Moringa_Dsc14_Week13_IP_Rprogramming_Jonah_Okiru_06_2022_part2

Jonah okiru

2022-06-05

```
#Load the data
library("data.table")
df <- read.csv("D:/R studio/week2R/online_shoppers_intention.csv")
```

1. Problem definition.

a) Specifying the question.

The sales and marketing team of the Russia brand, needs the insight of the customers behavior by groups and also to learn the characteristics of each customer groups. # b) The metric of success The metric of success of the model should be able to cluster the visitors of the webpage into two clusters based on the revenue class label. # c)The context The Russian brand with chain of retail stores in Russia, Belarus, Ukraine, Kazakhstan, China,Philippines and Armenia. The Brand sales and marketing team needs the insight of the behavior of their customers and specifically the characteristics of customer of groups. # d) Recording the experimental design. -Loading the data -Check the data -Clean the data -univariate analysis -Bivariate analysis -modelling - Conclusion -Recommendation -Challenge the solution -Follow up question

2. Data Sourcing.

- The data was sourced from the E-commerce customer datasets.

#3. Check for the Data

```
#Preview the first rows of the dataset
head(df,4)
```

```
##   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
##   ProductRelated ProductRelated_Duration BounceRates ExitRates PageValues
## 1              1                0.000000        0.20      0.20          0
## 2              2               64.000000        0.00      0.10          0
## 3              1               -1.000000        0.20      0.20          0
## 4              2                2.666667        0.05      0.14          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
##         VisitorType Weekend Revenue
## 1 Returning_Visitor   FALSE   FALSE
## 2 Returning_Visitor   FALSE   FALSE
## 3 Returning_Visitor   FALSE   FALSE
## 4 Returning_Visitor   FALSE   FALSE
```

```
#Check the number of columns in the dataset
ncol(df)
```

```
## [1] 18
```

The dataset has 18 columns

```
#Check the number of rows in the dataset
nrow(df)
```

```
## [1] 12330
```

The dataset has 12330 records

```
#check the datatype of each column
str(df)
```

```
## '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" "Returni
##  $ Weekend                : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ Revenue                : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

The dataset columns datatypes are as follows; 7 columns are of integer type, 7 columns are of numerical datatype, 2 columns are of string character datatype and lastly 2 columns are of logical datatypes.

```r
#Check for the missing values in the dataset
colSums(is.na(df))
```

```
##          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
```

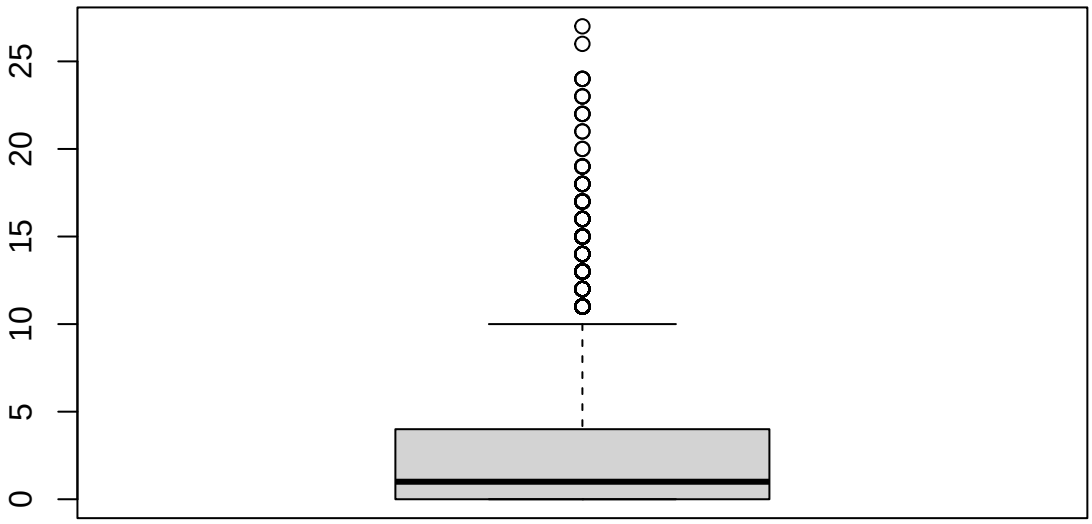
The columns of administrative, administrative_duration, informational, informational_duration, productrelated, productrelated_duration, bounce rates and exitrates each contains 14 missing values. The other remaining columns does not contain any missing values.

```r
#check for the duplicate in the dataset
duplicates <- df[duplicated(df), ]
#Check number of rows duplicated
nrow(duplicates)
```

```
## [1] 119
```

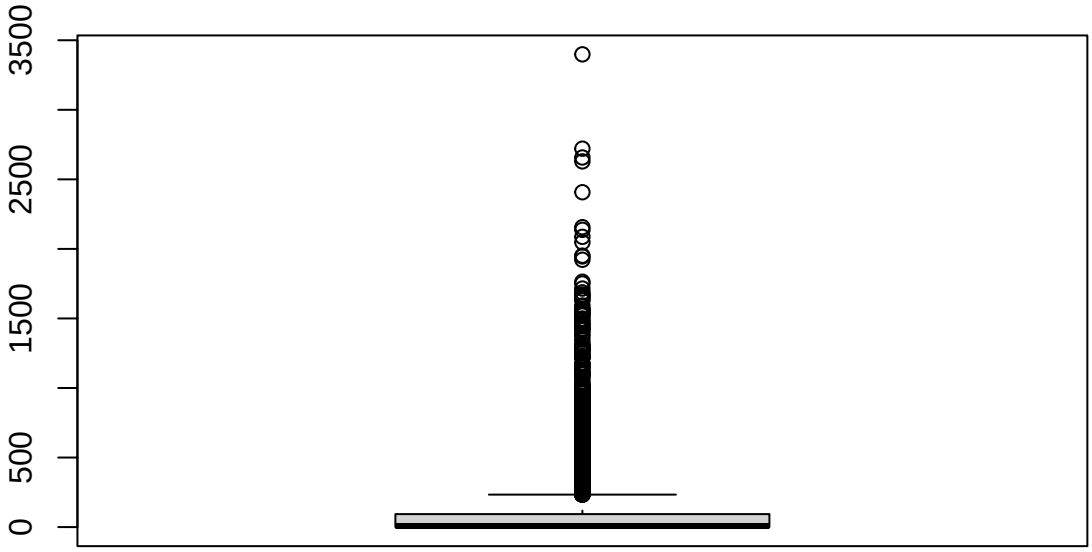
The dataset has 119 duplicate rows.

```r
#Check for the outlier in administartive columns
boxplot(df$Administrative)
```



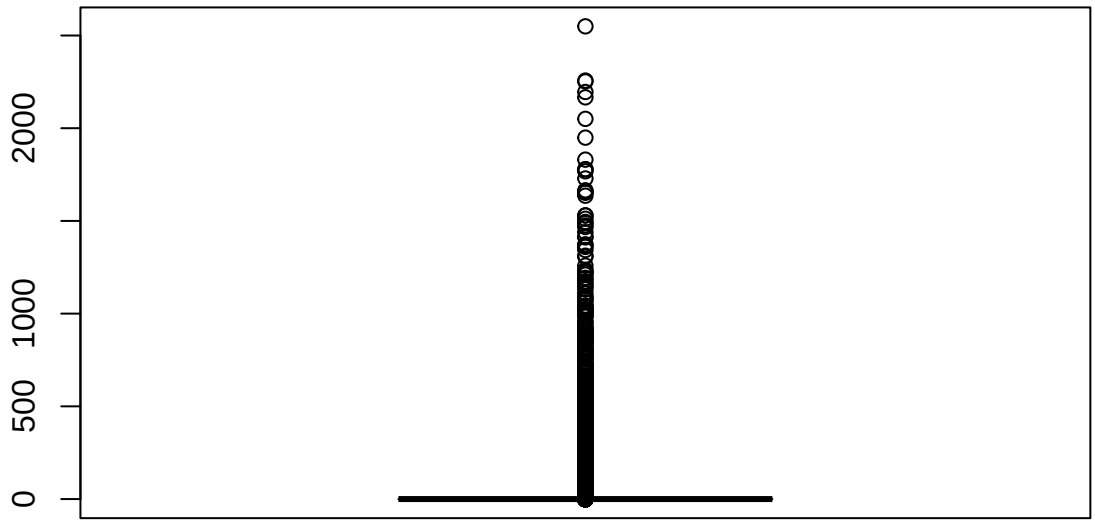
There is existence of outliers in the administrative column

