CONSUMER BEHAVIOR ANALYSIS

Objective

The Sales and Marketing team of Kira Plastinina Brand seeks to understand their customer's behavior from data collected from chain of retail stores in Russia, Ukraine, Kazakhstan, Belarus, China, Philippines, and Armenia.

Experimental Design

1.Exploratory Data Analysis
2.Modeling with KMeans and Hierarchical Clustering

```
#Loading libraries
packages<-function(x){
  x<-as.character(match.call()[[2]])
  if (!require(x,character.only=TRUE)){
    install.packages(pkgs=x,repos="http://cran.r-project.org")
    require(x,character.only=TRUE)
  }
}

packages(tidyverse) # data manipulation
## Loading required package: tidyverse
## -- Attaching packages ---------------------------------------
tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
packages(corrplot)
## Loading required package: corrplot
## corrplot 0.92 loaded
packages(gridExtra)
```

```
## Loading required package: gridExtra
##
## Attaching package: 'gridExtra'
## The following object is masked from 'package:dplyr':
##
##     combine
packages(GGally)
## Loading required package: GGally
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
packages(cluster) # clustering algorithms
## Loading required package: cluster
packages(factoextra)
## Loading required package: factoextra
## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa
```

Loading and Previewing Dataset

```
consumer_df <- read.csv("http://bit.ly/EcommerceCustomersDataset")
rmarkdown::paged_table(head(consumer_df,n=5))
rmarkdown::paged_table(tail(consumer_df,n=5))
#Checking the shape of the dataset
dim(consumer_df)
## [1] 12330    18
#the dataset has 18 columns and 12330 rows
#Checking the column names
colnames(consumer_df)
##  [1] "Administrative"          "Administrative_Duration"
##  [3] "Informational"           "Informational_Duration"
##  [5] "ProductRelated"          "ProductRelated_Duration"
##  [7] "BounceRates"             "ExitRates"
##  [9] "PageValues"              "SpecialDay"
## [11] "Month"                   "OperatingSystems"
## [13] "Browser"                 "Region"
## [15] "TrafficType"             "VisitorType"
## [17] "Weekend"                 "Revenue"
#Checking data types
sapply(consumer_df,class)
##          Administrative Administrative_Duration
Informational
##               "integer"               "numeric"
"integer"
##  Informational_Duration          ProductRelated
```

```
ProductRelated_Duration
##               "numeric"               "integer"
"numeric"
##             BounceRates               ExitRates
PageValues
##               "numeric"               "numeric"
"numeric"
##              SpecialDay                   Month
OperatingSystems
##               "numeric"             "character"
"integer"
##                 Browser                  Region
TrafficType
##               "integer"               "integer"
"integer"
##             VisitorType                 Weekend
Revenue
##             "character"               "logical"
"logical"
```

Data Cleaning

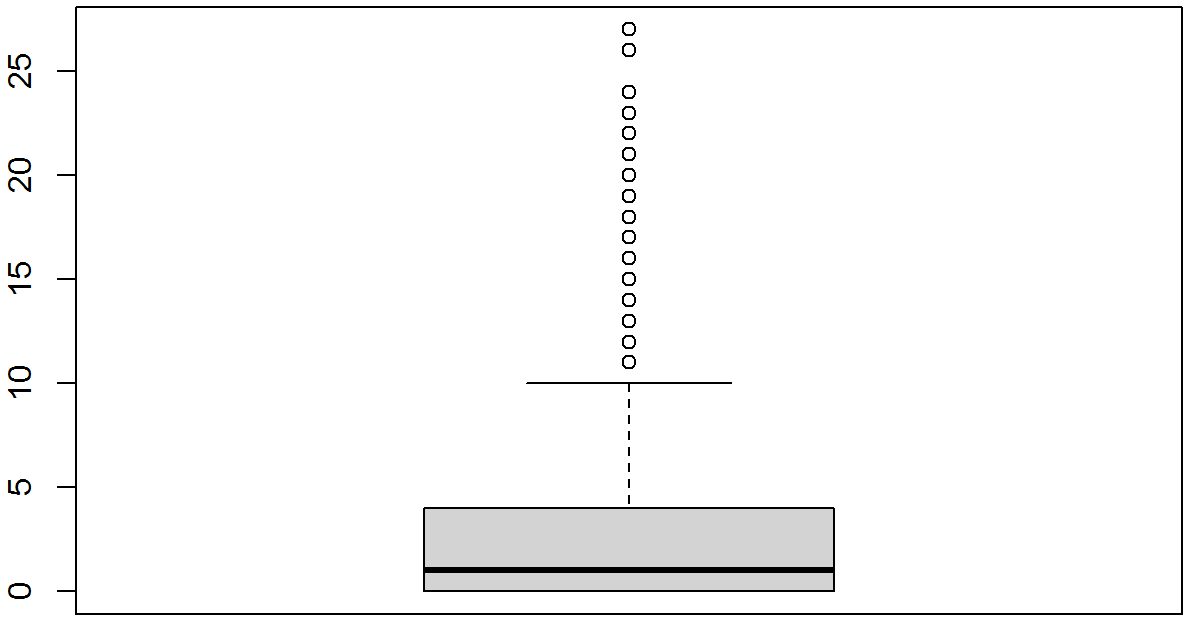
Checking for duplicated values

```
sum(duplicated(consumer_df))
## [1] 119
#Removing duplicates
consumerdf <-consumer_df[!duplicated(consumer_df), ]
#Checking if duplicates have been dropped successfully
sum(duplicated(consumerdf))
## [1] 0
```

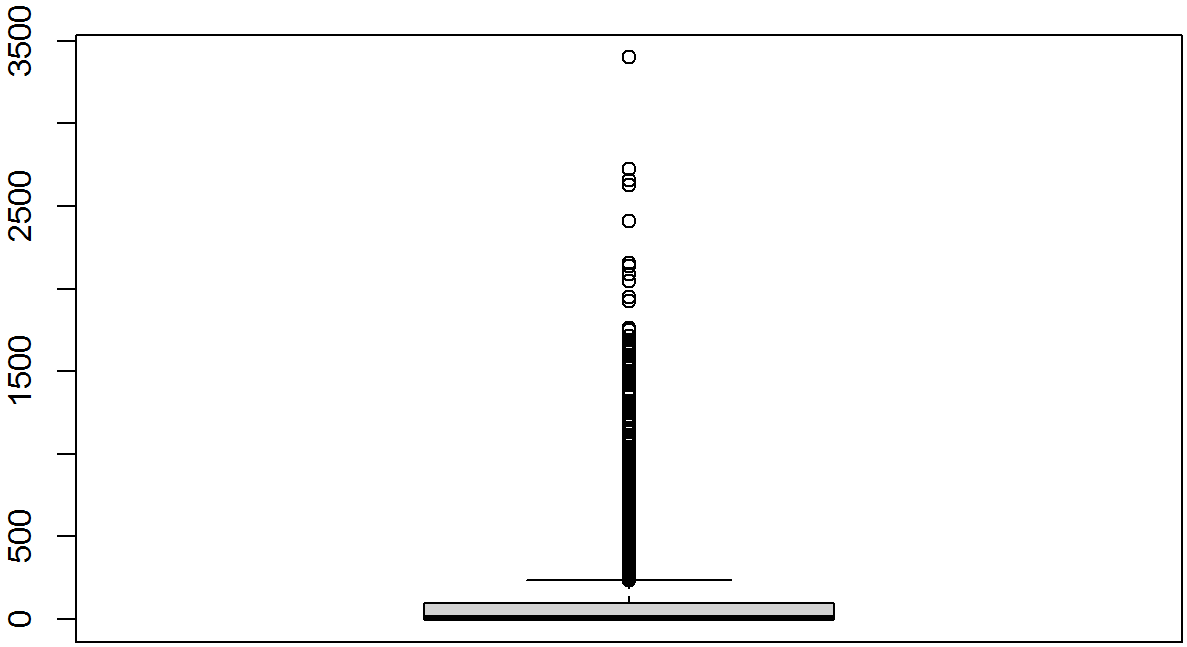
Dealing with missing values

```
colSums(is.na(consumerdf))
##          Administrative Administrative_Duration
Informational
##                      12                      12
12
##  Informational_Duration          ProductRelated
ProductRelated_Duration
##                      12                      12
12
##             BounceRates               ExitRates
PageValues
##                      12                      12
0
##              SpecialDay                   Month
```

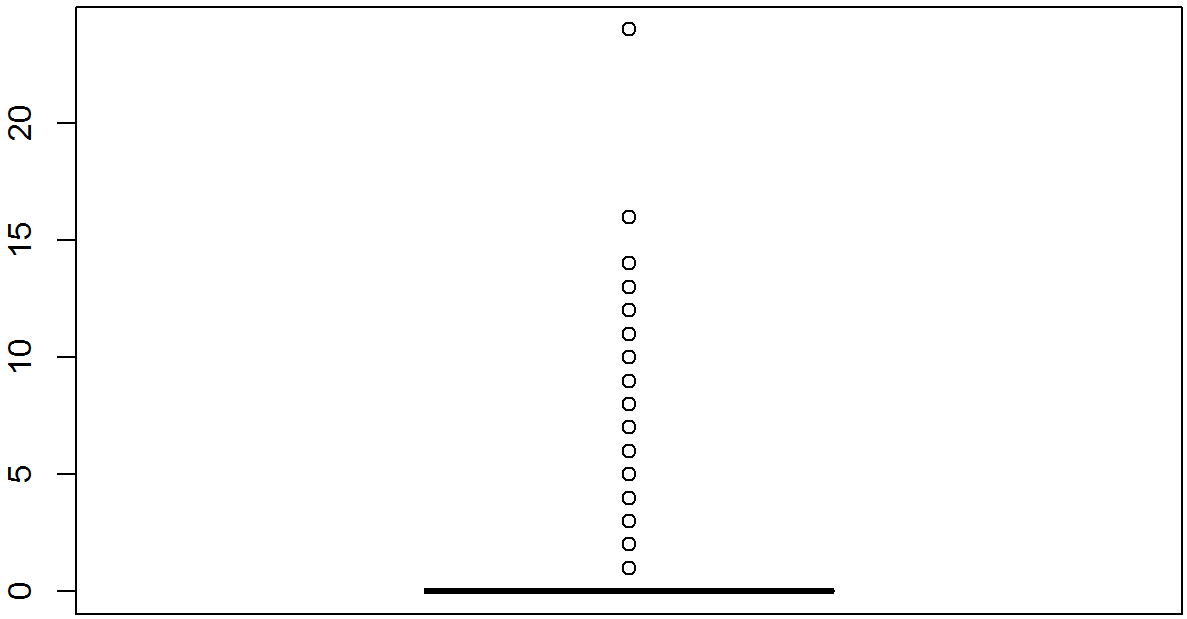
```
OperatingSystems
##                         0                         0
0
##                   Browser                    Region
TrafficType
##                         0                         0
0
##              VisitorType                   Weekend
Revenue
##                         0                         0
0
#Removing missing values
consumerdf <-na.omit(consumerdf,)
#Checking if missing values have been successfully dropped
sum(is.na(consumerdf))
## [1] 0
#Checking the shape of the data
dim(consumerdf)
## [1] 12199    18
boxplot(consumerdf$Administrative)
```



