Jackson Week 13 IP Part 2

Jackson Kyalo

9/3/2021

Define the Question

Kira Plastinina is a Russian brand that is sold through a defunct chain of retail stores in Russia, Ukraine, Kazakhstan, Belarus, China, Philippines, and Armenia. The brand's Sales and Marketing team would like to understand their customer's behavior from data that they have collected over the past year. More specifically, they would like to learn the characteristics of customer groups.

The metric for success

This project will be successful if we are able to determine which individuals are most likely to click on the ads.

The Outline context

The number of clicks an ad has helps understand how well the ad is being received by its audience. Ads that are targeted to the right audience receive the highest number of clicks. In our case determining the best audience for the ads will help company grow as well as increase the number of clicks and reach.

Experimental design

1. Define the Questions.
2. Import, load and preview the data.
3. Data Cleaning.
4. Data Analysis.
5. Conclusion and Recommendation.

Importing the libraries

```
#Import the data library
library(data.table)

## Warning: package 'data.table' was built under R version 4.0.5

library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tibble' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'purrr' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

## Warning: package 'stringr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()

library(ggplot2)
library(moments)
```

Load the dataset

```
#Load our data
ecomm=read.csv('http://bit.ly/EcommerceCustomersDataset')
```

Preview the data

```
# preview the head
head(ecomm)

##   Administrative Administrative_Duration Informational
Informational_Duration
## 1              0                       0             0
0
## 2              0                       0             0
0
## 3              0                      -1             0
```

```
-1
## 4              0                       0             0
0
## 5              0                       0             0
0
## 6              0                       0             0
0
##   ProductRelated ProductRelated_Duration BounceRates ExitRates PageValues
## 1              1                0.000000  0.20000000 0.2000000          0
## 2              2               64.000000  0.00000000 0.1000000          0
## 3              1               -1.000000  0.20000000 0.2000000          0
## 4              2                2.666667  0.05000000 0.1400000          0
## 5             10              627.500000  0.02000000 0.0500000          0
## 6             19              154.216667  0.01578947 0.0245614          0
##   SpecialDay Month OperatingSystems Browser Region TrafficType
## 1          0   Feb                1       1      1           1
## 2          0   Feb                2       2      1           2
## 3          0   Feb                4       1      9           3
## 4          0   Feb                3       2      2           4
## 5          0   Feb                3       3      1           4
## 6          0   Feb                2       2      1           3
##         VisitorType Weekend Revenue
## 1 Returning_Visitor   FALSE   FALSE
## 2 Returning_Visitor   FALSE   FALSE
## 3 Returning_Visitor   FALSE   FALSE
## 4 Returning_Visitor   FALSE   FALSE
## 5 Returning_Visitor    TRUE   FALSE
## 6 Returning_Visitor   FALSE   FALSE
```

Preview tail

```
tail(ecomm)

##       Administrative Administrative_Duration Informational
## 12325              0                       0             1
## 12326              3                     145             0
## 12327              0                       0             0
## 12328              0                       0             0
## 12329              4                      75             0
## 12330              0                       0             0
##       Informational_Duration ProductRelated ProductRelated_Duration
BounceRates
## 12325                      0             16                 503.000
0.000000000
## 12326                      0             53                1783.792
0.007142857
## 12327                      0              5                 465.750
0.000000000
## 12328                      0              6                 184.250
0.083333333
## 12329                      0             15                 346.000
```

```
0.000000000
## 12330                      0              3                  21.250
0.000000000
##        ExitRates PageValues SpecialDay Month OperatingSystems Browser
Region
## 12325 0.03764706    0.00000          0   Nov                2       2
1
## 12326 0.02903061   12.24172          0   Dec                4       6
1
## 12327 0.02133333    0.00000          0   Nov                3       2
1
## 12328 0.08666667    0.00000          0   Nov                3       2
1
## 12329 0.02105263    0.00000          0   Nov                2       2
3
## 12330 0.06666667    0.00000          0   Nov                3       2
1
##       TrafficType       VisitorType Weekend Revenue
## 12325           1 Returning_Visitor   FALSE   FALSE
## 12326           1 Returning_Visitor    TRUE   FALSE
## 12327           8 Returning_Visitor    TRUE   FALSE
## 12328          13 Returning_Visitor    TRUE   FALSE
## 12329          11 Returning_Visitor   FALSE   FALSE
## 12330           2       New_Visitor    TRUE   FALSE
```

Check the info

```
str(ecomm)
```

```
## 'data.frame':    12330 obs. of  18 variables:
##  $ Administrative         : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Administrative_Duration: num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ Informational          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration : num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ ProductRelated         : int  1 2 1 2 10 19 1 1 2 3 ...
##  $ ProductRelated_Duration: num  0 64 -1 2.67 627.5 ...
##  $ BounceRates            : num  0.2 0 0.2 0.05 0.02 ...
##  $ ExitRates              : num  0.2 0.1 0.2 0.14 0.05 ...
##  $ PageValues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialDay             : num  0 0 0 0 0 0 0.4 0 0.8 0.4 ...
##  $ Month                  : chr  "Feb" "Feb" "Feb" "Feb" ...
##  $ OperatingSystems       : int  1 2 4 3 3 2 2 1 2 2 ...
##  $ Browser                : int  1 2 1 2 3 2 4 2 2 4 ...
##  $ Region                 : int  1 1 9 2 1 1 3 1 2 1 ...
##  $ TrafficType            : int  1 2 3 4 4 3 3 5 3 2 ...
##  $ VisitorType            : chr  "Returning_Visitor" "Returning_Visitor"
"Returning_Visitor" "Returning_Visitor" ...
##  $ Weekend                : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ Revenue                : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

Check the shape

```
dim(ecomm)

## [1] 12330    18
```

Our code has 1000 rows and 10 columns

Data Cleaning

Missing values

```
#check for missing values
sum(is.na(ecomm))

## [1] 112
```

Our data has 112 missing values

```
#check the missing values in each column
colSums(is.na(ecomm))

##          Administrative Administrative_Duration           Informational
##                      14                      14                      14
##  Informational_Duration          ProductRelated ProductRelated_Duration
##                      14                      14                      14
##             BounceRates               ExitRates              PageValues
##                      14                      14                       0
##              SpecialDay                   Month        OperatingSystems
##                       0                       0                       0
##                 Browser                  Region             TrafficType
##                       0                       0                       0
##             VisitorType                 Weekend                 Revenue
##                       0                       0                       0

#We shall drop the missing values in each columns
df <- na.omit(ecomm)
colSums(is.na(df))

##          Administrative Administrative_Duration           Informational
##                       0                       0                       0
##  Informational_Duration          ProductRelated ProductRelated_Duration
##                       0                       0                       0
##             BounceRates               ExitRates              PageValues
##                       0                       0                       0
##              SpecialDay                   Month        OperatingSystems
##                       0                       0                       0
##                 Browser                  Region             TrafficType
##                       0                       0                       0
##             VisitorType                 Weekend                 Revenue
##                       0                       0                       0
```

Duplicates

```
#Check for duplicates
sum(duplicated(df))

## [1] 117
```

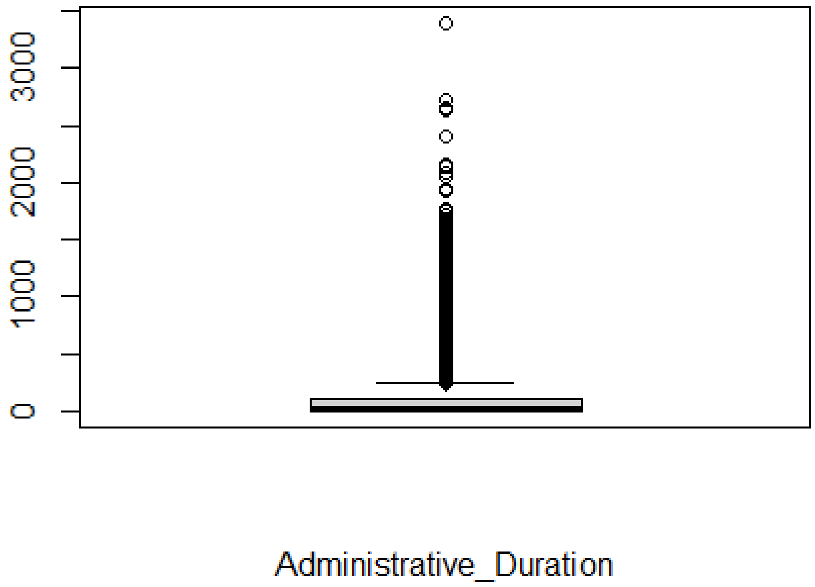
Our data has 117 duplicated rows. We shall drop all duplicates by selecting only the unique values

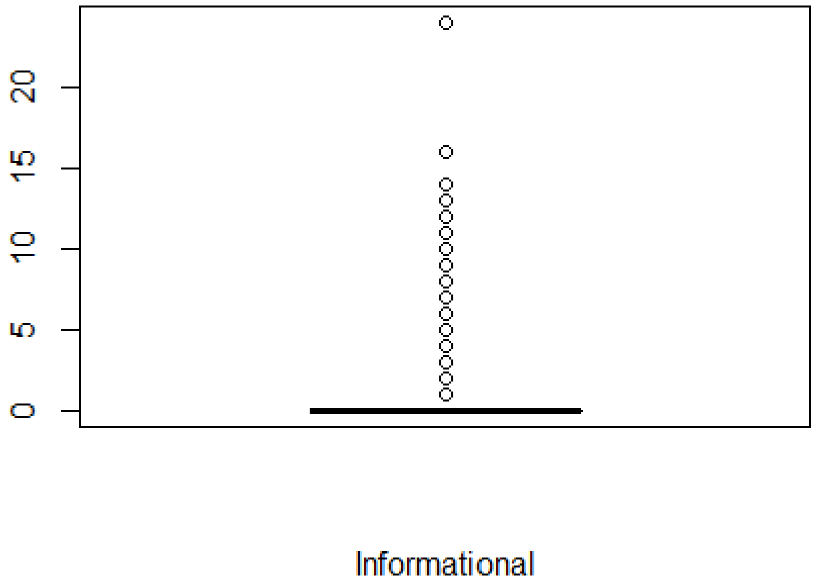
```
#selecting the unique values
df_new <-unique(df)
sum(duplicated(df_new))

## [1] 0

### Identify numeric cols
nums <- unlist(lapply(df_new, is.numeric))
y<- colnames(df_new[nums])
y

##  [1] "Administrative"          "Administrative_Duration"
##  [3] "Informational"           "Informational_Duration"
##  [5] "ProductRelated"          "ProductRelated_Duration"
##  [7] "BounceRates"             "ExitRates"
##  [9] "PageValues"              "SpecialDay"
## [11] "OperatingSystems"        "Browser"
## [13] "Region"                  "TrafficType"
```

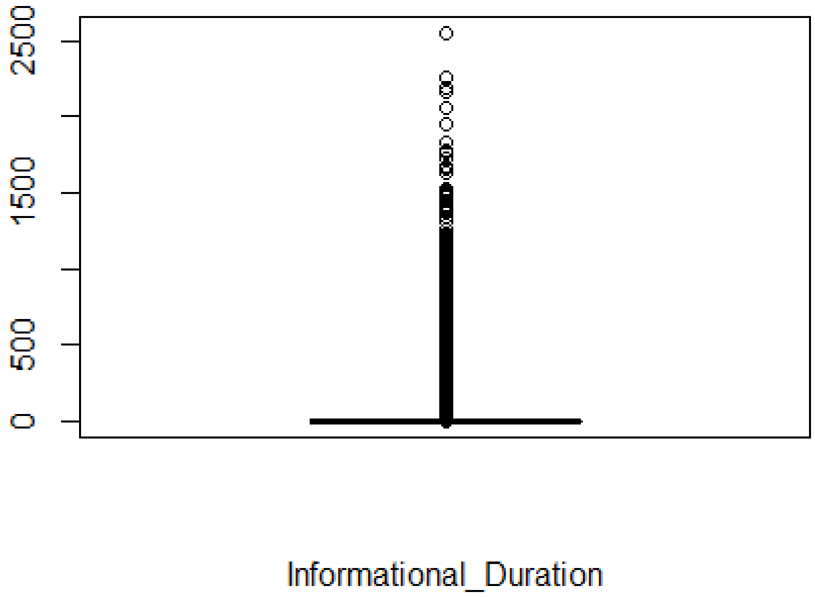
Check fo outliers

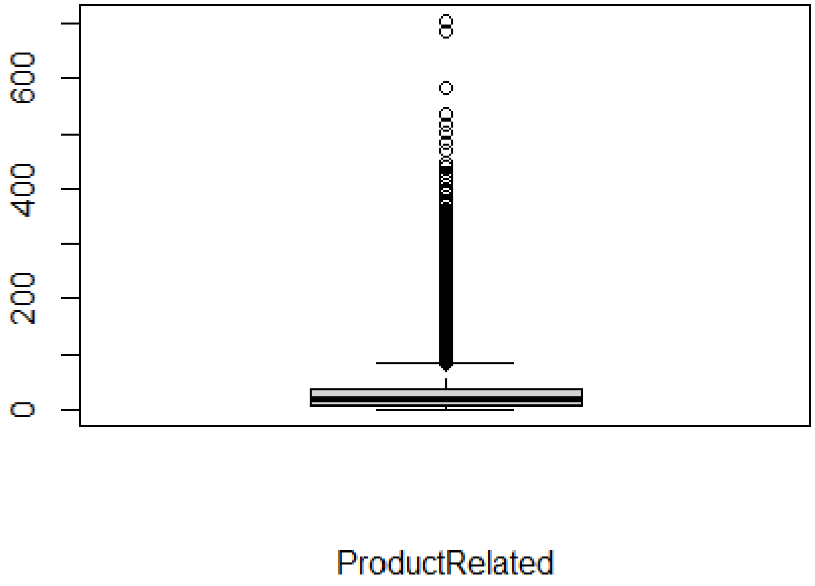
```
#Create a dataframe of numeric cols
num <-df_new[y]
head(num)