```
#Check for the existence of outlier in the administrative duration column
boxplot(df$Administrative_Duration)
```



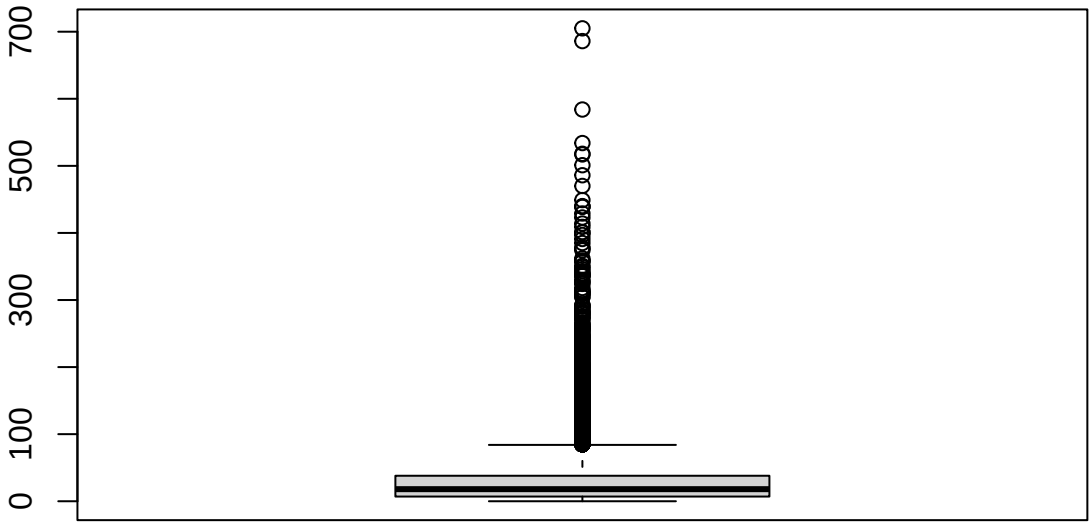
The outliers exist in the information duration column.

```
#Check for the existence of outliers in the Informational_Duration
boxplot(df$Informational_Duration)
```



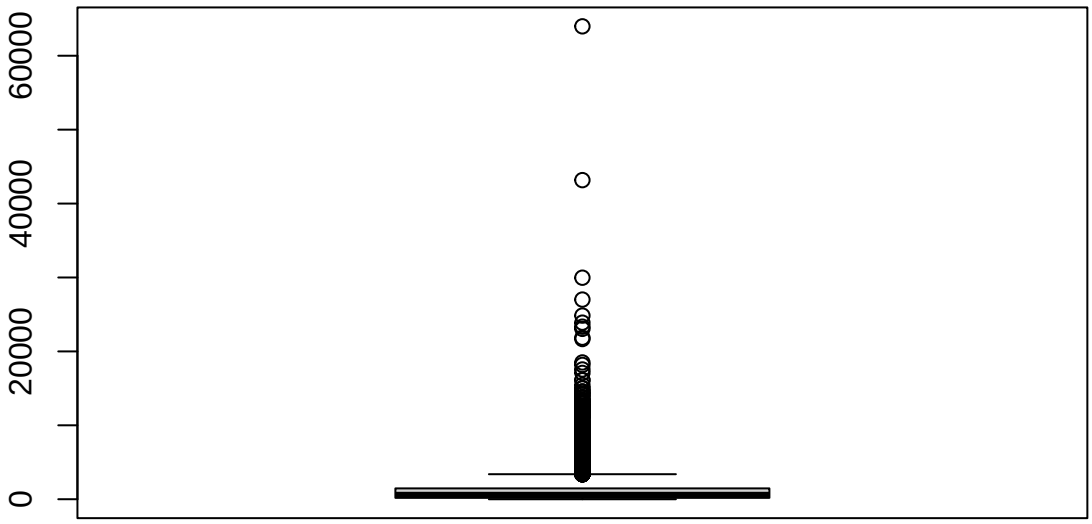
The outliers exist in the informational duration column.

```
#Check for the outliers in the ProductRelated
boxplot(df$ProductRelated)
```



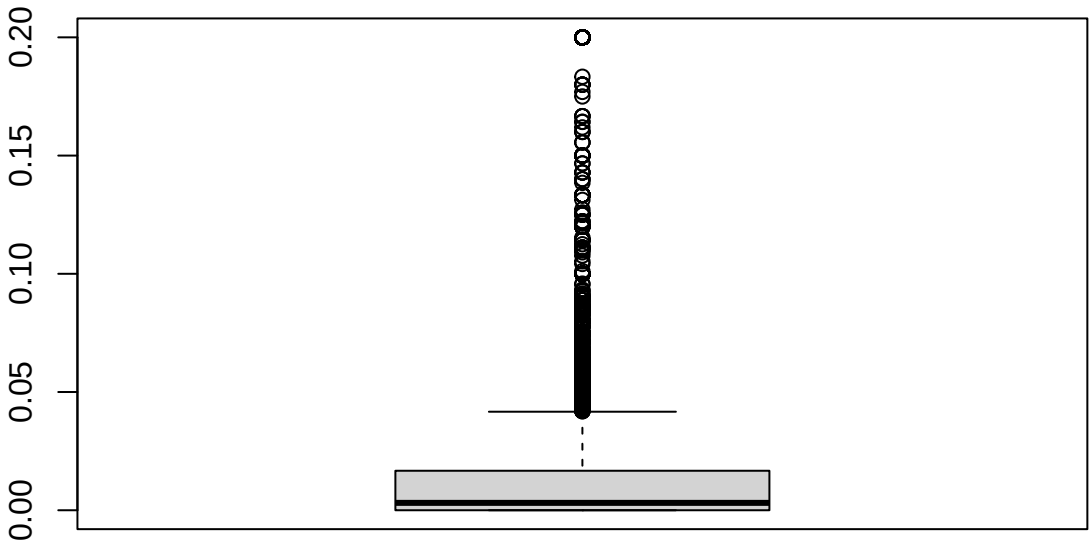
The outliers exist in the product related column.

```
#Check for the existence of outliers in the ProductRelated_Duration
boxplot(df$ProductRelated_Duration)
```



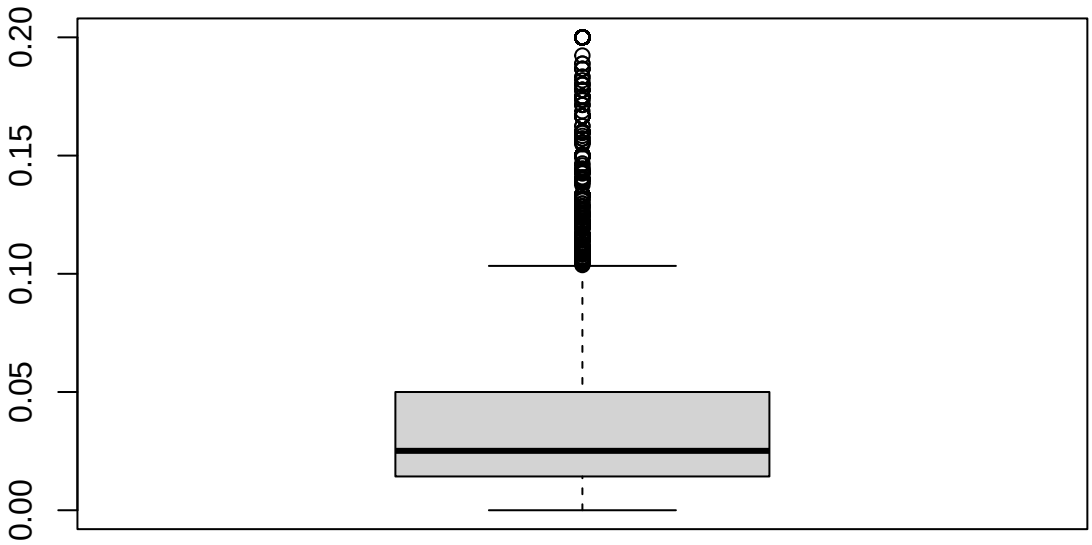
There is existence of outliers in the productrelated duration column

