Exploratory Data Analysis

Varun Bansal

2023-01-18

Setting the work directory.

Loading and attaching all the necessary packages.

```
#Load packages

if(!require(tinytex)){
  install.packages("tinytex")
  }
```

```
## Loading required package: tinytex
```

```
library("tinytex")

if (!require(dplyr)) {
  install.packages("dplyr")
  library(dplyr)
}
```

```
## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Basic Manipulation

Reading the text file and changing it to a data frame

```
df <- read.csv("Data_Centers.txt", header = TRUE, sep = ",")
```

Append the initials to all variables in the data frame

```
df_VB <- df
colnames(df_VB) <- paste(colnames(df_VB), "VB", sep = "_")
head(df_VB, 5)
```

```
##     Manufacturer_VB Server_VB     DC_VB SMBR_VB SMBT_VB Conn_VB
## 1              Lled    MG9696  Waterloo  102479   43473    6625
## 2            Ovonel    RX8838  Waterloo  103678   62534    7580
## 3              Lled    MB3406 Cambridge  102003   35916    5957
## 4              Lled    MB3406 Kitchener   98889   40245    6120
## 5 Highway-Passenger    DF6726 Cambridge  104907   25422    5839
```

Changing each character variable to a factor variable

```
# Changing each character variable to factor variable
df_VB <- as.data.frame(unclass(df_VB), stringsAsFactors = TRUE)

str(df_VB)
```

```
## 'data.frame':    90000 obs. of  6 variables:
##  $ Manufacturer_VB: Factor w/ 3 levels "Highway-Passenger",..: 2 3 2 2 1 1 1 2 2 3 ...
##  $ Server_VB      : Factor w/ 6 levels "DF6726","DJ3756",..: 4 6 3 3 1 1 1 3 4 5 ...
##  $ DC_VB          : Factor w/ 5 levels "Bridgeport","Cambridge",..: 5 5 2 4 2 4 3 5 2 4 ...
##  $ SMBR_VB        : int  102479 103678 102003 98889 104907 102659 106037 101077 101662 90592 ...
##  $ SMBT_VB        : int  43473 62534 35916 40245 25422 53168 59596 64132 42928 61989 ...
##  $ Conn_VB        : int  6625 7580 5957 6120 5839 7076 7258 7391 6608 6671 ...
```

Checking dimensions of the dataset

```
dim(df_VB)
```

```
## [1] 90000     6
```

The dataset is of dimension 90000 x 6 (90000 rows and 6 columns)

Summarizing Data

Means and Standard Deviations for Server Message Blocks Received.

```r
mean_smbr_VB <- mean(df_VB$SMBR_VB)
paste("Mean of Server Message Blocks Received:", mean_smbr_VB)
```

```
## [1] "Mean of Server Message Blocks Received: 100017.478544444"
```

```r
sd_smbr_VB <- sd(df_VB$SMBR_VB)
paste("Standard Deviation of Server Message Blocks Received:", sd_smbr_VB)
```

```
## [1] "Standard Deviation of Server Message Blocks Received: 10002.4583223398"
```

Calculating the coefficient of variation.

```r
cv_smbr_VB <- round((sd_smbr_VB / mean_smbr_VB) * 100, 3)
paste("Coefficient of variation of Server Message Blocks Received:", cv_smbr_VB)
```

```
## [1] "Coefficient of variation of Server Message Blocks Received: 10.001"
```

Calculating the mean and standard deviation for Server Message Blocks Transmitted.

```r
mean_smbt_VB <- mean(df_VB$SMBT_VB)
paste("Mean of Server Message Blocks Transmitted:", mean_smbt_VB)
```

```
## [1] "Mean of Server Message Blocks Transmitted: 49966.0049333333"
```

```r
sd_smbt_VB <- sd(df_VB$SMBT_VB)
paste("Standard Deviation of Server Message Blocks Transmitted:", sd_smbt_VB)
```

```
## [1] "Standard Deviation of Server Message Blocks Transmitted: 10024.435354702"
```

Calculating the coefficient of variation (rounded to 3 decimal places).