##   Administrative Administrative_Duration Informational
Informational_Duration
## 1              0                       0             0
0
## 2              0                       0             0
0
## 3              0                      -1             0
-1
## 4              0                       0             0
0
## 5              0                       0             0
0
## 6              0                       0             0
0
##   ProductRelated ProductRelated_Duration BounceRates ExitRates PageValues
## 1              1                0.000000  0.20000000 0.2000000          0
## 2              2               64.000000  0.00000000 0.1000000          0
## 3              1               -1.000000  0.20000000 0.2000000          0
## 4              2                2.666667  0.05000000 0.1400000          0
## 5             10              627.500000  0.02000000 0.0500000          0
## 6             19              154.216667  0.01578947 0.0245614          0
##   SpecialDay OperatingSystems Browser Region TrafficType
## 1          0                1       1      1           1
## 2          0                2       2      1           2
## 3          0                4       1      9           3
## 4          0                3       2      2           4
## 5          0                3       3      1           4
## 6          0                2       2      1           3
```

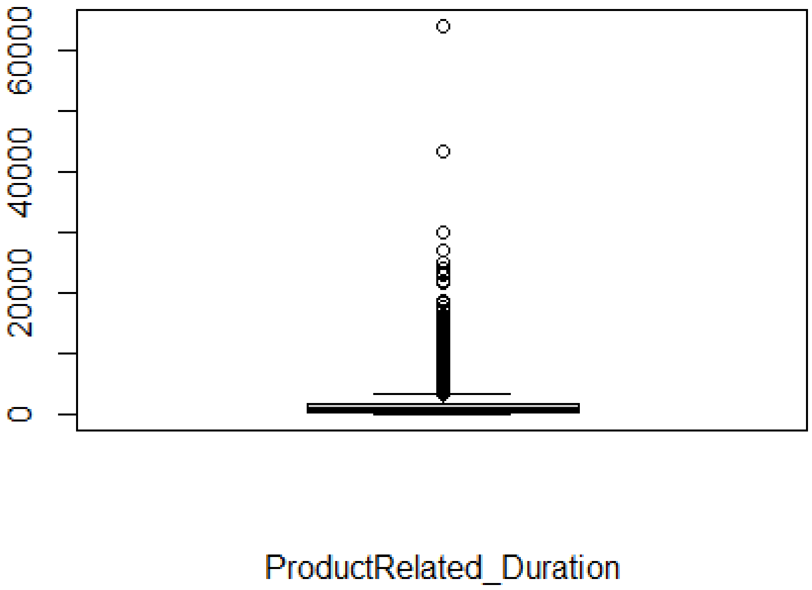
```
#Using boxplots to visulize the outliers
for(i in 2:ncol(num)) {
  boxplot(num[i], xlab=colnames(num[i]))
}
```

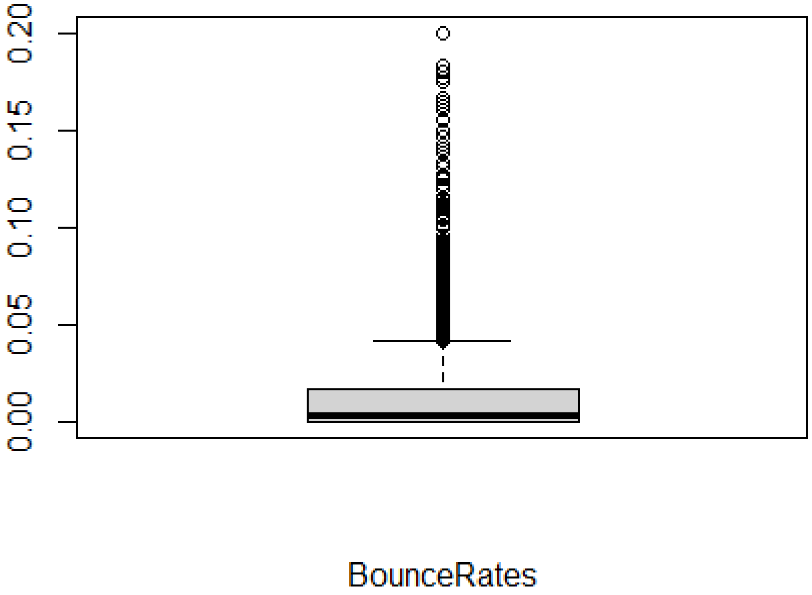

3000
2000
1000
0
Administrative_Duration

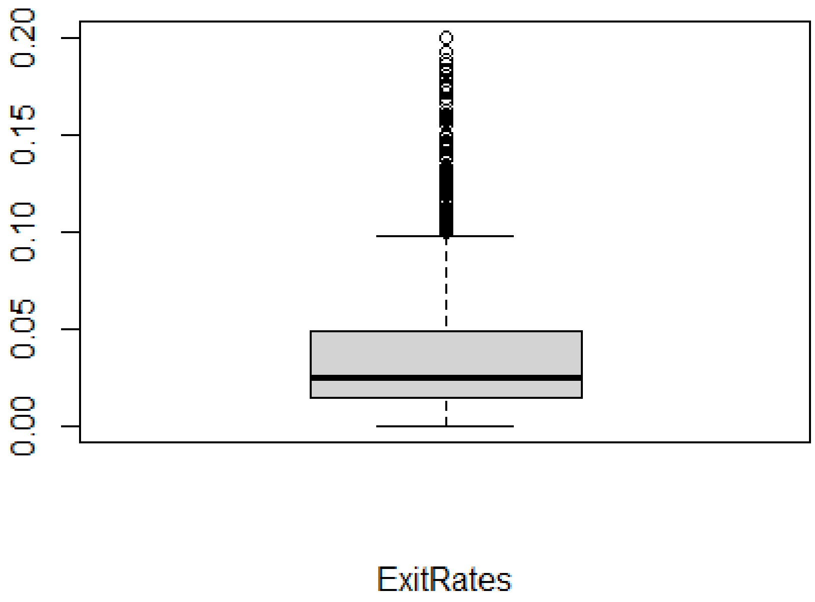

20
15
10
5
0
Informational

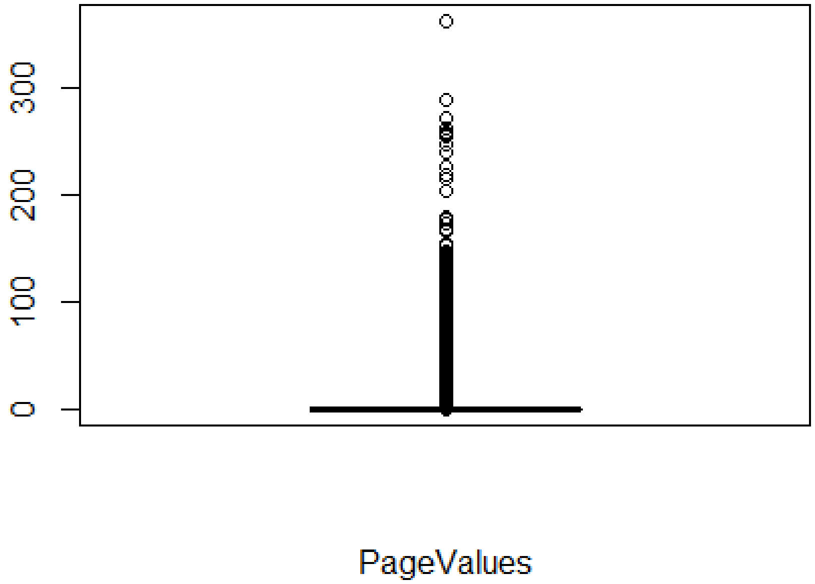

0
500
1500
2500
Informational_Duration

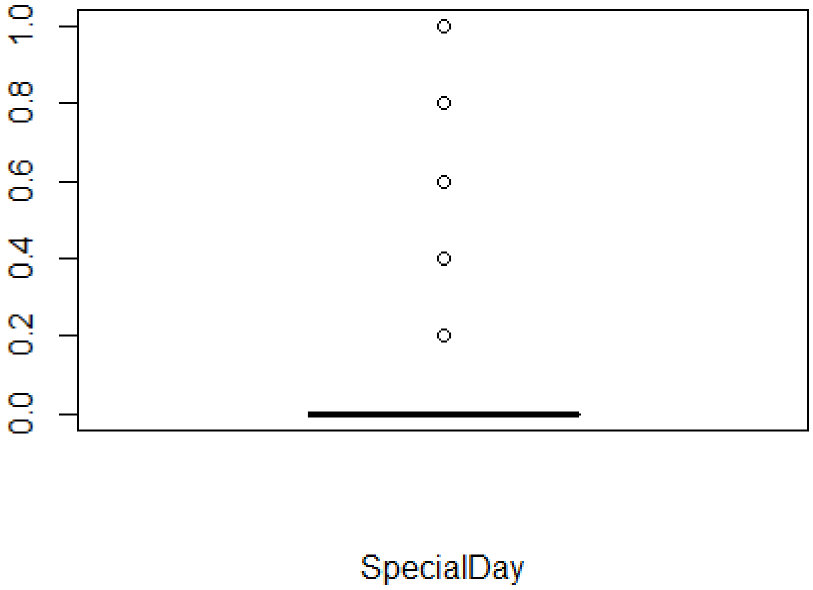

0
200
400
600
ProductRelated

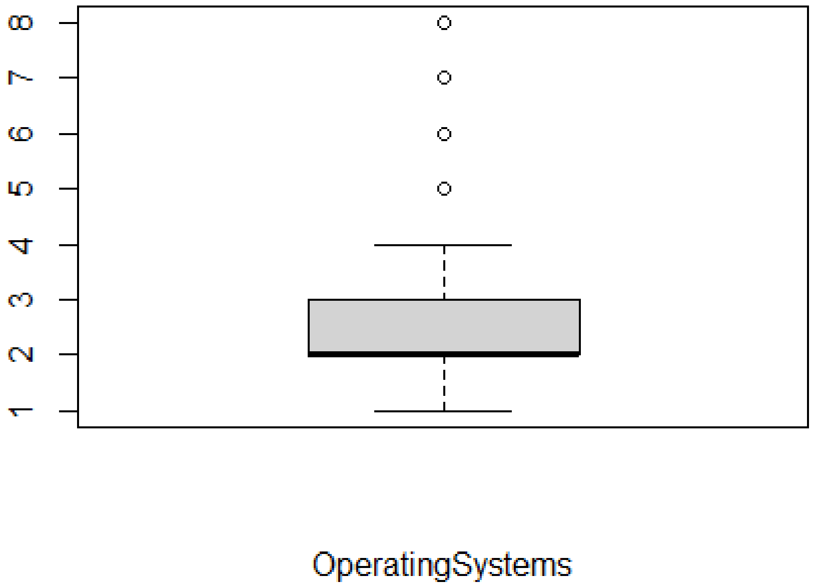

0
20000
40000
60000
ProductRelated_Duration

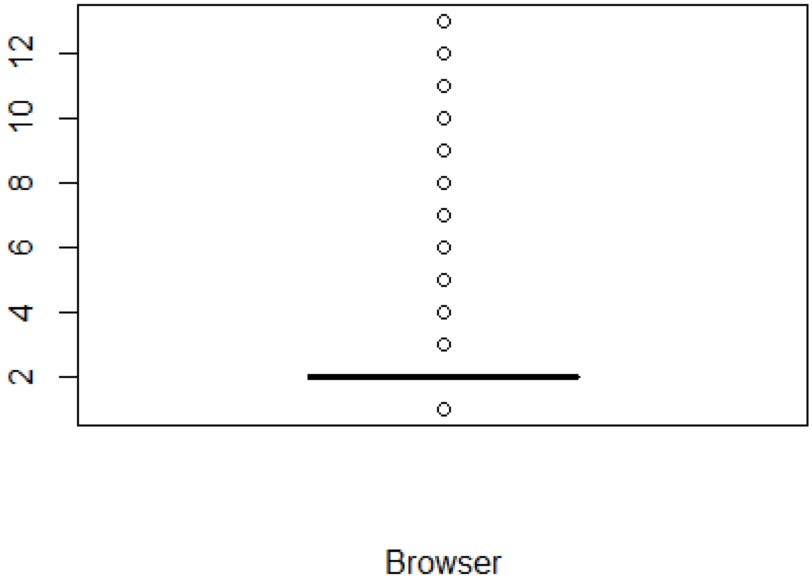

0.00
0.05
0.10
0.15
0.20
BounceRates

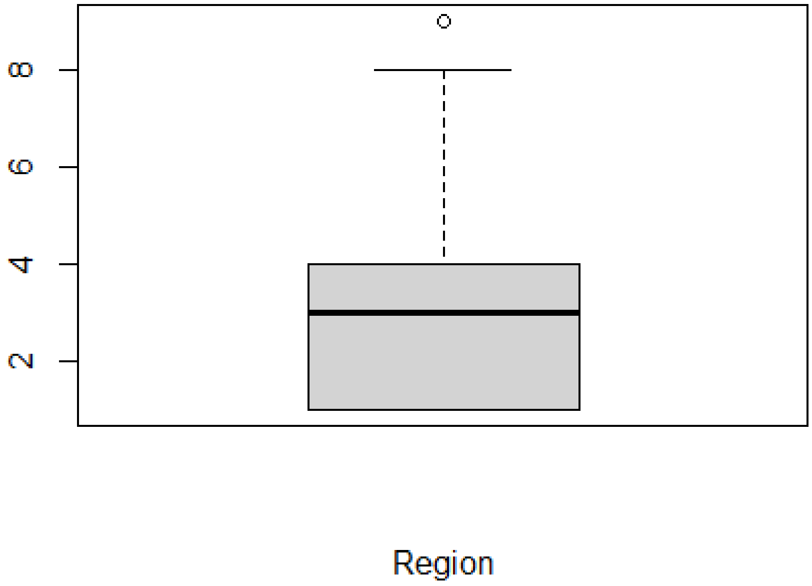

0.20
0.15
0.10
0.05
0.00
ExitRates

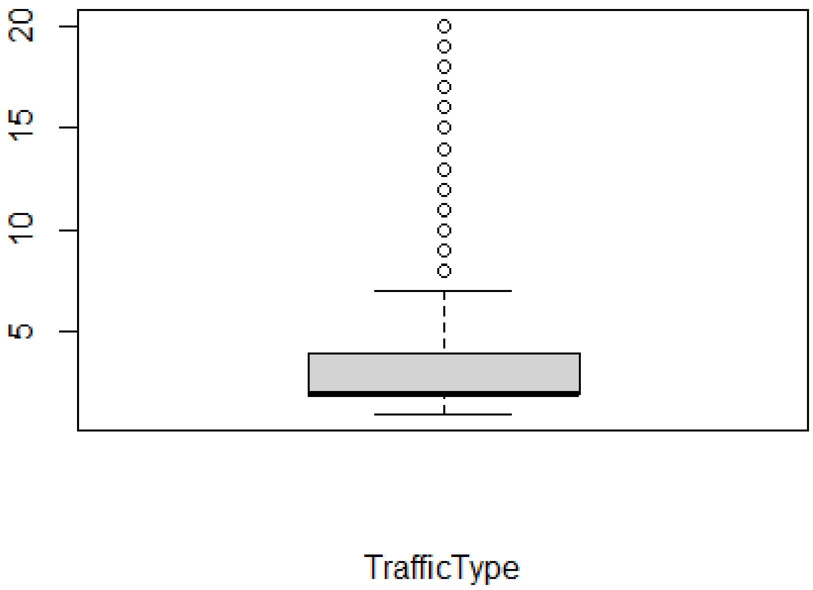

300
200
100
0
PageValues


0.0
0.2
0.4
0.6
0.8
1.0
SpecialDay


1
2
3
4
5
6
7
8
OperatingSystems


2
4
6
8
10
12
Browser


2
4
6
8
Region



Data Analysis

Univarient Analysis

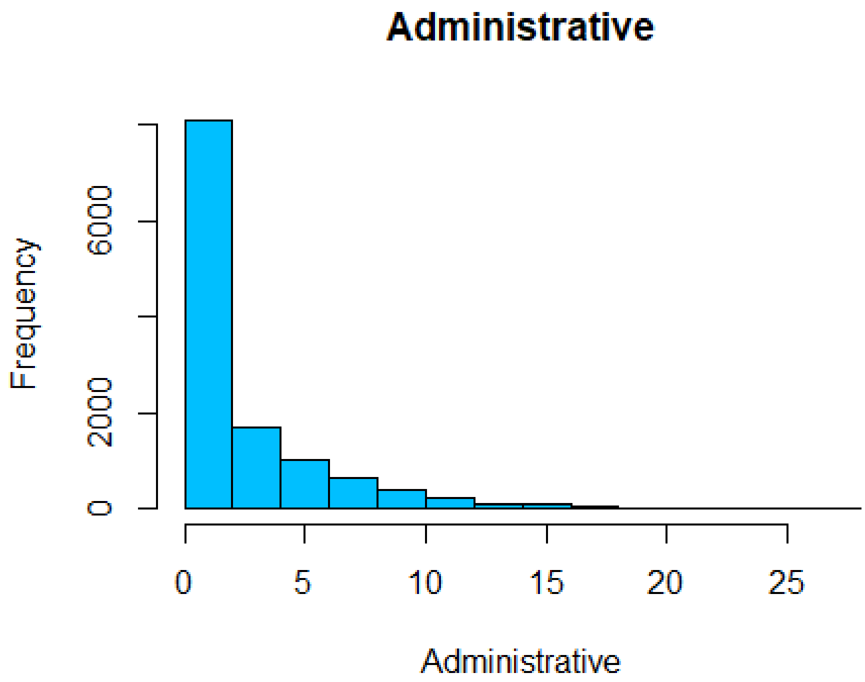
```
#Getting the statistical summaries of the data
summary(df_new)