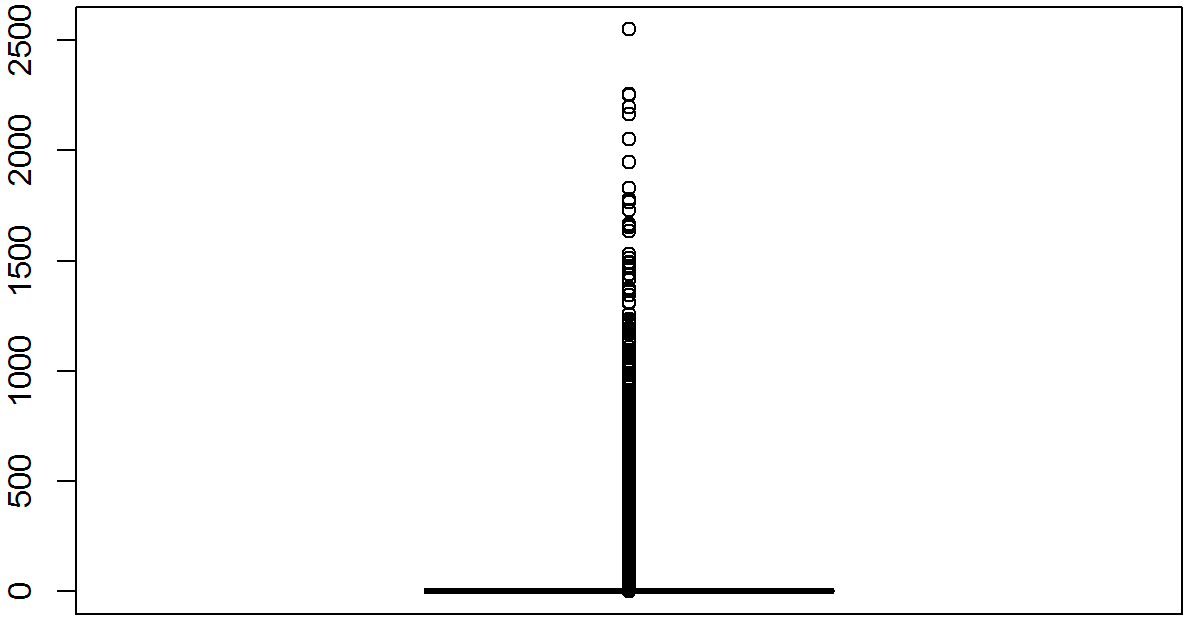

```
boxplot(consumerdf$Administrative_Duration)
```



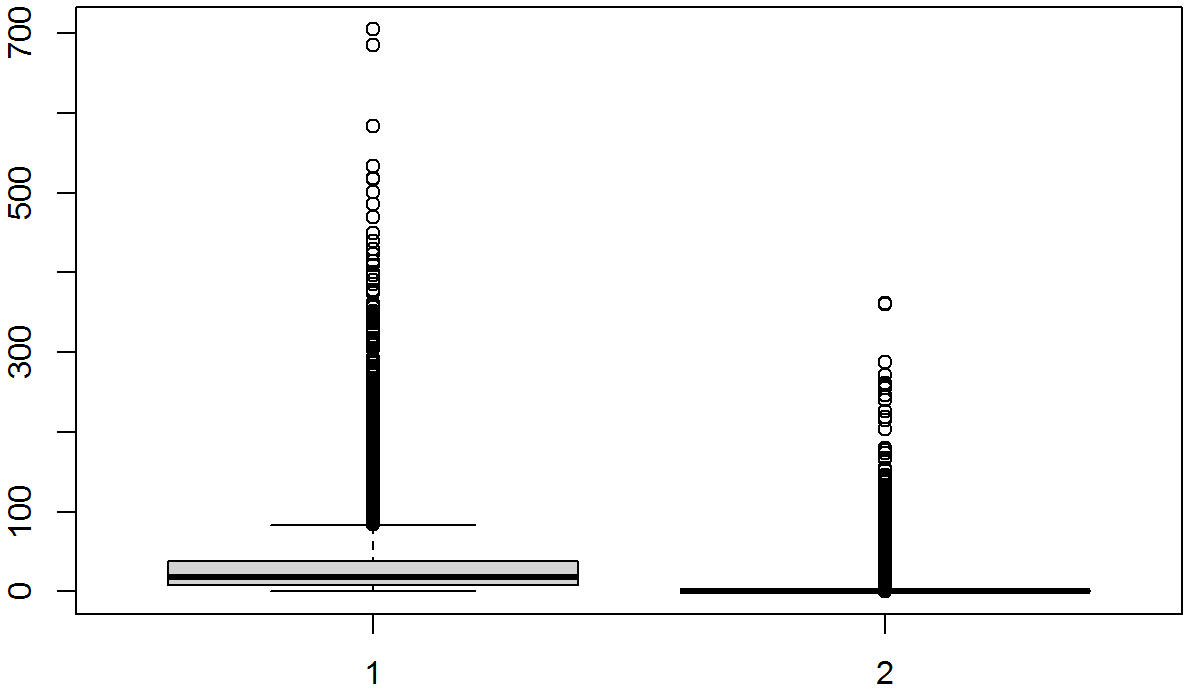

```
boxplot(consumerdf$Informational)
```



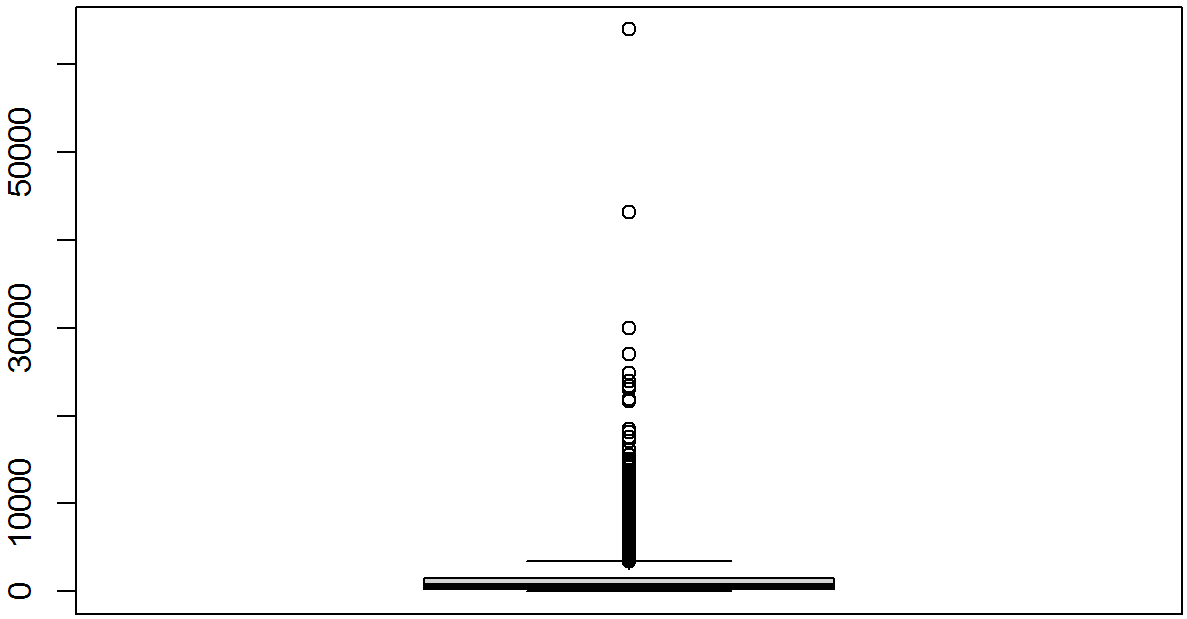

```
boxplot(consumerdf$Informational_Duration)
```



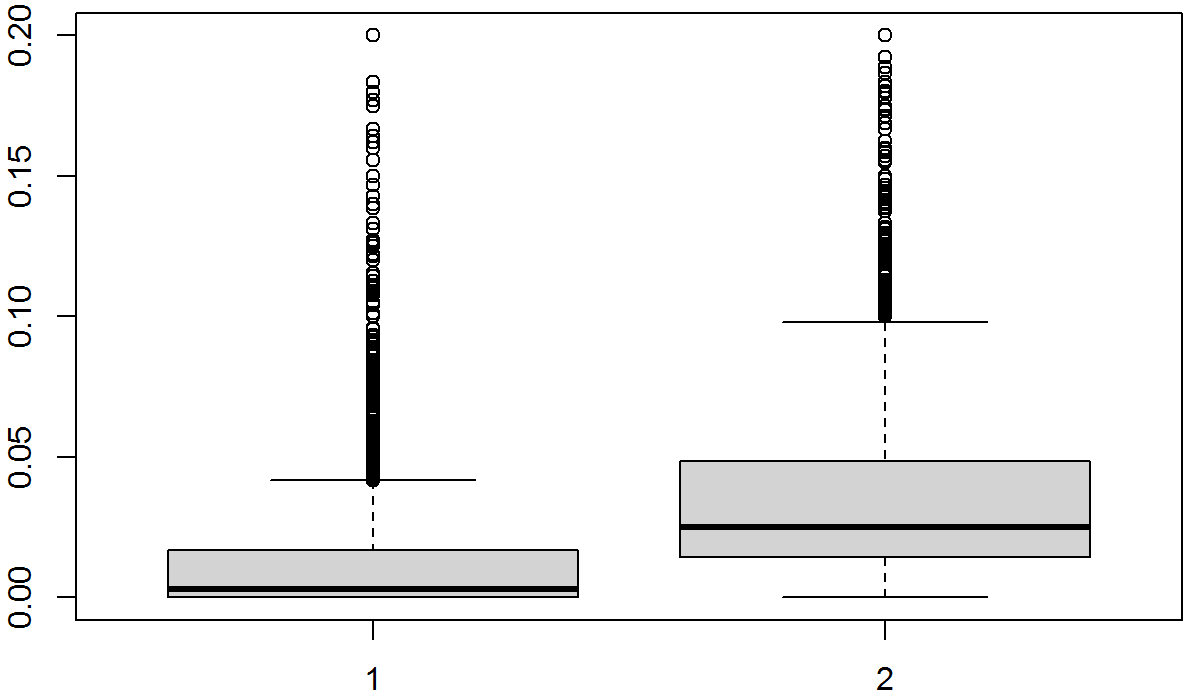

```
boxplot(consumerdf$ProductRelated,consumerdf$PageValues)
```