```
#Check for the existence of outlier in the BounceRates
boxplot(df$BounceRates)
```



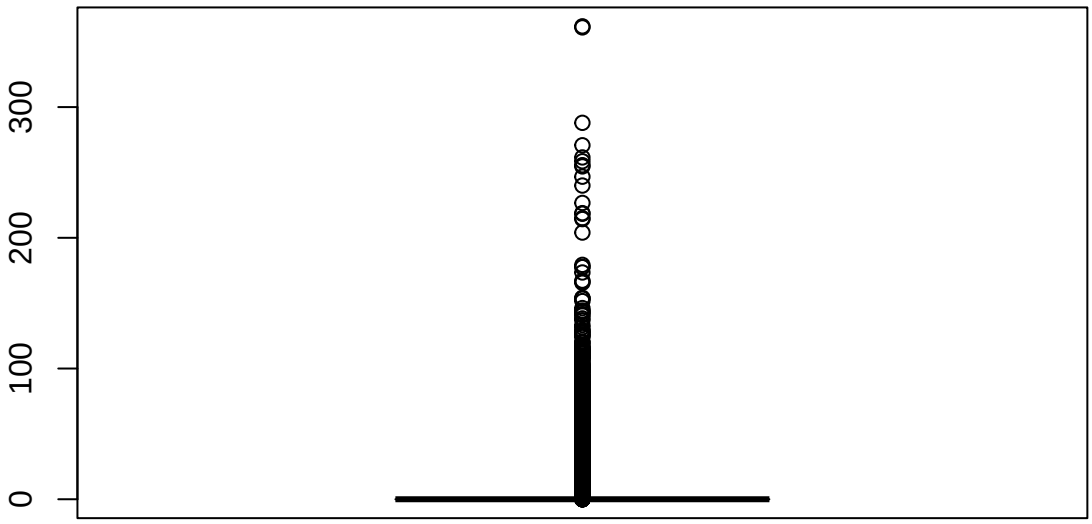
Outliers exists in the BounceRates columns

```
#Check for the existence of outliers in the ExitRates
boxplot(df$ExitRates)
```



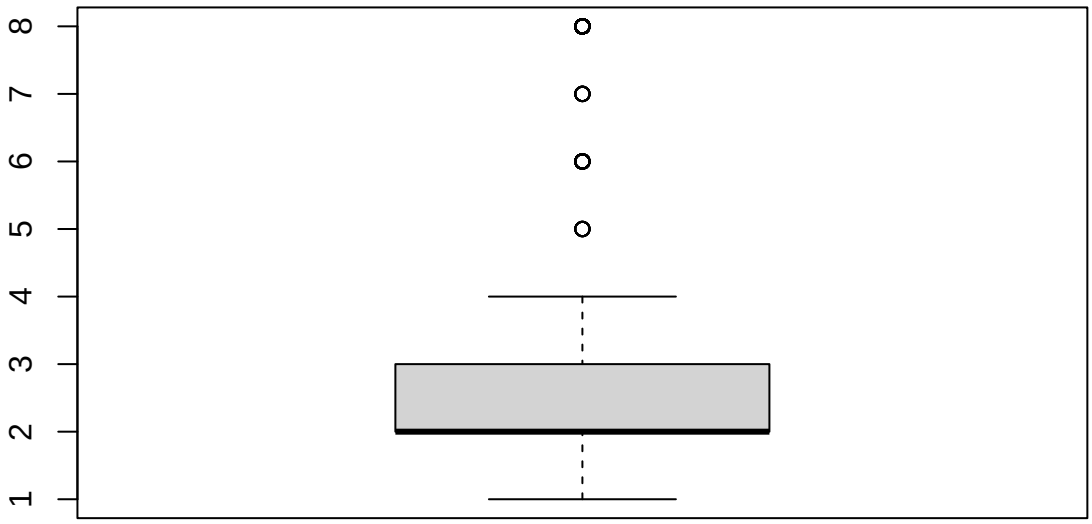
The outliers exists in the ExitRates columns

```
#Check for the outliers in the PageValues
boxplot(df$PageValues)
```



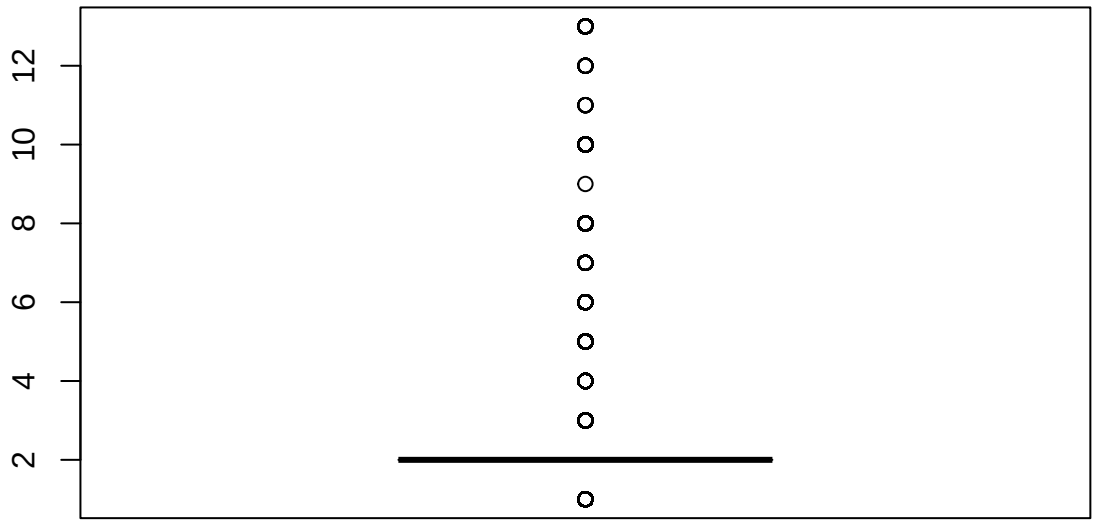
There was existence of the outliers in the SpecialDay column.

```
#Check for the outliers in the OperatingSystems
boxplot(df$OperatingSystems)
```



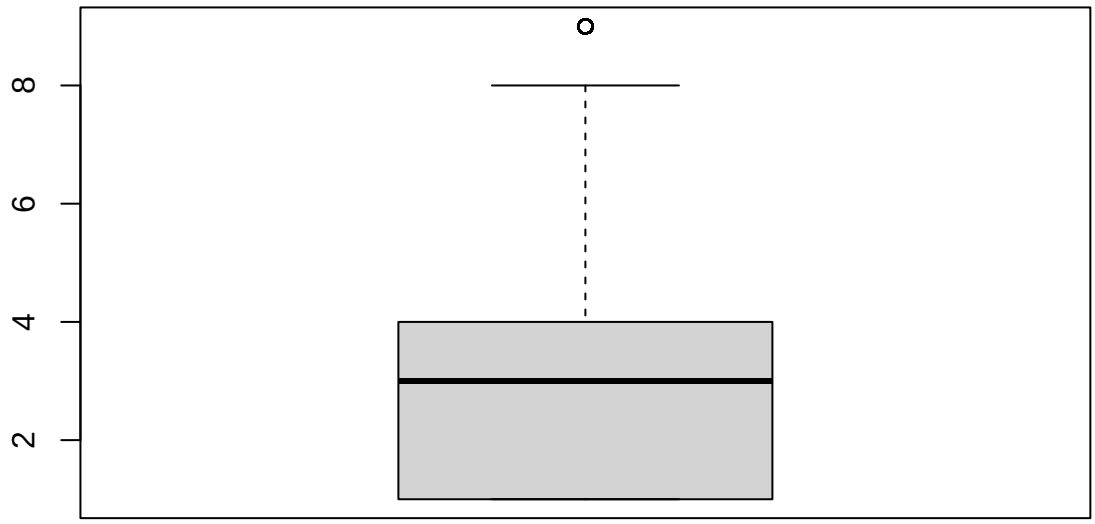
There was existence of the Outliers in the Operating Systems.