```r
cv_smbt_VB <- round((sd_smbt_VB / mean_smbt_VB) * 100, 3)
paste("Coefficient of variation of Server Message Blocks Transmitted:", cv_smbt_VB)
```

```
## [1] "Coefficient of variation of Server Message Blocks Transmitted: 20.063"
```

Looking for which variable has more variation (SMBT or SMBR)

```r
max(cv_smbr_VB,cv_smbt_VB)
```

```
## [1] 20.063
```

SMBT has more variation, as coefficient of variation of SMBT is double the coefficient of variation of SMBR.

Organizing Data

Summary Table

Table showing the average Server Message Blocks Transmitted by Manufacturer.

```
avg_smbt_df_VB <- aggregate(df_VB$SMBT_VB, by=list(df_VB$Manufacturer_VB),
                            FUN=function(x) round(mean(x), 2))
colnames(avg_smbt_df_VB) <- c("Manufacturer", "AverageSMBT")
avg_smbt_df_VB
```

```
##        Manufacturer AverageSMBT
## 1 Highway-Passenger    49916.14
## 2              Lled    50008.12
## 3            Ovonel    49973.76
```

Looking for which Manufacturer's Servers have, on average, transmitted the most server message blocks.

```
head( avg_smbt_df_VB[order(avg_smbt_df_VB$AverageSMBT, decreasing = TRUE), ], 1)
```

```
##   Manufacturer AverageSMBT
## 2         Lled    50008.12
```

Lled has transmitted the most server message blocks on an average.

Cross Tabulation

Table counting all Servers by Data Centre.

```
table_servers_by_DC_VB <- table(df_VB$Server_VB, df_VB$DC_VB)
count_servers_by_DC_VB <-margin.table(table_servers_by_DC_VB, 2)
count_servers_by_DC_VB
```

```
##
## Bridgeport  Cambridge     Elmira  Kitchener   Waterloo
##       8945      13582      17851      22403      27219
```

Changing the table to show the percentage of each Server in each Data Centre.

