Final Project

Mercan KARACABEY

05 01 2019

PART 1

1. What is your opinion about Python vs R debate? To what extent do you agree with the post on https://www.dataschool.io/python-or-r-for-data-science/? Be honest, you won't be penalized or rewarded for stating your opinions; only by the quality your arguments.
   - I started to learn programming language at university with Python, so it was hard for me to learn R programming.However, for people who have just begun programming, R may be a better choice for producing output by writing fewer codes.(It is quite important to obtain particularly visible outputs for new programmer :)) In terms of visualization, R is more successful than Python to produce more complex visuals. In terms of machine learning and statistical learning; Python is more successful in machine learning, because the more practical preparation of the learning models and the back-end libraries are more complex. R more successful in statistical learning. Because of the practical functions used in mathematical modeling can provide rapid solutions. In terms of integrated work with distributed architectures, python's methodologies are more successful. For the scientists who will produce outputs on mathematical models, R is more convenient for practicality. In summary, in my opinion, data engineers will prefer python and data scientists will prefer R.
2. What is your exploratory data analysis workflow? Suppose you are given a data set and a research question. Where do you start? How do you proceed? For instance, you are given the task to distribute funds from donations to public welfare projects in a wide range of subjects (e.g. education, gender equality, poverty, job creation, healthcare) with the objective of maximum positive impact on the society in general. Assume you have almost all the data you require. How do you measure impact? How do you form performance measures? What makes you think you find an interesting angle?

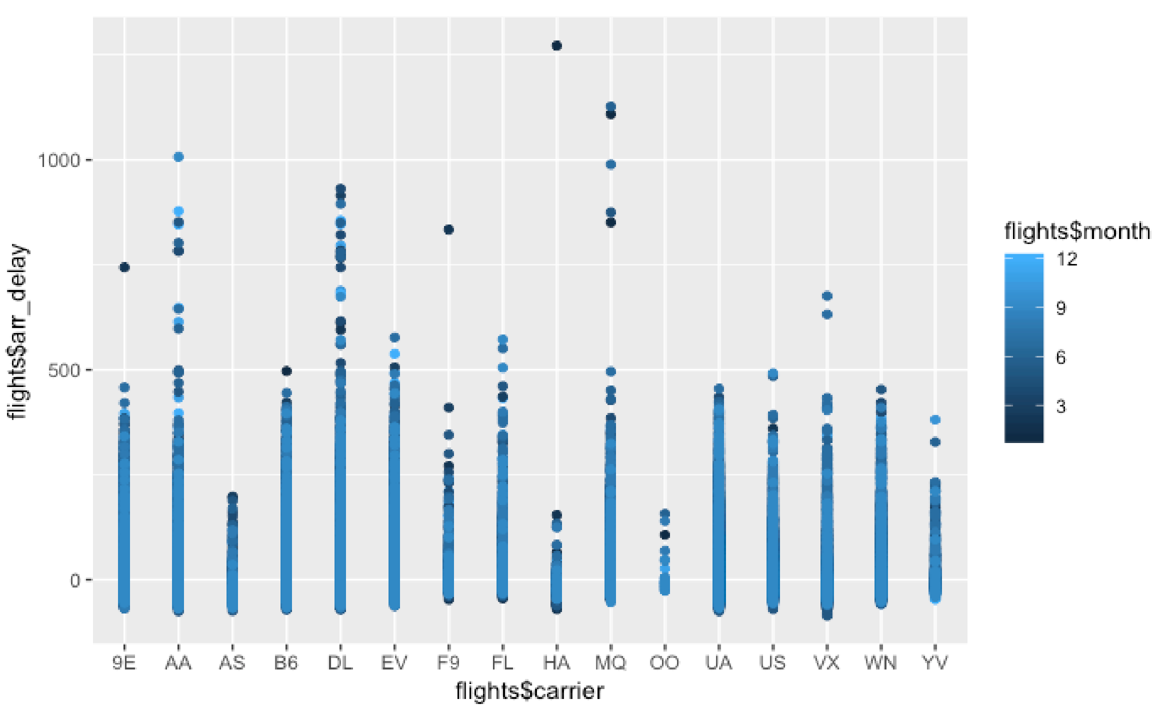
   Firstly, I analyze what types of data they contain, and the relationship between them if there are other l inked data. The operation is performed on the data by cleaning.Discrimination analysis, outlier analysis etc. can be carried out at this stage. Algorithms are deduced by deciding which machine learning or statistical learning is used.(Maybe used both of them.) The output generated by data visualization is interpreted by enriching the output. As it is a social issue about interpretation, I take the increasing benefit as a criterion for measurement.
   -
3. If you had to plot a single graph using the flights data what would it be? Why? Make your argument, actually code the plot and provide the output. (You can find detailed info about the movies data set in its help file. Use ?flights, after you load nycflights13 package.)

