Import and Export of India

Math2349 Assignment 3

Riya Minesh Amin s3807007

```
# libraries
library(dplyr)
library(tidyr)
library(stringr)
library(knitr)
library(ggplot2)
library(lubridate)
library(kableExtra)
library(outliers)
library(readr)
library(plotly)
library(tidyverse)
library(DT)
```

Executive Summary

India is one of the world's fastest developing nations, and the main component of any developing nation is trade between nations. The datasets consists of the commodities traded in the HS2 basket (import export data EDA | Kaggle. 2019. import export data EDA | Kaggle.). The datasets were used from Kaggle. The dataset consists of exchange values in the sum of US$ million for exporting and importing goods.There are two datasets imported in this report. The necessary libraries were loaded followed by the two dataset Import and Export csvs' of India. These dataset were than checked for any missing values, tidyness, outliers, classes of all the variables.

Data

```
# Data Import

Import <- read_csv("D:/2018-2010_import.csv")
```

```
## Parsed with column specification:
## cols(
##   HSCode = col_character(),
##   Commodity = col_character(),
##   value = col_double(),
##   country = col_character(),
##   year = col_double()
## )
```

```
Export <- read_csv("D:/2018-2010_export.csv")
```

```
## Parsed with column specification:
## cols(
##   HSCode = col_double(),
##   Commodity = col_character(),
##   value = col_double(),
##   country = col_character(),
##   year = col_double()
## )
```

Understand

- This section shows the class of the variables of both the dataset, dimensions, names, sapply and head of the dataset.

```
class(Import)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

```
class(Export)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

```
dim(Import)
```

```
## [1] 93095     5
```

```
dim(Export)
```

```
## [1] 137023      5
```

```
names(Import)
```

```
## [1] "HSCode"    "Commodity" "value"     "country"   "year"
```

```
names(Export)
```

```
## [1] "HSCode"    "Commodity" "value"     "country"   "year"
```

```
sapply(Import,class)
```

```
##      HSCode   Commodity       value     country        year
## "character" "character"   "numeric" "character"   "numeric"
```

```
sapply(Export,class)
```

```
##      HSCode   Commodity       value     country        year
##   "numeric" "character"   "numeric" "character"   "numeric"
```

```
head(Import)
```

HSCode <chr>
05
07
08
09
11
12

6 rows | 1-1 of 5 columns

```
head(Export)
```

HSCode <dbl>
2
3
4
6
7
8

6 rows | 1-1 of 5 columns

Changing the class

Here HSCode should be a Factor instead of Character.

```
Import$HSCode <- factor(Import$HSCode)
Export$HSCode <- factor(Export$HSCode)
```

- Above we can see that the dataset is tidy and each row consists of a single observation.

Mutating column HSCODE AND COMMODITY

```
#utility function used from stackoverflow
HSC1 <- function(cm,h){
  #cm <- gsub(pattern = ',',replacement = ';',x = cm)
  com <-substr(cm,start = 1,stop = 15)
  # com <- com[1]
  HSCode <- paste("HSCode",as.character(h),sep=': ')
  paste(com,HSCode,sep=' - ')
}

HSC2 <- function(cmv,hv){
  rtv <- character()
  for(i in 1:length(cmv)){
    rtv[i] <- HSC1(cmv[i],hv[i])
  }
  rtv
}
```

```
HSC_Commodity <- Import[,c(1,2)] %>% distinct()
HSC_Commodity <- HSC_Commodity %>% mutate(Commodity_HSCode = HSC2(Commodity,HSCode))
```

Scaning MISSING VALUES

```
colSums(is.na.data.frame(Import))
```

```
##    HSCode Commodity     value   country      year
##         0         0     14027         0         0
```

```
colSums(is.na.data.frame(Export))
```

```
##    HSCode Commodity     value   country      year
##         0         0     14038         0         0
```

Subsetting the blank columns

To remove the missing values the blank columns are filtered.

```
Import_na <- Import %>% filter(is.na(value))
Export_na <- Export %>% filter(is.na(value))

Import <- Import %>% filter(!is.na(value))
Export <- Export %>% filter(!is.na(value))
```

checking again the missing values

```
colSums(is.na.data.frame(Import))
```

```
##    HSCode Commodity     value   country      year
##         0         0         0         0         0
```

```
colSums(is.na.data.frame(Export))
```

```
##    HSCode Commodity     value   country      year
##         0         0         0         0         0
```

Analysing the data

Now to summarise the Import and Export per year a new data frame is formed for Import and Export each and mutated together to compare the same.