```
percentage_servers_by_DC_VB <- round((prop.table(count_servers_by_DC_VB))*100,3)
percentage_servers_by_DC_VB
```

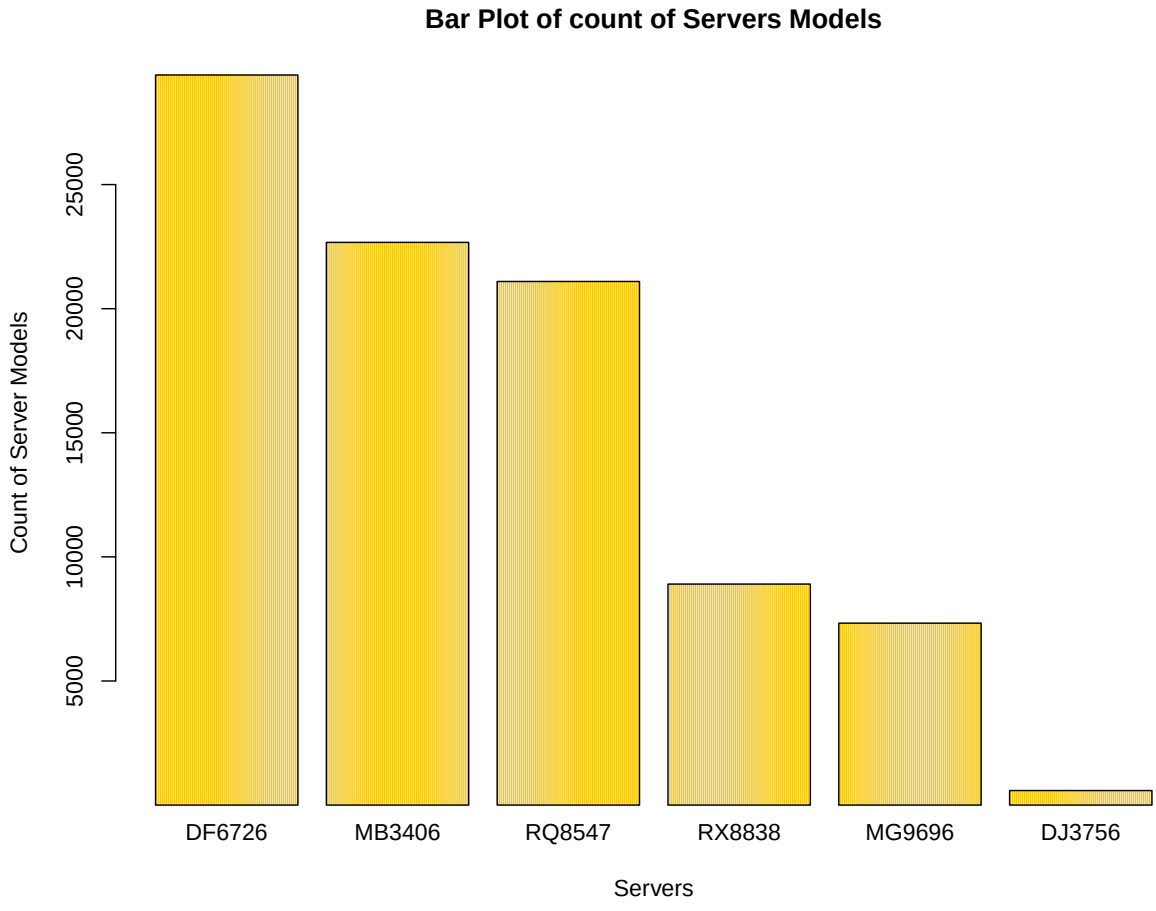
```
##
## Bridgeport  Cambridge     Elmira  Kitchener   Waterloo
##      9.939     15.091     19.834     24.892     30.243
```

Data Visualization

Bar Plot

Bar plot of count of Servers Models.

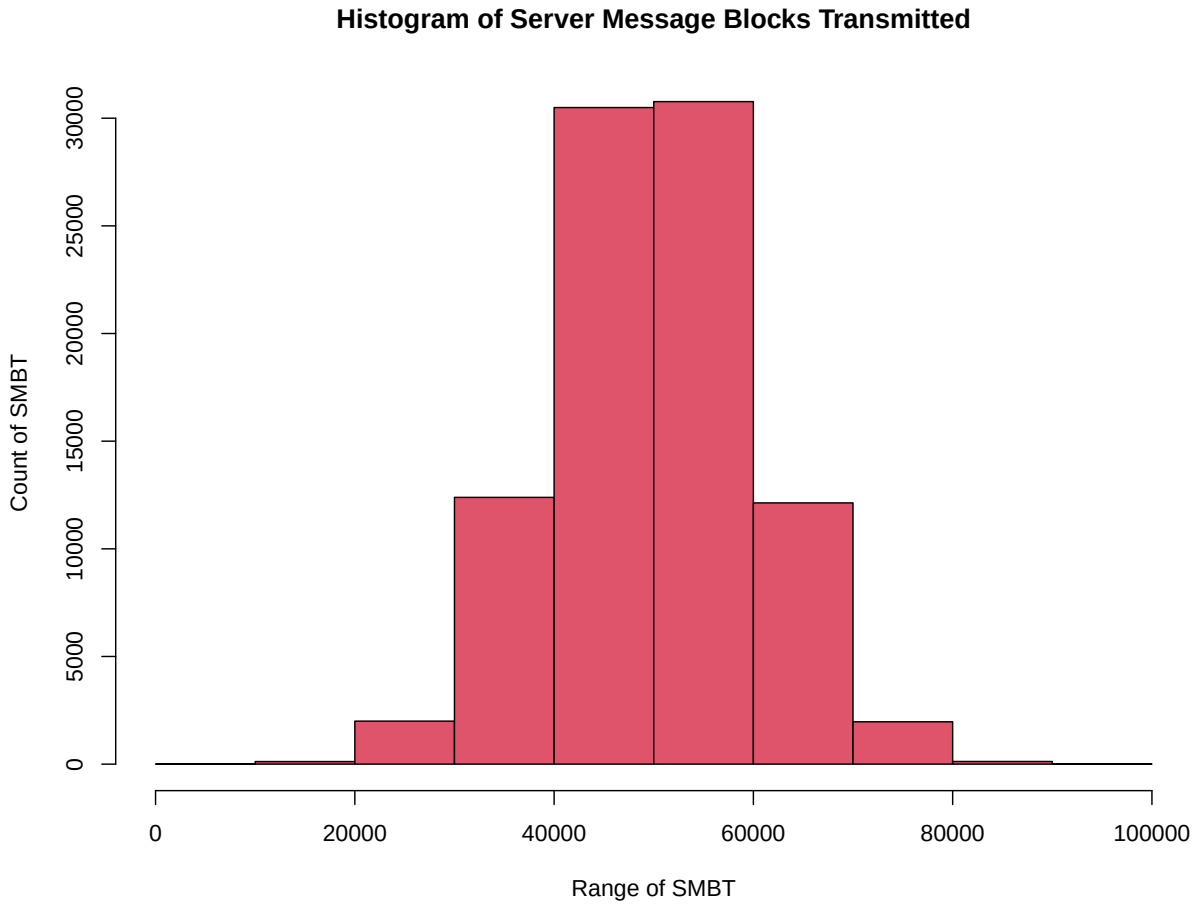
```
# Counting no. of servers for each server.
df_table_VB <- table(df_VB$Server_VB)