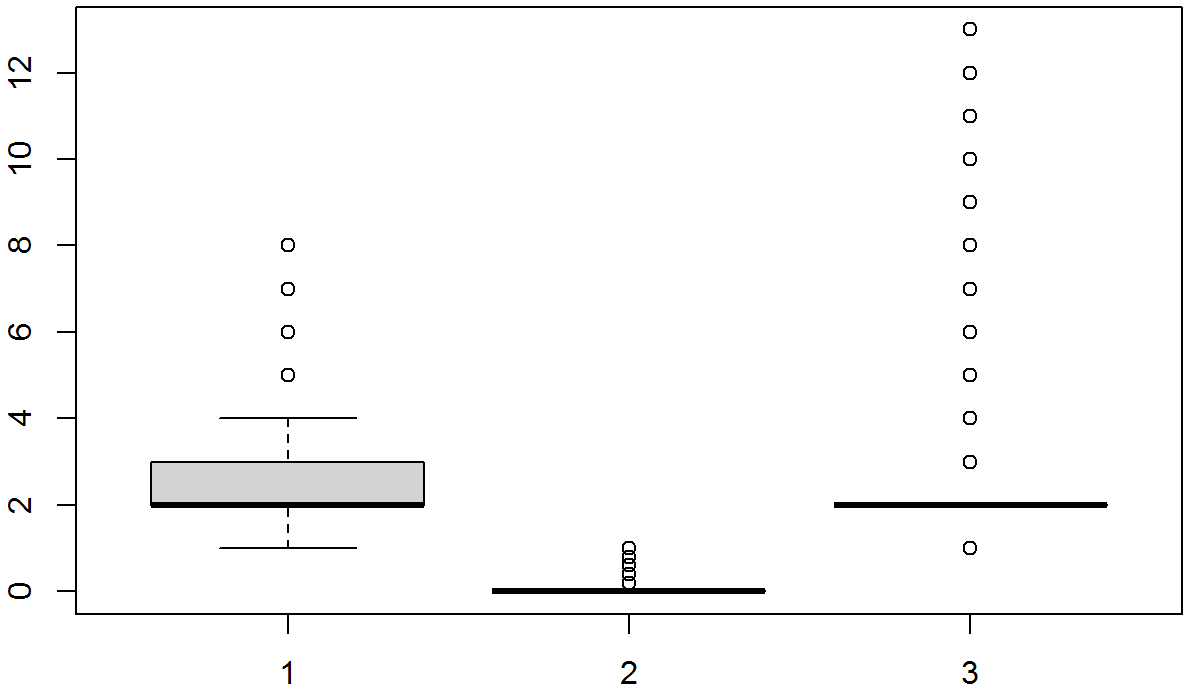

```
boxplot(consumerdf$ProductRelated_Duration)
```




```
boxplot(consumerdf$BounceRates,consumerdf$ExitRates)
```




```
boxplot(consumerdf$OperatingSystems,consumerdf$SpecialDay,consumerdf$B
rowser)
```



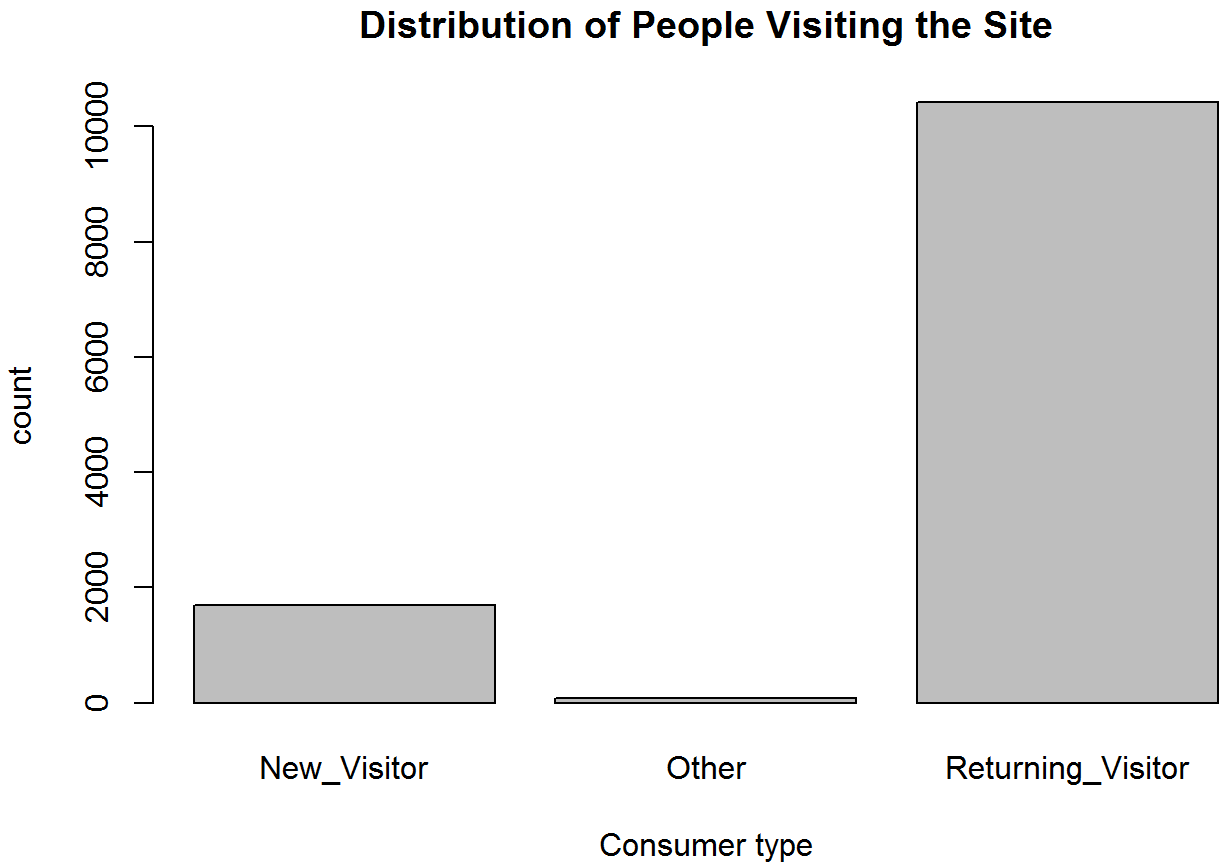
There is presence of outliers, however, the are legitimate data point and therefore will not be dropped. **Checking for anomalies**

```
unique(consumerdf$Revenue)
## [1] FALSE  TRUE
unique(consumerdf$Weekend)
## [1] FALSE  TRUE
unique(consumerdf$Month)
##  [1] "Feb"  "Mar"  "May"  "Oct"  "June" "Jul"  "Aug"  "Nov"  "Sep"
"Dec"
#Months January and April are not represented in the data
unique(consumerdf$Region)
## [1] 1 9 2 3 4 5 6 7 8
unique(consumerdf$VisitorType)
## [1] "Returning_Visitor" "New_Visitor"       "Other"
unique(consumerdf$TrafficType)
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 18 19 16 17 20
```

