ex1

Liav Alter

Thanks

Pisa Test

1.1

First we will want to load our Pisa test excel files. To do that, we'll define our work directory to be the folder that contains the 'pisa.xlsx' file, with the command wordkir. As the 'pisa.xlsx' contains 3 sheets, we will load each of them to a different reference.

```
workdir = "C:/Users/Liav/Desktop/Uni/R/targil1"
setwd(workdir)

pisa_math = read_excel("pisa.xlsx", sheet=1)

pisa_reading = read_excel("pisa.xlsx", sheet=2)
pisa_science = read_excel("pisa.xlsx", sheet=3)
```

To get a clue about how the data looks like, we can use the head command to pring the first rows of it. Let's make an example with pisa_math.

```
head(pisa_math)

## # A tibble: 6 x 3
##   Country                        Score  year
##   <chr>                          <chr> <dbl>
## 1 International Average (OECD) 490    2015
## 2  Albania                       413    2015
## 3  Algeria                       360    2015
## 4  Argentina                     409    2015
## 5  Australia                     494    2015
## 6  Austria                       497    2015
```

Since all the worksheets share the same columns names, to differ them, we can change some of the columns name.

```
colnames(pisa_math)[2] = "math_score"
colnames(pisa_reading)[2] = "reading_score"
colnames(pisa_science)[2] = "science_score"
```

1.2

We would like to check who were the leading countries in every field in 2015. To do so, we will sort by descending order the 2015 year subset. This command might seem very complicated, let's try to explain it simply:

1. head command (x, n= 3L) is the same head command we used before, but now we used "n=3L" to get the first 3 lines.

2. pisa_math[y, year=2015] command gives us a susbet of the pisa_math rows that their value in column 'year' equals to 2015. notice that pisa_math is the name of the dataframe, we can change it later to another name for the other dataframes.

3. order(pisa_math$math_score, decreasing = TRUE) returns the data decreasing order if the data frame be the pisa_math column.

To connect those three simple functions we can write the 3rd function instead of y, and the joined 2nd function (with 3 insted of y) instead of x, and now we got the 3 first countries orederd by scores, only when year was 2015.

```
head(subset(pisa_math[order(pisa_math$math_score, decreasing = TRUE),
], year == 2015), n=3L)

## # A tibble: 3 x 3
##   Country             math_score  year
##   <chr>               <chr>      <dbl>
## 1  Singapore          564         2015
## 2  Hong Kong, China  548         2015
## 3  Macau              544         2015

head(subset(pisa_science[order(pisa_science$science_score, decreasing =
TRUE), ], year == 2015), n=3L)

## # A tibble: 3 x 3
##   Country    science_score  year
##   <chr>      <chr>         <dbl>
## 1  Singapore 556            2015
## 2  Japan     538            2015
## 3  Estonia   534            2015

head(subset(pisa_reading[order(pisa_reading$reading_score, decreasing =
TRUE), ], year == 2015), n=3L)

## # A tibble: 3 x 3
##   Country             reading_score  year
##   <chr>               <chr>         <dbl>
## 1  Singapore          535            2015
## 2  Canada             527            2015
## 3  Hong Kong, China  527            2015
```

We can see that Singapore students has the best Pisa test grades in 2015, in all fields.

1.3

To continue working on this data set, we would like to merge the all three dataframes to one. For this, we can use the merge command. At first we merge the science and math grades columns. The command "all=TRUE" used to keep the NA value rows. The "sort=TRUE" command used to keep the new dataframe sorted by the country name.

After we merged science and math grade we can merge them again to the reading grade dataframe by the same method.

```
data = merge( merge( pisa_math, pisa_science, by = c("Country","year")
,all = TRUE, sort = TRUE), pisa_reading, by = c("Country","year"), all
= TRUE, sort=TRUE )
```

Here is ther head of our new merged dataframe:

```
##         Country year math_score science_score reading_score
## 1   Switzerland 2000       <NA>          <NA>           494
## 2   Switzerland 2003        527          <NA>           499
## 3   Switzerland 2006        530           512           499
## 4   Switzerland 2009        534           517           501
## 5   Switzerland 2012        531           515           509
## 6   Switzerland 2015        521           506           492
## 7       Albania 2000       <NA>          <NA>           349
```

Before we continue to answer the next questions, we would like to make sure that the columns type is still numeric, so we can apply arithmetic functions on them. For that, we'll use the following:

```
sapply(data, class)

##       Country          year    math_score science_score 
reading_score 
##   "character"     "numeric"   "character"   "character" 
"character"
```

We can see that the type of the values in the score columns are character. There for, we'll use sapply to apply the function as numeric on the 3rd to 5th columns.

```
data[, 3:5] <- sapply(data[, 3:5], as.numeric)
sapply(data, class)

##       Country          year    math_score science_score 
reading_score 
##   "character"     "numeric"     "numeric"     "numeric" 
"numeric"
```

And now we can see the type changed to numeric.

