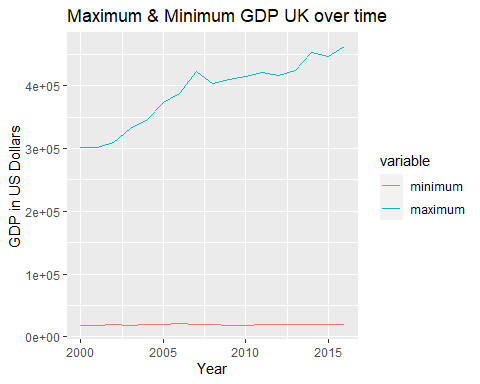
##   
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':  
##   
## smiths

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 —-

1. 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)

## Adding missing grouping variables: `year`

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

1. (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

1. 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)

## `summarise()` has grouped output by 'year'. You can override using the  
## `.groups` argument.

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

1. 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)

1. 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

1. 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

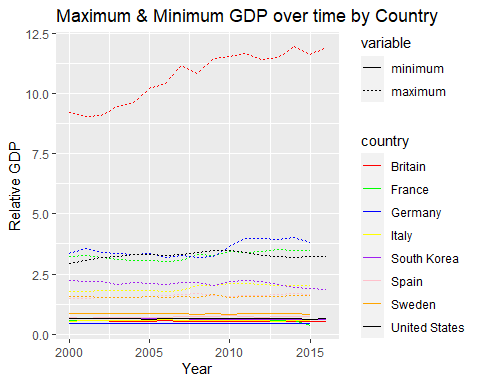
oecd\_3e\_m <- melt(oecd\_join, id.vars = c('country','year'))

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)

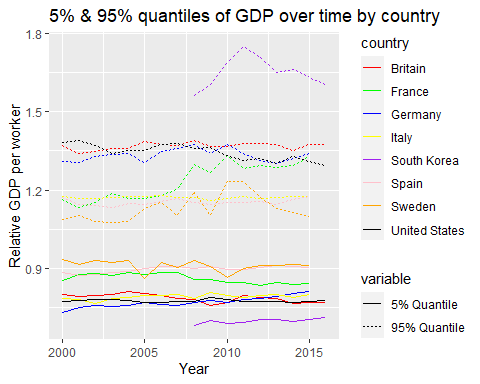
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 2600 63388 117000 370223 221000 23265300 140

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")

## `summarise()` has grouped output by 'country'. You can override using the  
## `.groups` argument.

## Warning: attributes are not identical across measure variables; they will be  
## dropped



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.