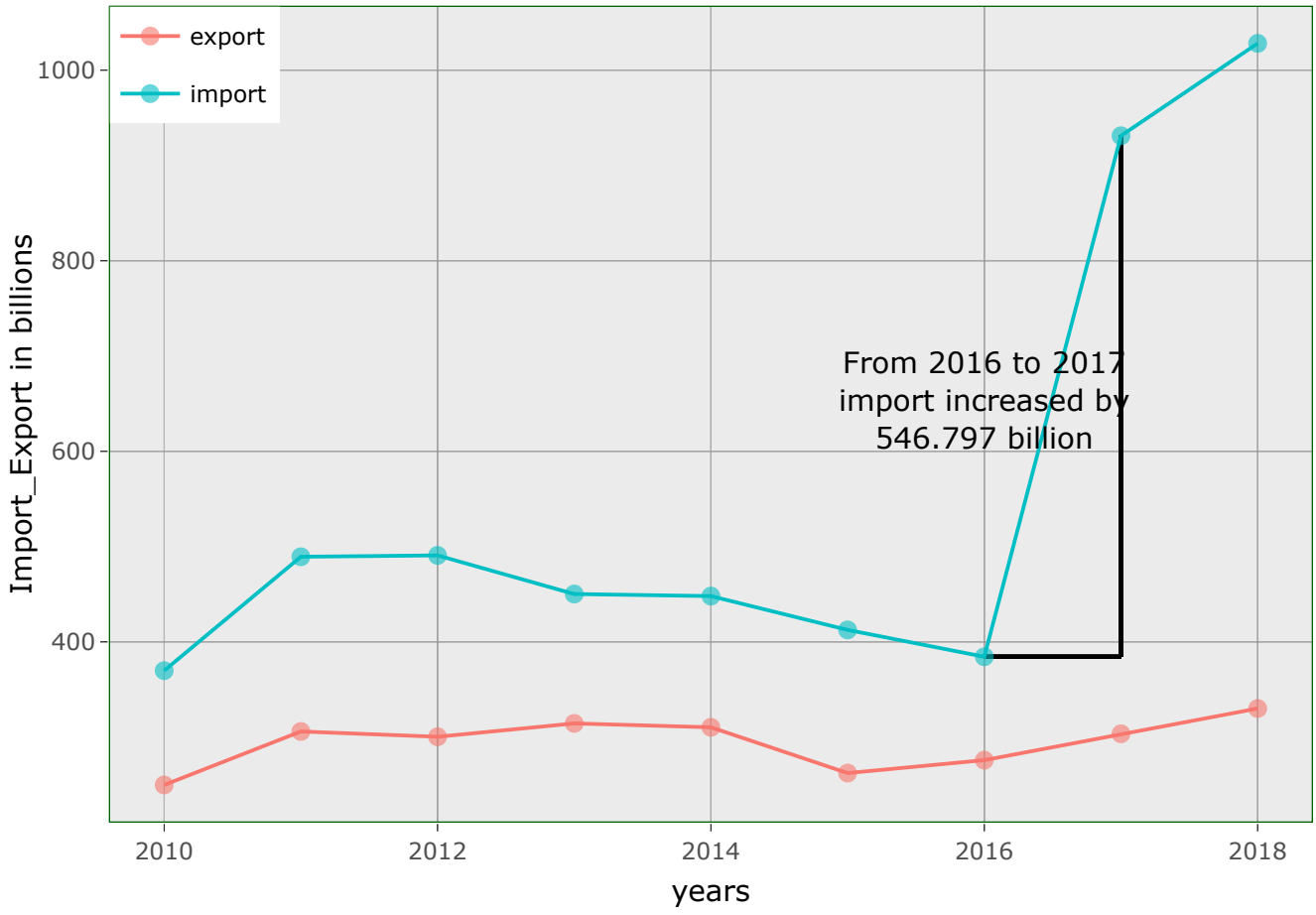
```
summary_Import <- Import %>% group_by(year) %>% summarise(import = sum(value))
summary_Export <- Export %>% group_by(year) %>% summarise(export = sum(value))
summary_Import_Export <- merge(summary_Import,summary_Export,x.by = year, y.by = year) %>% gathe
r('Type','Value',-year)

summary_Import_Export <- summary_Import_Export %>% mutate(V_in_billion = Value/1000)
summary_Import_Export
```

year <dbl>	Type <chr>	Value <dbl>	V_in_billion <dbl>
2010	import	369762.2	369.7622
2011	import	489311.8	489.3118
2012	import	490730.1	490.7301
2013	import	450193.0	450.1930
2014	import	448026.6	448.0266
2015	import	412537.5	412.5375
2016	import	384350.3	384.3503
2017	import	931148.0	931.1480
2018	import	1028142.7	1028.1427
2010	export	249801.2	249.8012

1-10 of 18 rows Previous **1** 2 Next

To summarise the above new dataframe, below the plot demonstates the over all imports and exports of India over years 2010 to 2018