1.4

We'll check the avarage pisa scores (average of all three fields scores) for each country on every year. To make it a fair competition, we will calculate only the rows that includes all three grades.

```
data$average_score <- rowMeans(data[,3:5], na.rm=TRUE)
```

Here is ther head of our new dataframe, with the average column:

```
##         Country year math_score science_score reading_score
average_score
## 1   Switzerland 2000         NA            NA           494
494.0000
## 2   Switzerland 2003        527            NA           499
513.0000
## 3   Switzerland 2006        530           512           499
513.6667
```

1.5

As beofre, let's use the subset and order functions to present the top average scores countries in 2006 and 2015. This time, we will use "[,c(1,6)]" slicing to show only the 1st, 2nd and 6th column.

```
head(subset(data[order(data$average_score, decreasing = TRUE), ], year
== 2015)[,c(1,2,6)], n=3L)
```

```
##               Country year average_score
## 348         Singapore 2015      551.6667
## 162  Hong Kong, China 2015      532.6667
## 204             Japan 2015      528.6667
```

```
head(subset(data[order(data$average_score, decreasing = TRUE), ], year
== 2006)[,c(1,2,6)], n=3L)
```

```
##               Country year average_score
## 129           Finland 2006      552.6667
## 159  Hong Kong, China 2006      541.6667
## 363       South Korea 2006      541.6667
```

Salaries

Now, we'll start working on the salaries file.

2.1 From a brief look on the data, we can see that the 2nd sheet on the sal.xlsx file is just an addition of a nonimnal salary column. Therefor, to save time and code, it will be a good idea to join the two dataframes by adding the "current" column from the 2nd dataframe to the first one. We' will use the same read_excel command from before to do so, and than we'll just add manually the missing column.

```
salaries = read_excel("sal.xlsx", sheet=1)
sal_nominal = read_excel("sal.xlsx", sheet=2)
salaries$current = sal_nominal$current
```

Here are some rows of our new dataframe:

```
## # A tibble: 3 x 16
##   TIME  `2000` `2005` `2006` `2007` `2008` `2009` `2010` `2011`
`2012`
##   <chr>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>
## 1  Por~     NA    100     98     96     95     97     98    100
86
## 2  Spa~     NA    100    101     99    103    106    106    100
95
## 3  Swe~     NA    100     NA    103     NA    104     NA    103
NA
## # ... with 6 more variables: `2013` <dbl>, `2014` <dbl>, `2015`
<dbl>,
## #   `2016` <dbl>, `2017` <dbl>, current <dbl>
```

Data complection and cleaning

Before we will start analyzing our data, we will want to have a look of it and see if there are missing values or any outliners that we need to take care of. After a quick look on the salaries dataframe, we can see that some of the salaries are missing. Take a look for example, on the 2006 and 2008 salary in Sweden. In order to fill the missing data, we can user linear regression and try to predict what were the salaries in the missing fields.

To fill out dataframe, I wrote myself the following function:

```
salary_predict <- function(row, i) {
df <- data.frame(x=1:13,y=100*(1:13))
df$y = as.vector(unlist(salaries[i,3:15]))
df$x= as.numeric(colnames(salaries[3:15]))
model <- lm(y~x+1,data=df)
df$y_hat <- predict.lm(model, newdata = df)
df= df %>% mutate(y = ifelse(is.na(y), y_hat, y))
return(df$y)
}
```

The function parameters are a row, and a row index. Since i wanted to predict more efficiently the missing values from years 2005-2017, and since later on we will not need it, i decided to ignore the “2000” column for now. The function creates a new dataframe with index values for cells. Afterwards, save inside them the Y vector which we want to predict (our explained variable), which in our case is the row’s 3rd to 15th values, which we had to unlist them and save as vector because of the predict function input demands. We did the same for our explaining variable, which is the series of years 2005-2017. model function creates a model, and y_hat is the vector of model predictions. Since we already have some data, we will take only the missing index and mutate them into our Y vector.

After we built the function, we will have to apply them on each row. For this, we can use a for loop that go through each row and replacing the values 3:15 in the function output.

```
for(i in 1:nrow(salaries)) {
    row <- salaries[i,]
    salaries[i,3:15] =  salary_predict(row,i)
}
```

And here is an example our filled data:

```
## # A tibble: 3 x 15
##   TIME  `2005` `2006` `2007` `2008` `2009` `2010` `2011` `2012`
`2013`
##   <chr>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>
## 1  Por~    100   98       96    95       97    98      100    86
84
## 2  Spa~    100  101       99   103      106   106      100    95
92
## 3  Swe~    100   99.3    103   103.     104   106.     103   110.
108
## # ... with 5 more variables: `2014` <dbl>, `2015` <dbl>, `2016`
<dbl>,
## #   `2017` <dbl>, current <dbl>
```

Note: Since our next question will be about years 2005-2017, i decided to drop 2000 salaries for now.