   I would like a plot a graph regarding carrier vs air delay. A plot is designed for the purpose of evaluating the carrier based on a minimum of flight delay.Without outliers, the most rapid carrier can be found.

```r
library(dplyr)
library(nycflights13)
library(ggplot2)
glimpse(flights)
```

```
## Observations: 336,776
## Variables: 19
## $ year           <int>  2013, 2013,  2013, 2013, 2013, 2013, 2013, 2013,...
## $ month          <int>  1, 1, 1, 1,  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ day            <int>  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
## $ dep_time       <int>  517, 533, 542, 544, 554, 554, 555, 557, 557, 55...
                          515, 529, 540, 545, 600, 558, 600, 600, 600, 60...
## $ dep_delay      <dbl>  2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2...
## $ arr_time       <int>  830, 850, 923, 1004, 812, 740, 913, 709, 838, 7...
## $ sched_arr_time <int>  819, 830, 850, 1022, 837, 728, 854, 723, 846, 7...
## $ arr_delay      <dbl>  11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -...
## $ carrier        <chr>  "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV",...
## $ flight         <int>  1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79...
## $ tailnum        <chr>  "N14228", "N24211", "N619AA", "N804JB", "N668DN...
## $ origin         <chr>  "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR"...
## $ dest           <chr>  "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL"...
## $ air_time       <dbl>  227, 227, 160, 183, 116, 150, 158, 53, 140, 138...
## $ distance       <dbl>  1400, 1416, 1089, 1576, 762, 719, 1065, 229, 94...
## $ hour           <dbl>  5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5,...
## $ minute         <dbl>  15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013...
```

```r
ggplot(flights, aes(flights$ carrier, flights$air_delay, colour = flights$months)) + geom_point()
```



PART 2

- *In previous analyzes, only export / import analysis was performed on a date basis.
- *In addition, Export / Import ratio depending on dollar exchange rate.
- *It was observed that both decreases due to the sudden increase in the dollar exchange rate.
- *Correlation table was created.(Consumer Price Index/ USD Rate / Import-Export Amounts)
- *The correlation values are shown in the table depending on the data in the confidence interval.(0.95)
- *The regression line is drawn. Analysis was made with and without confidence interval. In addition, drawing using the Loess method.

```
#required packages
library("tidyverse")
```

```
## -- Attaching packages ---------------------------------------------------------------------------

## v tibble     1.4.2     v purrr   0.2.5
## v tidyr      0.8.2     v stringr 1.3.1
## v readr      1.1.1     v forcats 0.3.0
## -- Conflicts ------------------------------------------------------------------------------------ t ## x dplyr::filter() masks stats::filter()

## x dplyr::lag()     masks stats::lag()
```

```
library("readxl")
library("ggplot2")
library("plotly")
```

```
##
```

```
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```
library("gapminder")
library("dplyr")
```

Download RDS Data from Github Page

```
tmp<-tempfile(fileext=".rds")
download.file("https://github.com/MEF-BDA503/gpj18-r_coders/blob/master/Data_Sources_Rds/imp_data_final
              destfile=tmp,mode = 'wb')
imp_data_final<-read_rds(tmp)
file.remove(tmp)
```

```
## [1] TRUE
```

```
tmp<-tempfile(fileext=".rds")
download.file("https://github.com/MEF-BDA503/gpj18-r_coders/blob/master/Data_Sources_Rds/exp_data_final
              destfile=tmp,mode = 'wb')
exp_data_final<-read_rds(tmp)
file.remove(tmp)
```

```
## [1] TRUE
```

```
tmp<-tempfile(fileext=".rds")
download.file("https://github.com/MEF-BDA503/gpj18-r_coders/blob/master/Data_Sources_Rds/imp_data.rds?r
              destfile=tmp,mode = 'wb')
imp_data<-read_rds(tmp)
file.remove(tmp)
```

```
## [1] TRUE
```

```
tmp<-tempfile(fileext=".rds")
download.file("https://github.com/MEF-BDA503/gpj18-r_coders/blob/master/Data_Sources_Rds/exp_data.rds?r
              destfile=tmp,mode = 'wb')
exp_data<-read_rds(tmp)
file.remove(tmp)
```

```
## [1] TRUE
```

```
tmp<-tempfile(fileext=".rds")
download.file("https://github.com/MEF-BDA503/gpj18-r_coders/blob/master/Data_Sources_Rds/Producer_Infla
              destfile=tmp,mode = 'wb')
producer_inf<-read_rds(tmp)
file.remove(tmp)
```

```
## [1] TRUE
```

```r
# Create a temporary file
tmp=tempfile(fileext=".xls")
# Download file from repository to the temp file
download.file("https://github.com/MEF-BDA503/gpj18-r_coders/blob/master/Data_Sources_Excel/export_impor
              destfile=tmp,mode='wb')
# Read that excel file. sectors
<- read_excel(tmp)
```

```
## readxl works best with a newer version of the tibble package.
## You currently have tibble v1.4.2.
## Falling back to column name repair from tibble <= v1.4.2.
## Message displays once per session.
```

```r
# Remove the temp file
file.remove(tmp)
```

```
## [1] TRUE
```

```r
tmp<-tempfile(fileext=".rds")
download.file("https://github.com/MEF-BDA503/gpj18-r_coders/blob/master/Data_Sources_Rds/US_Dollar_Mont
              destfile=tmp,mode = 'wb')
usd_rate<-read_rds(tmp)
file.remove(tmp)
```

```
## [1] TRUE
```