##  Administrative  Administrative_Duration Informational
##  Min.   : 0.00   Min.   :  -1.00         Min.   : 0.0000
##  1st Qu.: 0.00   1st Qu.:   0.00         1st Qu.: 0.0000
##  Median : 1.00   Median :   9.00         Median : 0.0000
##  Mean   : 2.34   Mean   :  81.68         Mean   : 0.5088
##  3rd Qu.: 4.00   3rd Qu.:  94.75         3rd Qu.: 0.0000
##  Max.   :27.00   Max.   :3398.75         Max.   :24.0000
##  Informational_Duration ProductRelated   ProductRelated_Duration
##  Min.   :  -1.00        Min.   :  0.00   Min.   :   -1.0
##  1st Qu.:   0.00        1st Qu.:  8.00   1st Qu.:  193.6
##  Median :   0.00        Median : 18.00   Median :  609.5
##  Mean   :  34.84        Mean   : 32.06   Mean   : 1207.5
##  3rd Qu.:   0.00        3rd Qu.: 38.00   3rd Qu.: 1477.6
##  Max.   :2549.38        Max.   :705.00   Max.   :63973.5
##   BounceRates         ExitRates          PageValues        SpecialDay
##  Min.   :0.00000   Min.   :0.00000   Min.   :  0.000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.01422   1st Qu.:  0.000   1st Qu.:0.00000
##  Median :0.00293   Median :0.02500   Median :  0.000   Median :0.00000
##  Mean   :0.02045   Mean   :0.04150   Mean   :  5.952   Mean   :0.06197
##  3rd Qu.:0.01667   3rd Qu.:0.04848   3rd Qu.:  0.000   3rd Qu.:0.00000  
##  Max.   :0.20000   Max.   :0.20000   Max.   :361.764   Max.   :1.00000  
##     Month           OperatingSystems    Browser           Region     
##  Length:12199       Min.   :1.000    Min.   : 1.000   Min.   :1.000  
##  Class :character   1st Qu.:2.000    1st Qu.: 2.000   1st Qu.:1.000  
##  Mode  :character   Median :2.000    Median : 2.000   Median :3.000  
##                     Mean   :2.124    Mean   : 2.358   Mean   :3.153  
##                     3rd Qu.:3.000    3rd Qu.: 2.000   3rd Qu.:4.000  
##                     Max.   :8.000    Max.   :13.000   Max.   :9.000  
##   TrafficType     VisitorType         Weekend         Revenue       
##  Min.   : 1.000   Length:12199       Mode :logical   Mode :logical  
##  1st Qu.: 2.000   Class :character   FALSE:9343      FALSE:10291    
##  Median : 2.000   Mode  :character   TRUE :2856      TRUE :1908     
##  Mean   : 4.075                                                     
##  3rd Qu.: 4.000                                                     
##  Max.   :20.000                                                     
```

```
#getting measure of dispersion fro each cols
#Create a function
library(moments)
summary.list = function(x)list(
  Mean=mean(x, na.rm=TRUE),
  Median=median(x, na.rm=TRUE),
  Skewness=skewness(x, na.rm=TRUE),
  Kurtosis=kurtosis(x, na.rm=TRUE),
  Variance=var(x, na.rm=TRUE),
  Std.Dev=sd(x, na.rm=TRUE),
  Coeff.Variation.Prcnt=sd(x, na.rm=TRUE)/mean(x, na.rm=TRUE)*100,
  Std.Error=sd(x, na.rm=TRUE)/sqrt(length(x[!is.na(x)]))
)

#Calling the function and applying the function
sapply(df_new[,c(y)], summary.list)
```