2.2

Now, we can find the nominal wages in each country in any year, by using the “current” column. Since the wage from 2006 to 2016. The forumla is going to be (Year.i*year.0)/nominal, while year 0 is 2017. To do this, we will use for a for loop again, this time on columns 2:13 (since the 13th column is 2017 wage)

```
for(i in 2:ncol(salaries)) {
  salaries[i] = ((salaries[i]/salaries$`2017`)*salaries$current)
}
salaries[16:18,]

## # A tibble: 3 x 15
##   TIME  `2005` `2006` `2007` `2008` `2009` `2010` `2011` `2012`
`2013`
##   <chr>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>
## 1  Por~ 43633. 42760. 41887. 41451. 42324. 42760. 43633. 37524.
36651.
```

```
## 2  Spa~ 52827. 53355. 52299. 54412. 55997. 55997. 52827. 50186.
48601.
## 3  Swe~ 35487. 35230. 36551. 36478. 36906. 37727. 36551. 38975.
38326.
## # ... with 5 more variables: `2014` <dbl>, `2015` <dbl>, `2016`
<dbl>,
## #   `2017` <dbl>, current <dbl>
```

Since Estonoia current wage in nominal terms is missing, , we got NA over the values. Since this row can help us at the moment, we can drop the Estonia salary row. We can also drop the nominal current salary, since it equals to the 2017 salary

```
## # A tibble: 1 x 14
##   TIME  `2005` `2006` `2007` `2008` `2009` `2010` `2011` `2012`
`2013`
##   <chr>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>
## 1  Est~     NA     NA     NA     NA     NA     NA     NA     NA
NA
## # ... with 4 more variables: `2014` <dbl>, `2015` <dbl>, `2016`
<dbl>,
## #   `2017` <dbl>
```

Data analyzing

2.3

To check the relative of a country from the rest of the world, we can add an average international rating row for each year. We'll it (OECD (Average International). To do so, we will create a vector with the row means, and add the title of it manually. Afterwords, we can add it manually to salaries dataframe.

```
row_name = "International Average (OECD)"
avg_vec = c(row_name, colMeans(salaries[2:ncol(salaries)]))
```

Now, we can join them together with the rbind command. We will use head(salaries) command to check out the top of our data as we did before.

```
salaries <- rbind(avg_vec, salaries)
colnames(salaries)[1] <- "Country"
head(salaries)

## # A tibble: 6 x 14
##   Country `2005` `2006` `2007` `2008` `2009` `2010` `2011` `2012`
`2013`
##   <chr>   <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>
<chr>
## 1 Intern~ 42590~ 42951~ 43062~ 43569~ 44379~ 44562~ 44183~ 43784~
43763~
## 2  Austr~ 51322~ 50295~ 51322~ 53374~ 52861~ 53888~ 54914~ 54914~
56967~
```

```
## 3  Austr~ 49334~ 49334~ 49827~ 49827~ 50814~ 51307~ 50814~ 50321~
49827~
## 4  Denma~ 52535~ 53060~ 53060~ 54111~ 57263~ 56212~ 55162~ 53586~
54636~
## 5  Finla~ 43893~ 45210~ 45210~ 46088~ 46088~ 46966~ 46527~ 45649~
45210~
## 6  France 39396~ 39396~ 39002~ 38214~ 38214~ 38608~ 38214~ 37820~
37426~
## # ... with 4 more variables: `2014` <chr>, `2015` <chr>, `2016`
<chr>,
## #   `2017` <chr>
```

2.4

Now we can, for example, analyse year 2010. Let's check which salary was the closest to the average in 2010, and check which one was the most far from it (from both under and above) To do so, we will go throguh the following levels: 1. Create a reference with the salaries$'2010' values 2. Reduce from it the OCED (remember that this is the first row now), and get the absolut value of to get only the distance. 3. The maximum and minimum of the original are the most far values from average, and the minumim of the absolut value reduced column is the closest to average.