```
#Check for the outliers in the
boxplot(df$Browser)
```



There was existence of outliers in the Browser columns.

```
#Check for the existence of outliers in the Region
boxplot(df$Region)
```



There was only one outlier in the Region column

#4. Data cleaning.

```
#Remove the missing data
df <- na.omit(df)
#Check if the missing values have been removed in the dataset
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
```

The rows containing the missing values were removed since the percentage of the missing records in the dataset were 0.1% of the data, hence removing them could not have any impact on the dataset.

```
#Remove the duplicate records in the dataset by applying unique function to
# the datasets.
df1 <- unique(df)
#Check if the duplicate rows is dropped.
df1[duplicated(df1), ]
```

```
##  [1] Administrative          Administrative_Duration Informational
##  [4] Informational_Duration  ProductRelated          ProductRelated_Duration
##  [7] BounceRates             ExitRates               PageValues
## [10] SpecialDay              Month                   OperatingSystems
## [13] Browser                 Region                  TrafficType
## [16] VisitorType             Weekend                 Revenue
## <0 rows> (or 0-length row.names)
```

The duplicates are removed from the dataset in order to avoid chaos in during the analysis of the data.

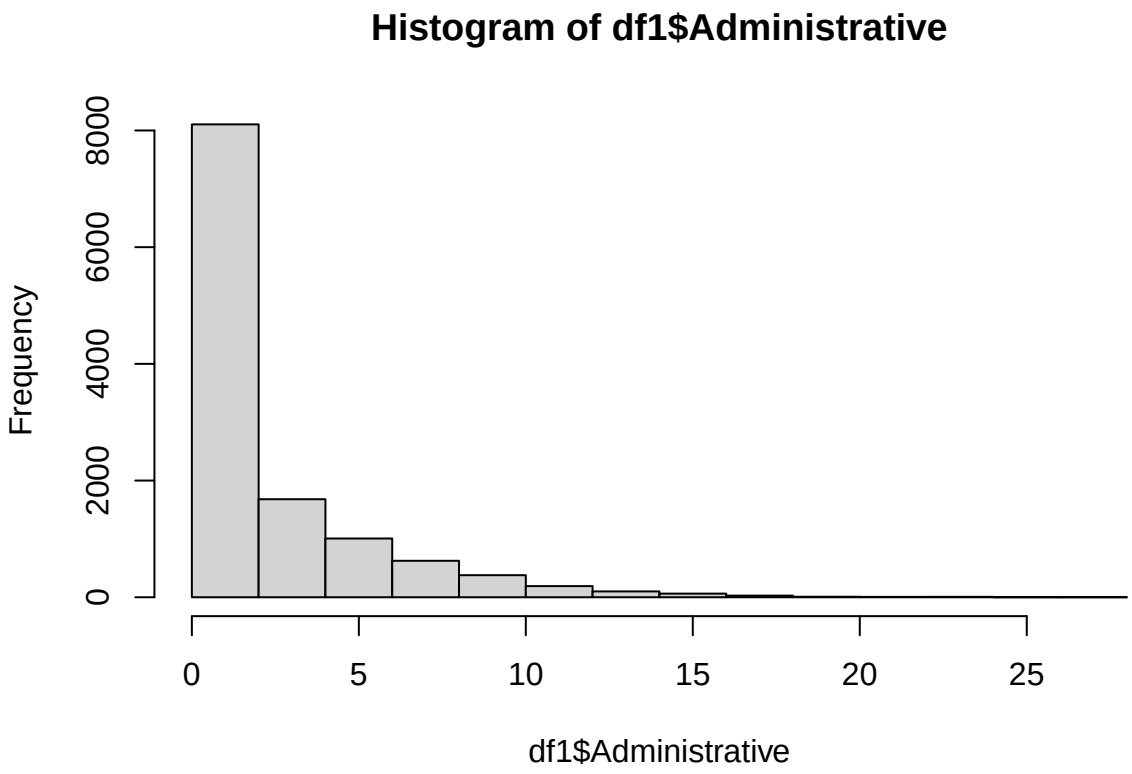
```
#Dealing with the outliers
#The outliers in the datasets were not dropped.
```

The reasons for retaining the outliers were due to the following reasons 1. For the administration, administration duration, informational, informational, informational duration, product related and product related duration since they represent the different types of pages visited by the visitor during the session and the total time spend on each of the page categories. The outliers were retained since the number of pages visited by the visitor and the time spend on the page varies by each visitor during the session. 2. The bounce rate, exit rate, and page value represents the metrics measured by the Google analytic for each page in e-commerce site. The outliers were retained since each page has different metric measurement. 3.The spacial day feature represents the closeness of the site visiting time to specific special day. The outliers of these column were retained since the closeness of the visitng sites varies with the type of special day. 4. The outliers for the other columns were retained since there column values are categorical.

#5. Exploratory Data Analysis

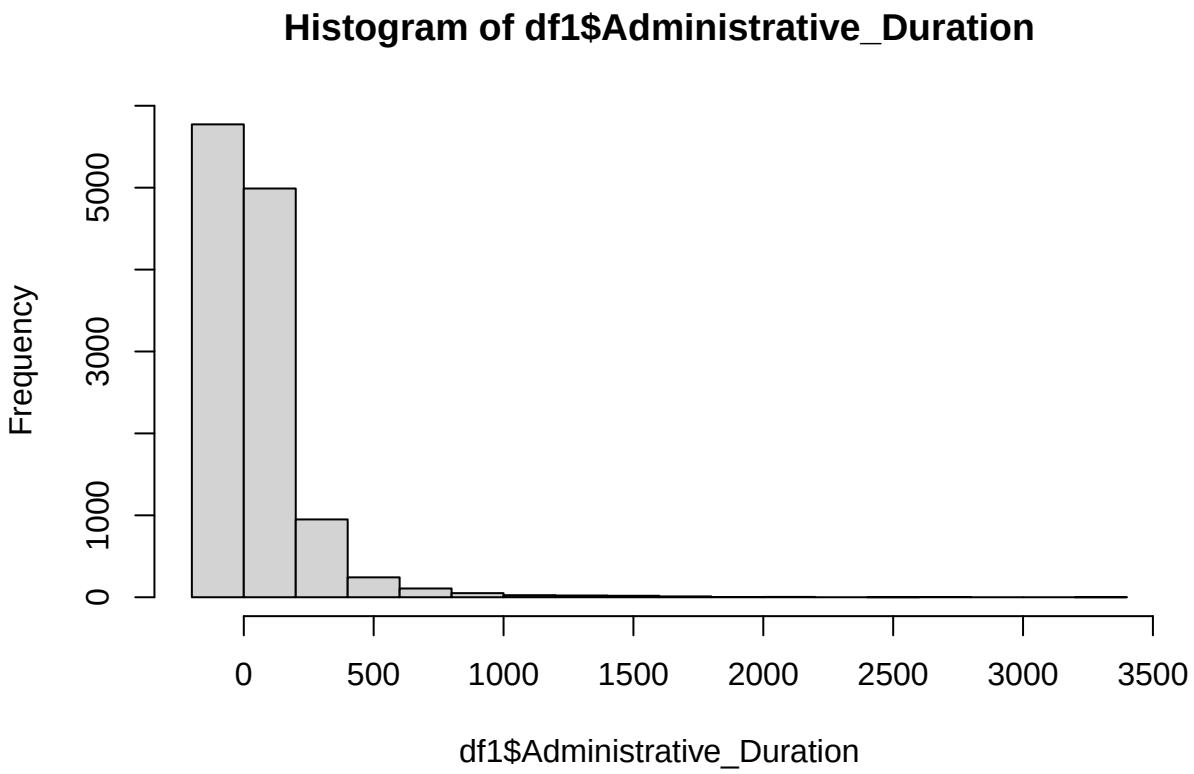
1. a). Univariate data Analysis(Numerical)