```
##                       Administrative Administrative_Duration Informational
## Mean                  2.340028       81.68214                0.5088122    
## Median                1              9                       0            
## Skewness              1.946248       5.59021                 4.013451     
## Kurtosis              7.636106       53.09389                29.64254     
## Variance              11.09457       31516.25                1.62771      
## Std.Dev               3.330851       177.5282                1.275817     
## Coeff.Variation.Prcnt 142.3424       217.3402                250.7442     
## Std.Error             0.03015735     1.60733                 0.01155118   
##                       Informational_Duration ProductRelated
## Mean                  34.83734               32.05845      
## Median                0                      18            
## Skewness              7.537435               4.332134      
## Kurtosis              78.46409               34.04903      
## Variance              20010.51               1989.241      
## Std.Dev               141.4585               44.60091      
## Coeff.Variation.Prcnt 406.0543               139.1237
## Std.Error             1.280758               0.4038142
##                       ProductRelated_Duration BounceRates  ExitRates
## Mean                  1207.508                0.02044674   0.04149678
## Median                609.5417                0.002930403  0.025
## Skewness              7.251403                3.152874     2.233125
## Kurtosis              139.5908                12.25506     7.624252
## Variance              3686121                 0.002061387  0.0021388
## Std.Dev               1919.927                0.0454025    0.04624716
## Coeff.Variation.Prcnt 158.9991                222.0526     111.4476
## Std.Error             17.38292                0.0004110718 0.0004187193
##                       PageValues SpecialDay  OperatingSystems Browser
## Mean                  5.9525     0.06197229  2.124354         2.358144
## Median                0          0           2                2
## Skewness              6.348663   3.284481    2.031955         3.215653
## Kurtosis              67.94031   12.78605    13.26887         15.53659
## Variance              348.1132   0.03988432  0.8226229        2.926075
## Std.Dev               18.65779   0.1997106   0.9069856        1.710578
## Coeff.Variation.Prcnt 313.4446   322.2579    42.69465         72.53914
## Std.Error             0.1689266  0.001808169 0.008211799      0.01548748
##                       Region     TrafficType
## Mean                  3.153291   4.074596
## Median                3          2
## Skewness              0.9787304  1.958522
## Kurtosis              2.840195   6.466127
## Variance              5.771712   16.12675
## Std.Dev               2.402439   4.015813
## Coeff.Variation.Prcnt 76.18829   98.55732
## Std.Error             0.02175155 0.03635895
```

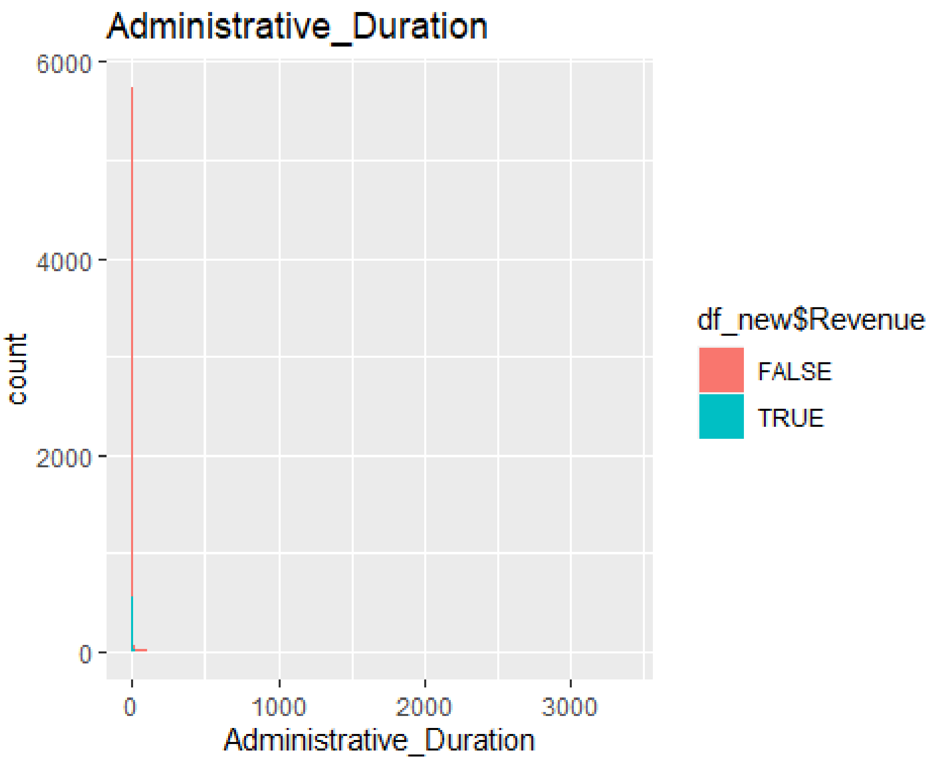
#Plots

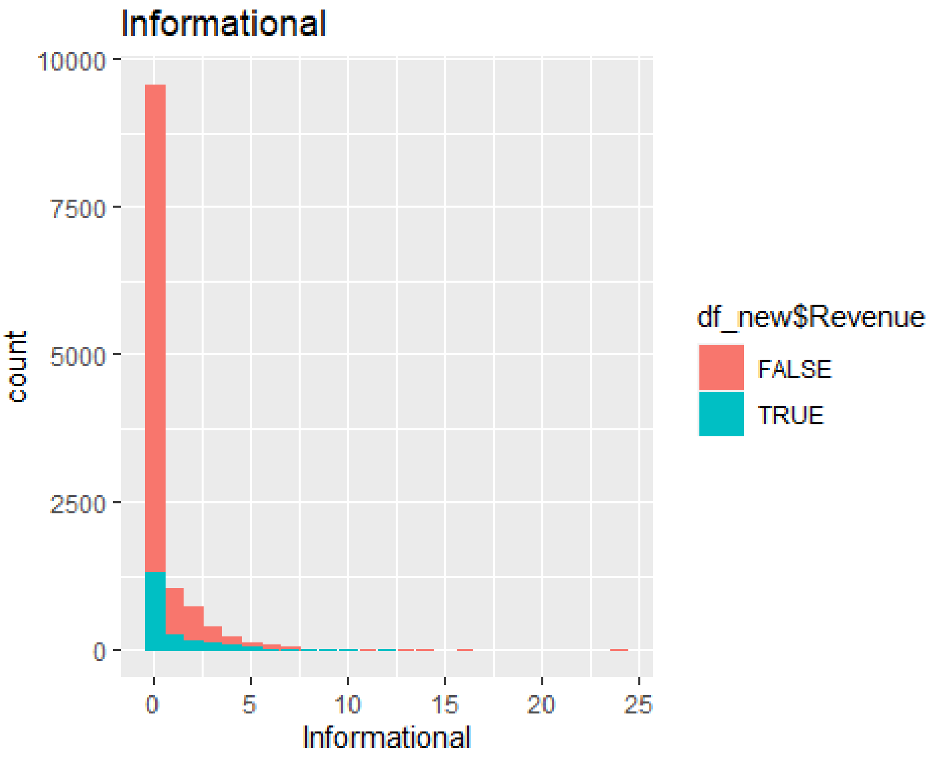
```
library(tidyverse)
# Histograms for Area Income
hist(df_new$Administrative,
  main = "Administrative",
  xlab = "Administrative",
  col = "deepskyblue")
```

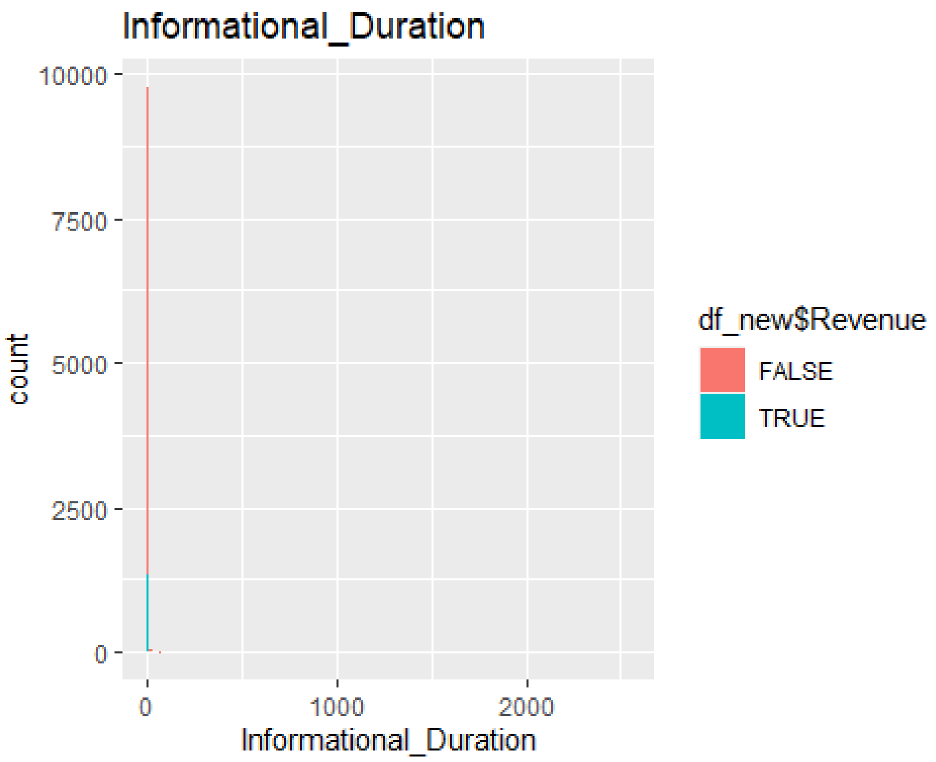


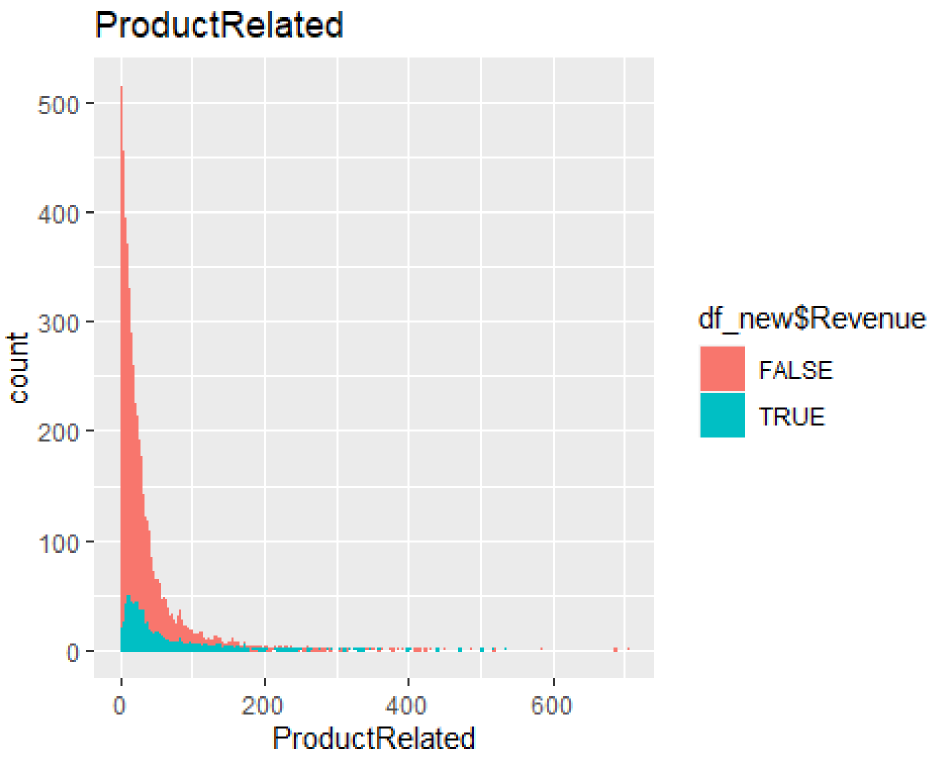
## Bivarient Analysis

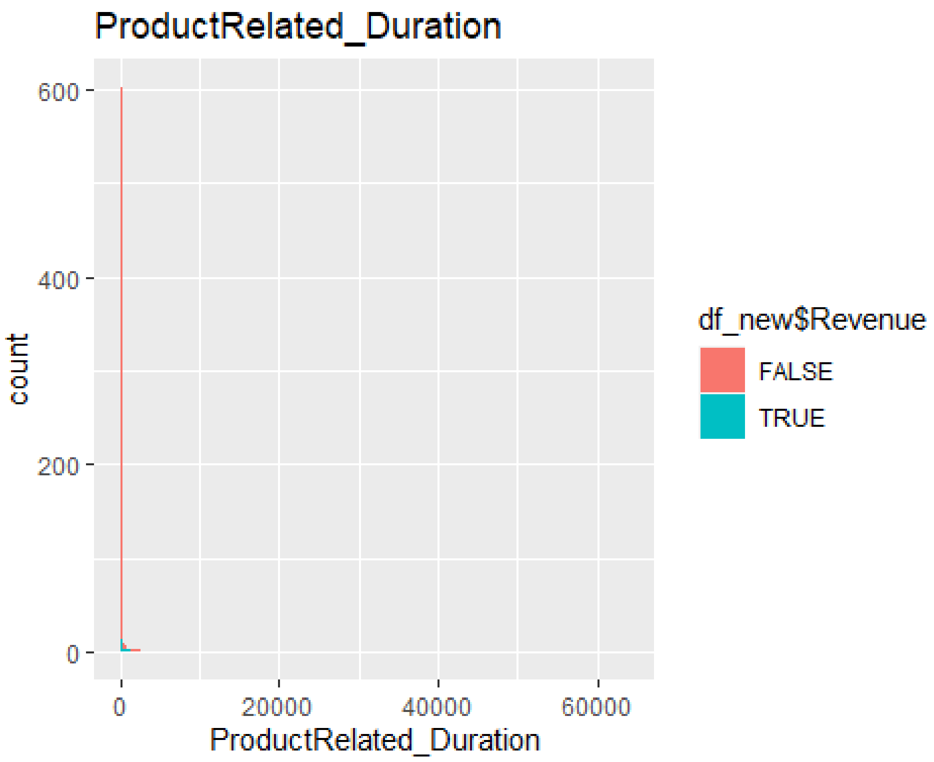
```
#We shall use loops to visuize how each column behave aganist revenue
for(i in 2:ncol(num)) {                                    # Printing ggplot within for-loop
  print(ggplot(num, aes(x= num[ , i],fill = df_new$Revenue, color = df_new$Revenue, )) +
          geom_bar()+labs(title = 'df_new[i]')+labs(title=colnames(df_new[i]), x=colnames(df_new[i])))
}
```

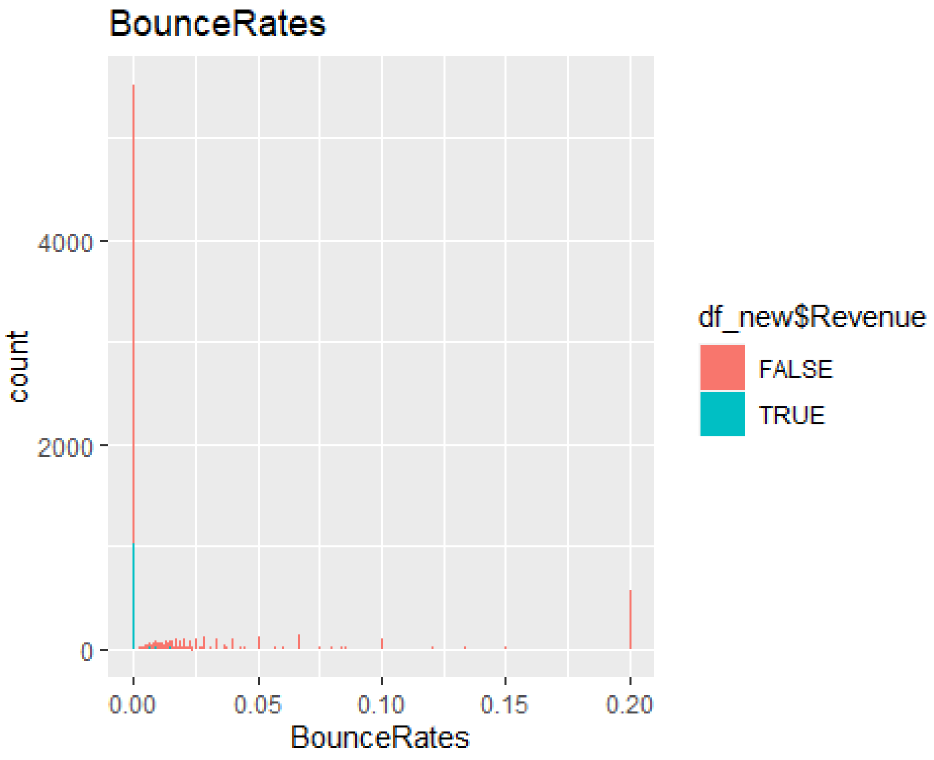

Administrative_Duration
6000
4000
2000
0
count
0
1000
2000
3000
Administrative_Duration
df_new$Revenue
FALSE
TRUE

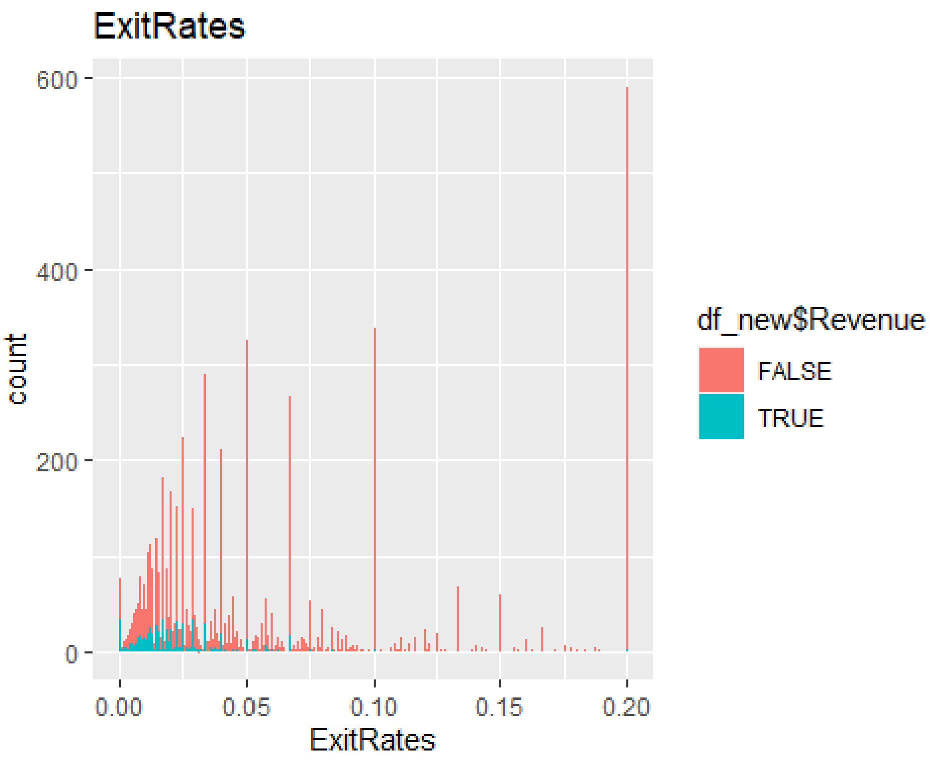

Informational
10000
7500
5000
2500
0
count
0
5
10
15
20
25
Informational
df_new$Revenue
FALSE
TRUE

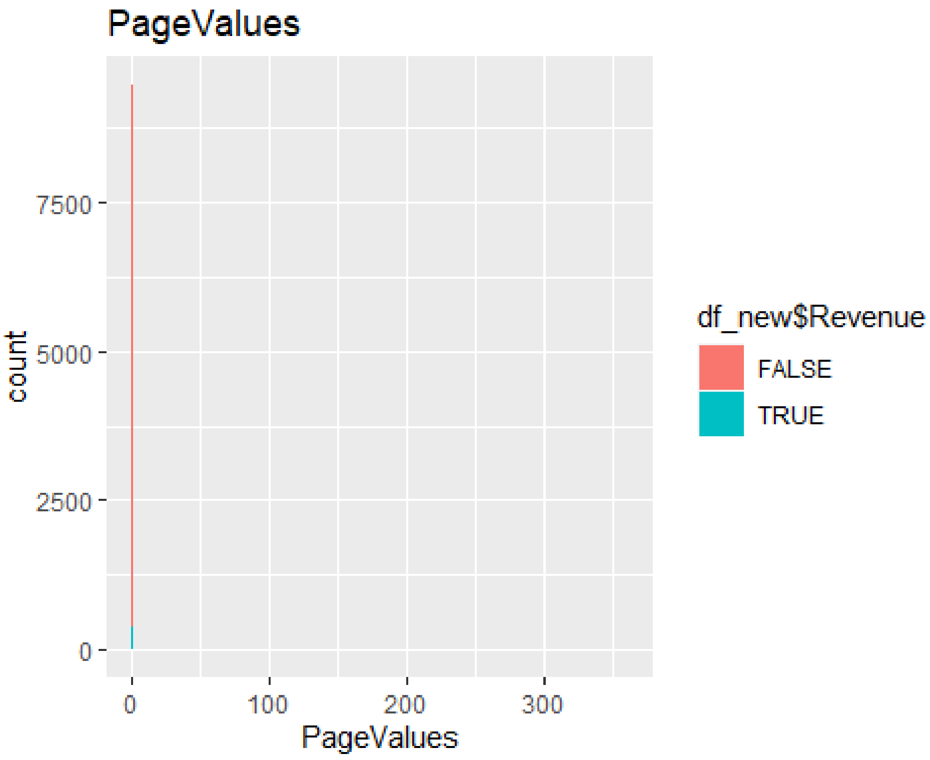

Informational_Duration
10000
7500
5000
2500
0
count
0
1000
2000
Informational_Duration
df_new$Revenue
FALSE
TRUE

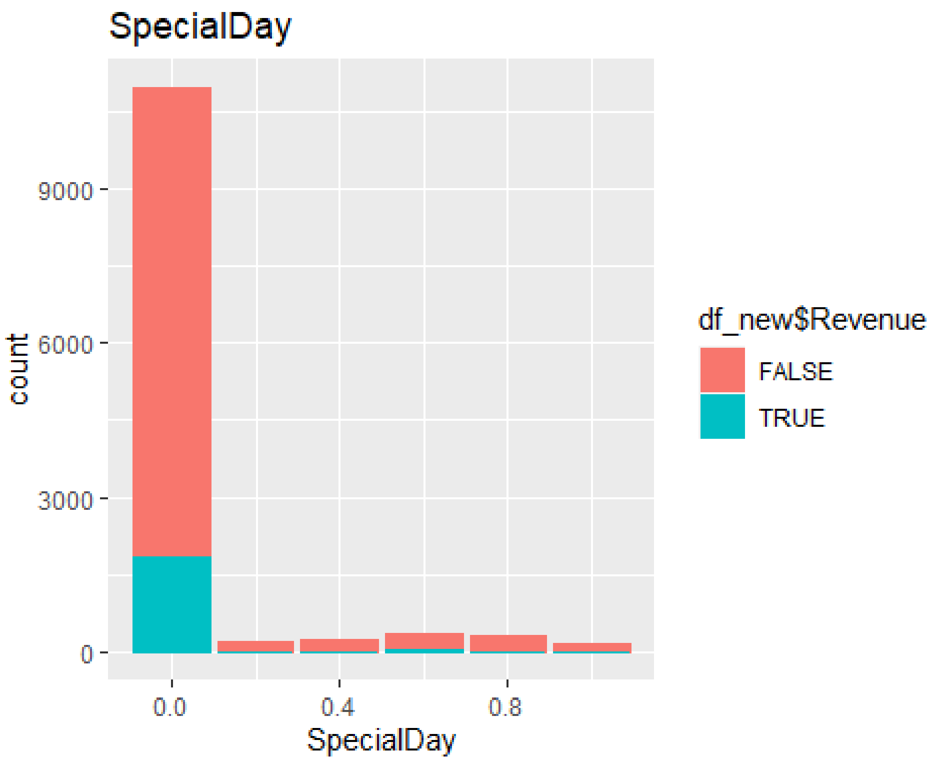

ProductRelated
500
400
300
200
100
0
count
0
200
400
600
ProductRelated
df_new$Revenue
FALSE
TRUE

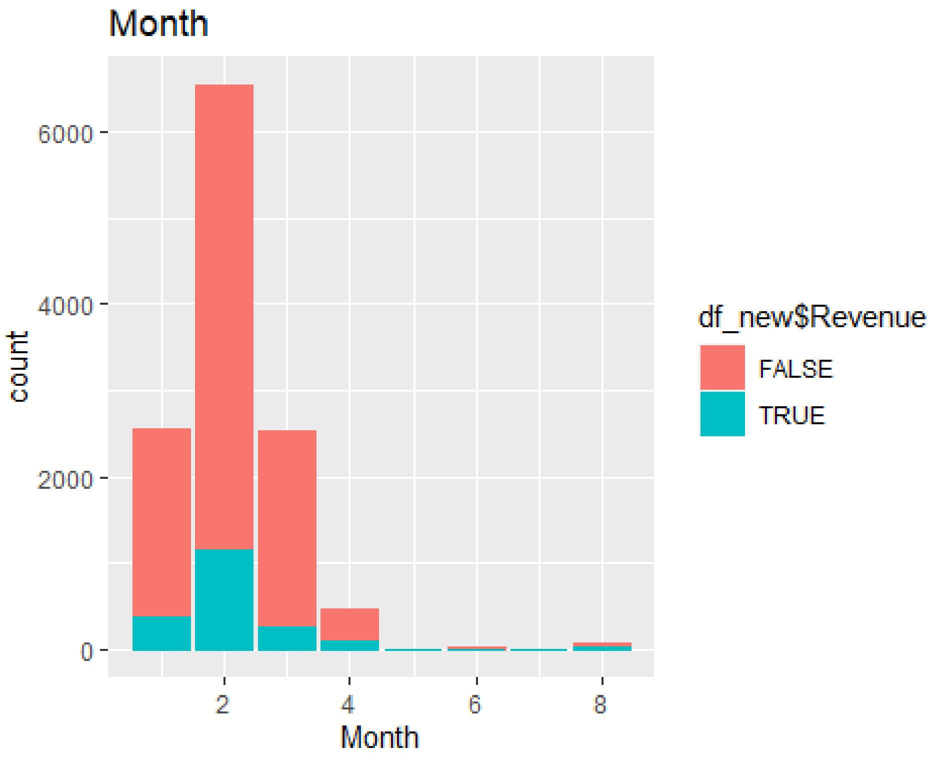

ProductRelated_Duration
600
400
200
0
count
0
20000
40000
60000
ProductRelated_Duration
df_new$Revenue
FALSE
TRUE

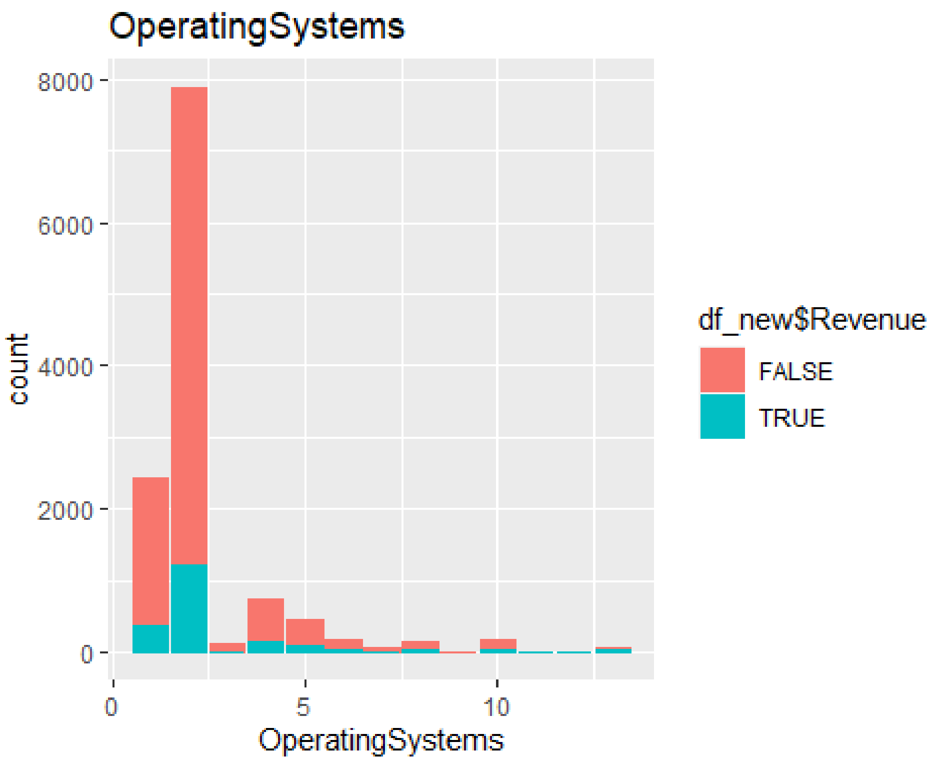

BounceRates
4000
2000
0
count
0.00
0.05
0.10
0.15
0.20
BounceRates
df_new$Revenue
FALSE
TRUE

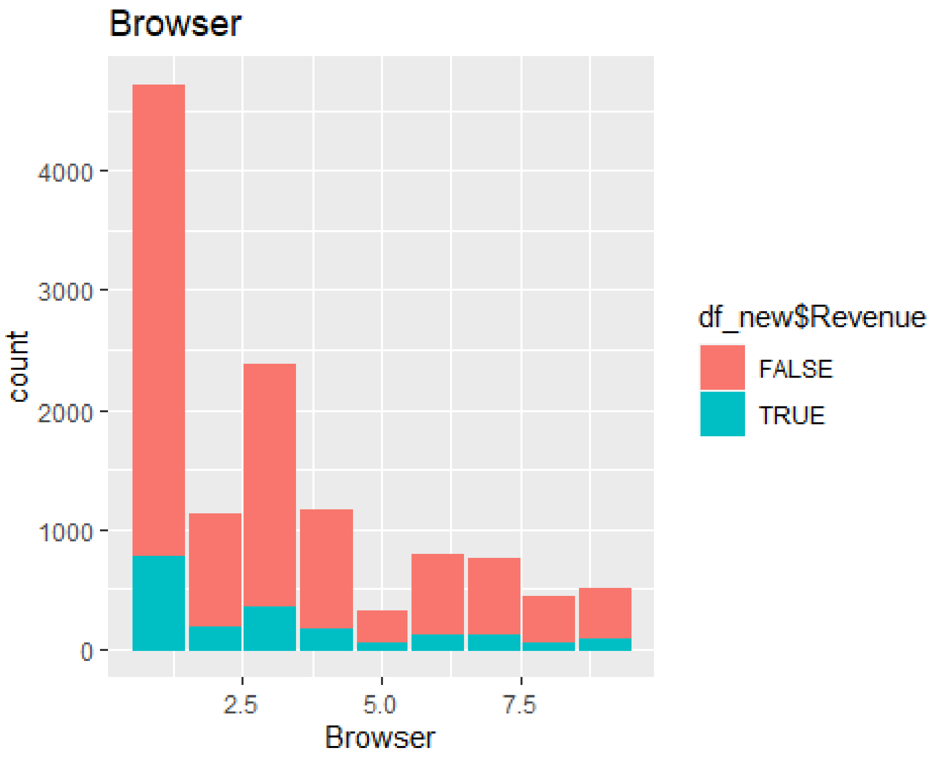

ExitRates
count
600
400
200
0
0.00
0.05
0.10
0.15
0.20
ExitRates
df_new$Revenue
FALSE
TRUE

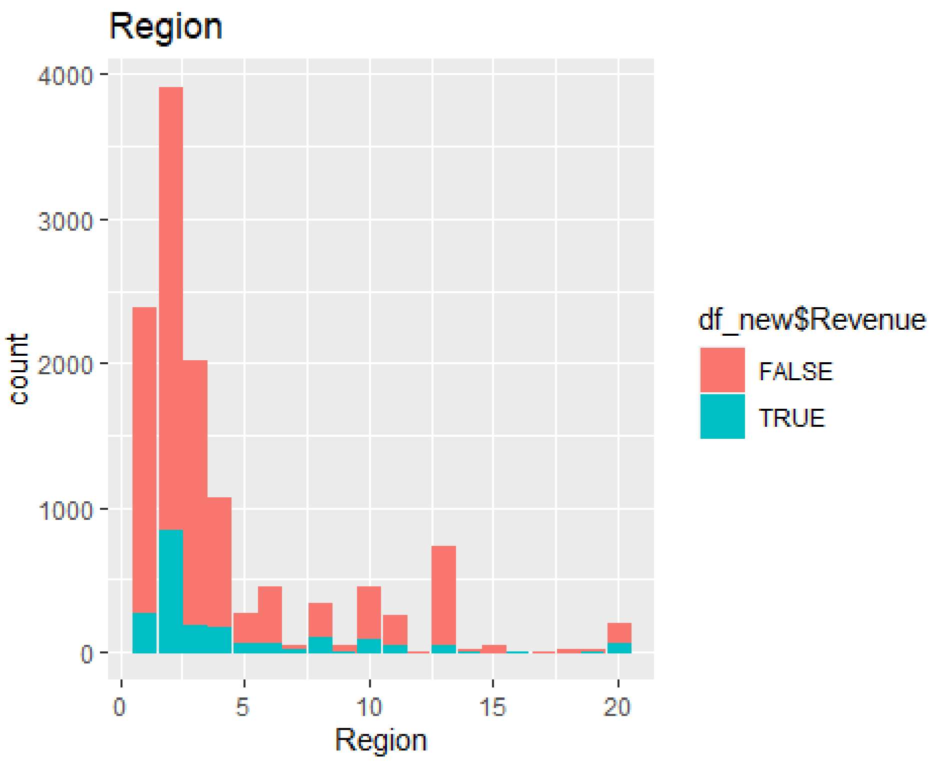

PageValues
count
7500
5000
2500
0
0
100
200
300
PageValues
df_new$Revenue
FALSE
TRUE


SpecialDay
count
9000
6000
3000
0
0.0
0.4
0.8
SpecialDay
df_new$Revenue
FALSE
TRUE


Month
count
6000
4000
2000
0
2
4
6
8
Month
df_new$Revenue
FALSE
TRUE

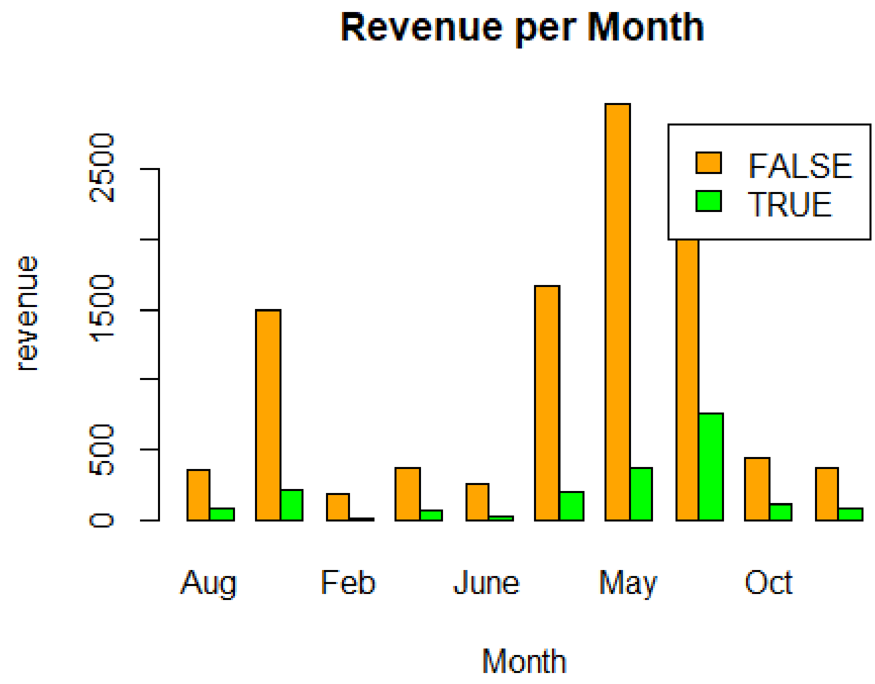

OperatingSystems
count
8000
6000
4000
2000
0
0
5
10
OperatingSystems
df_new$Revenue
FALSE
TRUE


Browser
count
4000
3000
2000
1000
0
2.5
5.0
7.5
Browser
df_new$Revenue
FALSE
TRUE



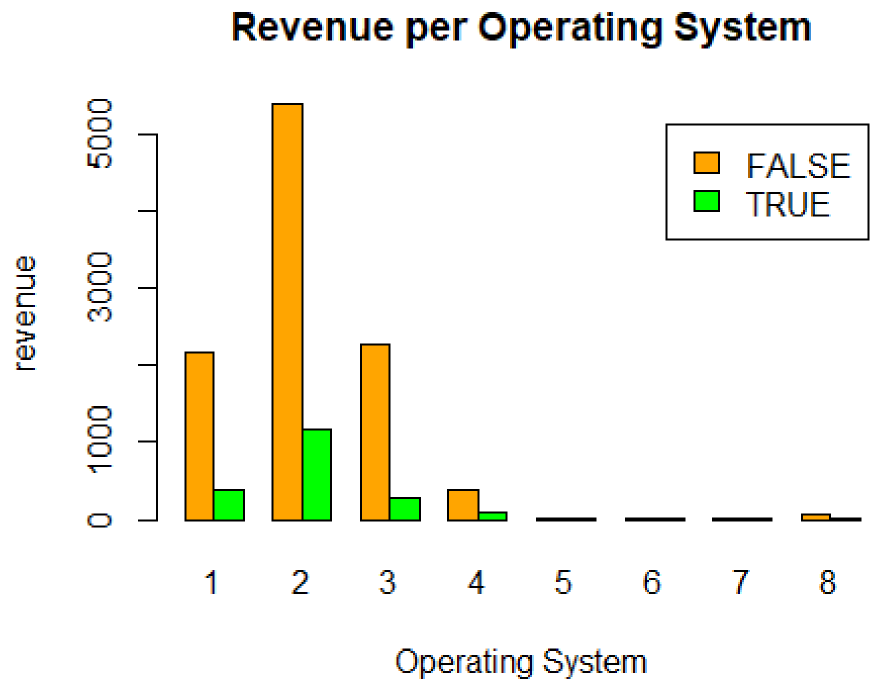
Categorical months