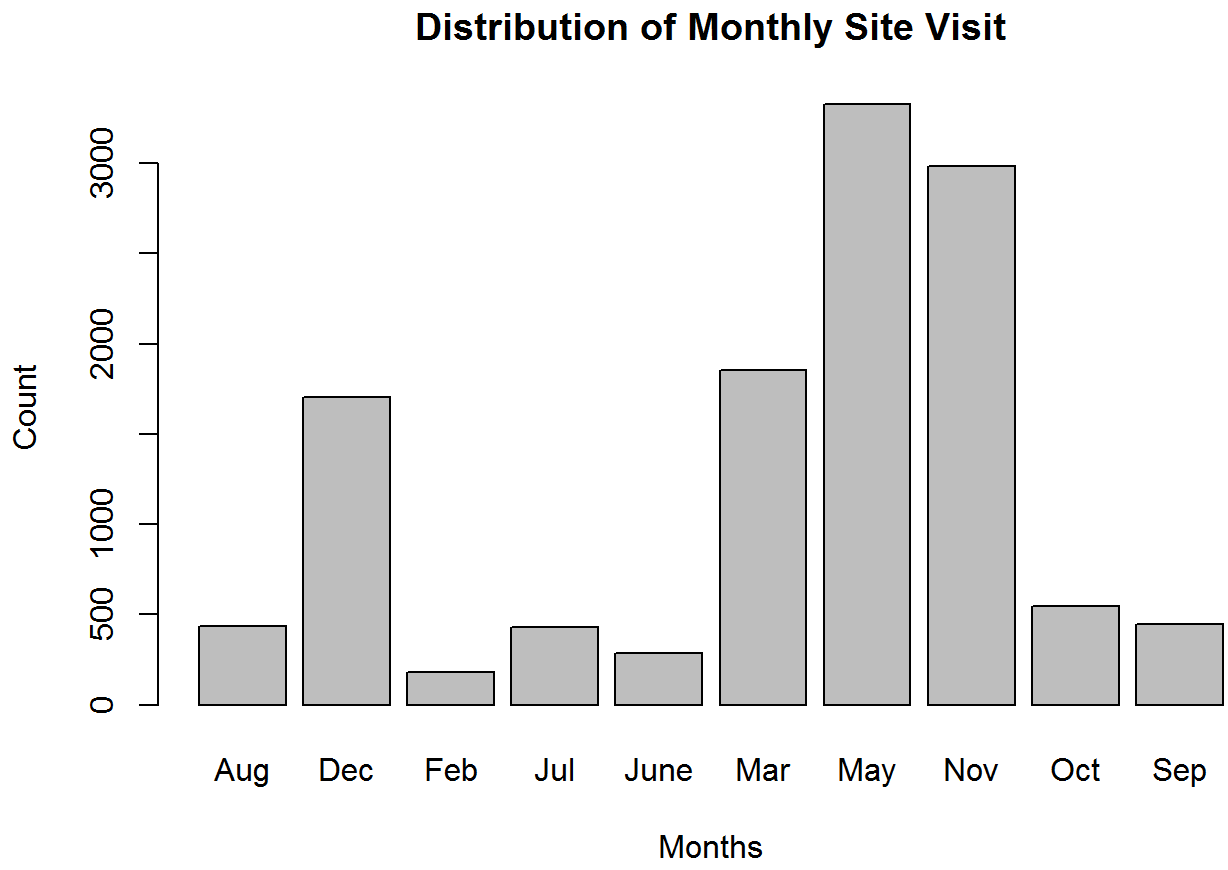
Exploratory Data Analysis

Univariate Analysis

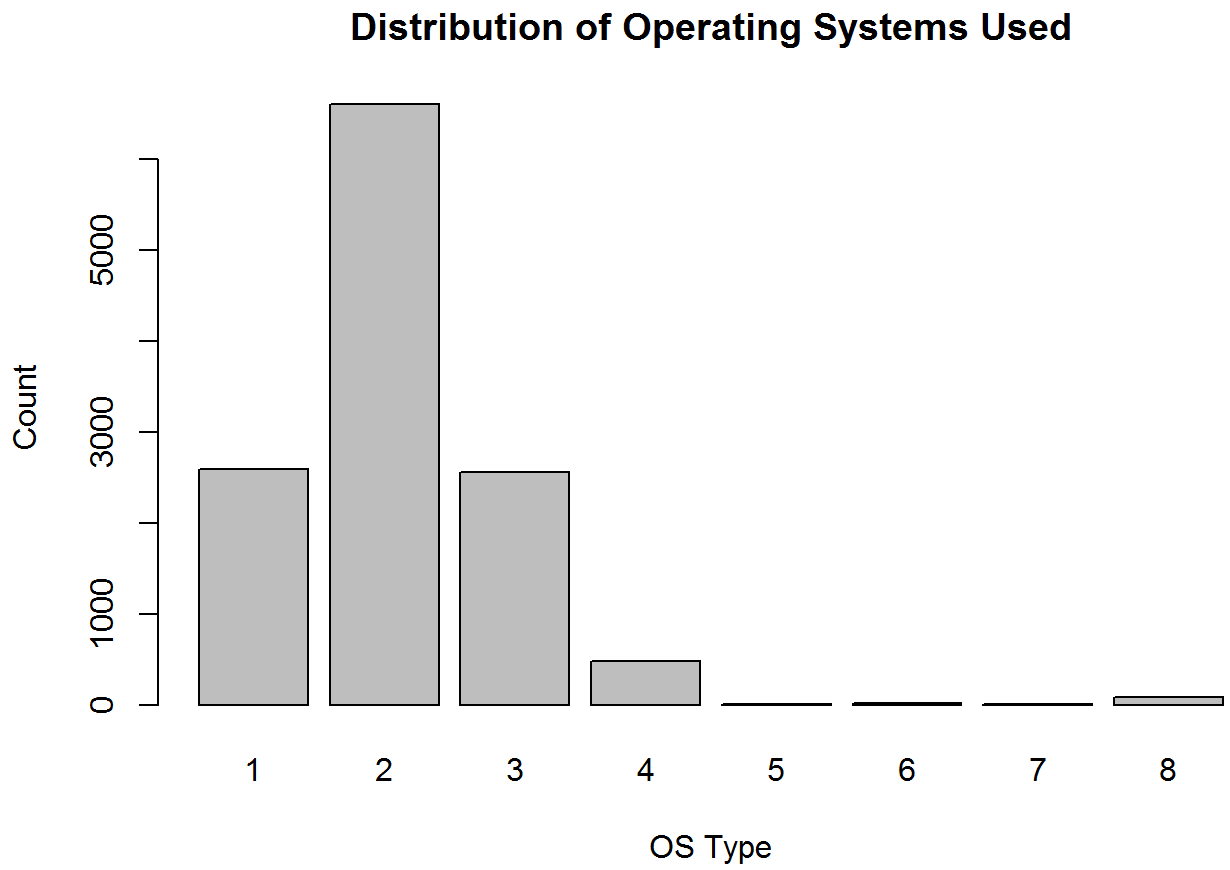
```
summary(consumerdf[,c(2,4,6,7,8)])
##  Administrative_Duration Informational_Duration
ProductRelated_Duration
##  Min.   :  -1.00         Min.   :  -1.00        Min.   :   -1.0
##  1st Qu.:   0.00         1st Qu.:   0.00        1st Qu.:  193.6
##  Median :   9.00         Median :   0.00        Median :  609.5
##  Mean   :  81.68         Mean   :  34.84        Mean   : 1207.5
##  3rd Qu.:  94.75         3rd Qu.:   0.00        3rd Qu.: 1477.6
##  Max.   :3398.75         Max.   :2549.38        Max.   :63973.5
##   BounceRates        ExitRates
##  Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.01422
##  Median :0.00293   Median :0.02500
##  Mean   :0.02045   Mean   :0.04150
##  3rd Qu.:0.01667   3rd Qu.:0.04848
##  Max.   :0.20000   Max.   :0.20000
x=table(consumerdf$VisitorType)
barplot(x, xlab="Consumer type",ylab="count",main = "Distribution of
People Visiting the Site")
```



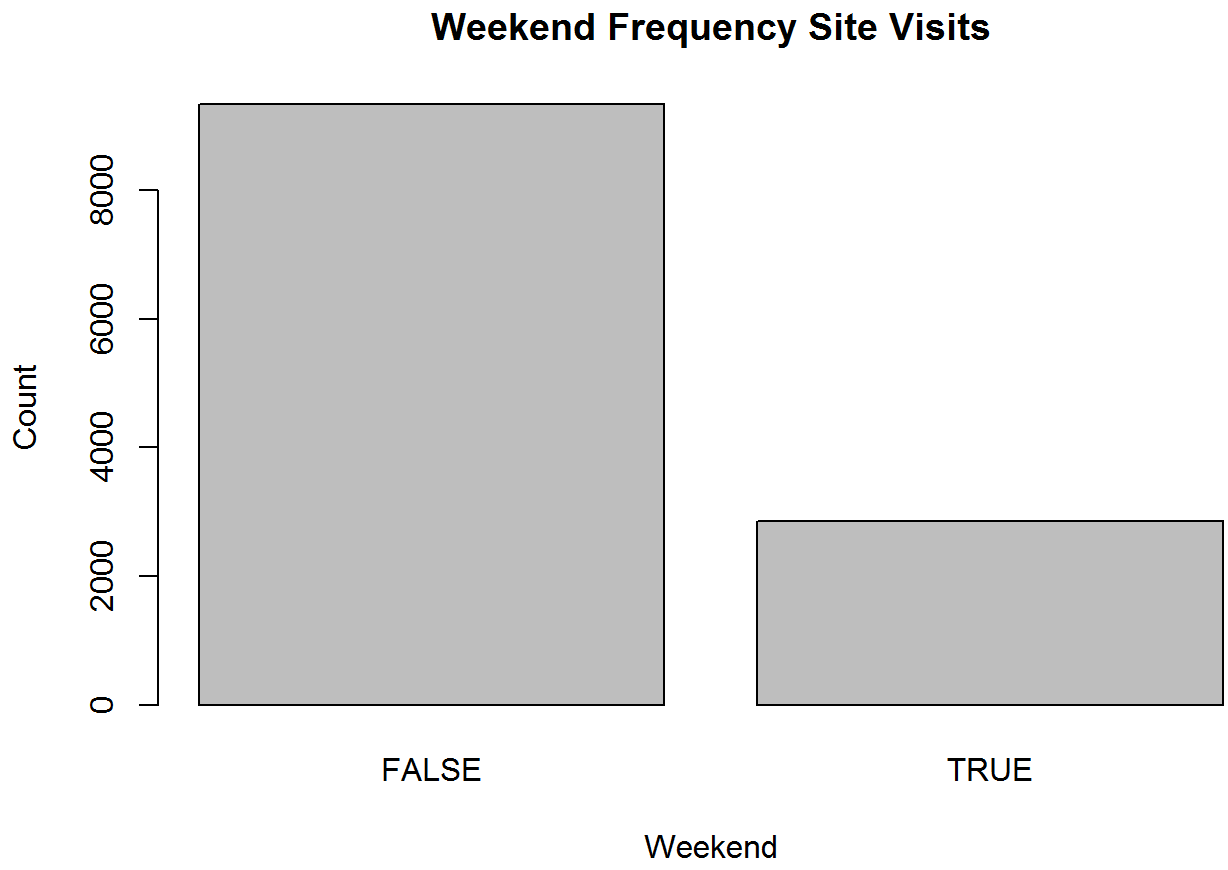

```
#Majority of the respondents were return visitors to the site
x=table(consumerdf$Month)
barplot(x,xlab="Months",ylab="Count", main="Distribution of Monthly
Site Visit")
```



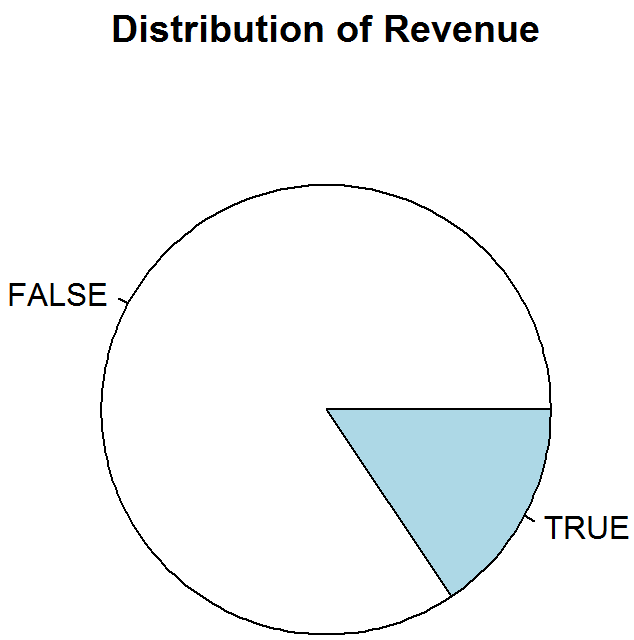

```
#More people visited the sites in the months of May, November, March
and December
x=table(consumer_df$OperatingSystems)
barplot(x,xlab="OS Type",ylab="Count",main="Distribution of Operating
Systems Used")
```



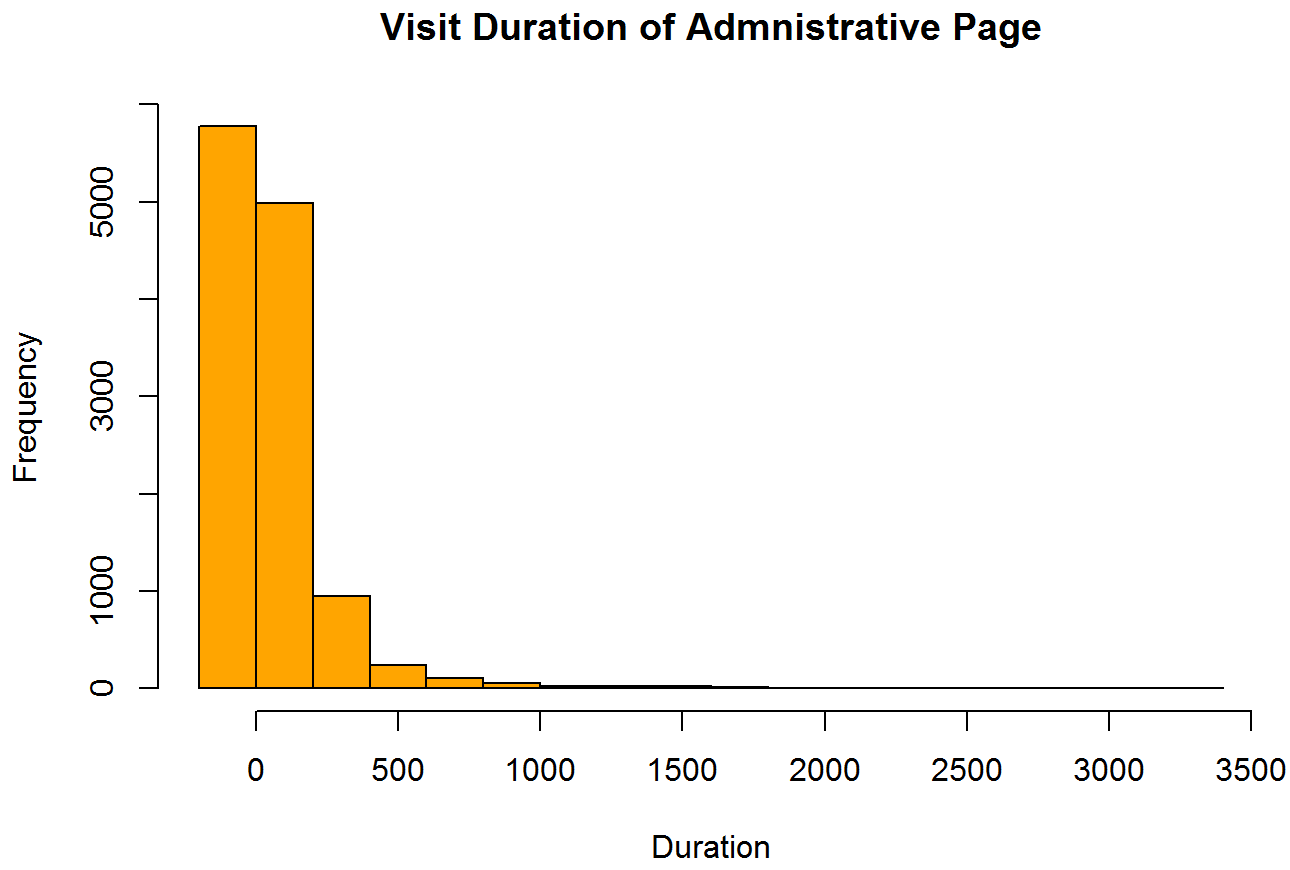

```
# OS type 2 Users visited the site more conpared to OS types 5 & 6
x=table(consumerdf$Weekend)
barplot(x,xlab = "Weekend",ylab="Count",main = "Weekend Frequency Site
Visits")
```