```
#Administrative histogram
hist(df1$Administrative)
```



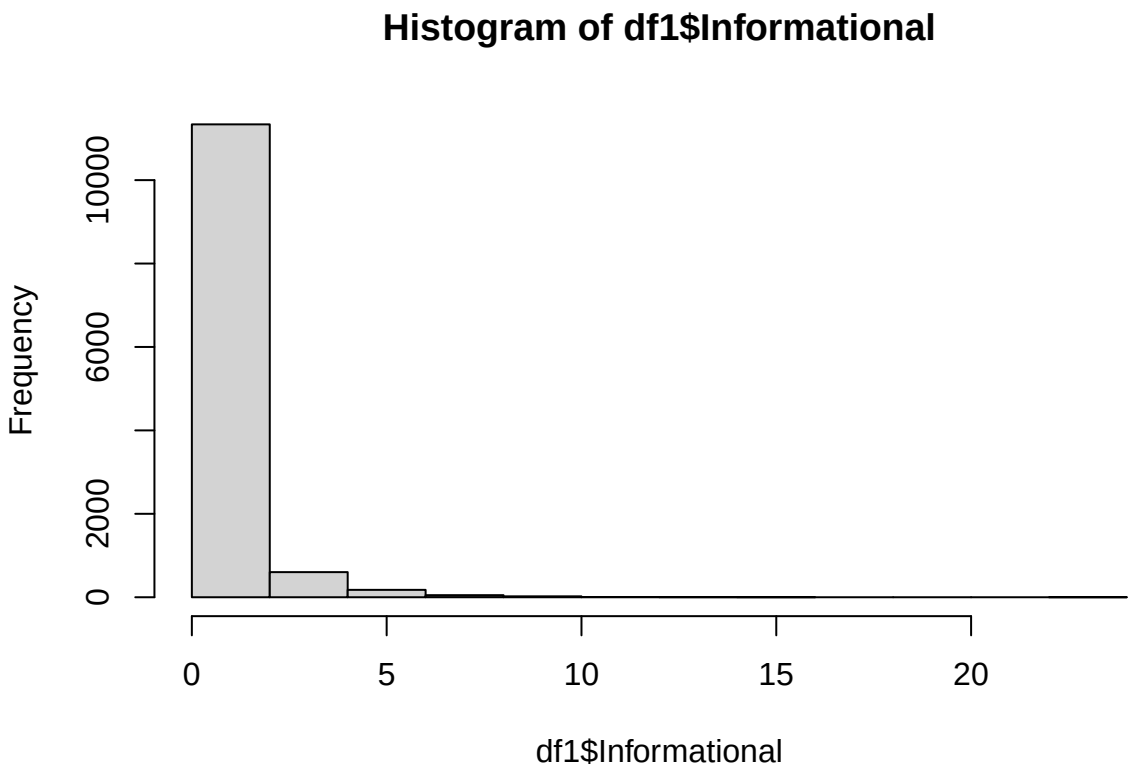
The number of pages visited by the visitors of administrative web page were as follows; 0 and 5 pages was the most, followed by 5 to 10 and then the rests follow in the same pecking order.

```
#Administrative_Duration histogram
hist(df1$Administrative_Duration)
```



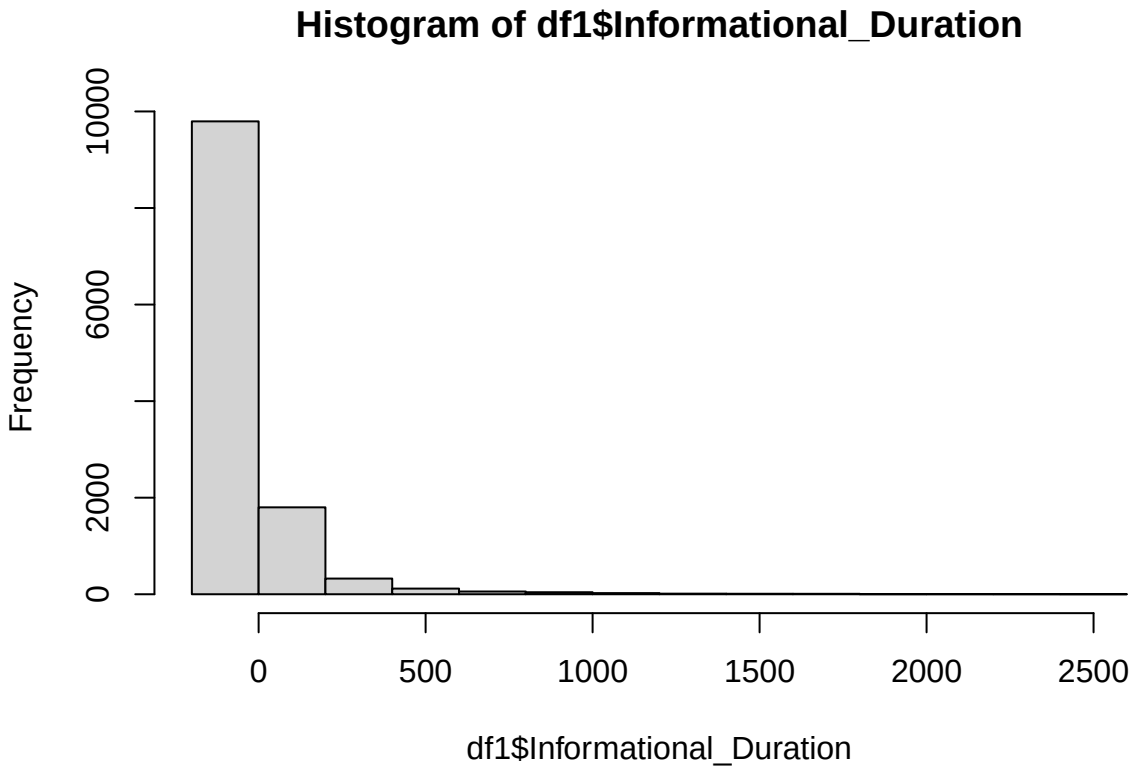
The duration spent by the visitors of the administrative page were as follows; 0 duration was the most, followed by 0 to 500, then 500 to 1000 and then rest follows

```
#Informational histogram
hist(df1$Informational)
```



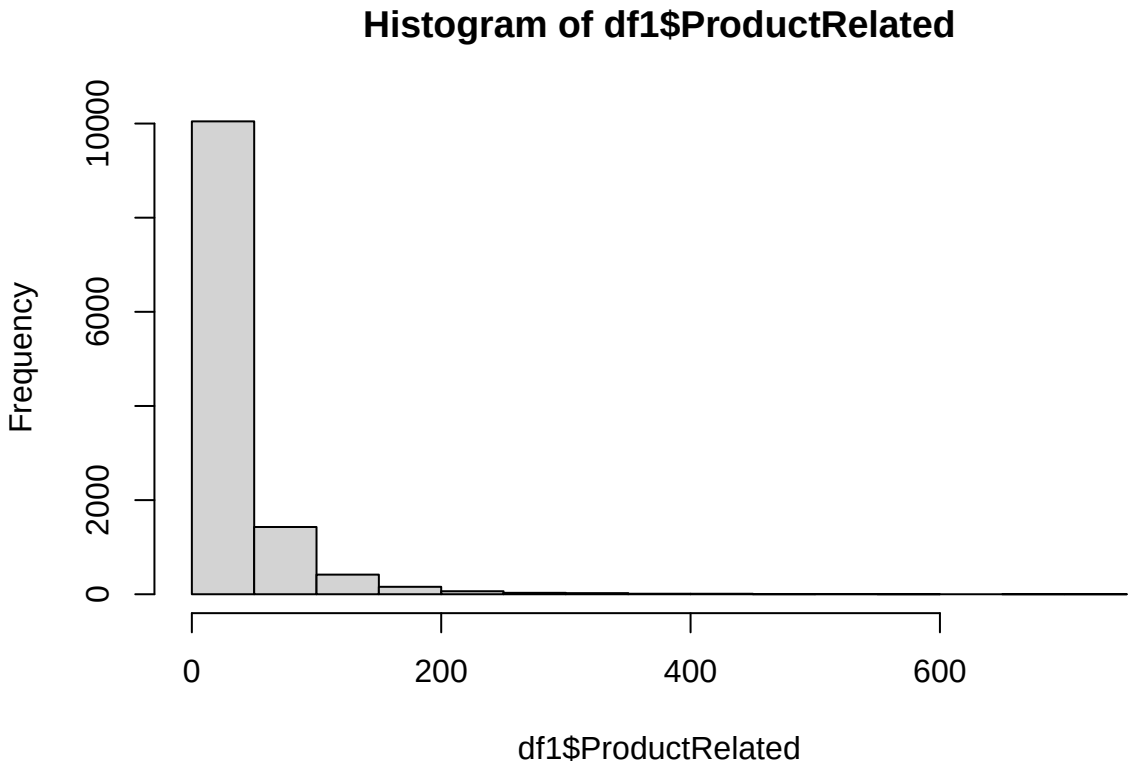
The number of pages visited by the visitors of informational web page were as follows; 0 t0 5 pages were the most followed by 5 to 10 pages.