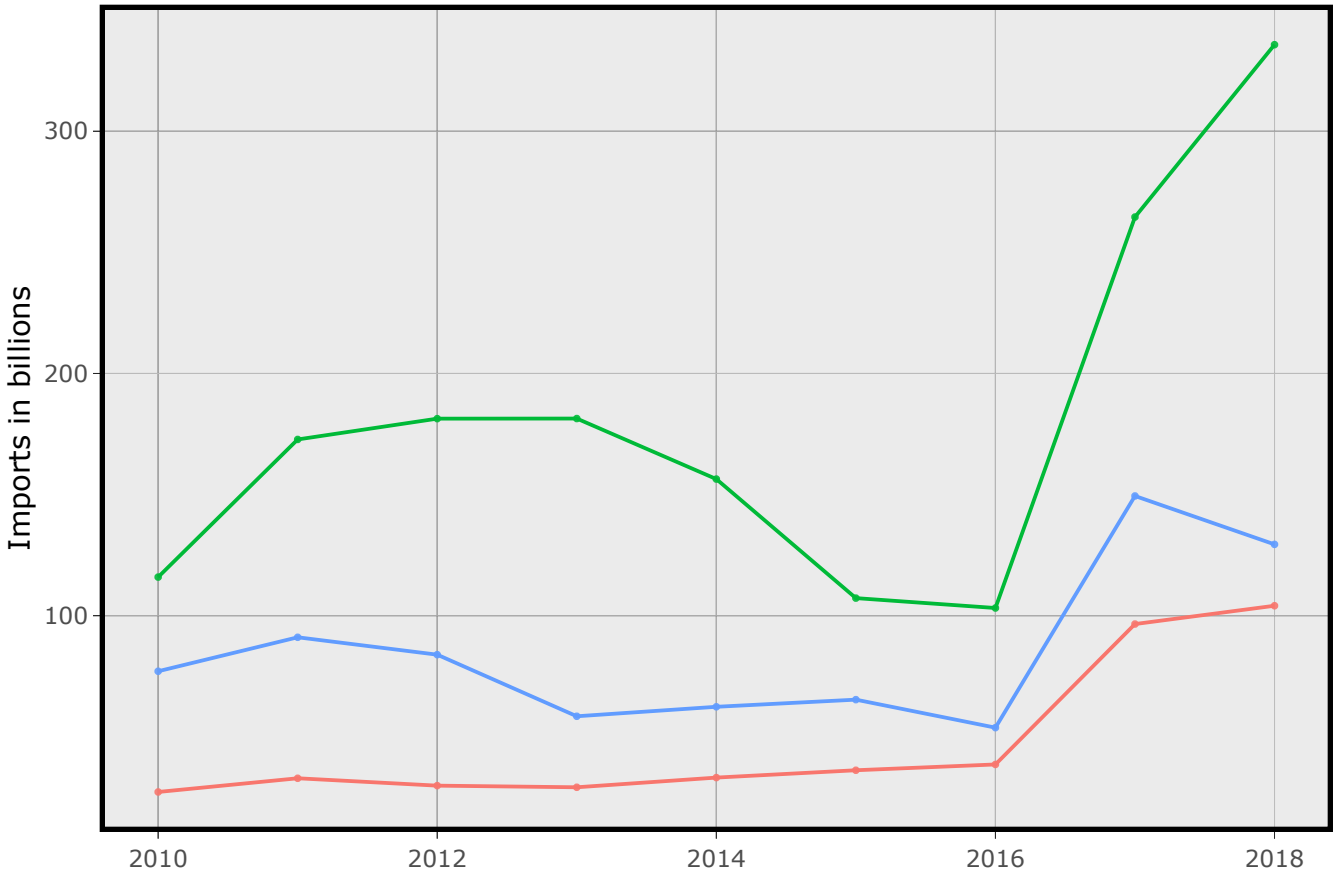
```
Import_Export_G <- ggplot(summary_Import_Export)+
  geom_segment(aes(x = 2017, y = 384350.35/1000, xend= 2017,yend = 931148.0/1000), arrow = arrow
(length = unit(0.1, "inches")))+
  geom_segment(aes(x = 2016, y = 384350.3/1000, xend= 2017,yend = 384350.35/1000), linetype =1)+
    geom_point(aes(x=year,y=V_in_billion,color=Type),size = 2, alpha = 0.6)+
    geom_line(aes(x=year,y=V_in_billion,color=Type))+
  annotate('text',x = 2016,y = 650,
               label = "From 2016 to 2017\nimport increased by\n546.797 billion")+
  labs(x = "years", y="Import_Export in billions")+
  theme(legend.title = element_blank(),
        panel.border = element_rect(colour = "dark green", fill=NA, size=0.5))


ggplotly(Import_Export_G) %>% layout(legend = list(x = 0, y = 1,
                                     bordercolor="white",
                                     borderwidth=0.5))
```



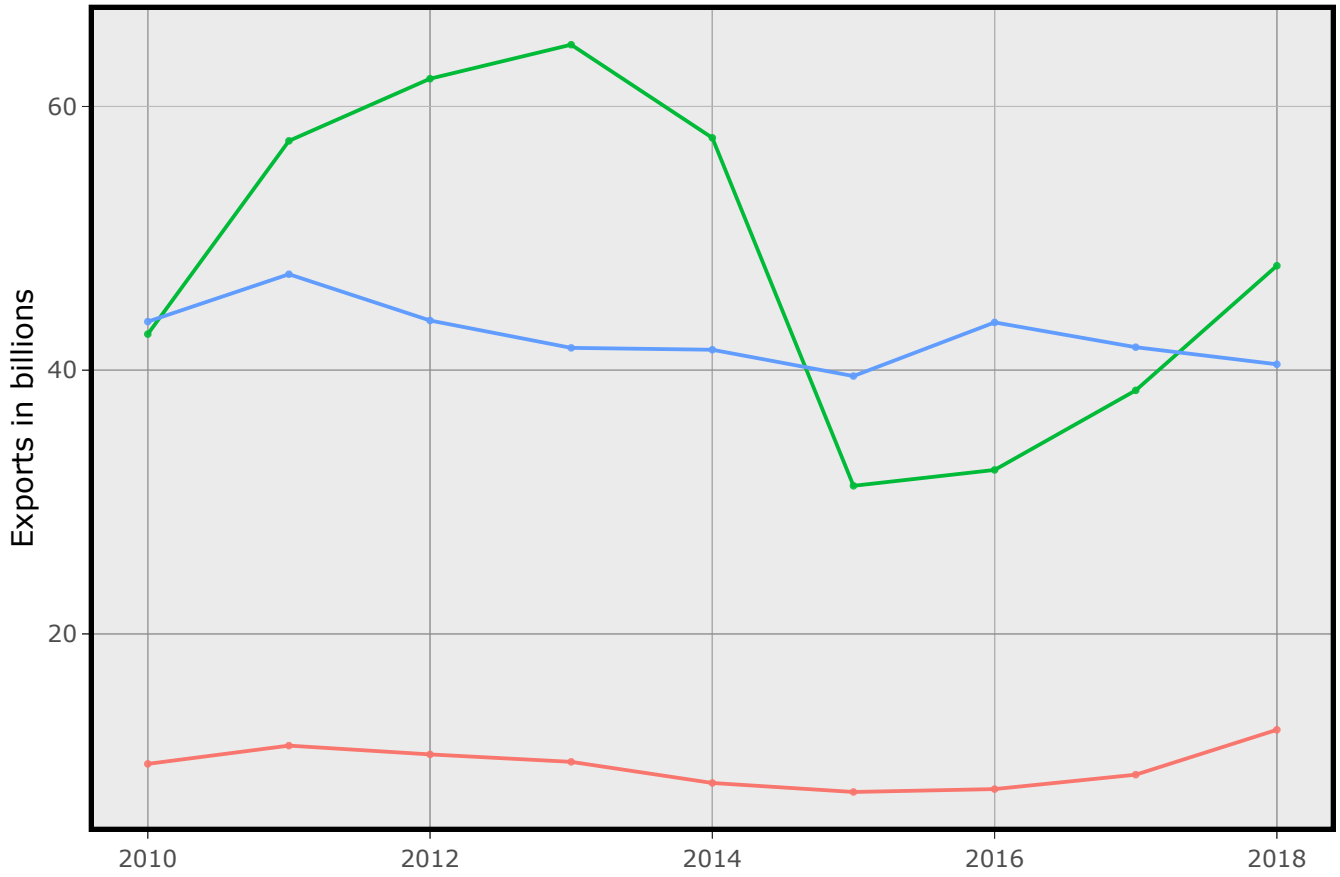
Analysing cont..