Format Data

```r
names(exp_data_final)[names(exp_data_final) == 'Date'] <- 'Export_Date'
names(exp_data)[names(exp_data) == 'Date'] <- 'Export_Date'
names(imp_data_final)[names(imp_data_final) == 'Date'] <- 'Import_Date'
names(imp_data_final)[names(imp_data_final) == 'Export_Total_Amount']<-'Import_Total_Amount'
names(imp_data)[names(imp_data) == 'Date'] <- 'Import_Date'

library("dplyr")
exp_data <- inner_join(exp_data,sectors, by=c("Sector_Type_Code"="Sub_Sector_Type_Code"))

imp_data <- inner_join(imp_data,sectors, by=c("Sector_Type_Code"="Sub_Sector_Type_Code"))

exp_data$Export_Year<-as.numeric(format(exp_data$Export_Date,"%Y"))
exp_data$Export_Year_Month<-format(exp_data$Export_Date,"%Y-%m")
exp_data_final$Export_Year<-as.numeric(format(exp_data_final$Export_Date,"%Y"))
exp_data_final$Export_Year_Month<-format(exp_data_final$Export_Date,"%Y-%m")

imp_data$Import_Year<-as.numeric(format(imp_data$Import_Date,"%Y"))
imp_data$Import_Year_Month<-format(imp_data$Import_Date,"%Y-%m")
imp_data_final$Import_Year<-as.numeric(format(imp_data_final$Import_Date,"%Y"))
imp_data_final$Import_Year_Month<-format(imp_data_final$Import_Date,"%Y-%m")

imp_data<- imp_data %>%
  select (Import_Date,Sector_Type_Code,Sector_Type_Code.y,Main_Sector_Flag,Sector_Name_Eng,
          Amount,Import_Year,Import_Year_Month)
exp_data<- exp_data %>%
  select (Export_Date,Sector_Type_Code,Sector_Type_Code.y,Main_Sector_Flag,Sector_Name_Eng,
          Amount,Export_Year,Export_Year_Month)
```

```r
colnames(imp_data)[colnames(imp_data) == 'Amount'] <- 'Import_Amount'
colnames(exp_data)[colnames(exp_data) == 'Amount'] <- 'Export_Amount'
colnames(imp_data)[colnames(imp_data) == 'Sector_Type_Code'] <- 'Sub_Sector_Type_Code'
colnames(exp_data)[colnames(exp_data) == 'Sector_Type_Code'] <- 'Sub_Sector_Type_Code'
colnames(imp_data)[colnames(imp_data) == 'Sector_Type_Code.y'] <- 'Sector_Type_Code'
colnames(exp_data)[colnames(exp_data) == 'Sector_Type_Code.y'] <- 'Sector_Type_Code'
imp_data$Import_Amount[is.na(imp_data$Import_Amount)] <- 0
imp_data_final$Import_Total_Amount[is.na(imp_data_final$Import_Total_Amount)] <- 0
exp_data$Export_Amount[is.na(exp_data$Export_Amount)] <- 0
exp_data_final$Export_Total_Amount[is.na(exp_data_final$Export_Total_Amount)] <- 0



        exp_data_final <- exp_data_final %>%
           filter(Export_Date<'2018-11-01')

exp_data <- exp_data %>%
  filter(Export_Date<'2018-11-01')

        imp_data_final <- imp_data_final %>%
           filter(Import_Date<'2018-11-01')

imp_data <- imp_data %>%
  filter(Import_Date<'2018-11-01')

saveRDS(imp_data,file="imp_data_v2.rds")
saveRDS(imp_data_final,file="imp_data_final_v2.rds")
saveRDS(exp_data,file="exp_data_v2.rds")
saveRDS(exp_data_final,file="exp_data_final_v2.rds")
```

Format Data

```
For exploratory data analysis, I put this stage(Data preparation)
```{r, warning=FALSE}
names(exp_data_final)[names(exp_data_final) == 'Date'] <- 'Export_Date'
names(exp_data)[names(exp_data) == 'Date'] <- 'Export_Date'
names(imp_data_final)[names(imp_data_final) == 'Date'] <- 'Import_Date'
names(imp_data_final)[names(imp_data_final) == 'Export_Total_Amount']<-'Import_Total_Amount'
names(imp_data)[names(imp_data) == 'Date'] <- 'Import_Date'

library("dplyr")
exp_data <- inner_join(exp_data,sectors, by=c("Sector_Type_Code"="Sub_Sector_Type_Code"))

imp_data <- inner_join(imp_data,sectors, by=c("Sector_Type_Code"="Sub_Sector_Type_Code"))

exp_data$Export_Year<-as.numeric(format(exp_data$Export_Date,"%Y"))
exp_data$Export_Year_Month<-format(exp_data$Export_Date,"%Y-%m")
exp_data_final$Export_Year<-as.numeric(format(exp_data_final$Export_Date,"%Y"))
exp_data_final$Export_Year_Month<-format(exp_data_final$Export_Date,"%Y-%m")

imp_data$Import_Year<-as.numeric(format(imp_data$Import_Date,"%Y"))
imp_data$Import_Year_Month<-format(imp_data$Import_Date,"%Y-%m")
imp_data_final$Import_Year<-as.numeric(format(imp_data_final$Import_Date,"%Y"))
imp_data_final$Import_Year_Month<-format(imp_data_final$Import_Date,"%Y-%m")