```
#Informational_Duration
hist(df1$Informational_Duration)
```



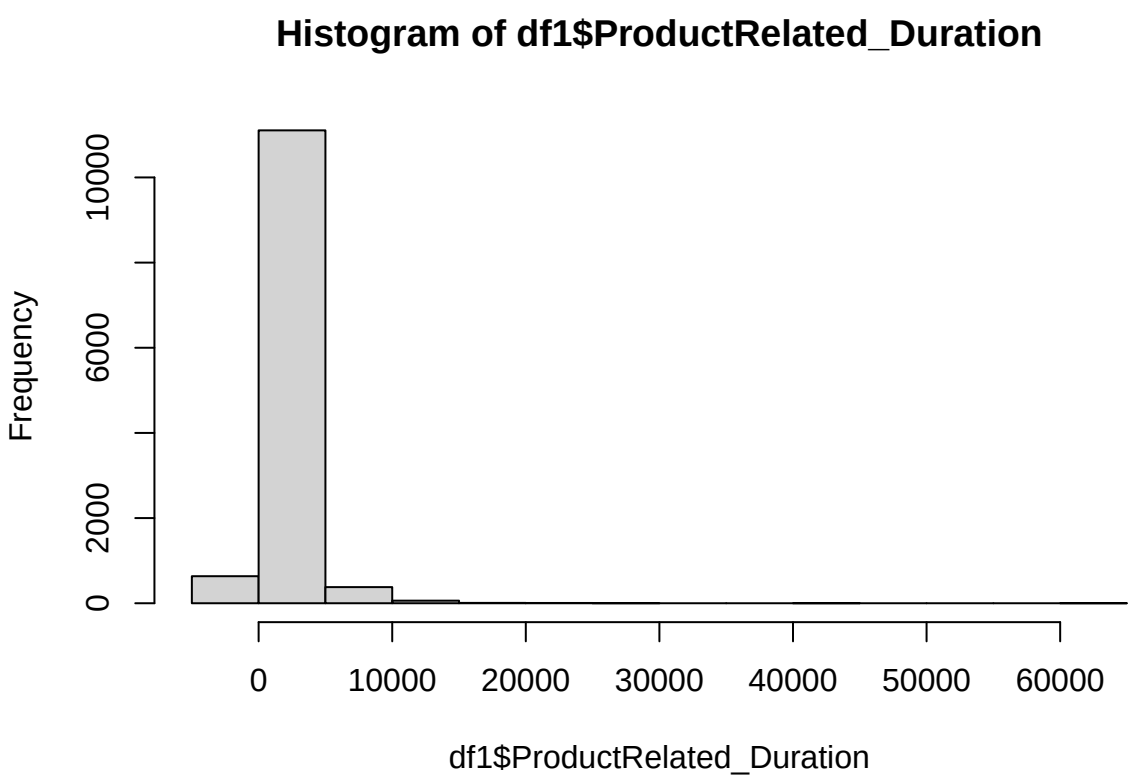
The duration spent by the visitors to the informational page were as follows; less than 0 was the most, followed, 0 to 500, then 500 to 1000

```
#ProductRelated histogram
hist(df1$ProductRelated)
```



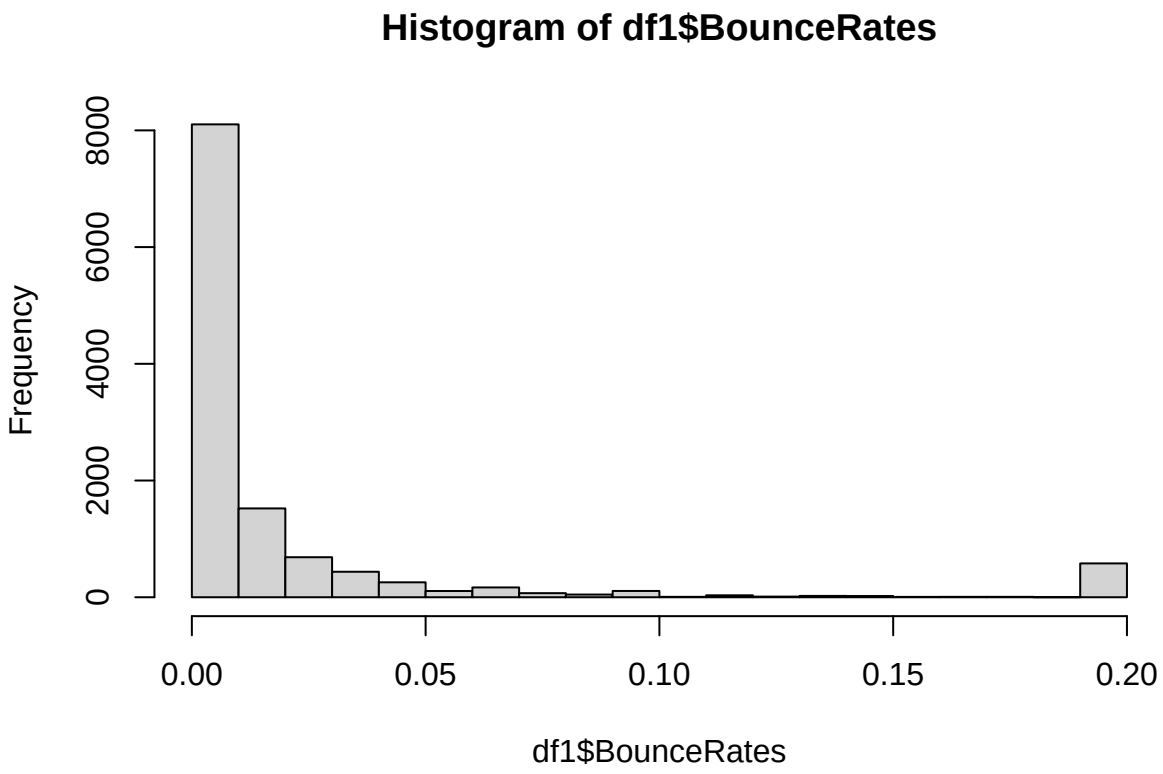
The number of pages visited by the visitors of the product related page were as follows; 0 to 200 was the most, followed by 200 to 400 and then the rest.