Notice that on the 'which' command (that returns the index that inside the brackets) we add +1 to the index because we sliced the salaries$2010 vector

```
column = as.numeric(salaries$`2010`)
salaries[which(salaries$`2010`==max(column)),]

## # A tibble: 1 x 14
##   Country `2005` `2006` `2007` `2008` `2009` `2010` `2011` `2012`
`2013`
##   <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>
## 1  Luxem~ 94924. 94924. 94924. 1.02e5 1.08e5 1.06e5 1.03e5 1.06e5
1.07e5
## # ... with 4 more variables: `2014` <dbl>, `2015` <dbl>, `2016`
<dbl>,
## #   `2017` <dbl>

salaries[which(salaries$`2010`==min(column)),]

## # A tibble: 1 x 14
##   Country `2005` `2006` `2007` `2008` `2009` `2010` `2011` `2012`
`2013`
##   <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>
## 1  Hunga~ 18744. 18370. 17620. 17245. 15370. 14808. 14246. 13309.
12746.
## # ... with 4 more variables: `2014` <dbl>, `2015` <dbl>, `2016`
<dbl>,
## #   `2017` <dbl>
```

```
salaries[which (abs(column[2:nrow(salaries)]-column[1]) ==
min(abs(column[2:nrow(salaries)]-column[1])))+1,]

## # A tibble: 1 x 14
##   Country `2005` `2006` `2007` `2008` `2009` `2010` `2011` `2012`
`2013`
##   <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>
## 1  Portu~ 43633. 42760. 41887. 41451. 42324. 42760. 43633. 37524.
36651.
## # ... with 4 more variables: `2014` <dbl>, `2015` <dbl>, `2016`
<dbl>,
## #   `2017` <dbl>
```

We got that the highest salary was at Luxembourg, the lowest was in at Hungary, and the closest to average Portugal

2.5

Here are the highest earned countries in 2015.

```
head(subset(salaries[order(salaries$`2015`, decreasing = TRUE),
c(1,12)]), n=5L)

## # A tibble: 5 x 2
##   Country         `2015`
##   <chr>            <dbl>
## 1  Luxembourg    110112.
## 2  Germany        73978.
## 3  United States  63184.
## 4  Ireland        58953.
## 5  Australia      58507.
```

Let's check their rank in the Pisa test.

```
twenty_fifteen = data[which(data$year==2015),]
twenty_fifteen = merge(
salaries[(which(colnames(salaries)==c("Country","2015")))],
twenty_fifteen, by = c("Country") ,all = TRUE, sort = TRUE)
head(twenty_fifteen[order(twenty_fifteen$`2015`, decreasing = TRUE),],n
= 5)

##            Country      2015 year math_score science_score
reading_score
## 41      Luxembourg 110111.54 2015        486           483
481
## 25         Germany  73977.59 2015        506           509
509
## 71   United States  63184.36 2015        470           496
497
## 31         Ireland  58952.52 2015        504           503
521
```

```
## 6       Australia  58507.16 2015        494            510
503
##    average_score
## 41      483.3333
## 25      508.0000
## 71      487.6667
## 31      509.3333
## 6       502.3333
```

We can see that there is no high correlation between the salaries in 2015 to the Pisa scores.

2.6

Let’s check the change in salary between 2005 and 2017. To do so, we can copy the 2005, 2017 columns to a new dataframe and calculate the percent of change. Afterwards we can use the head and order function to show the top 5, as we did before.

```
changes = salaries[c(1,2,ncol(salaries))]
changes$change = (changes$`2017`/changes$`2005`)*100
head(changes[order(changes$change, decreasing = TRUE),],n = 5)
```

```
## # A tibble: 5 x 4
##   Country `2005` `2017` change
##   <chr>    <dbl>  <dbl>  <dbl>
## 1  Latvia 13716. 19696.   144.
## 2  Israel 23367. 33442.   143.
## 3  Poland 20433. 25553.   125.
## 4  Mexico 32478. 40595.   125.
## 5  Sweden 35487. 43827.   124.
```

We can see that Latvia has the biggest growth in the teachers salary, and right after, Israel. One reason to that might be the an increase in the price index or inflation

Salaries vs. Test grades

3.1

To check the influence of the salaries on the test grades, let us first merge our two data set and drop the NA values. To do so, we first need to gather the data. To do so we will use the command gather from “tidyr” library. gather() arguments are first, the name of the dataframe, second, the name you want for you new category column (that’s called the key), and third is the name you want for your new value column (called value). After that are any column that you want to gather into the new key and value columns.Because we want all the year columns gathered but not the country columns, we will add the argument “-Country” (pay attention to the minus sign).

After we gathered the data, we can easily merge it to the grades data, since now their shape is similar. notice that rows that aren't filled in both dataframes are removed automatically.

In addition, we use the same functions as before to get the OECD average grades of each year. Notice that some of them are missing values, i decided to drop them.