Top 3 Imported and Exported Commodity in India over years.

```
Top_Import <- Import %>% group_by(HSCode) %>% summarise(value = sum(value)) %>% top_n( 3,value)
#Top_Import <- merge(Top_Import,HSC_Commodity,by='HSCode')
Top_Import1 <- Import %>% filter(HSCode %in% Top_Import$HSCode) %>%
  group_by(year,HSCode) %>%
  summarise(v_in_billion = sum(value)/1000)

Top_Import1 <- merge(Top_Import1,HSC_Commodity,by='HSCode')


ImportG2 <- ggplot(Top_Import1)+
  geom_point(aes(x=year,y=v_in_billion,color=Commodity_HSCode),size = 0.5, alpha = 0.9)+
  geom_line(aes(x=year,y=v_in_billion,color=Commodity_HSCode))+
  labs(x = "", y="Imports in billions")+
  theme(panel.border = element_rect(colour = "black", fill=NA, size=2))
ggplotly(ImportG2) %>% hide_legend()
```



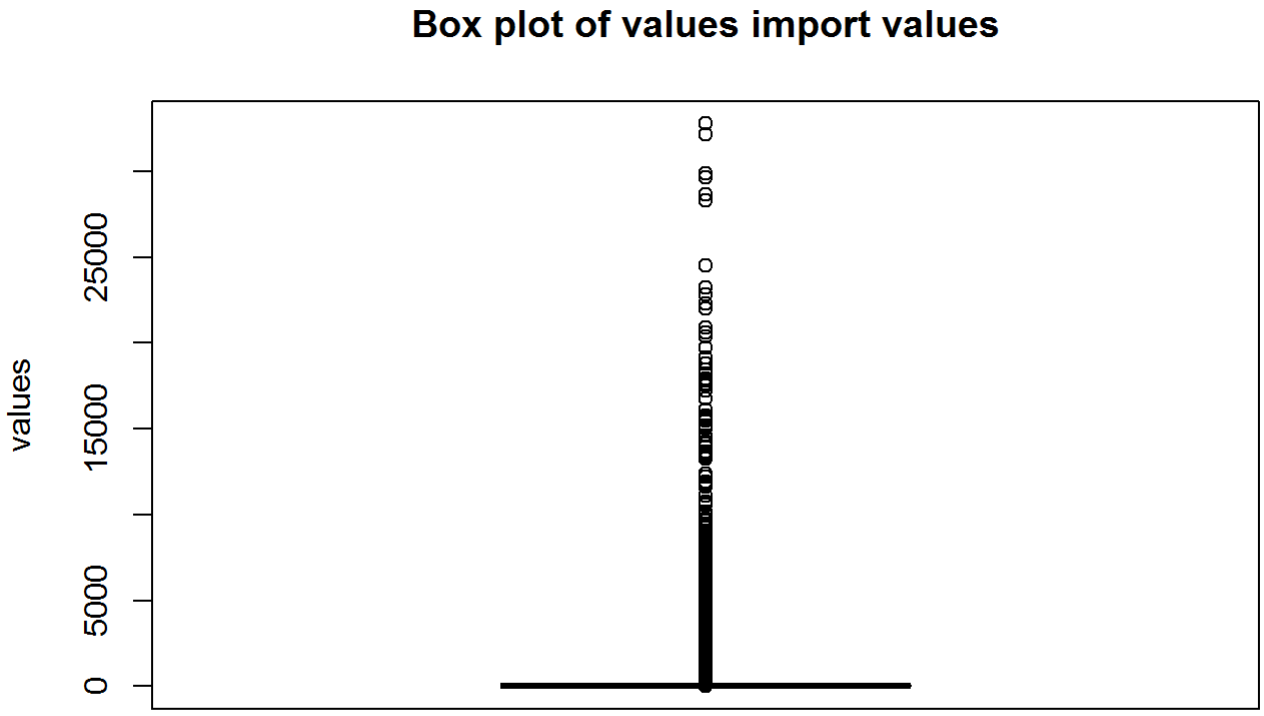
```
Top_Export <- Export %>% group_by(HSCode) %>% summarise(value = sum(value)) %>% top_n( 3,value)
Top_Export2 <- Export %>% filter(HSCode %in% Top_Import$HSCode) %>%
  group_by(year,HSCode) %>%
  summarise(v_in_billion = sum(value)/1000)
Top_Export2 <- merge(Top_Export2,HSC_Commodity,by='HSCode')

ExportG2 <- ggplot(Top_Export2)+
  geom_point(aes(x=year,y=v_in_billion,color=Commodity_HSCode),size = 0.5, alpha = 0.9)+
  geom_line(aes(x=year,y=v_in_billion,color=Commodity_HSCode))+
  labs(x = "", y="Exports in billions")+
  theme(panel.border = element_rect(colour = "black", fill=NA, size=2))

ggplotly(ExportG2) %>% hide_legend()
```