# Ordering by highest count of Server Model
df_table_VB <- df_table_VB[order(df_table_VB,decreasing=TRUE)]

# Creating the bar plot of count of Servers Models
barplot(df_table_VB,
        col=55,
        density = 70, angle = 90,
        main="Bar Plot of count of Servers Models",
        xlab = "Servers",
        ylab = "Count of Server Models",
        ylim = range(df_table_VB)
        )
```



Histogram

Histogram of Server Message Blocks Transmitted.

```
hist(df_VB$SMBT_VB,
     col=2,
     breaks=10,
     xlab="Range of SMBT",
     ylab="Count of SMBT",
     main="Histogram of Server Message Blocks Transmitted")
```

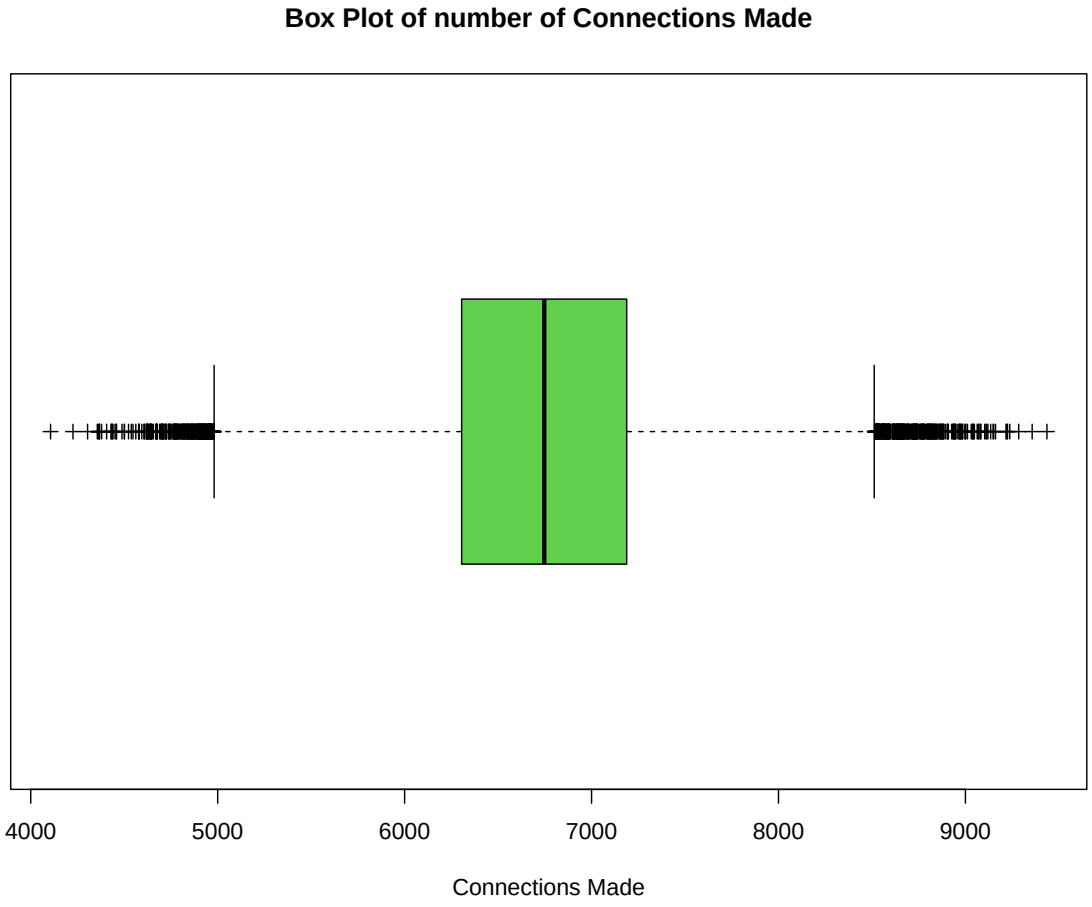


Box plot

Horizontal box plot of number of Connections Made.