```
# Visualize revenue against months
barplot(table(df_new$Revenue, df_new$Month), main = "Revenue per Month", col
= c("orange", "green"), beside = TRUE,
legend = rownames(table(df_new$Revenue, df_new$Month)), ylab="revenue", xlab
= "Month")
```



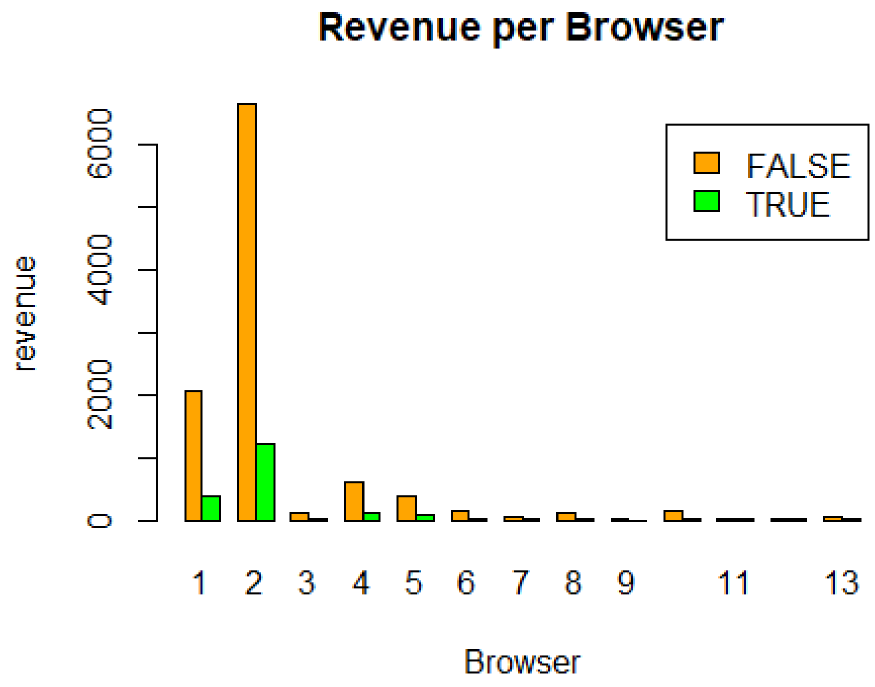
November returns the highest number of revenues while February returns the lowest.