imp_data<- imp_data %>%
  select (Import_Date,Sector_Type_Code,Sector_Type_Code.y,Main_Sector_Flag,Sector_Name_Eng,
         Amount,Import_Year,Import_Year_Month)
exp_data<- exp_data %>%
  select (Export_Date,Sector_Type_Code,Sector_Type_Code.y,Main_Sector_Flag,Sector_Name_Eng,
         Amount,Export_Year,Export_Year_Month)

colnames(imp_data)[colnames(imp_data) == 'Amount'] <- 'Import_Amount'
colnames(exp_data)[colnames(exp_data) == 'Amount'] <- 'Export_Amount'
```
```

```
colnames(imp_data)[colnames(imp_data) == 'Sector_Type_Code'] <- 'Sub_Sector_Type_Code'
colnames(exp_data)[colnames(exp_data) == 'Sector_Type_Code'] <- 'Sub_Sector_Type_Code'
colnames(imp_data)[colnames(imp_data) == 'Sector_Type_Code.y'] <- 'Sector_Type_Code'
colnames(exp_data)[colnames(exp_data) == 'Sector_Type_Code.y'] <- 'Sector_Type_Code'
imp_data$Import_Amount[is.na(imp_data$Import_Amount)] <- 0
imp_data_final$Import_Total_Amount[is.na(imp_data_final$Import_Total_Amount)] <- 0
exp_data$Export_Amount[is.na(exp_data$Export_Amount)] <- 0
exp_data_final$Export_Total_Amount[is.na(exp_data_final$Export_Total_Amount)] <- 0


exp_data_final <- exp_data_final %>%
  filter(Export_Date<'2018-11-01')

exp_data <- exp_data %>%
  filter(Export_Date<'2018-11-01')

imp_data_final <- imp_data_final %>%
  filter(Import_Date<'2018-11-01')

imp_data <- imp_data %>%
  filter(Import_Date<'2018-11-01')

saveRDS(imp_data,file="imp_data_v2.rds")
saveRDS(imp_data_final,file="imp_data_final_v2.rds")
saveRDS(exp_data,file="exp_data_v2.rds")
saveRDS(exp_data_final,file="exp_data_final_v2.rds")
```

Prepare Data and Import&Export Line Graph

**In previous analyzes, only export / import analysis was performed on a date basis.*
**In addition, Export / Import ratio depending on dollar exchange rate.*
**It was observed that both decreases due to the sudden increase in the dollar exchange rate.*
**Correlation table was created.(Consumer Price Index/ USD Rate / Import-Export Amounts)*
**The correlation values are shown in the table depending on the data in the confidence interval.(0.95)*
**The regression line is drawn. Analysis was made with and without confidence interval. In addition, drawing using the Loess method*

```
```{r, warning=FALSE }
imp_and_exp_data <- inner_join(exp_data, imp_data, by=c("Export_Date" =
"Import_Date","Sub_Sector_Type_Code"="Sub_Sector_Type_Code"))

imp_and_exp_data_bymonth <- aggregate(cbind(Import_Amount, Export_Amount) ~ Export_Date, data =
imp_and_exp_data, sum)
imp_and_exp_data_bymonth <- gather(imp_and_exp_data_bymonth,
                               value = "value",
                               key = "type",
                               Export_Amount, Import_Amount)

#Rename column names
colnames(imp_and_exp_data_bymonth) <- c("Date","Type","Amount")

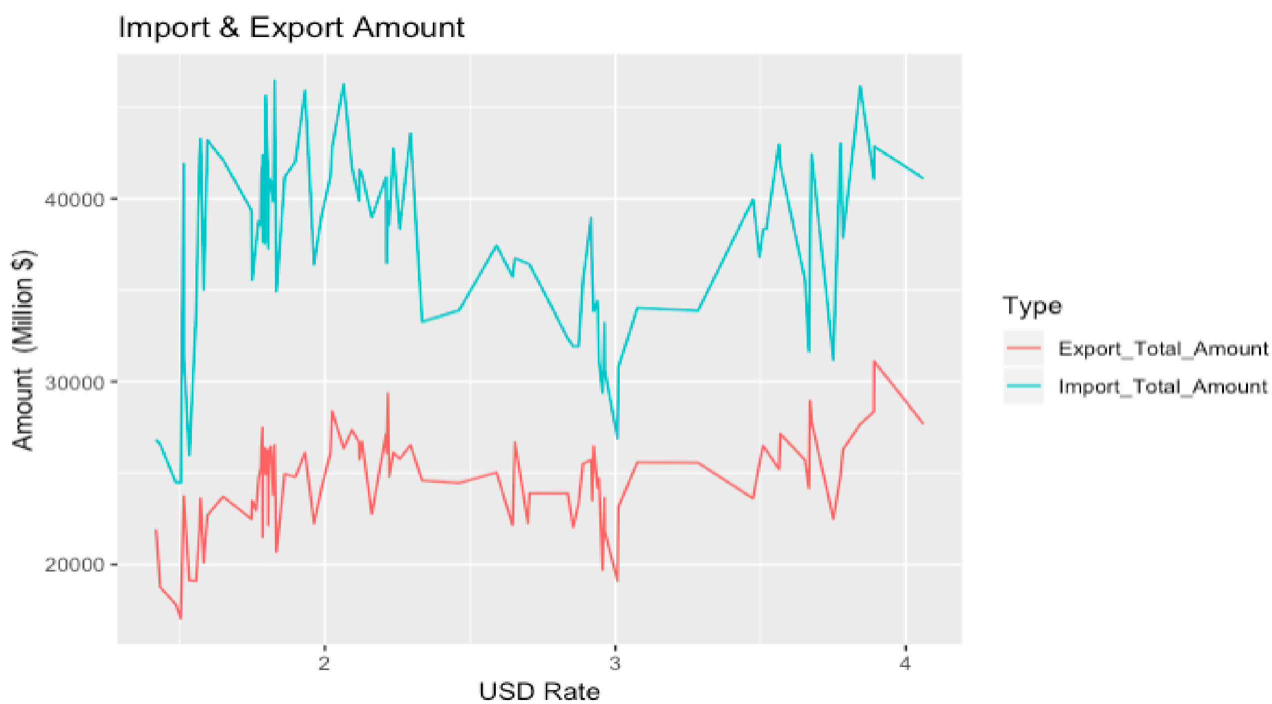
#Remove Empty Dates
imp_and_exp_data_bymonth <- imp_and_exp_data_bymonth %>%
  filter(Date<'2018-11-01')