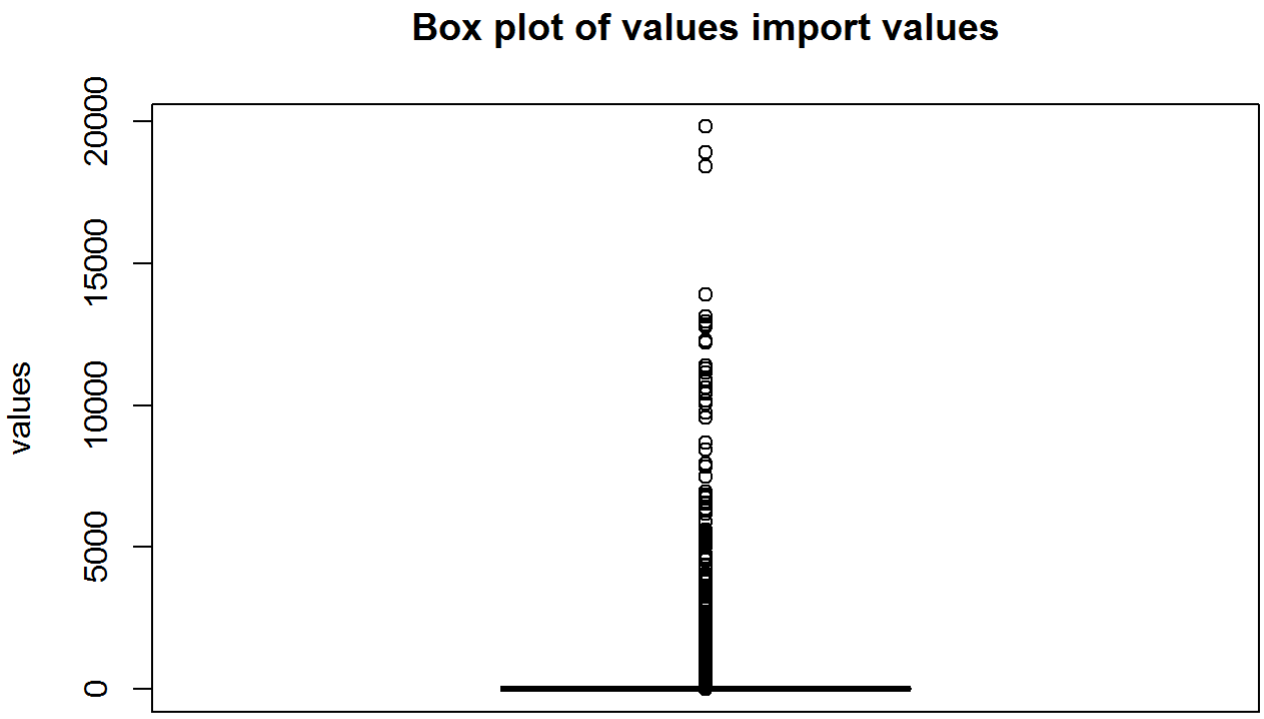
Outliers Detection

Below outliers have been detected in the import and export Values

```
Import$value %>% boxplot(main="Box plot of values import values", ylab="values", col= "grey")
```




```
Export$value %>% boxplot(main="Box plot of values import values", ylab="values", col= "grey")
```



To replace the outliers, Capping/Winsorising is used.

Capping or winsorizing involves replacing the outliers that are not outliers with the nearest neighbors.

```
# Capping is use to replace the outliers.
#function to define cap below is used from module notes and stackoverflow

cap <- function(x){
    quantiles <- quantile( x, c(.05, 0.25, 0.75, .95 ) )
    x[ x < quantiles[2] - 1.5*IQR(x) ] <- quantiles[1]
    x[ x > quantiles[3] + 1.5*IQR(x) ] <- quantiles[4]
    x
}
Import_values_capped <- Import$value %>% cap()
Export_values_capped <- Export$value %>% cap()


#subset using value
import_value_sub <- Import %>% dplyr::select(value)
export_value_sub <- Export %>% dplyr::select(value)

#descriptive statistics
summary(import_value_sub)
```

```
##      value
##  Min.   :    0.00
##  1st Qu.:    0.03
##  Median :    0.38
##  Mean   :   63.29
##  3rd Qu.:    4.91
##  Max.   :32781.57
```

```
summary(export_value_sub)
```

```
##      value
##  Min.   :    0.00
##  1st Qu.:    0.03
##  Median :    0.36
##  Mean   :   21.57
##  3rd Qu.:    3.77
##  Max.   :19805.17
```

```
#apply user defined function "cap" to value_sub
Import_values_capped <- sapply(import_value_sub, FUN = cap)
Export_values_capped <- sapply(export_value_sub, FUN = cap)

#check statistics
summary(Import_values_capped)
```

```
##      value
##  Min.   :  0.00
##  1st Qu.:  0.03
##  Median :  0.38
##  Mean   : 24.19
##  3rd Qu.:  4.91
##  Max.   :131.51
```