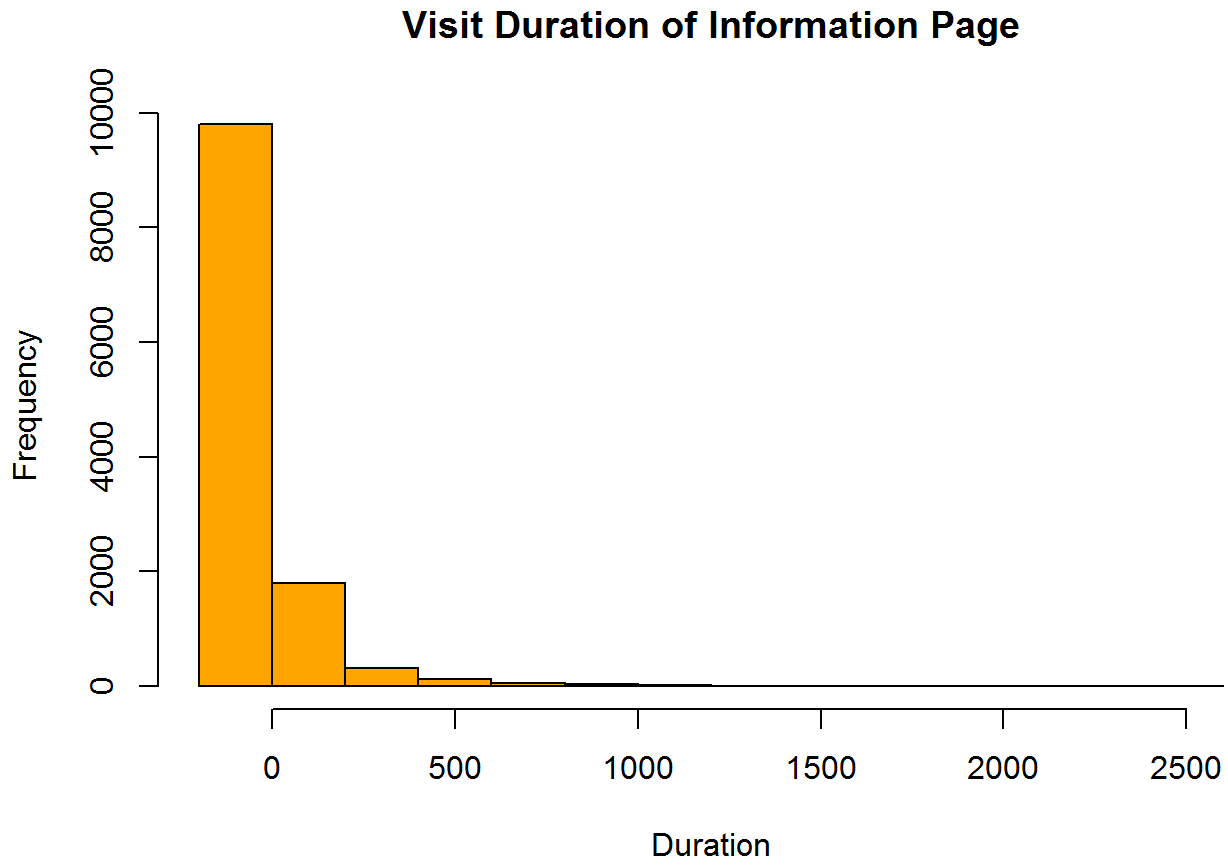

```
#More people visited the sites during weekdays compared to weekends.
However it cannot be comparable since the are more weekdays than there
are weekends
x=table(consumerdf$Revenue)
pie(x,main = "Distribution of Revenue")
```



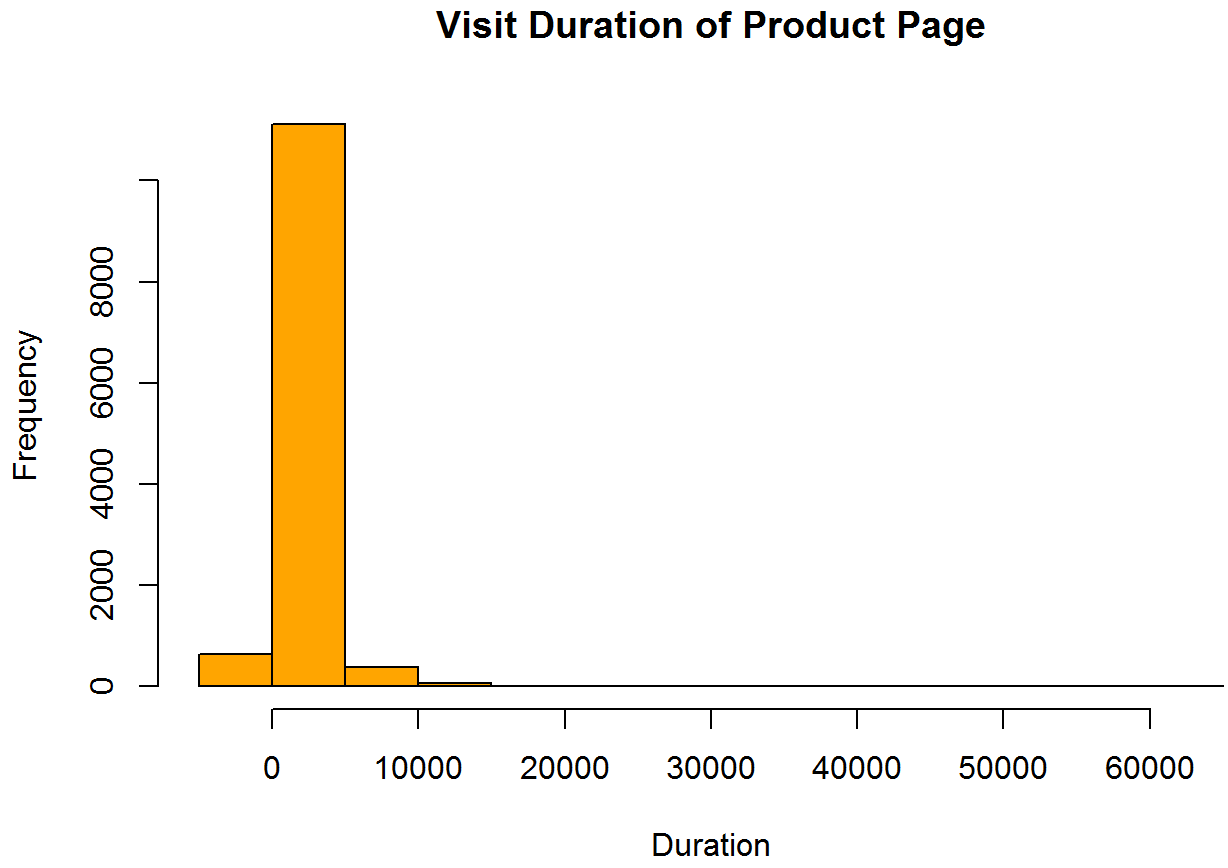

```
#Despite visiting the site, few people ended up spending on the
products
table(consumerdf$Revenue)
##
## FALSE  TRUE
## 10291  1908
hist(consumerdf$Administrative_Duration,main = "Visit Duration of
Admnistrative Page", xlab = "Duration",col = "Orange")
```




```
hist(consumerdf$Informational_Duration,main = "Visit Duration of
Information Page", xlab = "Duration",col = "Orange")
```



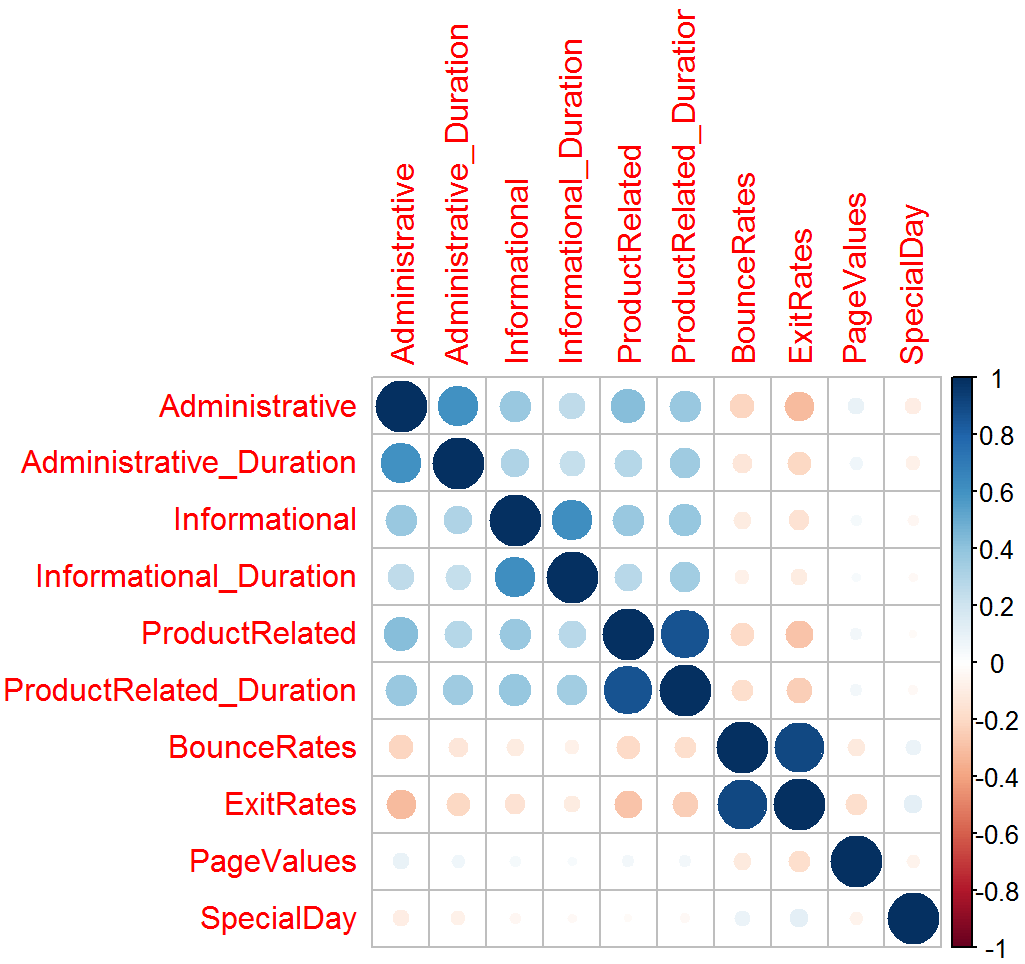

```
hist(consumerdf$ProductRelated_Duration ,main = "Visit Duration of
Product Page", xlab = "Duration",col = "Orange")
```