imp_and_exp_data_final_1 <- inner_join(exp_data_final, imp_data_final, by=c("Export_Date" = "Import_Date"))


imp_and_exp_data_bymonth_final <- aggregate(cbind(Import_Total_Amount, Export_Total_Amount) ~ USD_Rate.y, data
= imp_and_exp_data_final_1, sum)
imp_and_exp_data_bymonth_final <- gather(imp_and_exp_data_bymonth_final,
                               value = "value",
                               key = "type",
                               Export_Total_Amount, Import_Total_Amount)

#Rename column names
colnames(imp_and_exp_data_bymonth_final) <- c("USD_Rate","Type","Amount")
```
```

```
p<-ggplot(imp_and_exp_data_bymonth_final,
       aes(x=USD_Rate,
           y=Amount/1000,
           color=Type)) +
   geom_line()+
   scale_size_area("Nitrogen") +
   xlab("USD Rate") +
   ylab("Amount  (Million $)") +
   ggtitle("Import & Export Amount")
 style(p, text = row.names(imp_and_exp_data_bymonth))
 p
```



- Corelation Matrix

```{r, warning=FALSE }
temp_table <- imp_and_exp_data_final_1 %>%
select(Export_Total_Amount,Consumer_Price_Index_Monthly_Change.x,Consumer_Price_Index_Yearly_Change.y,Import_Total_Amount,USD_Rate.y)


names(temp_table)[names(temp_table) == "Export_Total_Amount"] <- "Exp.Amount"

names(temp_table)[names(temp_table) == "Import_Total_Amount"] <- "Imp.Amount"

names(temp_table)[names(temp_table) == "Consumer_Price_Index_Monthly_Change.x"] <- "ConsumerPriceX"


names(temp_table)[names(temp_table) == "Consumer_Price_Index_Yearly_Change.y"] <- "ConsumerPriceY"

names(temp_table)[names(temp_table) == "USD_Rate.y"] <- "USD_R"


corelation_matrix <- cor(temp_table)
corelation_matrix
```

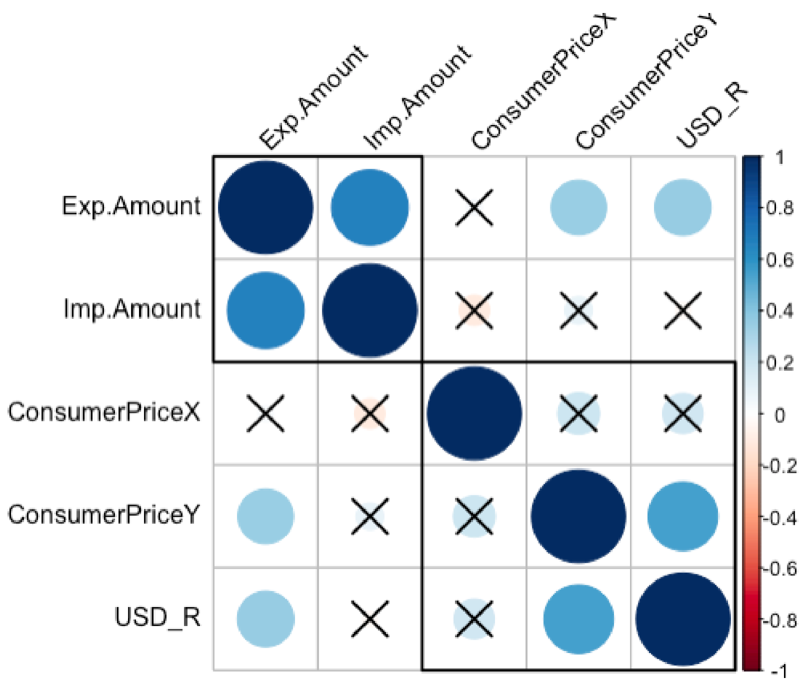
	Exp.Amount	ConsumerPriceX	ConsumerPriceY	Imp.Amount	USD_R
Exp.Amount	1.0000000	0.0298634	0.34818339	0.66498247	0.35976687
ConsumerPriceX	0.0298634	1.0000000	0.19873285	-0.10411749	0.18414595
ConsumerPriceY	0.3481834	0.1987328	1.00000000	0.08591141	0.54789133
Imp.Amount	0.6649825	-0.1041175	0.08591141	1.00000000	-0.02853854
USD_R	0.3597669	0.1841459	0.54789133	-0.02853854	1.00000000