```
# Visualize revenue against Operating System
barplot(table(df_new$Revenue, df_new$OperatingSystems),
        main = "Revenue per Operating System",
        col = c('Orange', "green"), beside = TRUE,
        legend = rownames(table(df_new$Revenue, df_new$OperatingSystems)),
        ylab="revenue",
        xlab = "Operating System")
```



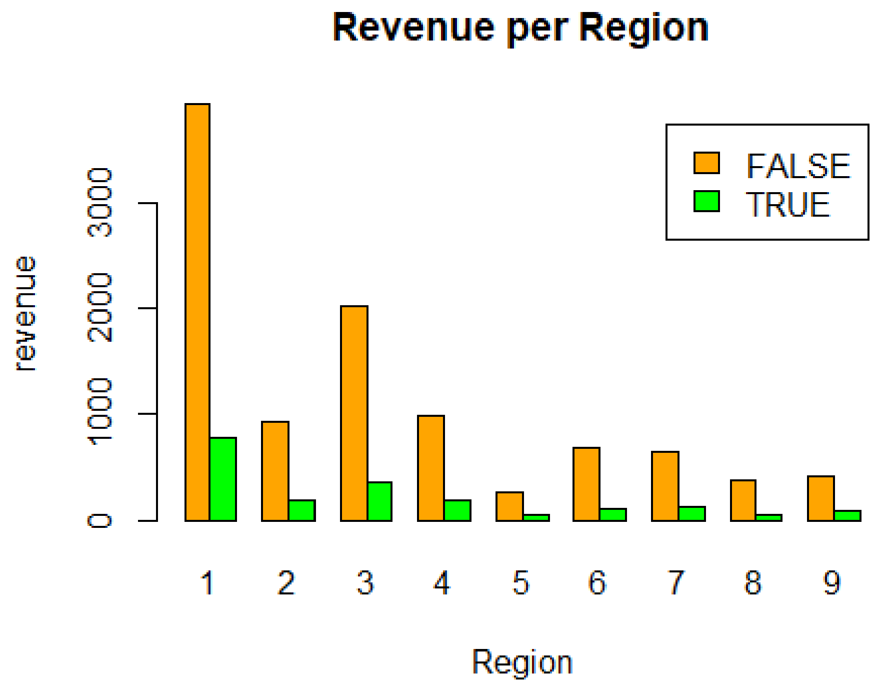
Operating System 2 returns the highest number of revenue while OS 5, 6, and 7 return the lowest.

```
# plotting the distribution of Revenue per Browser
barplot(table(df_new$Revenue, df_new$Browser),
        main = "Revenue per Browser",
        col = c("orange", "green"),
        beside = TRUE,
        legend = rownames(table(df_new$Revenue, df_new$Browser)),
        ylab="revenue",
        xlab = "Browser")
```



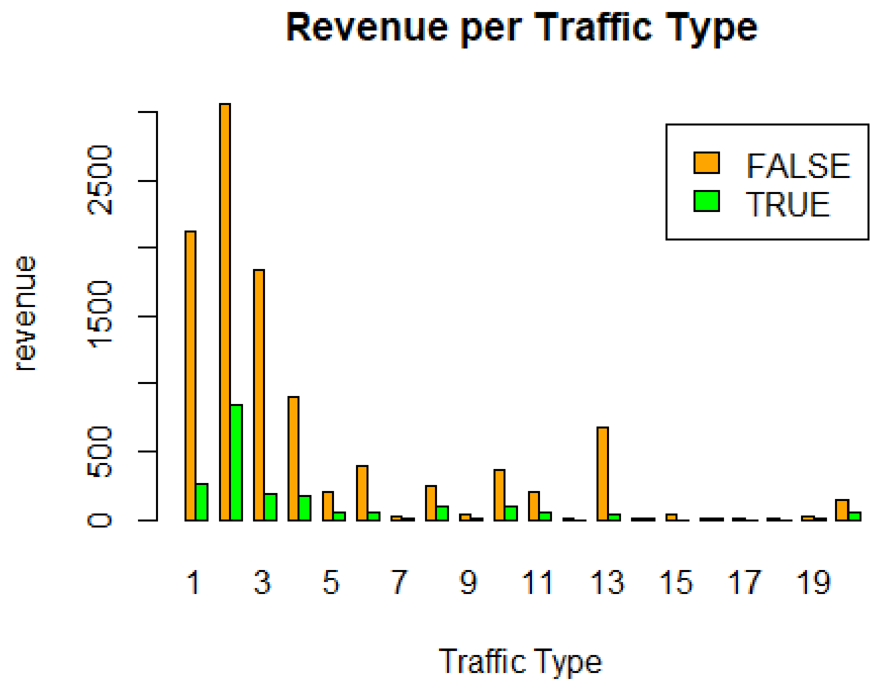
Browser 2 returns the highest number of revenue while 3, 7, 9, 11, and 12 return the lowest.

```
# plotting the distribution of Revenue per Region
barplot(table(df_new$Revenue, df_new$Region),
        main = "Revenue per Region",
        col = c("orange", "green"), beside = TRUE,
        legend = rownames(table(df_new$Revenue, df_new$Region)),
        ylab="revenue", xlab = "Region")
```



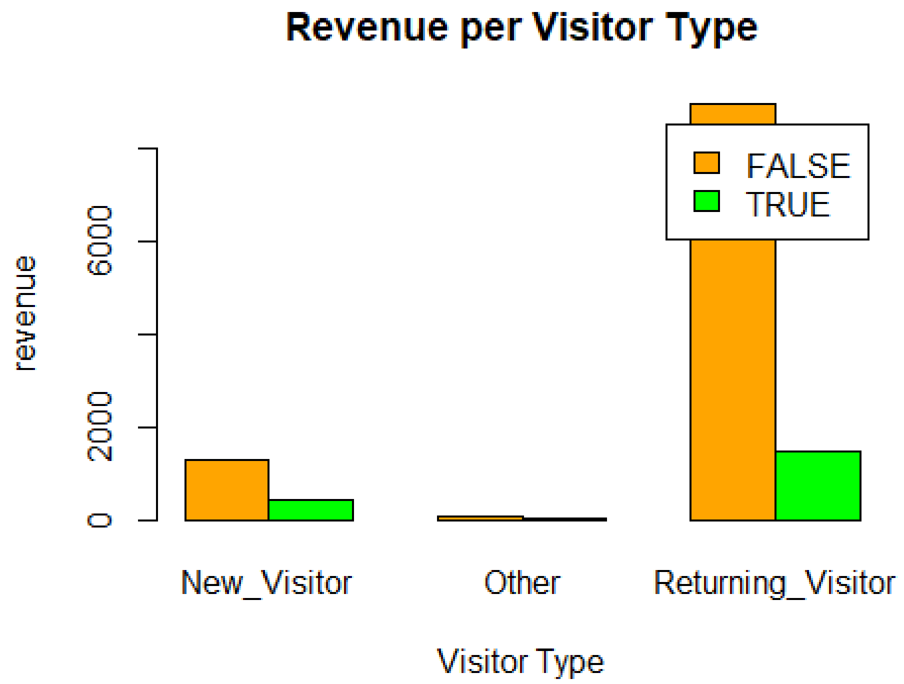
Region 1 returns the highest number of revenue, Region 5 and 8 returns the lowest.

```
# plotting the distribution of Revenue per Traffic Type
barplot(table(df_new$Revenue, df_new$TrafficType),
        main = "Revenue per Traffic Type",
        col = c("orange", "green"), beside = TRUE,
        legend = rownames(table(df_new$Revenue, df_new$TrafficType)),
        ylab="revenue", xlab = "Traffic Type")
```



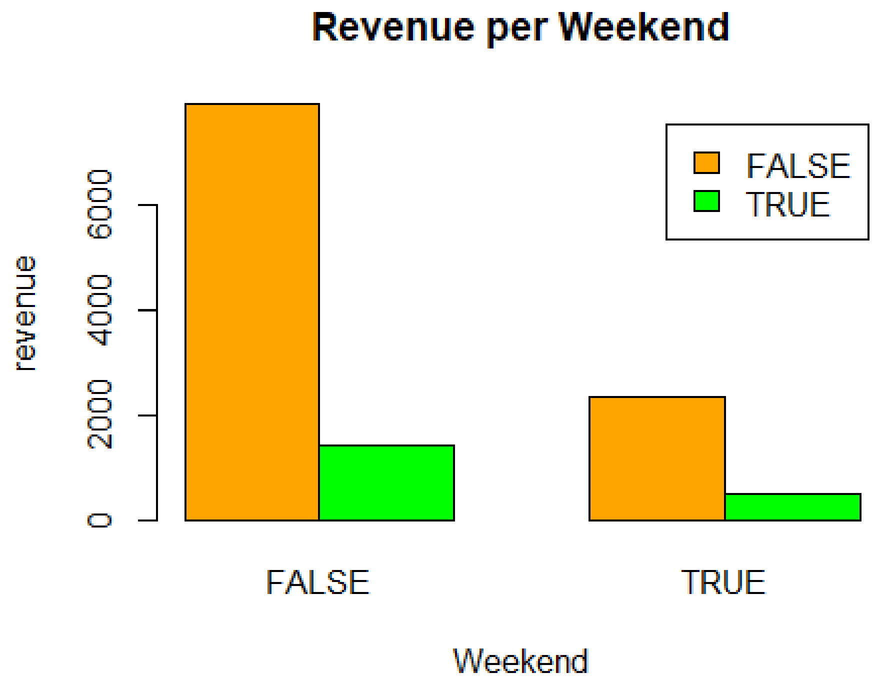
Traffic 2 has the highest number of revenues, 12, 14 and 18 return the lowest.