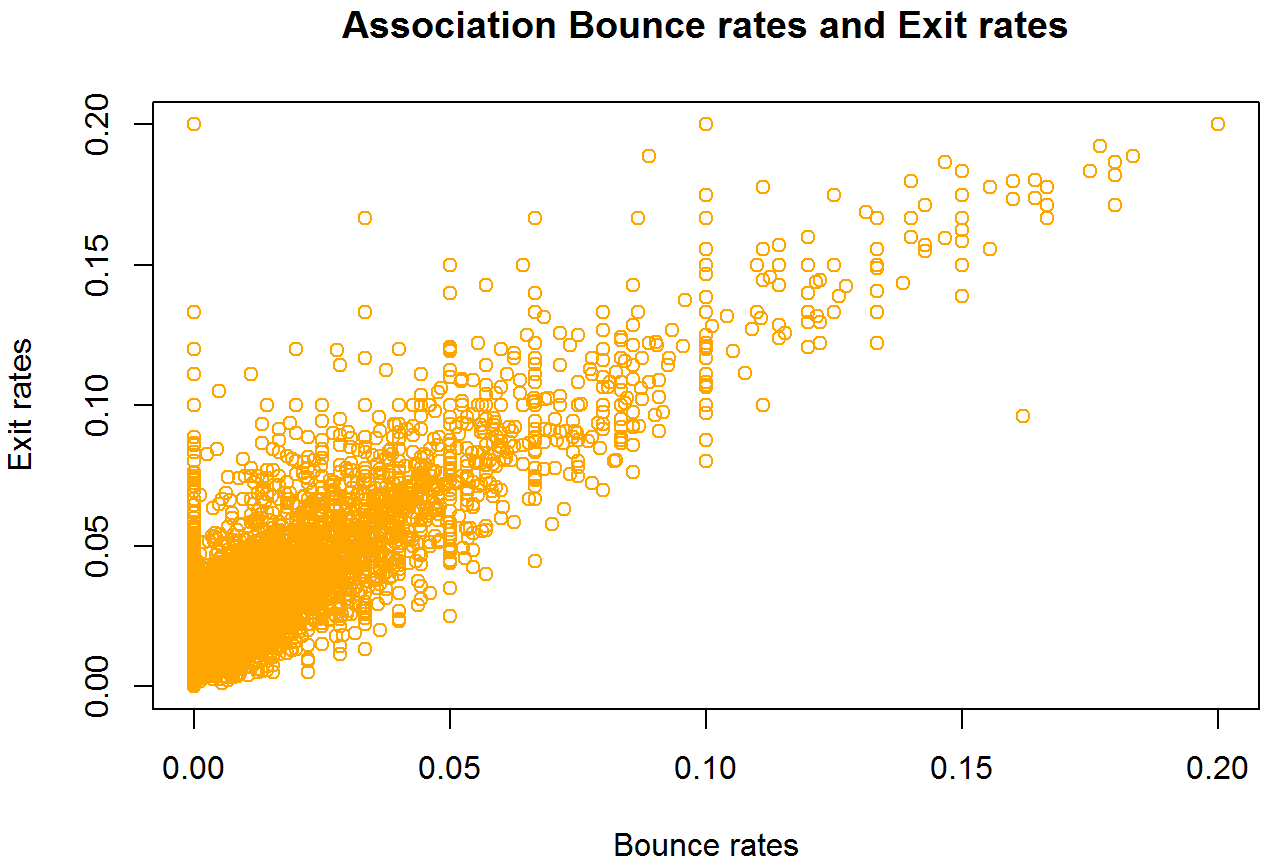
Bivariate Analysis

```
r <-table(consumerdf$Revenue)
t <-table(consumerdf$Weekend)
barplot(r,t,height = r)
```

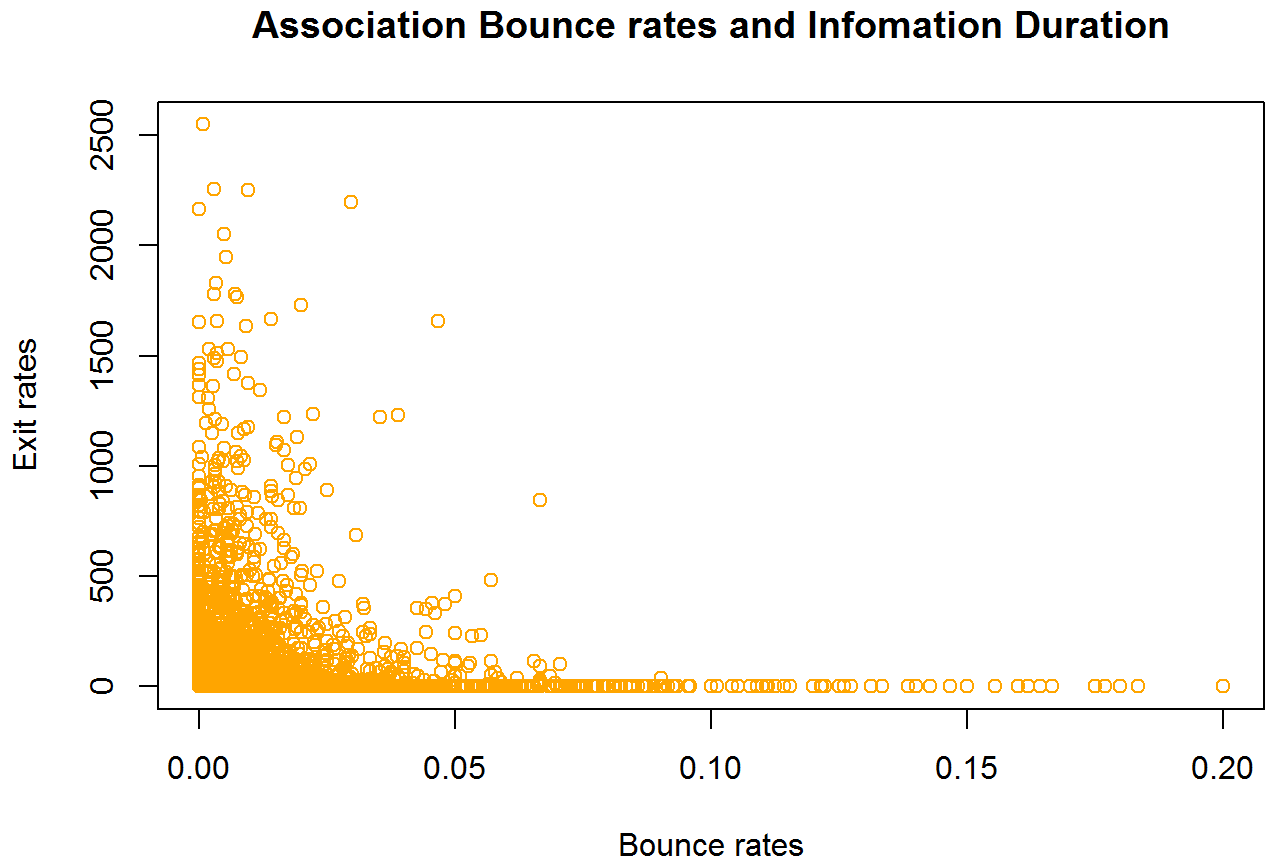

```
library(corrplot)
corrplot(cor(consumerdf[,c(1,2,3,4,5,6,7,8,9,10)]))
```



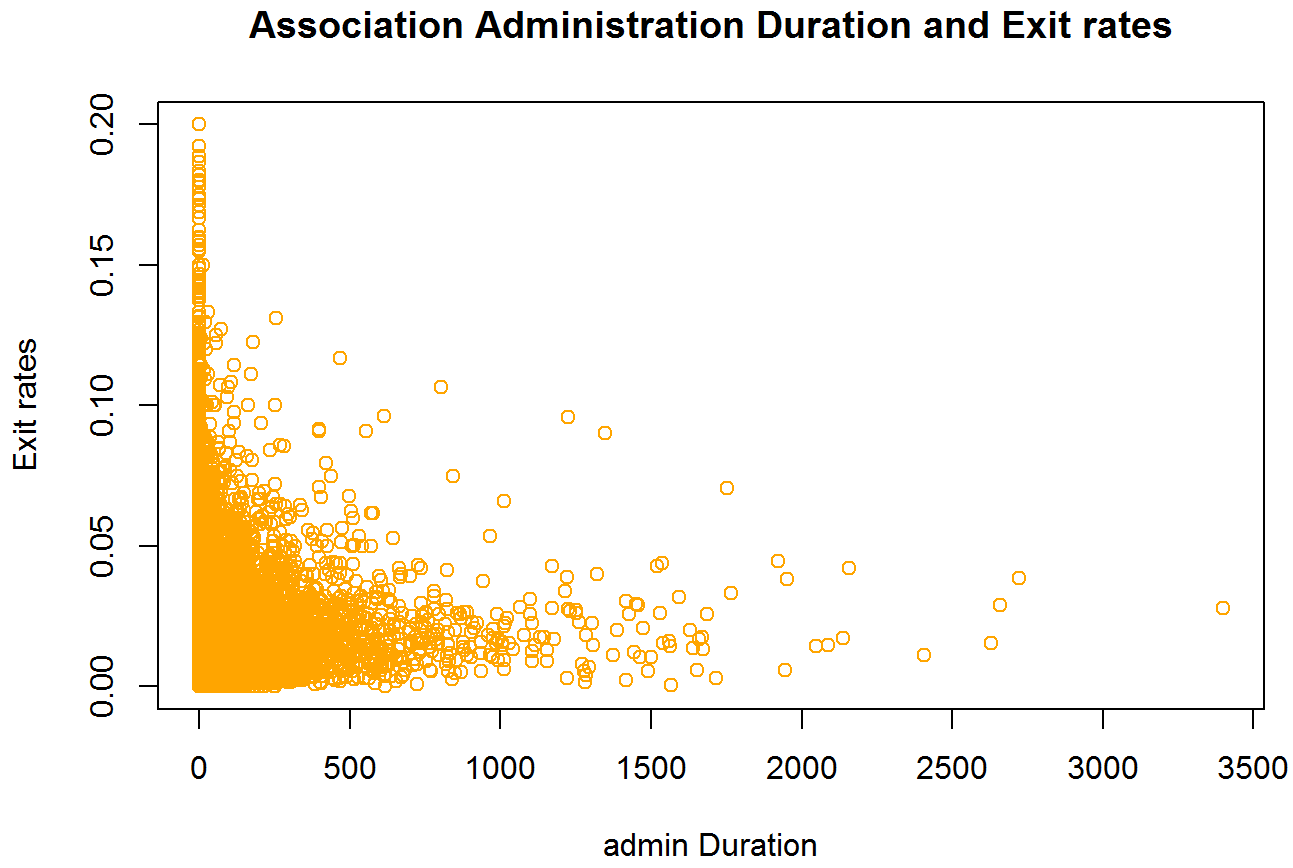

```
plot(consumer_df$BounceRates,consumer_df$ExitRates,xlab = "Bounce
rates",ylab = "Exit rates",main="Association Bounce rates and Exit
rates",col = "orange")
```