```
#ProductRelated_Duration Histogram
hist(df1$ProductRelated_Duration)
```



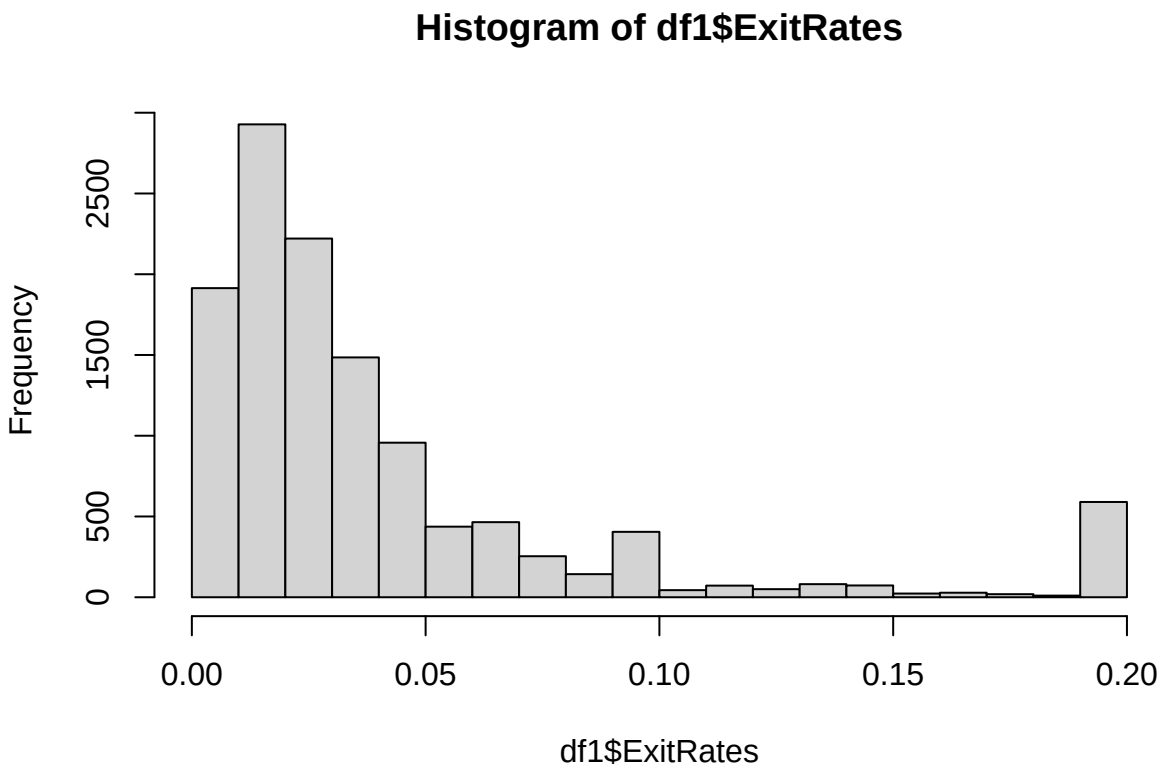
The duration spent by the visitors of the product related webpage were as follows; 0 to 1000 were the most, followed by less than 0, then 1000 to 1500.

```
#BounceRates Histogram
hist(df1$BounceRates)
```



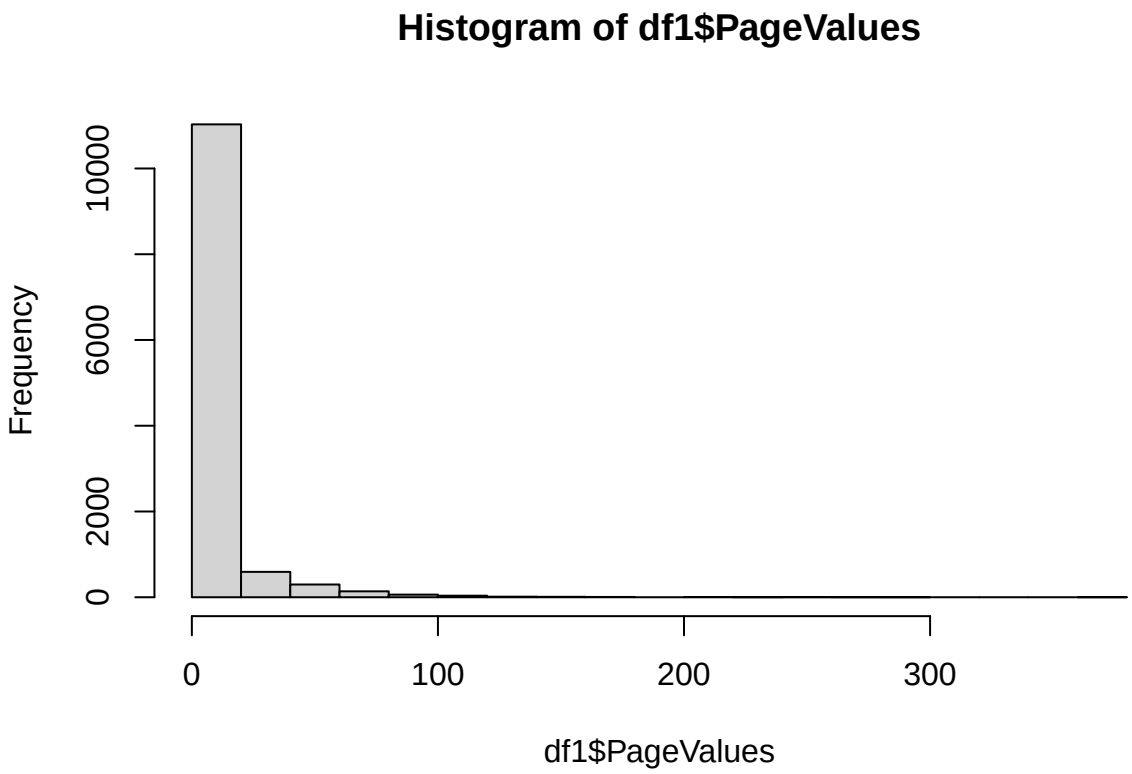
The most bounce rate was between 0.00 to 0.05

```
#ExitRates histograms
hist(df1$ExitRates)
```



The most exit rate was between 0.00 to 0.05, followed by 0.05 to 0.10

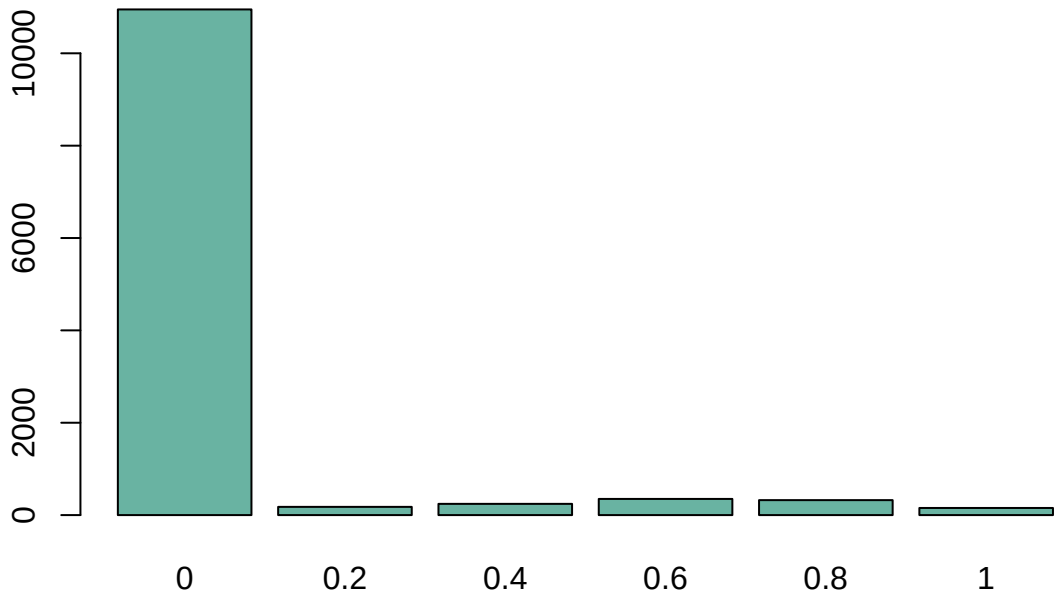
```
#PageValues histogram
hist(df1$PageValues)
```



The highest page value was between 0 to 20, followed by 20 to 40

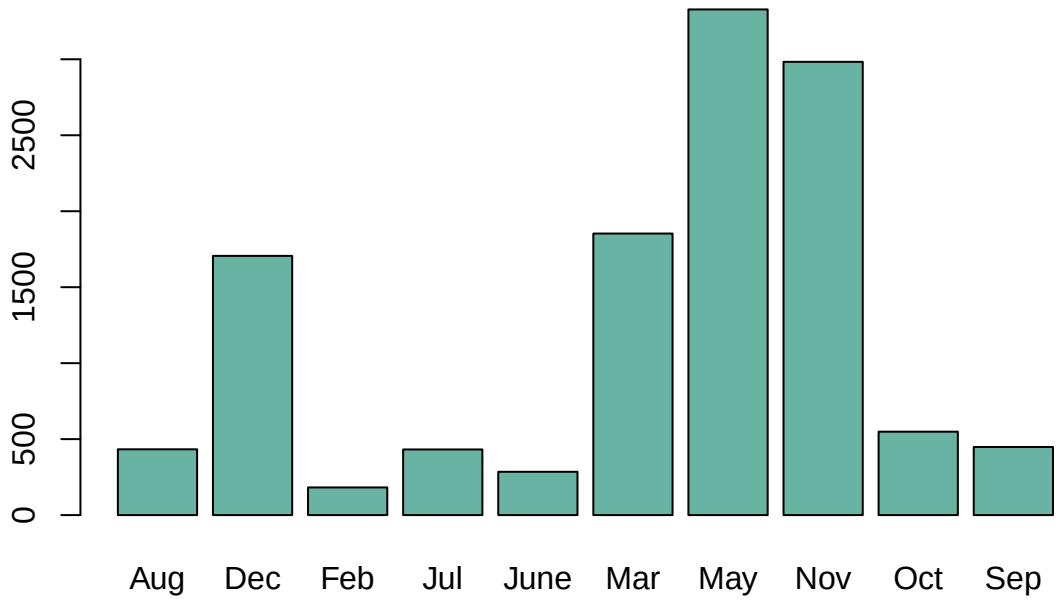
b). Univariate data Analysis(Categorical)