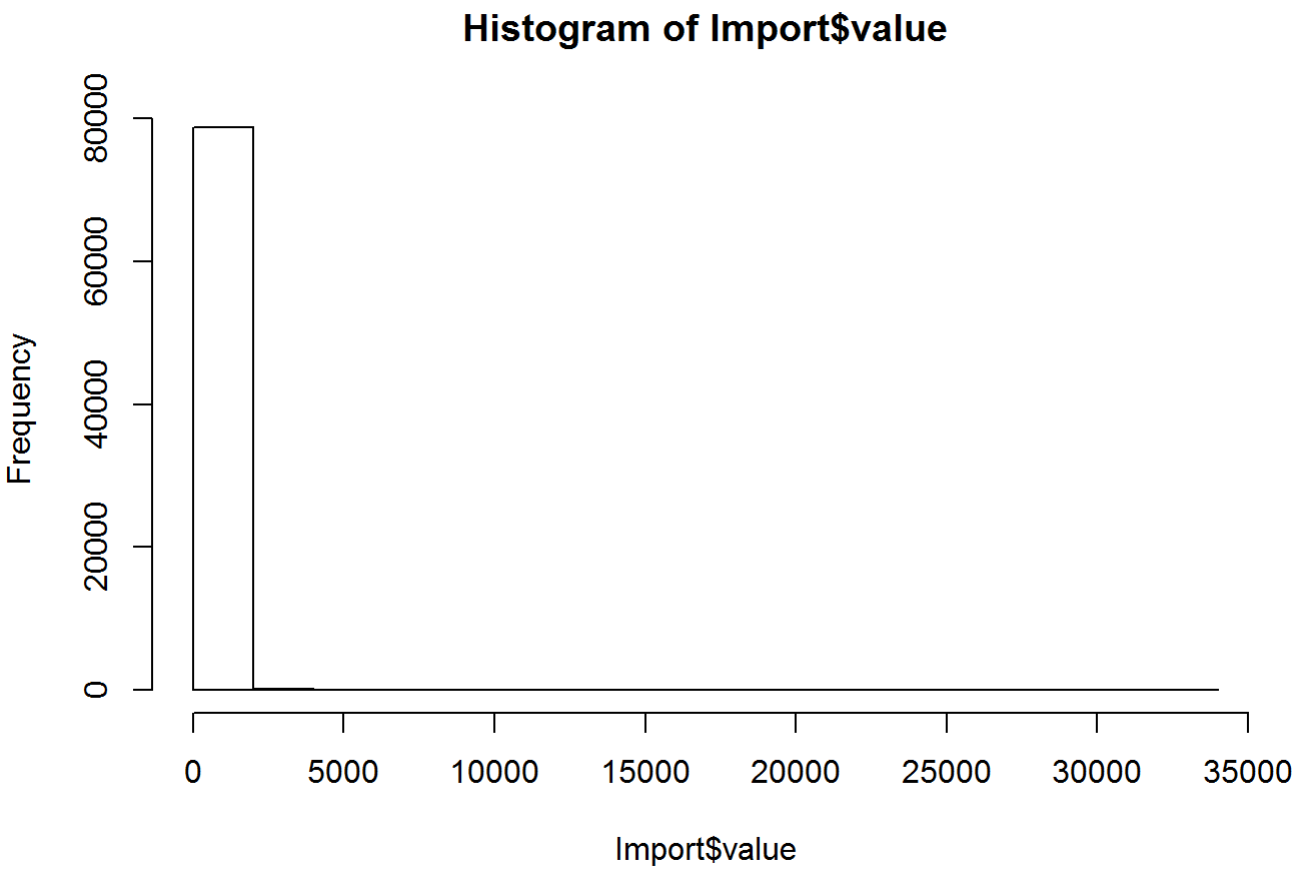
```
summary(Export_values_capped)
```

```
##      value
##  Min.   : 0.00
##  1st Qu.: 0.03
##  Median : 0.36
##  Mean   :10.94
##  3rd Qu.: 3.77
##  Max.   :60.83
```

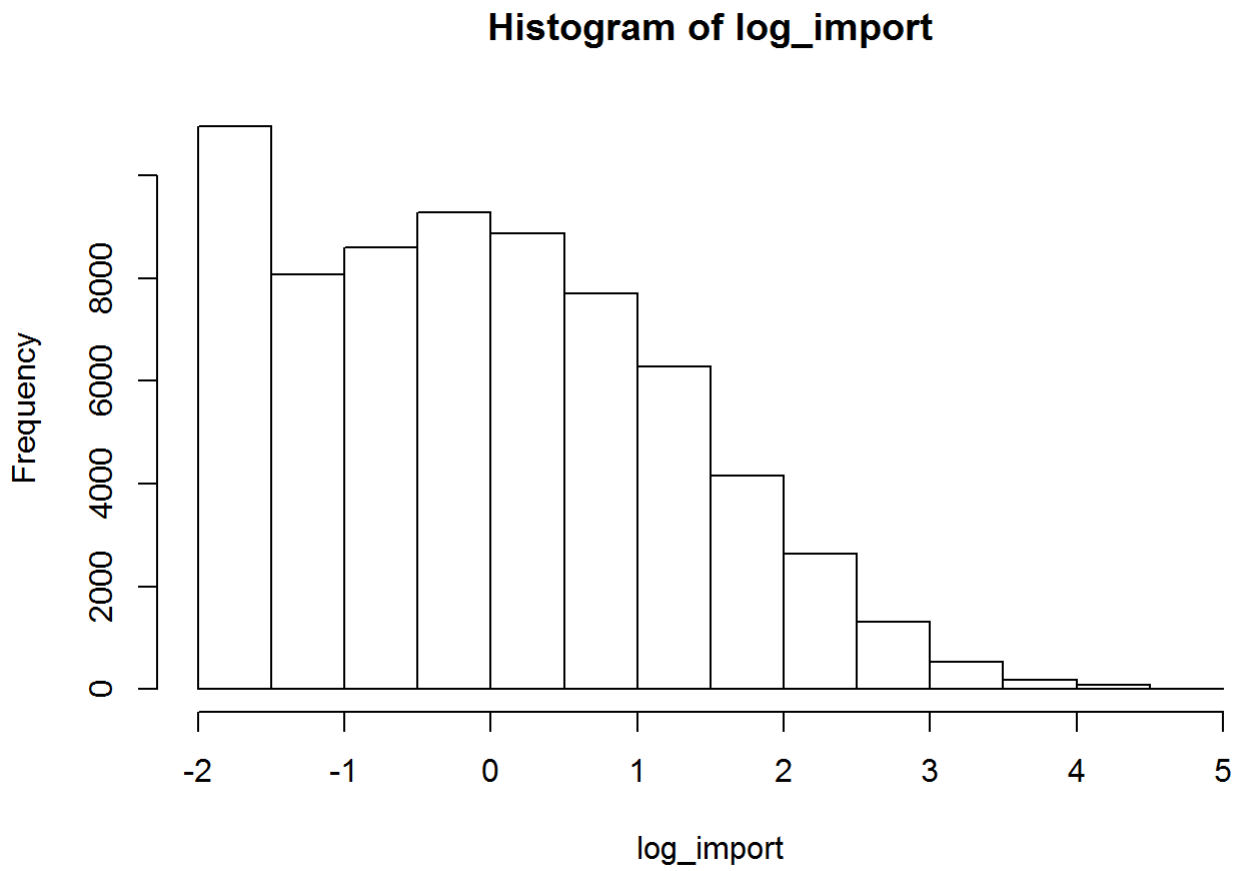
Transforming

- As the data set have high values (in billions) both log10 and ln have been performed as there are no zero or negative values to check which suits the best

```
#The Log Tranformation
#Hypothetical data on the Import data

hist(Import$value)
```