```
# plotting the distribution of Revenue per Visitor Type
barplot(table(df_new$Revenue, df_new$VisitorType),
        main = "Revenue per Visitor Type",
        col = c("orange", "green"), beside = TRUE,
        legend = rownames(table(df_new$Revenue, df_new$VisitorType)),
        ylab="revenue", xlab = "Visitor Type")
```



Returning visitors brought more revenue with new vistors generating around 1000.

```
# plotting the distribution of Revenue per Weekend
barplot(table(df_new$Revenue, df_new$Weekend),
        main = "Revenue per Weekend",
        col = c("orange", "green"),beside = TRUE,
        legend = rownames(table(df_new$Revenue, df_new$Weekend)),
        ylab="revenue", xlab = "Weekend")
```



More revenue was generated during the weekdays than the weekends.

```
#check the correlation
cor(df_new[,unlist(lapply(df_new, is.numeric))])
```

```
##                           Administrative Administrative_Duration
Informational
## Administrative               1.000000000             0.600409653
0.37528761
## Administrative_Duration      0.600409653             1.000000000
0.30143630
## Informational                0.375287611             0.301436296
1.00000000
## Informational_Duration       0.254786021             0.237189860
0.61867795
## ProductRelated               0.428191515             0.286783914
0.37260472
## ProductRelated_Duration      0.371027224             0.353513793
0.38608372
## BounceRates                 -0.213666635            -0.137333397   -
0.10950530
## ExitRates                   -0.311274132            -0.202024452   -
0.15956681
## PageValues                   0.096920968             0.066168365
0.04739015
## SpecialDay                  -0.097072098            -0.074736885   -
0.04937677
```

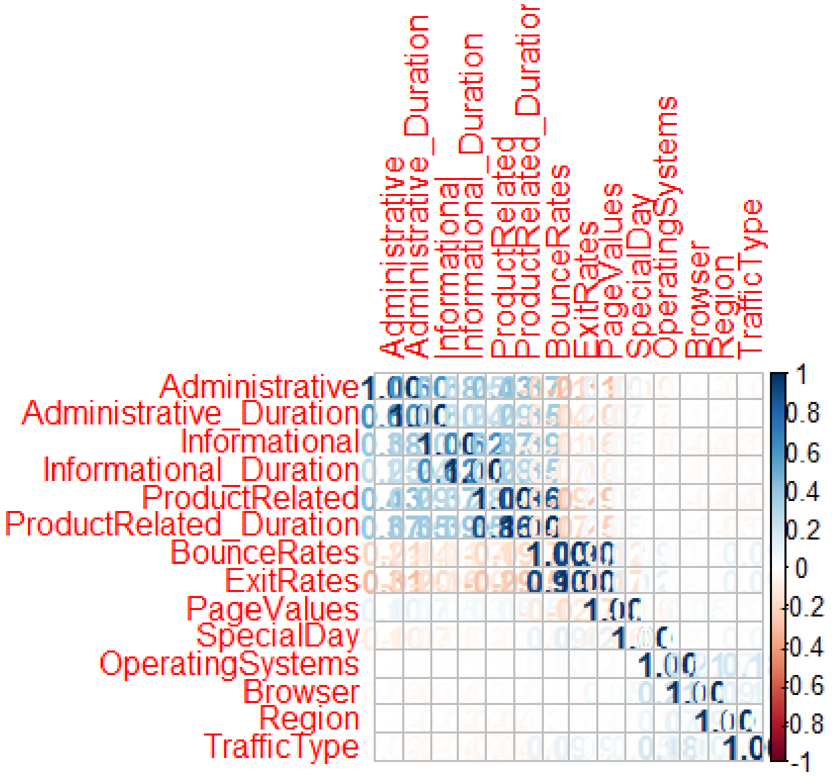
```
## OperatingSystems          -0.006697922           -0.007610715   -
0.00962587
## Browser                   -0.025763658           -0.015833675   -
0.03876681
## Region                    -0.007262053           -0.006723711   -
0.03047732
## TrafficType               -0.034784126           -0.015075015   -
0.03518669
##                         Informational_Duration ProductRelated
## Administrative                      0.254786021    0.428191515
## Administrative_Duration             0.237189860    0.286783914
## Informational                       0.618677947    0.372604721
## Informational_Duration              1.000000000    0.279061948
## ProductRelated                      0.279061948    1.000000000
## ProductRelated_Duration             0.346580691    0.860308186
## BounceRates                        -0.070159472   -0.193515772
## ExitRates                          -0.102932678   -0.286163211
## PageValues                          0.030064160    0.054115494
## SpecialDay                         -0.031293040   -0.025930622
## OperatingSystems                   -0.009749983    0.004090351
## Browser                            -0.019609349   -0.013706213
## Region                             -0.027920098   -0.040106501
## TrafficType                        -0.025163571   -0.044344333
##                         ProductRelated_Duration  BounceRates    ExitRates
## Administrative                      0.371027224 -0.213666635 -0.311274132
## Administrative_Duration             0.353513793 -0.137333397 -0.202024452
## Informational                       0.386083717 -0.109505298 -0.159566815
## Informational_Duration              0.346580691 -0.070159472 -0.102932678
## ProductRelated                      0.860308186 -0.193515772 -0.286163211
## ProductRelated_Duration             1.000000000 -0.174375499 -0.245334012
## BounceRates                        -0.174375499  1.000000000  0.903358192
## ExitRates                          -0.245334012  0.903358192  1.000000000
## PageValues                          0.050840624 -0.115991977 -0.173571542
## SpecialDay                         -0.038210652  0.087839995  0.116783762
## OperatingSystems                    0.002775788  0.026839839  0.016482012
## Browser                            -0.007838332 -0.016018380 -0.003565541
## Region                             -0.034862498  0.001432015 -0.001837556
## TrafficType                        -0.037506944  0.089199039  0.087386232
##                          PageValues   SpecialDay OperatingSystems
Browser
## Administrative            0.09692097 -0.097072098     -0.006697922 -
0.025763658
## Administrative_Duration   0.06616837 -0.074736885     -0.007610715 -
0.015833675
## Informational             0.04739015 -0.049376774     -0.009625870 -
0.038766808
## Informational_Duration    0.03006416 -0.031293040     -0.009749983 -
0.019609349
## ProductRelated            0.05411549 -0.025930622      0.004090351 -
0.013706213
```

```
## ProductRelated_Duration  0.05084062 -0.038210652      0.002775788 -
0.007838332
## BounceRates             -0.11599198  0.087839995      0.026839839 -
0.016018380
## ExitRates               -0.17357154  0.116783762      0.016482012 -
0.003565541
## PageValues               1.00000000 -0.064532709      0.018583782
0.045845065
## SpecialDay              -0.06453271  1.000000000      0.012757766
0.003465984
## OperatingSystems         0.01858378  0.012757766      1.000000000
0.212244823
## Browser                  0.04584506  0.003465984      0.212244823
1.000000000
## Region                   0.01059087 -0.016452464      0.071953240
0.091889464
## TrafficType              0.01223694  0.052827944      0.182874100
0.102886237
##                               Region TrafficType
## Administrative          -0.007262053 -0.03478413
## Administrative_Duration -0.006723711 -0.01507502
## Informational           -0.030477323 -0.03518669
## Informational_Duration  -0.027920098 -0.02516357
## ProductRelated          -0.040106501 -0.04434433
## ProductRelated_Duration -0.034862498 -0.03750694
## BounceRates              0.001432015  0.08919904
## ExitRates               -0.001837556  0.08738623
## PageValues               0.010590868  0.01223694
## SpecialDay              -0.016452464  0.05282794
## OperatingSystems         0.071953240  0.18287410
## Browser                  0.091889464  0.10288624
## Region                   1.000000000  0.04252523
## TrafficType              0.042525234  1.00000000
```

```
#install.packages("corrplot")
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
#
## Let’s build a correlation matrix to understand the relation between each
attributes
corrplot(cor(num), method = 'number')
```



```
#drop cols highly correlated
col_drop <- c("Administrative_Duration", "Informational_Duration",
"ProductRelated_Duration", "ExitRates")
df_new <- df_new[, !names(df_new) %in% col_drop]
head(df_new)
```

```
##   Administrative Informational ProductRelated BounceRates PageValues
SpecialDay
## 1              0             0              1  0.20000000          0
0
## 2              0             0              2  0.00000000          0
0
## 3              0             0              1  0.20000000          0
0
## 4              0             0              2  0.05000000          0
0
## 5              0             0             10  0.02000000          0
0
## 6              0             0             19  0.01578947          0
0
##   Month OperatingSystems Browser Region TrafficType       VisitorType
Weekend
## 1   Feb                1       1      1           1 Returning_Visitor
FALSE
## 2   Feb                2       2      1           2 Returning_Visitor
FALSE
## 3   Feb                4       1      9           3 Returning_Visitor
```

```
FALSE
## 4    Feb                3       2      2           4 Returning_Visitor
FALSE
## 5    Feb                3       3      1           4 Returning_Visitor
TRUE
## 6    Feb                2       2      1           3 Returning_Visitor
FALSE
##   Revenue
## 1   FALSE
## 2   FALSE
## 3   FALSE
## 4   FALSE
## 5   FALSE
## 6   FALSE
```

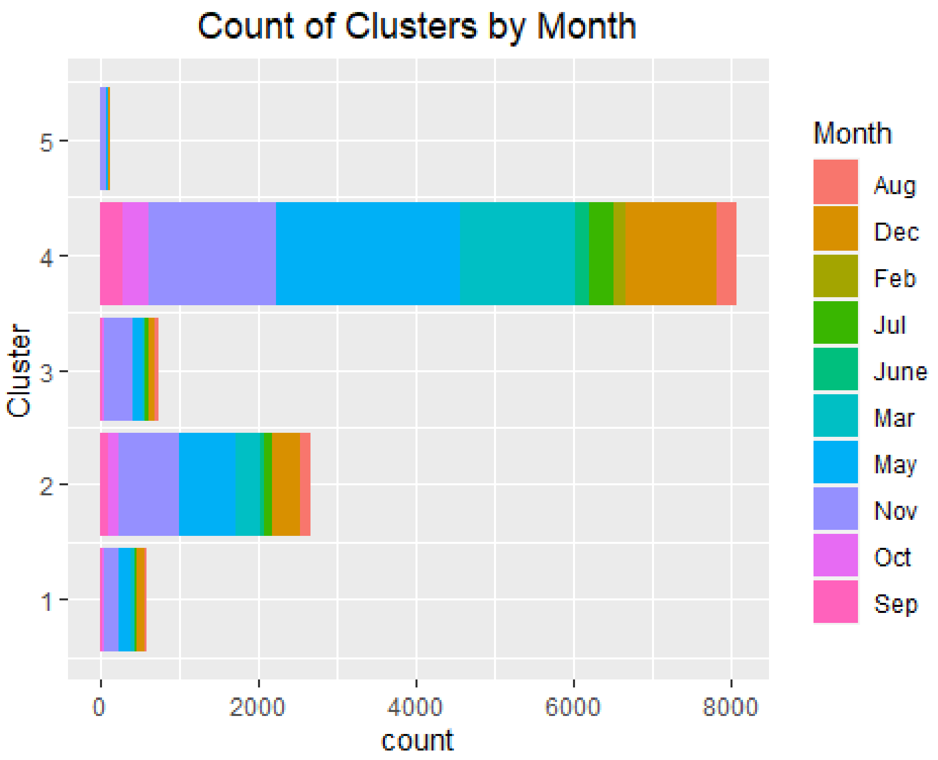
Modelling

```
#Check head
head(df_new)
```

```
##   Administrative Informational ProductRelated BounceRates PageValues
SpecialDay
## 1              0             0              1  0.20000000          0
0
## 2              0             0              2  0.00000000          0
0
## 3              0             0              1  0.20000000          0
0
## 4              0             0              2  0.05000000          0
0
## 5              0             0             10  0.02000000          0
0
## 6              0             0             19  0.01578947          0
0
##   Month OperatingSystems Browser Region TrafficType       VisitorType
Weekend
## 1    Feb                1       1      1           1 Returning_Visitor
FALSE
## 2    Feb                2       2      1           2 Returning_Visitor
FALSE
## 3    Feb                4       1      9           3 Returning_Visitor
FALSE
## 4    Feb                3       2      2           4 Returning_Visitor
FALSE
## 5    Feb                3       3      1           4 Returning_Visitor
TRUE
## 6    Feb                2       2      1           3 Returning_Visitor
FALSE
##   Revenue
## 1   FALSE
## 2   FALSE
## 3   FALSE
## 4   FALSE
## 5   FALSE
## 6   FALSE

#selecting data without revenue
data<-df_new[,-14]
head(data)