Let's have a look on the gathered salaries dataframe:

```
gathered_salaries = gather(data = salaries, key = "year", value =
salary, -Country)
head(gathered_salaries[order(gathered_salaries$Country, decreasing =
FALSE),])

## # A tibble: 6 x 3
##   Country    year  salary
##   <chr>      <chr>  <dbl>
## 1  Australia 2005  51322.
## 2  Australia 2006  50296.
## 3  Australia 2007  51322.
## 4  Australia 2008  53375.
## 5  Australia 2009  52862.
## 6  Australia 2010  53888.

total = merge(data, gathered_salaries, by=c("Country","year"))
```

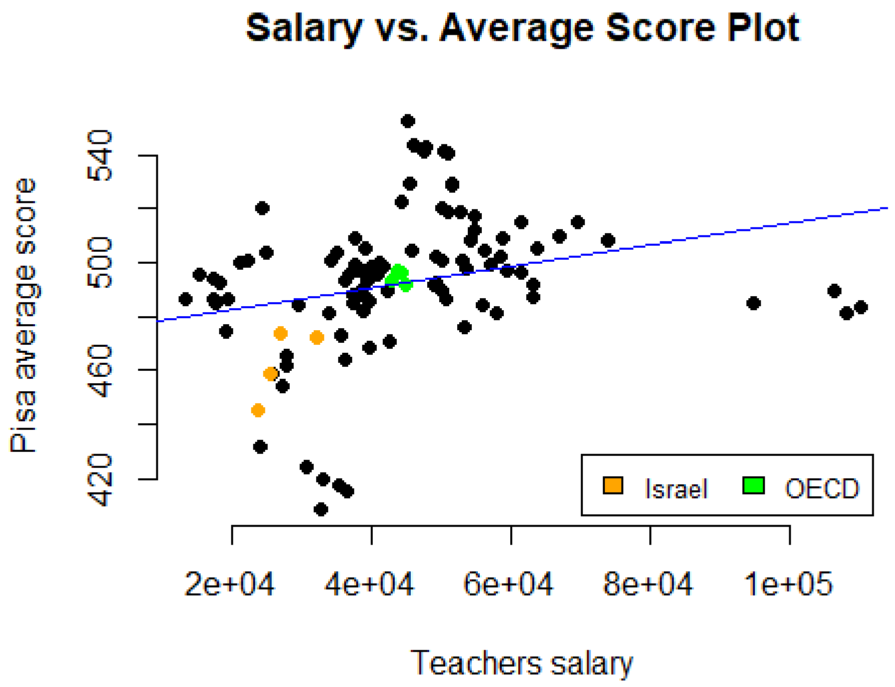
And here is the result of the final merge:

```
##      Country year math_score science_score reading_score
average_score
## 1  Australia 2006        520           527           513
520.0000
## 2  Australia 2009        514           527           515
518.6667
## 3  Australia 2012        504           521           512
512.3333
## 4  Australia 2015        494           510           503
502.3333
## 5    Austria 2006        505           511           490
502.0000
## 6    Austria 2009        496           494           470
486.6667
##      salary
## 1 50295.63
## 2 52861.74
## 3 54914.62
## 4 58507.16
## 5 49334.32
## 6 50814.35
```

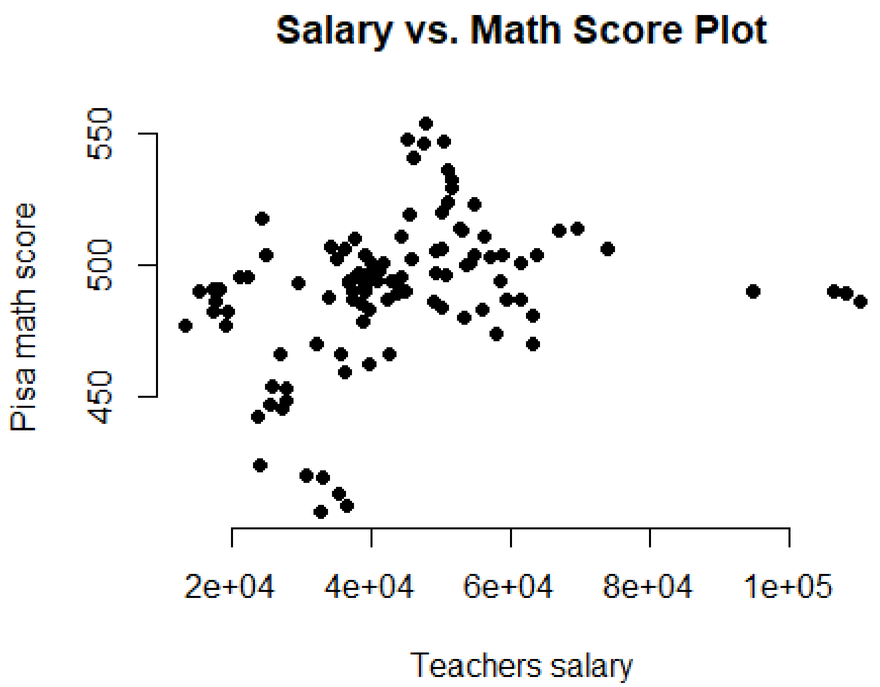
3.2-3

Let’s draw some graphs.