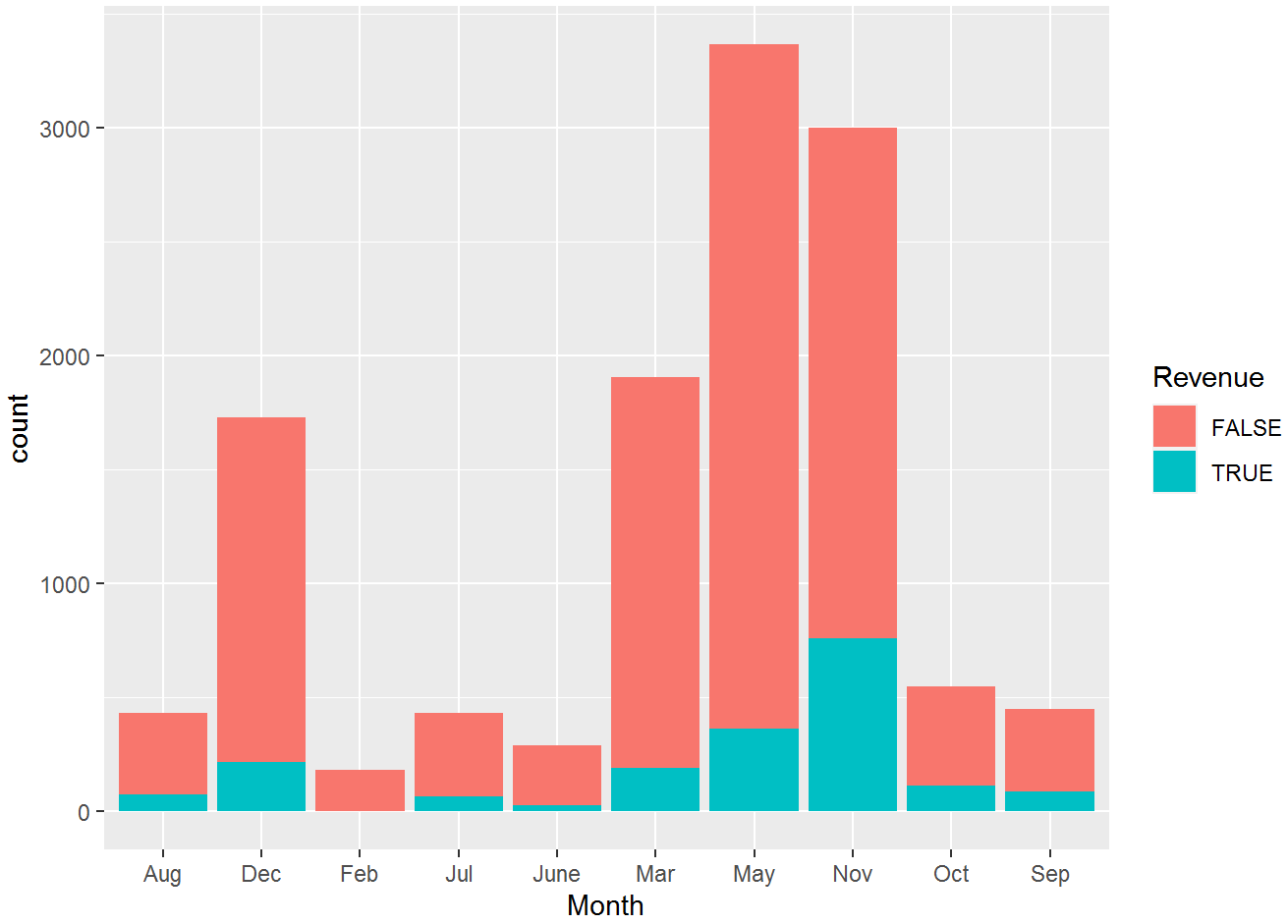

```
#There is a positive corrilation between bounce and exit rates
plot(consumer_df$BounceRates,consumer_df$Informational_Duration,xlab =
"Bounce rates",ylab = "Exit rates",main="Association Bounce rates and
Infomation Duration",col = "orange")
```



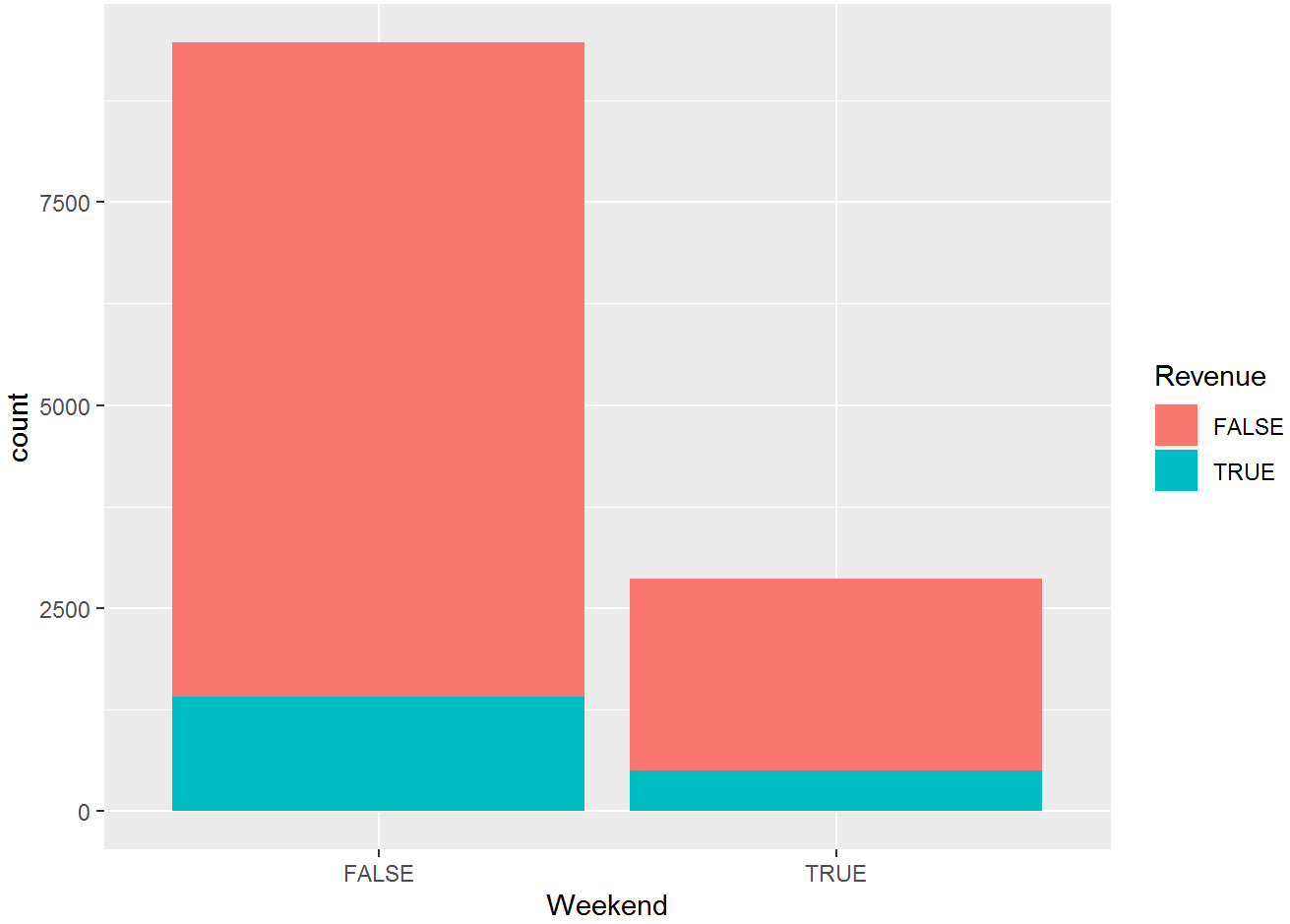

```
plot(consumer_df$Administrative_Duration,consumer_df$ExitRates,xlab =
"admin Duration",ylab = "Exit rates",main="Association Administration
Duration and Exit rates",col = "orange")
```




```
library(ggplot2)
ggplot(consumer_df, aes(x = Month,fill =Revenue),
       title(main ="Month vs revenue status" )) +
  geom_bar()
```




```
ggplot(consumer_df, aes(x = Weekend,fill =Revenue),
       title(main ="Product Site vs revenue status" )) +
  geom_bar()
```



Modeling ### Data Preparation

```
#Dropping the label column
consumerdf$Revenue <- NULL
colnames(consumerdf)
##  [1] "Administrative"          "Administrative_Duration"
##  [3] "Informational"           "Informational_Duration"
##  [5] "ProductRelated"          "ProductRelated_Duration"
##  [7] "BounceRates"             "ExitRates"
##  [9] "PageValues"              "SpecialDay"
## [11] "Month"                   "OperatingSystems"
## [13] "Browser"                 "Region"
## [15] "TrafficType"             "VisitorType"
## [17] "Weekend"
#Encoding character data typed columns
consumerdf$Month <- factor(consumerdf$Month)
consumerdf$Month <- as.numeric(consumerdf$Month)
consumerdf$VisitorType <- as.factor(consumerdf$VisitorType)
consumerdf$VisitorType <- as.numeric(consumerdf$VisitorType)
consumerdf$Weekend <- ifelse(consumerdf$Weekend==FALSE,0,1)
head(consumerdf)
##   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
VisitorType
## 1          0     3                 1       1      1           1
3
## 2          0     3                 2       2      1           2
3
## 3          0     3                 4       1      9           3
3
## 4          0     3                 3       2      2           4
3
## 5          0     3                 3       3      1           4
3
## 6          0     3                 2       2      1           3
3
##   Weekend
## 1       0
## 2       0
## 3       0
## 4       0
## 5       1
## 6       0
```