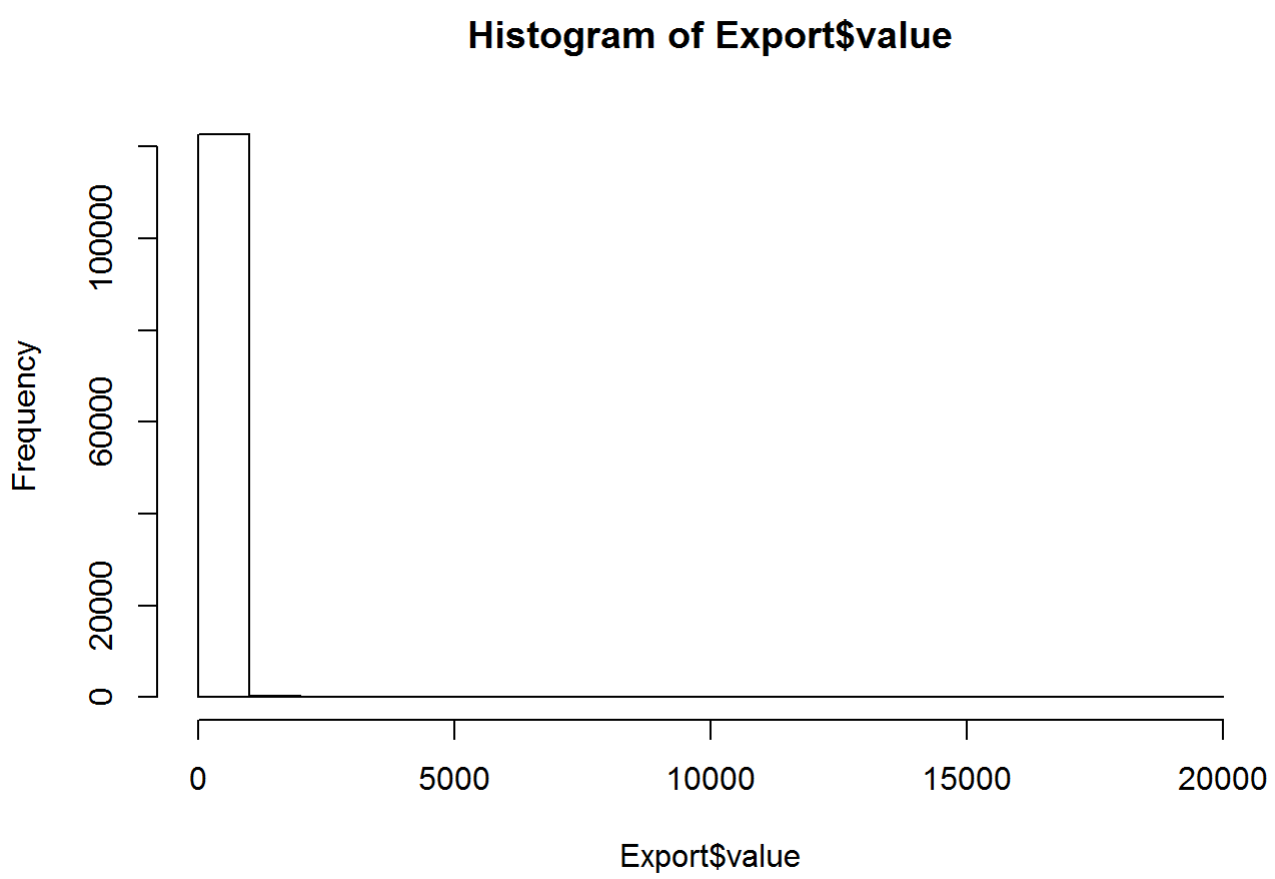



```
#we apply the log10() transformation

log_import <- log10(Import$value)
hist(log_import)
```