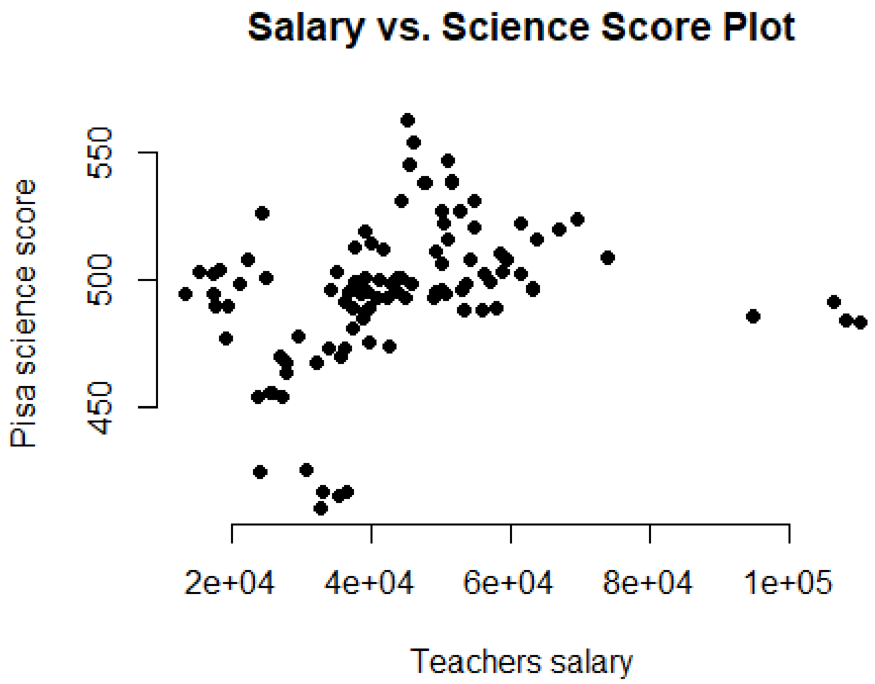
```
#Salary vs average plot
total$Country = trim(total$Country)
plot(x = total$salary, y = total$average_score, main = "Salary vs.
Average Score Plot", ylab = "Pisa average score", xlab = "Teachers
salary",
     pch =19, frame = FALSE, col =
ifelse( (total$Country)=="Israel", "orange", ifelse(
(total$Country==c("International Average (OECD)")), "green", "black")))

#Add plot legend
#legend(100,100,legend=c("Israel", "OECD average"),col=c("orange",
"green"))

legend("bottomright", inset=.02,
   c("Israel","OECD"), fill=c("orange","green"), horiz=TRUE, cex=0.8)

#Regression line
abline(lm(total$average_score ~ total$salary, data = total), col =
"blue")
```



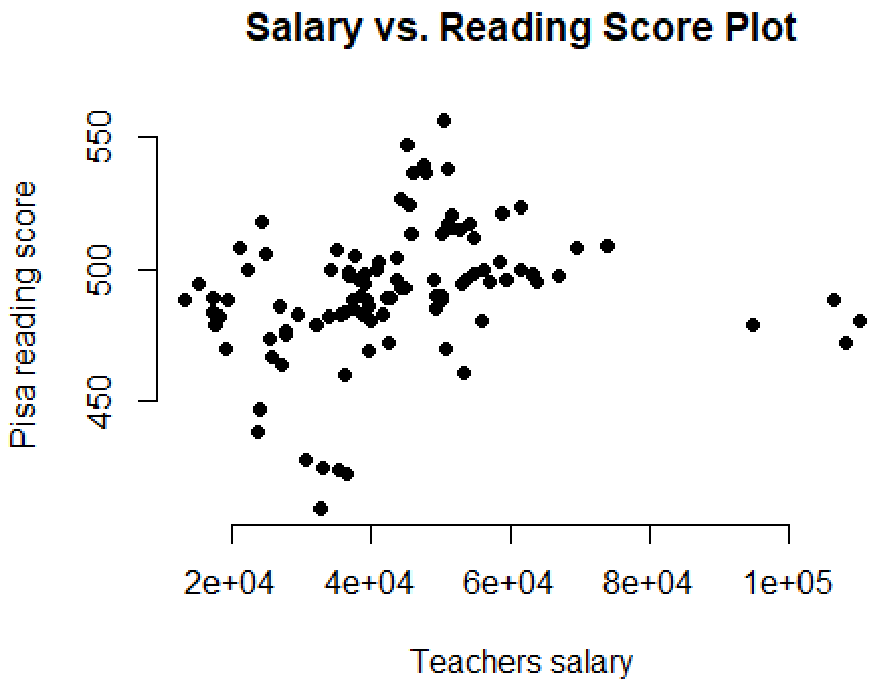
```
#Salary vs math plot
plot(x = total$salary, y = total$math_score, main = "Salary vs. Math
Score Plot",
     ylab = "Pisa math score", xlab = "Teachers salary",
     pch =19, frame = FALSE)
```



```
#Salary vs science plot
plot(x= total$salary, y = total$science_score, main = "Salary vs.
Science Score Plot",
     ylab = "Pisa science score", xlab = "Teachers salary",
     pch =19, frame = FALSE)
```



```
#Salary vs reading plot
plot(x = total$salary, y = total$reading_score, main = "Salary vs.
Reading Score Plot",
     ylab = "Pisa reading score", xlab = "Teachers salary",
     pch =19, frame = FALSE)
```