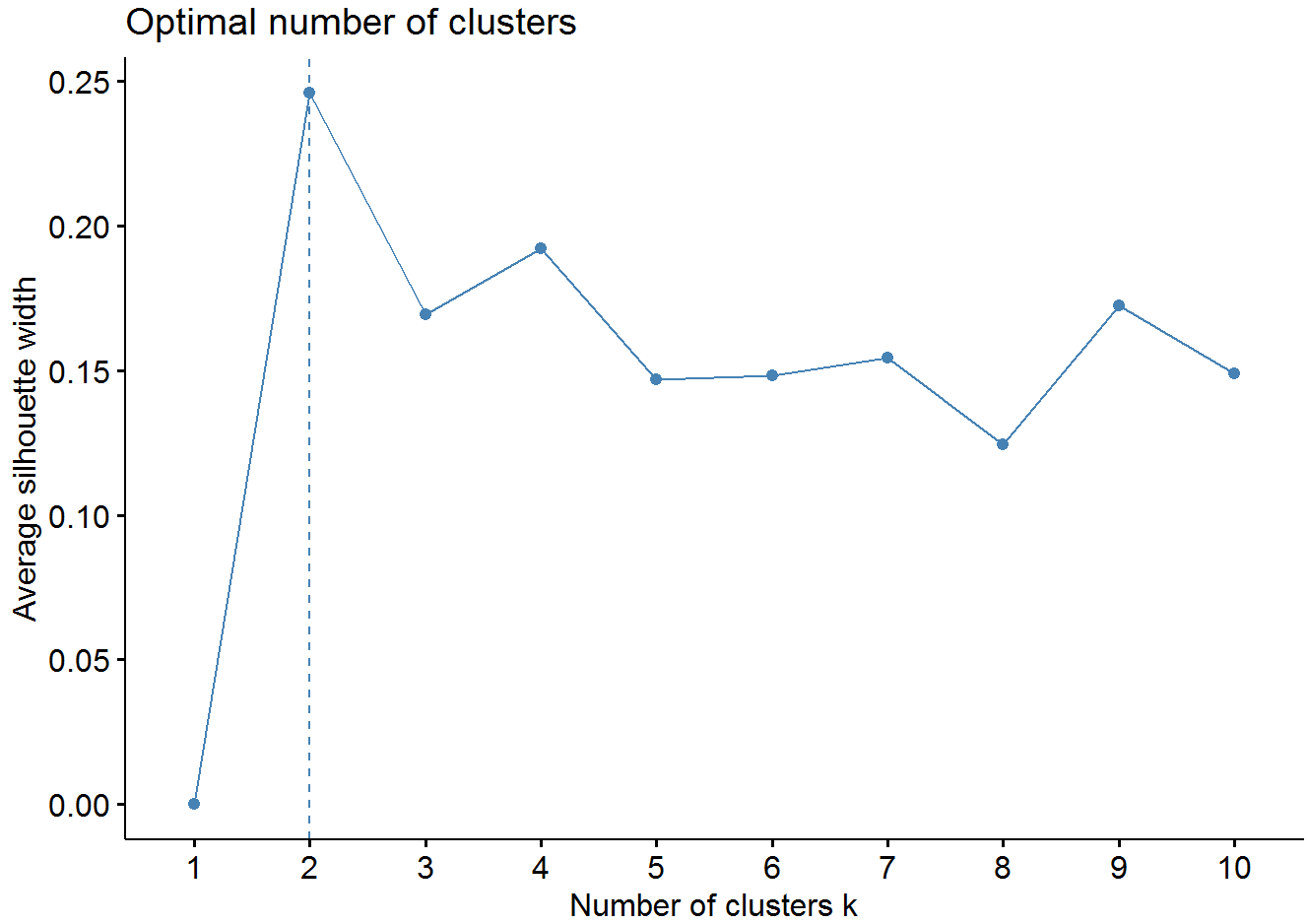
Data scaling

```
modeling.data <-scale(consumerdf)
```

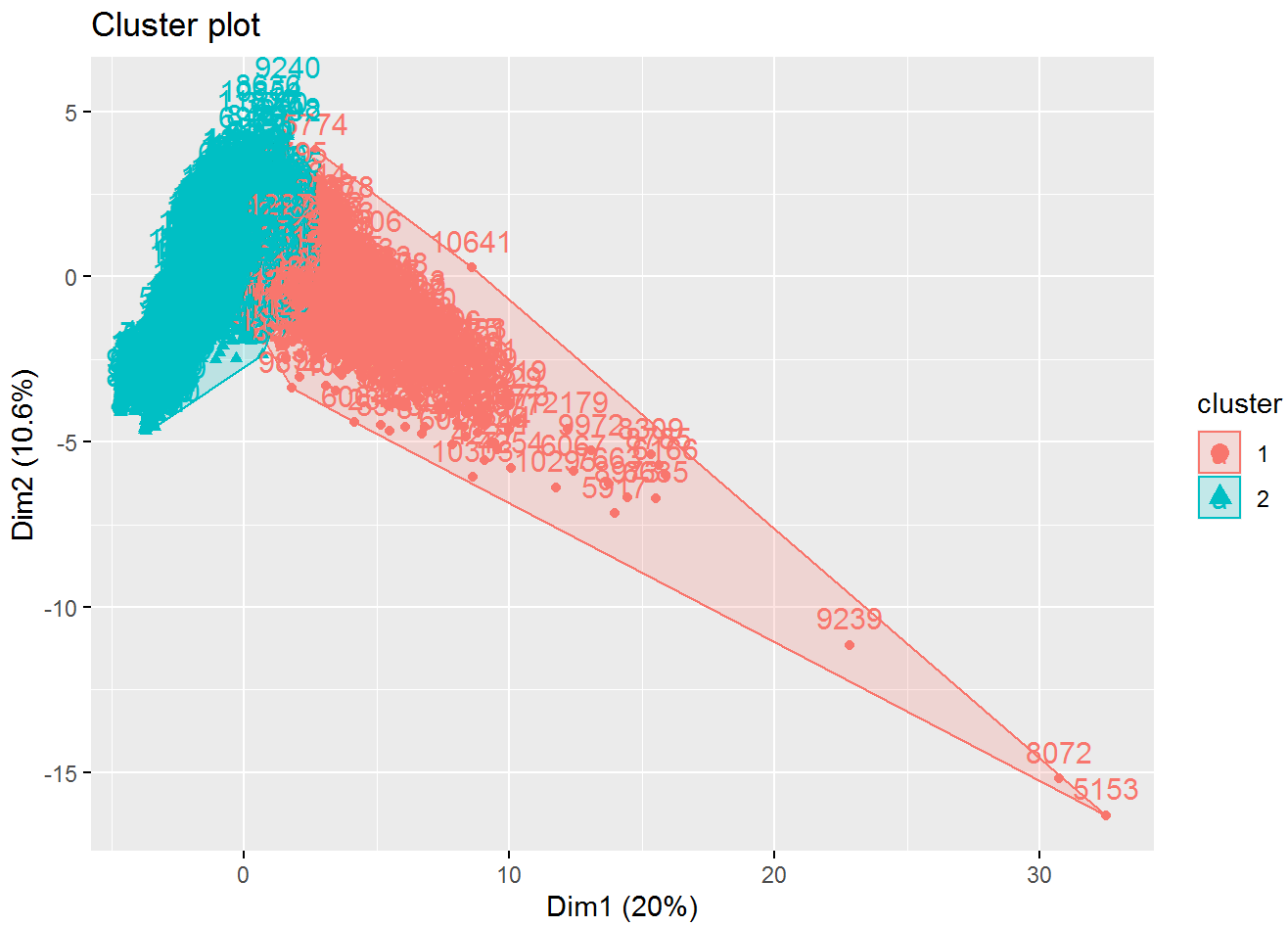
Implementing solution with K-Means

K-Means Clustering

```
#Identifying optimal k
fviz_nbclust(x = modeling.data,FUNcluster = kmeans, method =
'silhouette')
```




```
#The best k is 2
#clustering with kmeans
modelled <-kmeans(modeling.data,centers = 2,nstart = 25)
#Visualizing the clusters
fviz_cluster(modelled, data = modeling.data)
```



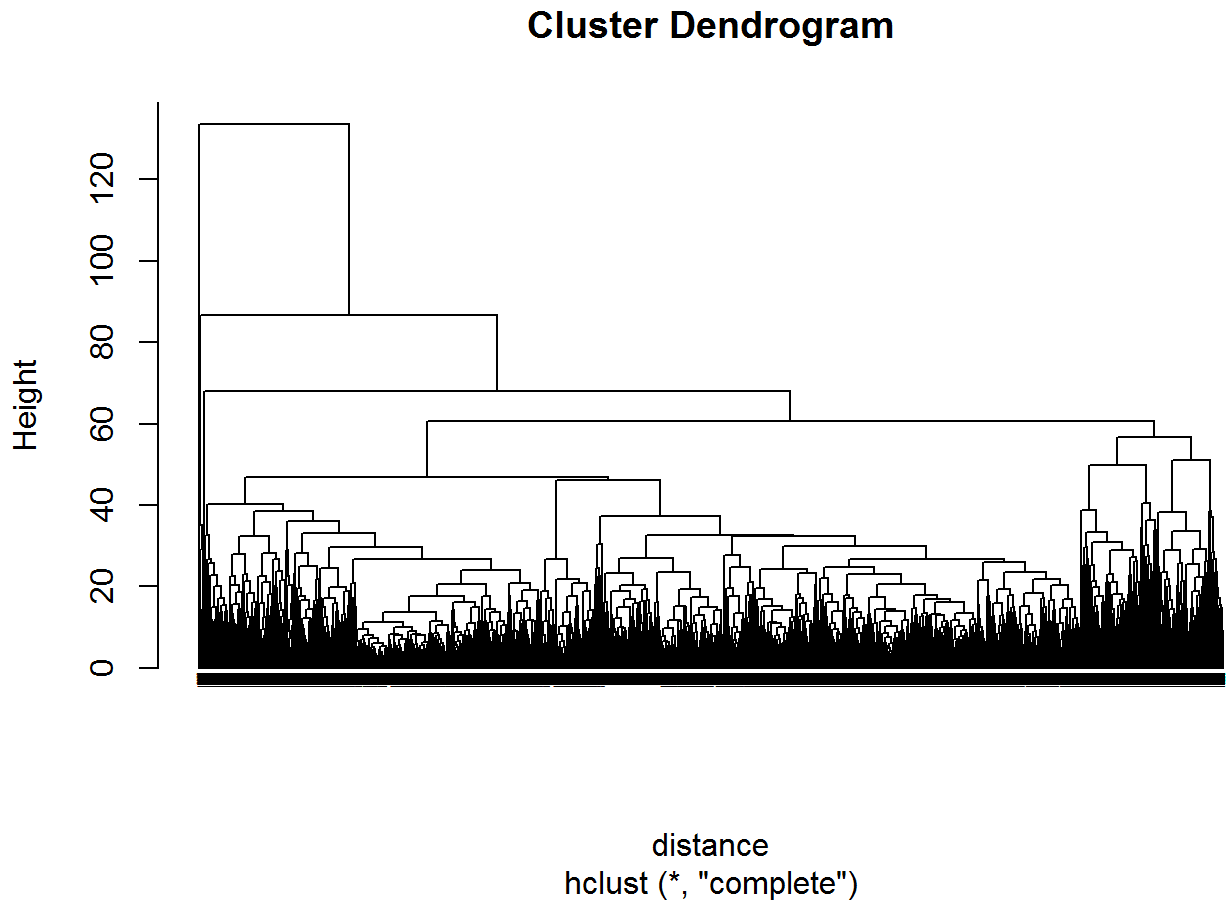

```
#Checking the size of each cluster
modelled$size
## [1]  1927 10272
# One cluster has 1927 data points while the other has 10272 data
points.
```

Challenging solution with Hierachical Clustering

Hierarchical Clustering

```
#Calculating the distance
distance <- dist(modeling.data,method = "manhattan")
#Hierarchical clustering
```

```
model2 <-hclust(distance)
#Visualizing the dendogram
plot(model2, cex = 0.2, hang = -5)
```




```
model2$size
## NULL
```