Corelation Matrix - Graph

````
```{r, warning=FALSE }
library(corrplot)

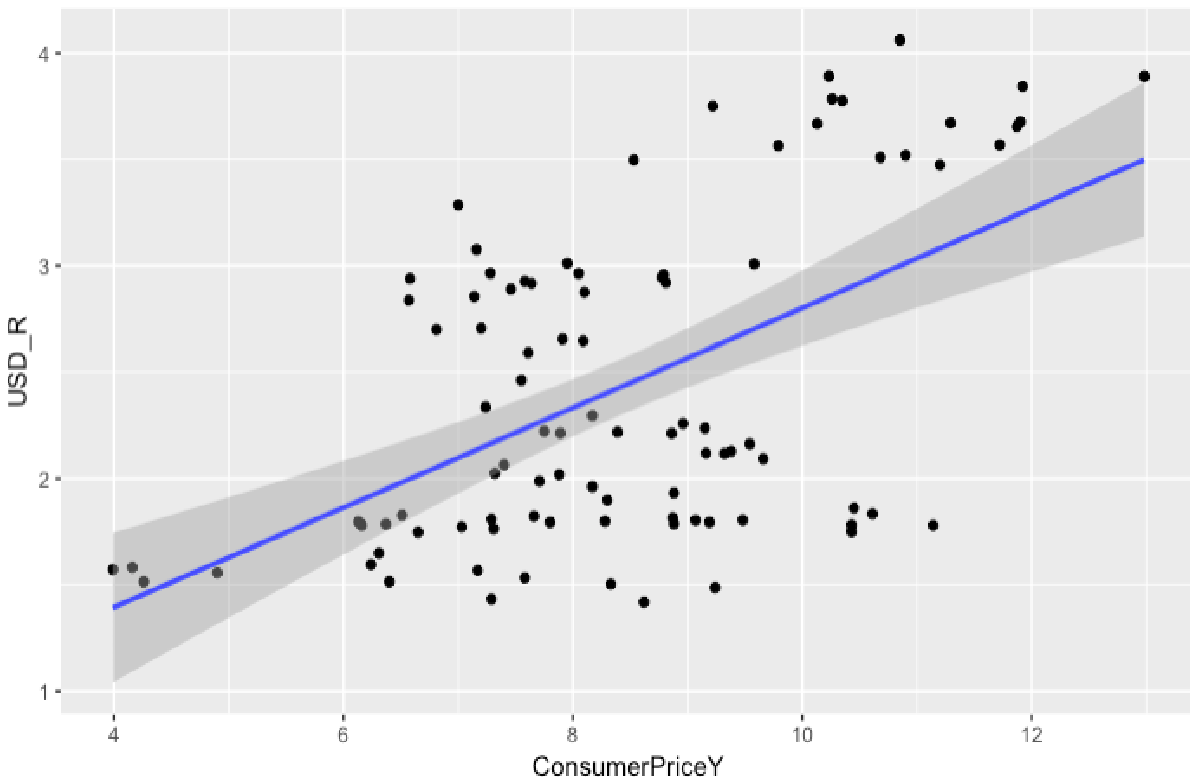
cor.mtest <- function(mat, conf.level = 0.95) {
  mat <- as.matrix(mat)
  n <- ncol(mat)
  p.mat <- lowCI.mat <- uppCI.mat <- matrix(NA, n, n)
  diag(p.mat) <- 0
  diag(lowCI.mat) <- diag(uppCI.mat) <- 1
  for (i in 1:(n - 1)) {
    for (j in (i + 1):n) {
      tmp <- cor.test(mat[, i], mat[, j], conf.level = conf.level)
      p.mat[i, j] <- p.mat[j, i] <- tmp$p.value
      lowCI.mat[i, j] <- lowCI.mat[j, i] <- tmp$conf.int[1]
      uppCI.mat[i, j] <- uppCI.mat[j, i] <- tmp$conf.int[2]
    }
  }
  return(list(p.mat, lowCI.mat, uppCI.mat))
}

res <- cor.mtest(temp_table, 0.95)
corrplot(corelation_matrix, method = "circle", order = "hclust", p.mat = res[[1]], sig.level = 0.05, addrect = 2, tl.col="black", tl.srt=45)
```
````