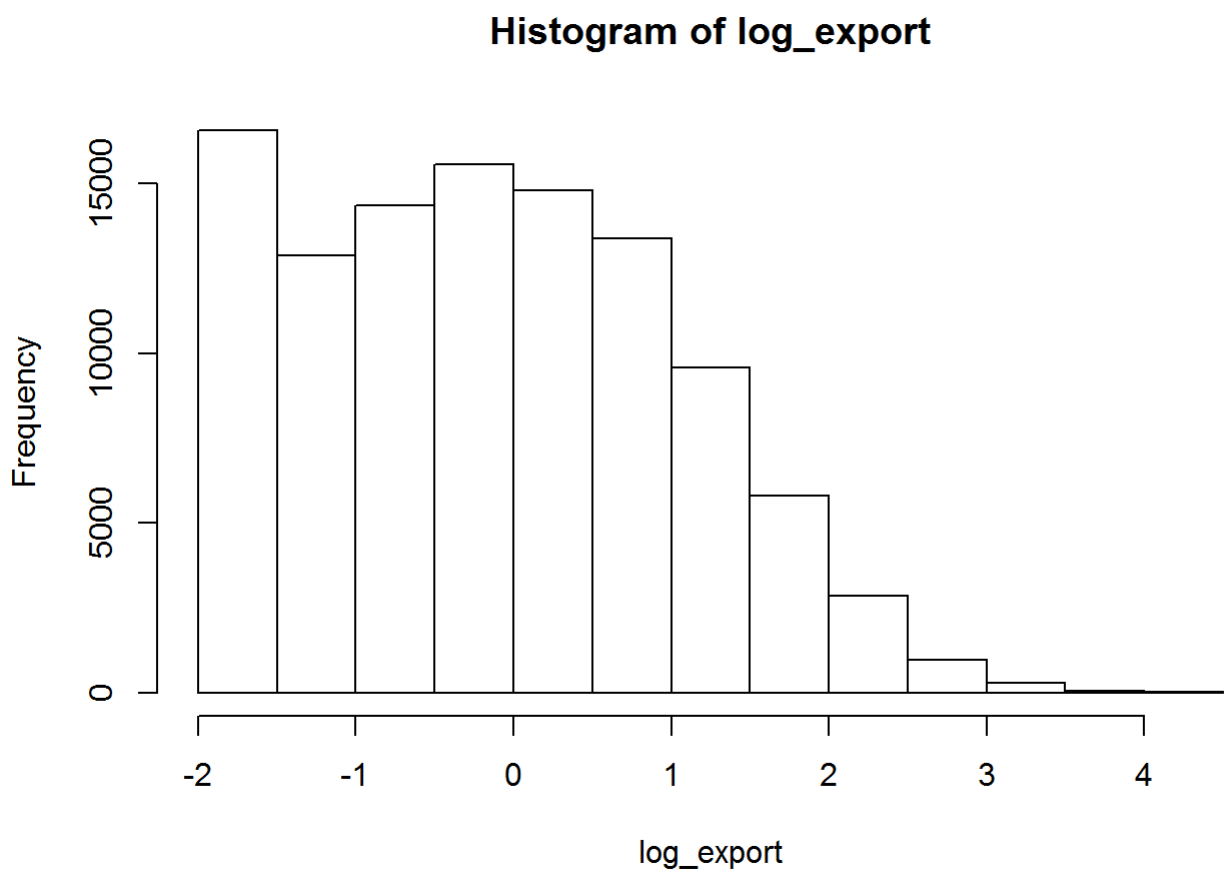



```
#Lets check ln transformation
ln_import <- log(Import$value)
hist(ln_import)
```

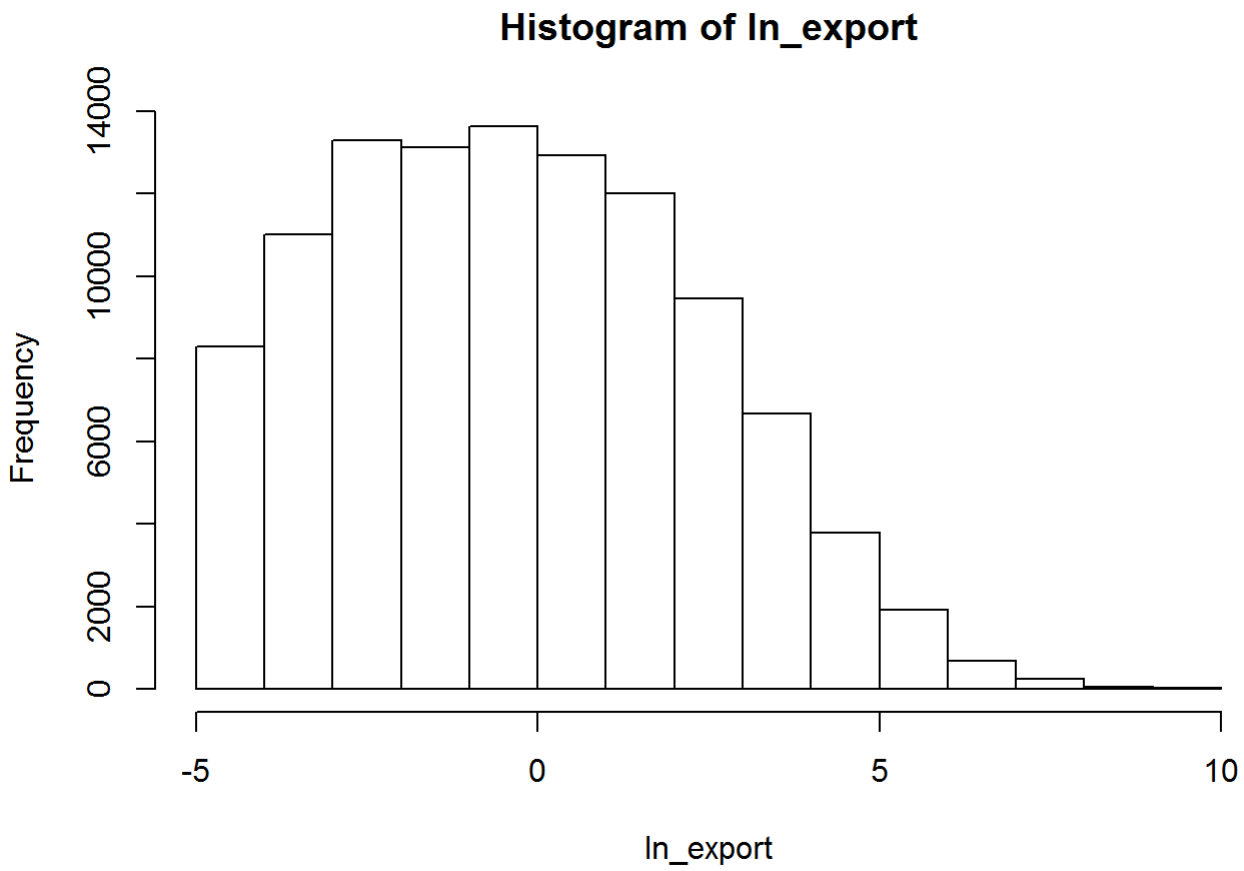

```
#The Log Tranformation
#Hypothetical data on the Export data

hist(Export$value)
```




```
#we apply the log10() transformation

log_export <- log10(Export$value)
hist(log_export)
```




```
#Lets check ln transformation
ln_export <- log(Export$value)
hist(ln_export)
```



we see above that ln shape is slightly less right skewed than ln transformation for exports. Where as for Imports log10 is preferred.