```
boxplot(df_VB$Conn_VB,
        main="Box Plot of number of Connections Made",
        xlab="Connections Made",
        col=19,
```

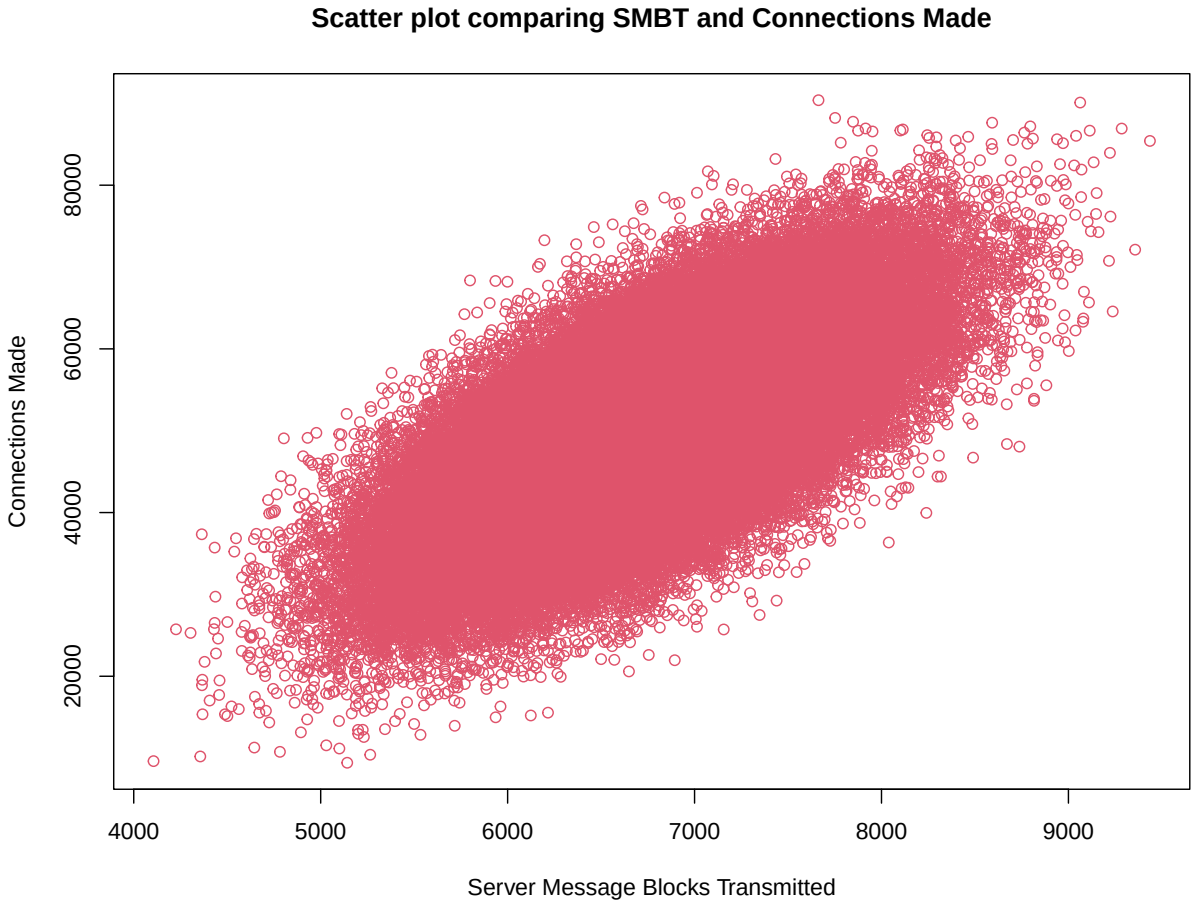
```
        horizontal=TRUE,
        pch=3)
```



Scatter Plot

Scatter plot comparing Server Message Blocks Transmitted and Connections Made.

```
plot(df_VB$SMBT_VB ~ df_VB$Conn_VB,
     data=df_VB,
     col=2,
     pch=1,
     main="Scatter plot comparing SMBT and Connections Made",
     xlab="Server Message Blocks Transmitted",
     ylab="Connections Made")
```



Server Message Blocks Transmitted and Connections Made have a linear relationship. It looks like it has a moderate positive correlation.