##   Administrative Informational ProductRelated BounceRates PageValues
SpecialDay
## 1              0             0              1  0.20000000          0
0
## 2              0             0              2  0.00000000          0
0
## 3              0             0              1  0.20000000          0
0
## 4              0             0              2  0.05000000          0
0
## 5              0             0             10  0.02000000          0
0
## 6              0             0             19  0.01578947          0
0
##   Month OperatingSystems Browser Region TrafficType       VisitorType
Weekend
## 1   Feb                1       1      1           1 Returning_Visitor
FALSE
## 2   Feb                2       2      1           2 Returning_Visitor
FALSE
## 3   Feb                4       1      9           3 Returning_Visitor
FALSE
## 4   Feb                3       2      2           4 Returning_Visitor
FALSE
## 5   Feb                3       3      1           4 Returning_Visitor
TRUE
## 6   Feb                2       2      1           3 Returning_Visitor
FALSE

# Create custom function to fix data types and round
to_numeric_and_round_func <- function(x){
  round(as.numeric(as.character(x)),2)
}
# Mutate the columns to proper data type
data <- data %>%
  mutate_at(vars(-one_of("Month", "Region", "VisitorType", "Weekend")),
to_numeric_and_round_func)

# create clean data with no NA
clean_data <- data %>%
  drop_na()
```

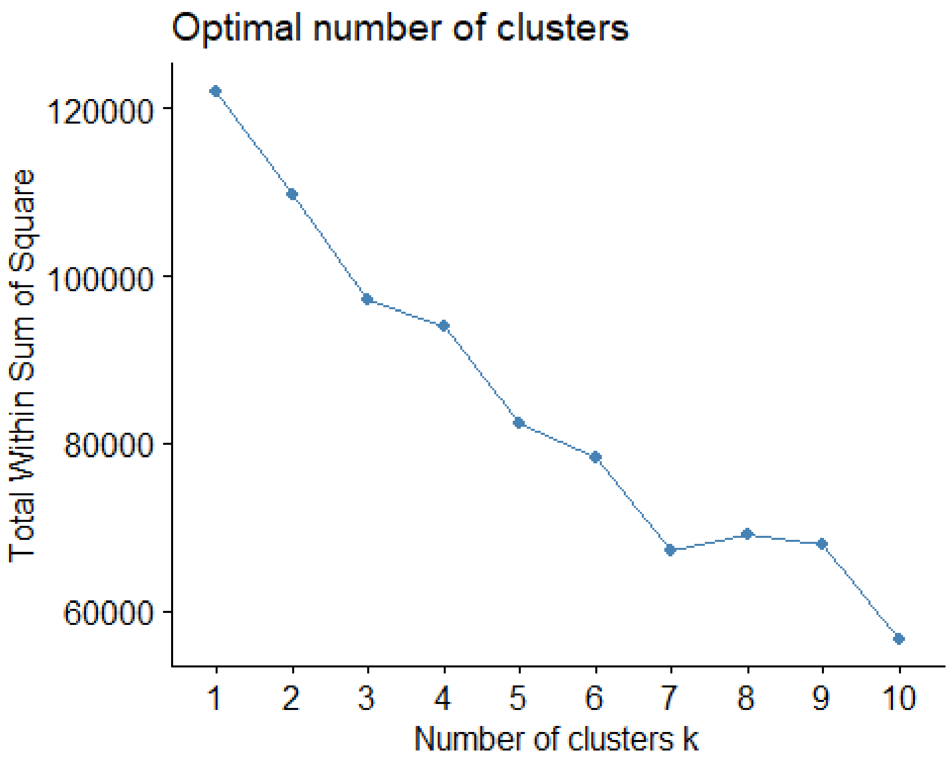
##Kmeans

```
# Set seed
set.seed(1234)
col.names<-c("Month", "VisitorType", "Weekend")

# Cluster Analysis - kmeans
kmeans_basic <- kmeans(clean_data[, !names(data) %in% col.names], centers =
5)
kmeans_basic_table <- data.frame(kmeans_basic$size, kmeans_basic$centers)
kmeans_basic_df <- data.frame(Cluster = kmeans_basic$cluster, data)
# head of df
head(kmeans_basic_df)

##   Cluster Administrative Informational ProductRelated BounceRates
PageValues
## 1       4              0             0              1        0.20
0
## 2       4              0             0              2        0.00
0
## 3       4              0             0              1        0.20
0
## 4       4              0             0              2        0.05
0
## 5       4              0             0             10        0.02
0
## 6       4              0             0             19        0.02
0
##   SpecialDay Month OperatingSystems Browser Region TrafficType
## 1          0   Feb                1       1      1           1
## 2          0   Feb                2       2      1           2
## 3          0   Feb                4       1      9           3
## 4          0   Feb                3       2      2           4
## 5          0   Feb                3       3      1           4
## 6          0   Feb                2       2      1           3
##         VisitorType Weekend
## 1 Returning_Visitor   FALSE
## 2 Returning_Visitor   FALSE
## 3 Returning_Visitor   FALSE
## 4 Returning_Visitor   FALSE
## 5 Returning_Visitor    TRUE
## 6 Returning_Visitor   FALSE

# Visulize the clusters per month
ggplot(data = kmeans_basic_df, aes(y = Cluster)) +
  geom_bar(aes(fill = Month)) +
  ggtitle("Count of Clusters by Month") +
  theme(plot.title = element_text(hjust = 0.5))
```



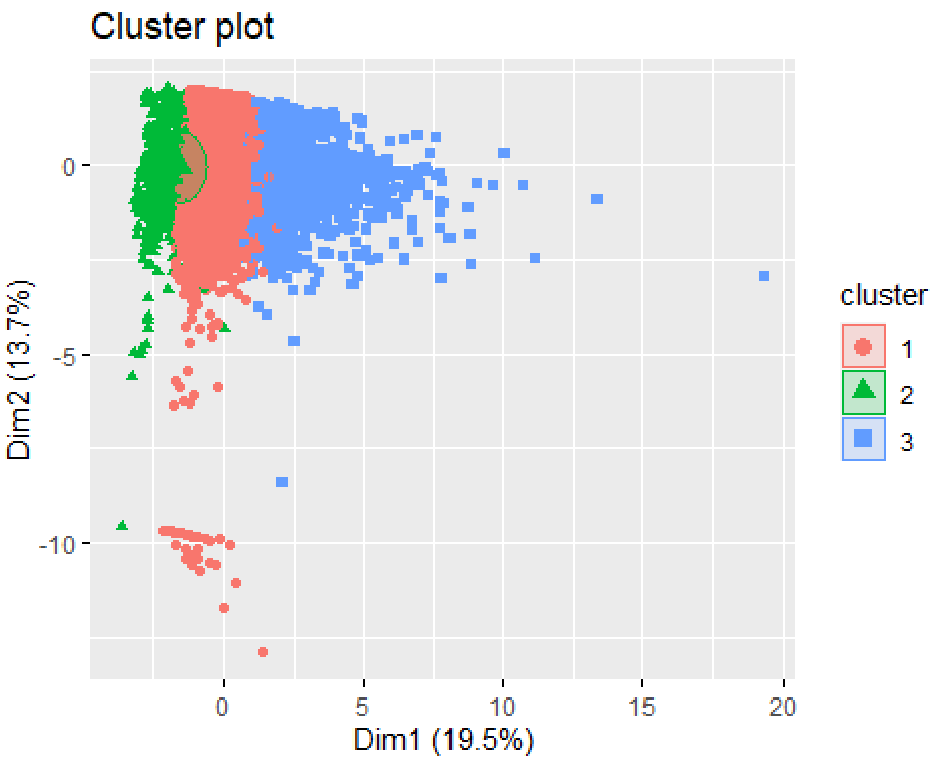
elbow method

```
library(factoextra)

## Warning: package 'factoextra' was built under R version 4.0.5

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

data_norm <-scale(clean_data[, !names(data) %in% col.names])
# Get the optimum number of clusters
fviz_nbclust(data_norm, kmeans, method = "wss")
```



The optimum clusters from the above is 3.

```r
#kmeans
kmeans_fancy <- kmeans(data_norm, 3, nstart = 20)

# plot the clusters
fviz_cluster(kmeans_fancy, data = data_norm, geom = c("point"),ellipse.type =
"euclid")
```



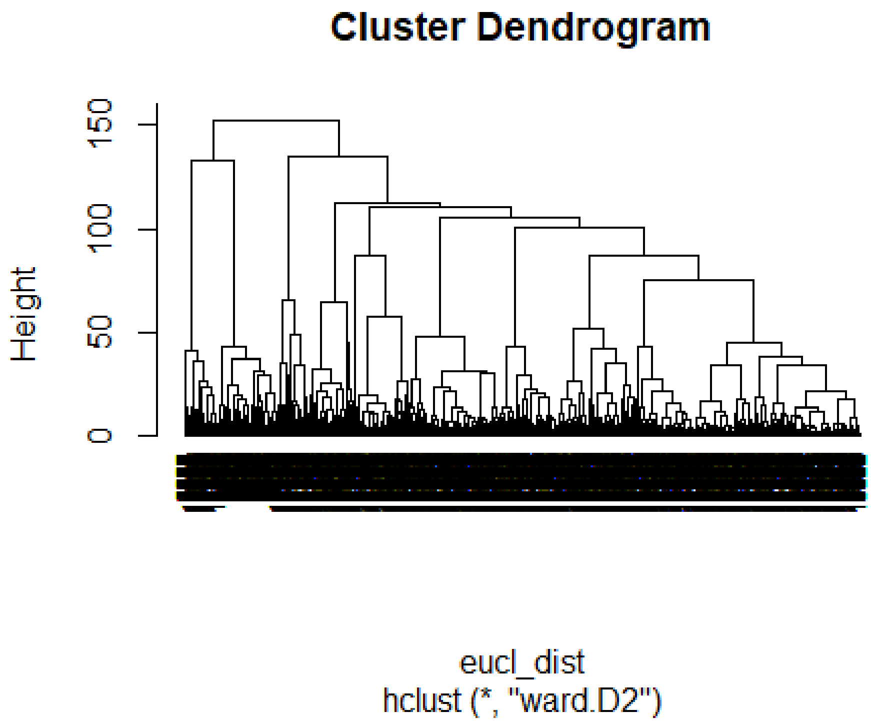
```
#Check the size of each cluster
kmeans_fancy $size

## [1] 8885 1561 1753
```

The first cluster has 8885 values, second has 1561 and third has 1753 values

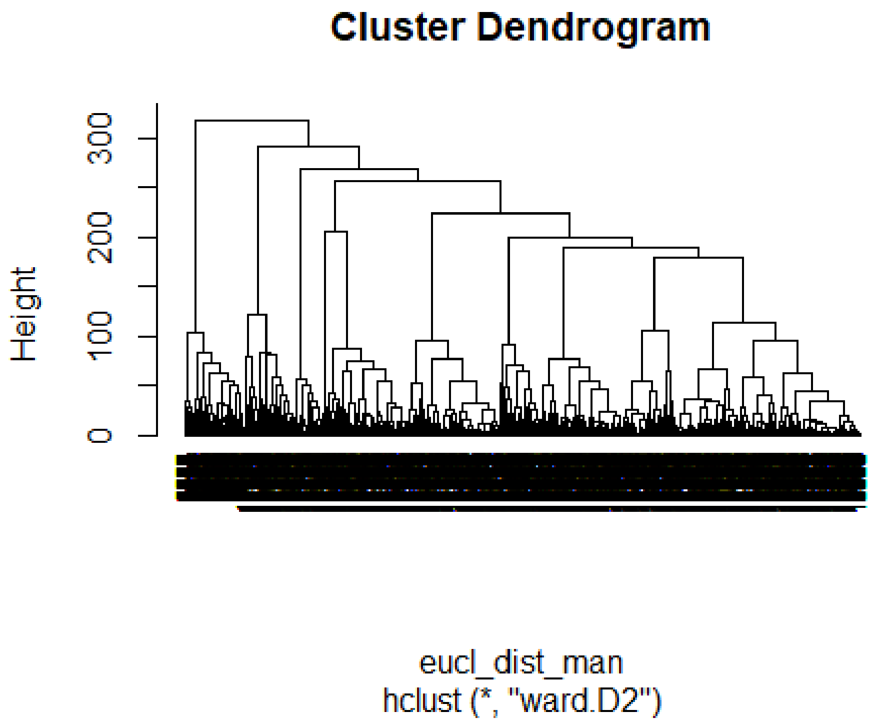
```
# check their response to revenue
table(kmeans_fancy$cluster, df_new$Revenue)

##
##      FALSE TRUE
##    1  7519 1366
##    2  1506   55
##    3  1266  487

data %>%
  mutate(Cluster = kmeans_fancy$cluster) %>%
  group_by(Cluster) %>%
  summarize_all('median')

## # A tibble: 3 x 14
##   Cluster Administrative Informational ProductRelated BounceRates
PageValues
##     <int>          <dbl>         <dbl>          <dbl>       <dbl>
<dbl>
## 1       1              1             0             16           0          0
## 2       2              0             0              5        0.07          0
## 3       3              7             2             70        0
2.09
## # ... with 8 more variables: SpecialDay <dbl>, Month <chr>,
## #   OperatingSystems <dbl>, Browser <dbl>, Region <int>, TrafficType
<dbl>,
## #   VisitorType <chr>, Weekend <lgl>
```

Hierarchical clustering

```
library(cluster)

## Warning: package 'cluster' was built under R version 4.0.5

# compute the euclidean distance using euclidean metric
eucl_dist<- dist(data_norm, method = "euclidean")
#compute hierarchical clustering using the Ward method
res_hc<- hclust(eucl_dist, method =  "ward.D2")
res_hc

##
## Call:
## hclust(d = eucl_dist, method = "ward.D2")
##
## Cluster method   : ward.D2
## Distance         : euclidean
## Number of objects: 12199

# plot the obtained dendrogram
plot(res_hc, cex = 0.6, hang = -1)
```



```
# compute the euclidean distance using manhattan metric
eucl_dist_man<- dist(data_norm, method = "manhattan")
#compute hierarchical clustering using the Ward method
res_hc_man<- hclust(eucl_dist_man, method =  "ward.D2")
res_hc_man

##
## Call:
## hclust(d = eucl_dist_man, method = "ward.D2")
##
## Cluster method   : ward.D2
## Distance         : manhattan
## Number of objects: 12199

# plot the obtained dendrogram
plot(res_hc_man, cex = 0.6, hang = -1)
```


Cluster Dendrogram
Height
0
100
200
300
eucl_dist_man
hclust (*, "ward.D2")