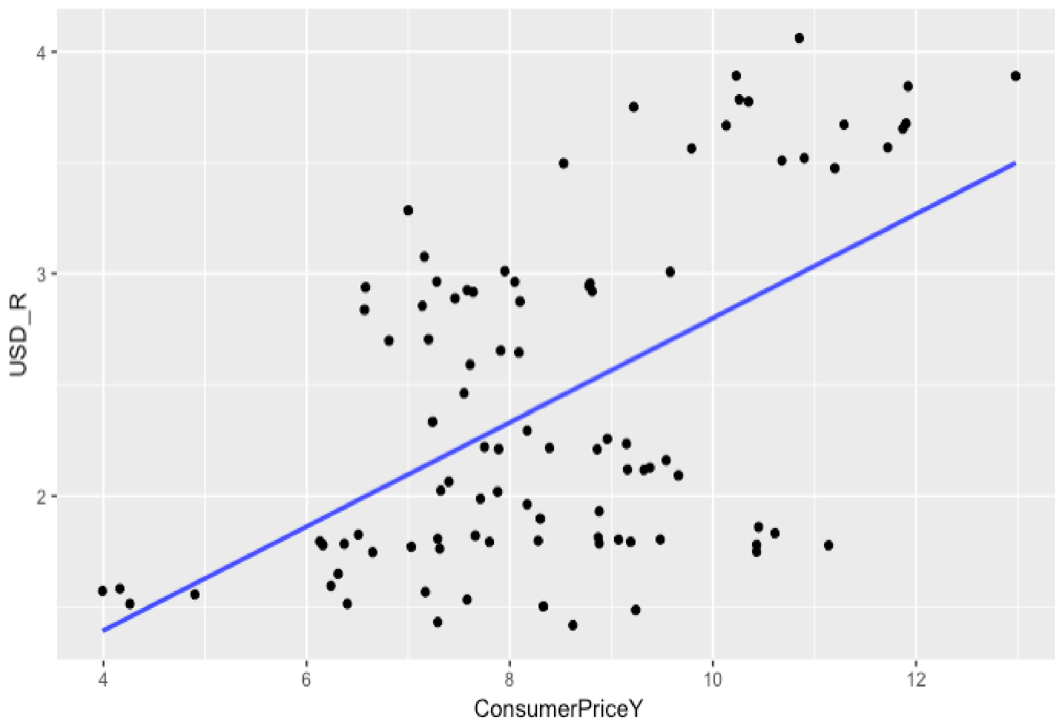
Regression Graph

````
```{r, warning=FALSE }
# Regresyon doğrusu eklemek
ggplot(temp_table, aes(x=ConsumerPriceY, y=USD_R)) +
geom_point()+
geom_smooth(method=lm)
```
````



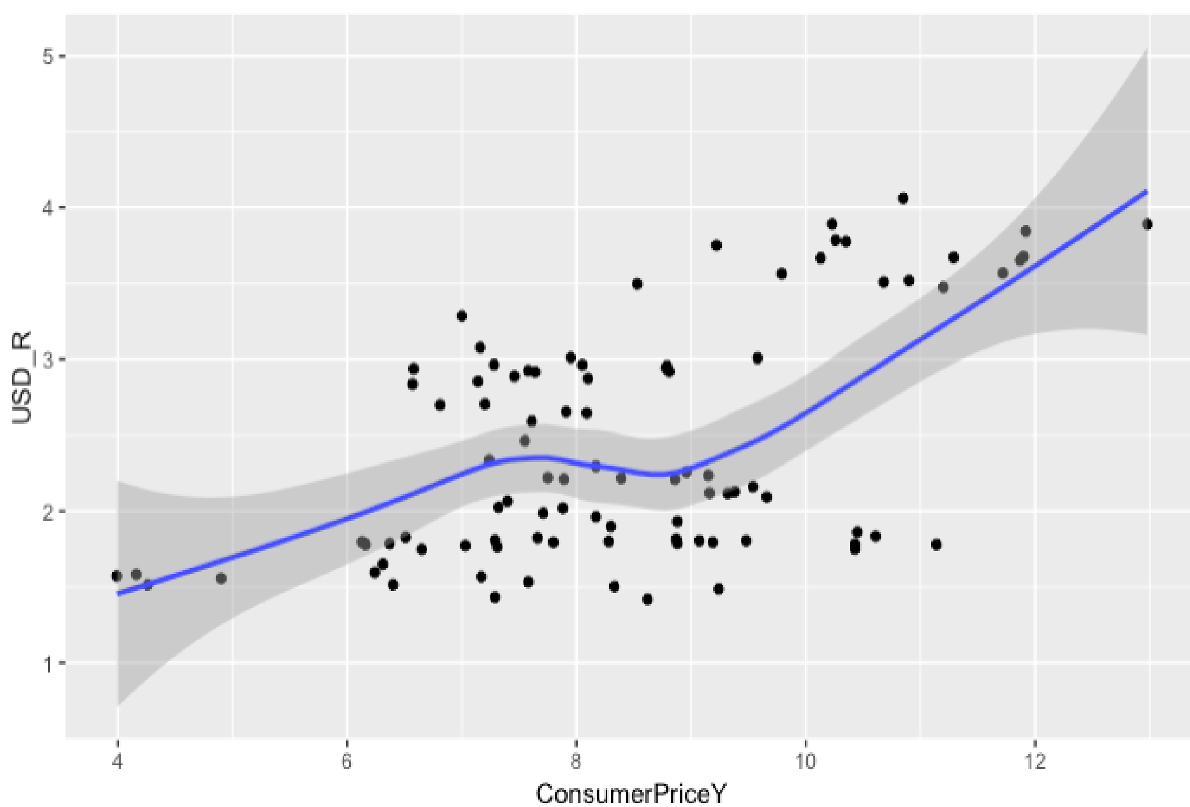
Remove Confidence Interval and Draw Graph

````
```{r, warning=FALSE }
# Güven aralığı kaldırıldığında
ggplot(temp_table, aes(x=ConsumerPriceY, y=USD_R)) +
geom_point()+
geom_smooth(method=lm, se=FALSE)
```
````