3.4

Conclusions

We can see that there is some correlation between the teachers salary and the grades. We can get some conclusions from those graphs. First, in richer countries, where teachers earn more, it is possible that more money was invested in education. It is also possible that students from reacher countries has more tools to succeed, like private teachers or after-hours classess. But, we can not forget that the data was collected over more than 10 years, and for my opinion it is possible that the teachers salary is getting higher every year beacause of Infaltion, and that the grades are getting higher because more thecnology is envolved this days in schools and students houses, thus a student can study math or read more at home from various websites. In conclusion, for my opinion the correlation is not strong enough, but it is possible that there is a connection betwwen teachers salary and Pisa test, as written above.

4.1

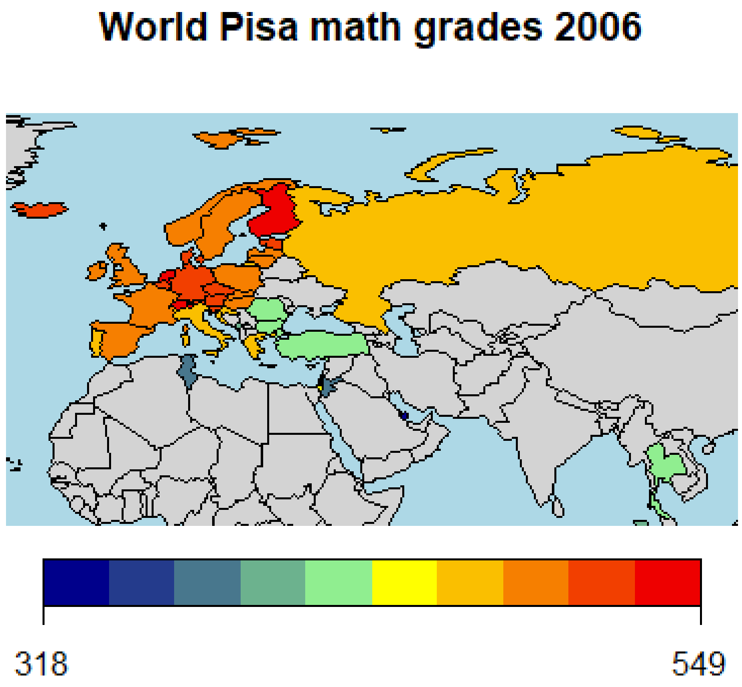
Grades by region

During the work on the grades tables, I was wondering if there is any connection between geographic region and math grade. To do so, i decided to color the world map by the Pisa math score value. And Here is the results. More specifically, i decided to check regions around the medditerenian sea. Here are the results:

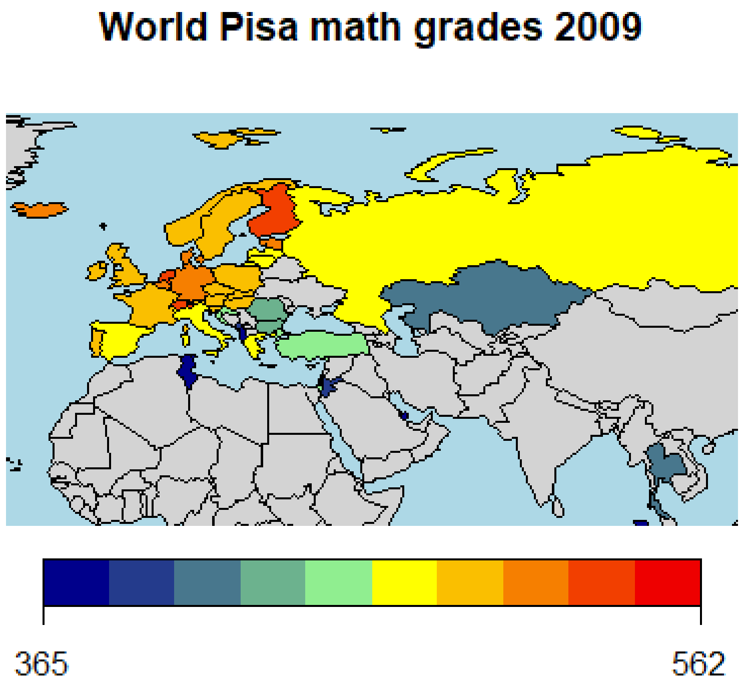
```
twenty_six = data[which(data$year==2006),]
twenty_six = merge(
salaries[(which(colnames(salaries)==c("Country","2006")))], twenty_six,
by = c("Country") ,all = TRUE, sort = TRUE)
twenty_six = twenty_six[-c(5,13,74,75),]
twenty_six$Country = (trim(twenty_six$Country))


twenty_nine = data[which(data$year==2009),]
twenty_nine = merge(
salaries[(which(colnames(salaries)==c("Country","2009")))],
twenty_nine, by = c("Country") ,all = TRUE, sort = TRUE)
twenty_nine = twenty_nine[-c(5,13,74,75),]
twenty_nine$Country = (trim(twenty_nine$Country))

twenty_fifteen$Country = (trim(twenty_fifteen$Country))

#create a map-shaped window
mapDevice('x11')
#join to a coarse resolution map
spdf <- joinCountryData2Map(twenty_fifteen, joinCode="NAME",
nameJoinColumn="Country")
spdf2 <- joinCountryData2Map(twenty_six, joinCode="NAME",
nameJoinColumn="Country")
spdf3 <- joinCountryData2Map(twenty_nine, joinCode="NAME",
nameJoinColumn="Country")
```

It is very interesting to see the changes along the years in european countries, and how for example, countries like Germany Sweden and Finland improved over the years, and the neighbours like spain and portugal were left behind.

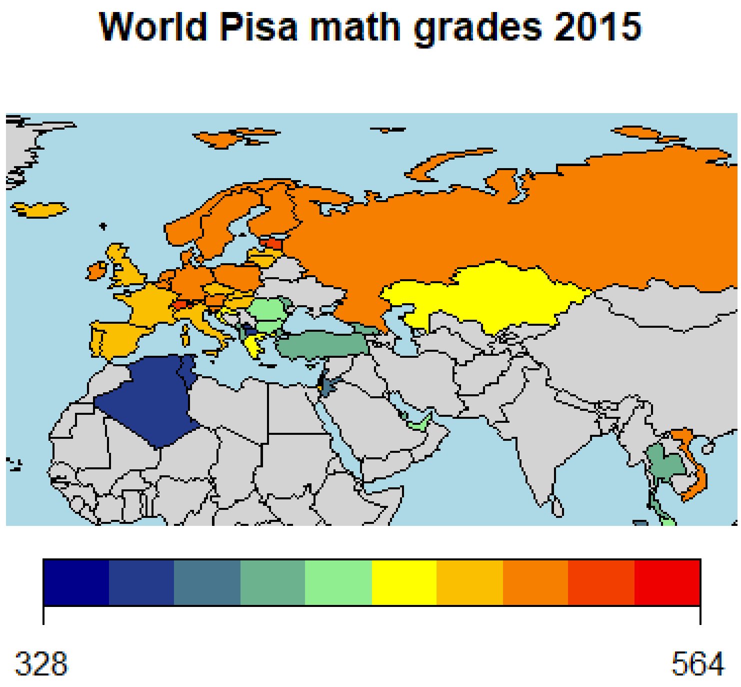
```
mapCountryData(spdf2, nameColumnToPlot="math_score",
catMethod="fixedWidth", colourPalette = "diverging", mapTitle = "World
Pisa math grades 2006",  numCats = 10, oceanCol = "lightblue",
missingCountryCol = "lightgrey", mapRegion = 'Eurasia', borderCol =
'black')
```



```
mapCountryData(spdf3, nameColumnToPlot="math_score",
catMethod="fixedWidth", colourPalette = "diverging", mapTitle = "World
Pisa math grades 2009", numCats = 10, oceanCol = "lightblue",
missingCountryCol = "lightgrey",mapRegion = 'Eurasia', borderCol =
'black')
```



```
mapCountryData(spdf, nameColumnToPlot="math_score",
catMethod="fixedWidth", colourPalette = "diverging", mapTitle = "World
Pisa math grades 2015", numCats = 10, oceanCol = "lightblue",
missingCountryCol = "lightgrey", mapRegion = 'Eurasia', borderCol =
'black')
```



Special Thanks

I would like to thanks Dr Yuval Benjamini from the Hebrew Univuersity Jerusalem Institute for the data and all the questions asked about it.

Special thanks to Mr. Adi Ziv for the support and guidance.