Loess method

```
```{r, warning=FALSE }
# Loess metodu
ggplot(temp_table, aes(x=ConsumerPriceY, y=USD_R)) +
geom_point()+
geom_smooth()
```
```



PART 3

```
tmp<-tempfile(fileext=".rds")
download.file("https://github.com/MEF-BDA503/pj18-efehandanisman/blob/master/timeshighereducation/ranki
              destfile=tmp,mode = 'wb')
education_data <-read_rds(tmp)
```

b) The correlation between: As a student in the industry, the number of students, the number of students, the number of studies and the ranking was investigated.

```
education_data$scores_research_rank <-suppressWarnings(as.numeric(education_data$scores_research_rank))
education_data$rank <- suppressWarnings(as.numeric(education_data$rank))
education_data$scores_industry_income <- suppressWarnings(as.numeric(education_data$scores_industry_income))
education_data$stats_number_students <- suppressWarnings(as.numeric(education_data$stats_number_students))

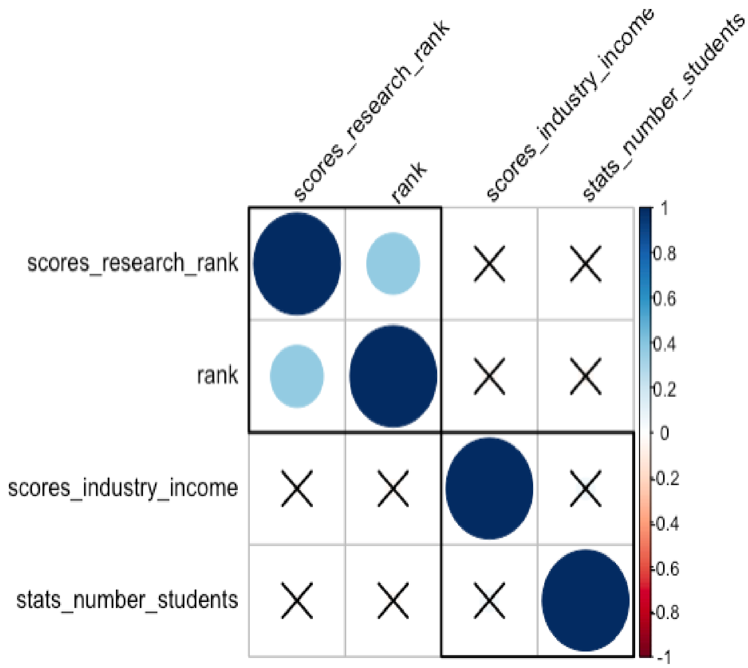
nEducationData <- education_data %>%
select(scores_research_rank,rank,scores_industry_income,stats_number_students)


corelation_matrix <- cor(nEducationData)


library(corrplot)
```

```
cor.mtest <- function(mat, conf.level = 0.95) {
  mat <- as.matrix(mat)
  n <- ncol(mat)
  p.mat <- lowCI.mat <- uppCI.mat <- matrix(NA, n, n)
  diag(p.mat) <- 0
  diag(lowCI.mat) <- diag(uppCI.mat) <- 1
  for (i in 1:(n - 1)) {
    for (j in (i + 1):n) {
      tmp <- cor.test(mat[, i], mat[, j], conf.level = conf.level)
      p.mat[i, j] <- p.mat[j, i] <- tmp$p.value
      lowCI.mat[i, j] <- lowCI.mat[j, i] <- tmp$conf.int[1]
      uppCI.mat[i, j] <- uppCI.mat[j, i] <- tmp$conf.int[2]
    }
  }
  return(list(p.mat, lowCI.mat, uppCI.mat))
}


res <- cor.mtest(nEducationData, 0.25)
corrplot(corelation_matrix, method = "circle", order = "hclust", p.mat = res[[1]], sig.level = 0.05, addrect = 2,
 tl.col="black", tl.srt=45)
```




```
```{r part3-b-2, echo=FALSE}
p <- ggplot(education_data, aes(x= education_data$scores_research_rank  , y=  education_data$rank)) +
  xlab("scores_research_rank") +
  ylab("rank") +
  ggtitle("Rank/Scores_research_rank")+
  geom_point(stat='identity', aes(col=education_data$stats_number_students), size=6)
p + theme(axis.text.x = element_text(angle = 90, hjust = 1))
ggplotly(p)
```
```

Values that appear to be correlated with a small scale are shown on the graph. However, it is not possible to comment on a relationship from the output of the graph.