```
#SpecialDay barplot
frequency_SpecialDay <- table(df1$SpecialDay)
barplot(height=frequency_SpecialDay, names = df1$name, col = "#69b3a2")
```



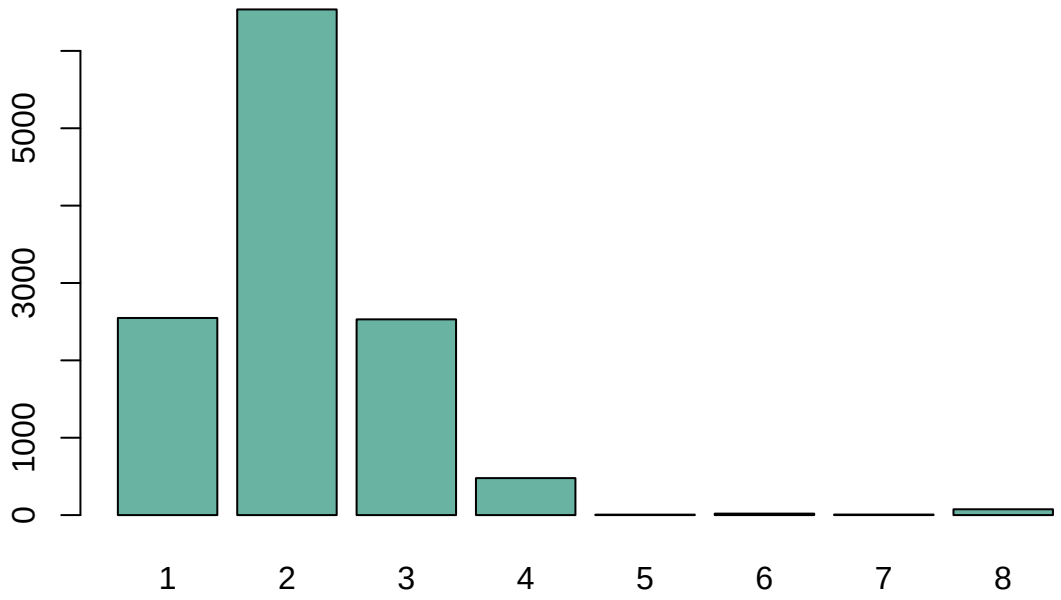
From the special day frequency bar graph zero had the most, followed by 0.8, 0.6 0.4, 0.2 and 0.1.

```
#Month barplot
frequency_Month <- table(df1$Month)
barplot(height=frequency_Month, names = df1$name, col = "#69b3a2")
```



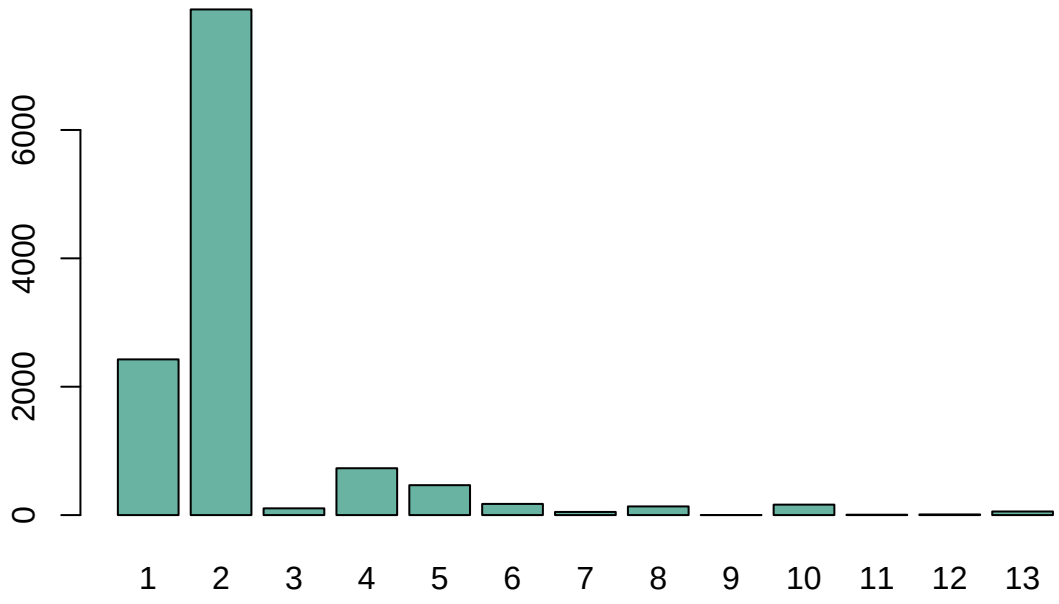
from the month bar chart the month with most frequency is may, followed by november march , december and then the rests follow.

```
##OperatingSystems barplot
frequency_OperatingSystems <- table(df1$OperatingSystems)
barplot(height=frequency_OperatingSystems, names = df1$name, col = "#69b3a2")
```



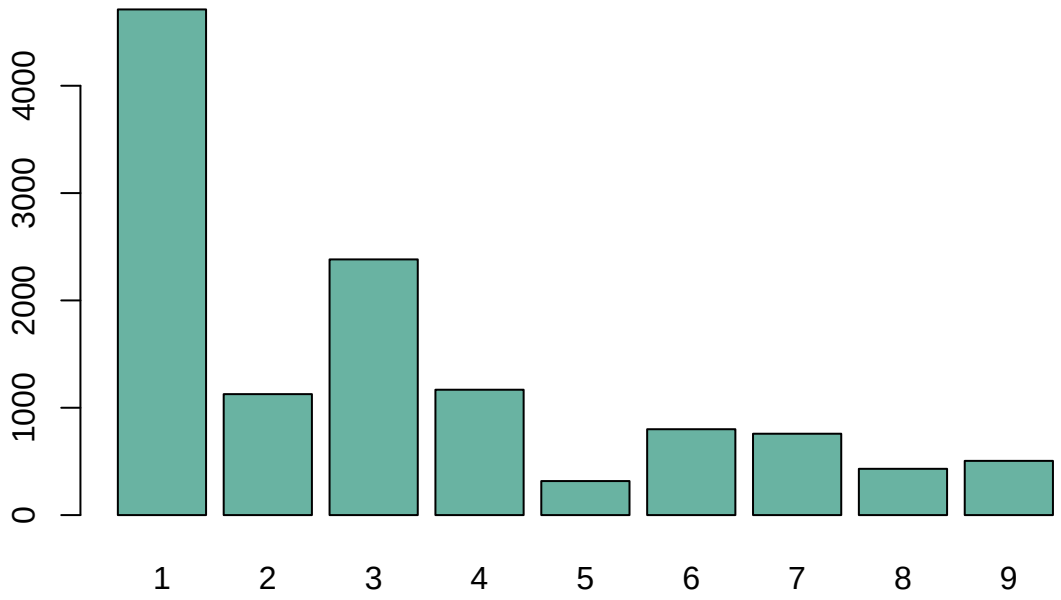
The operating with the most frequency was operating system number 2 followed by number 3 and 1.

```
##Browser barplot
frequency_Browser <- table(df1$Browser)
barplot(height=frequency_Browser, names = df1$name, col = "#69b3a2")
```



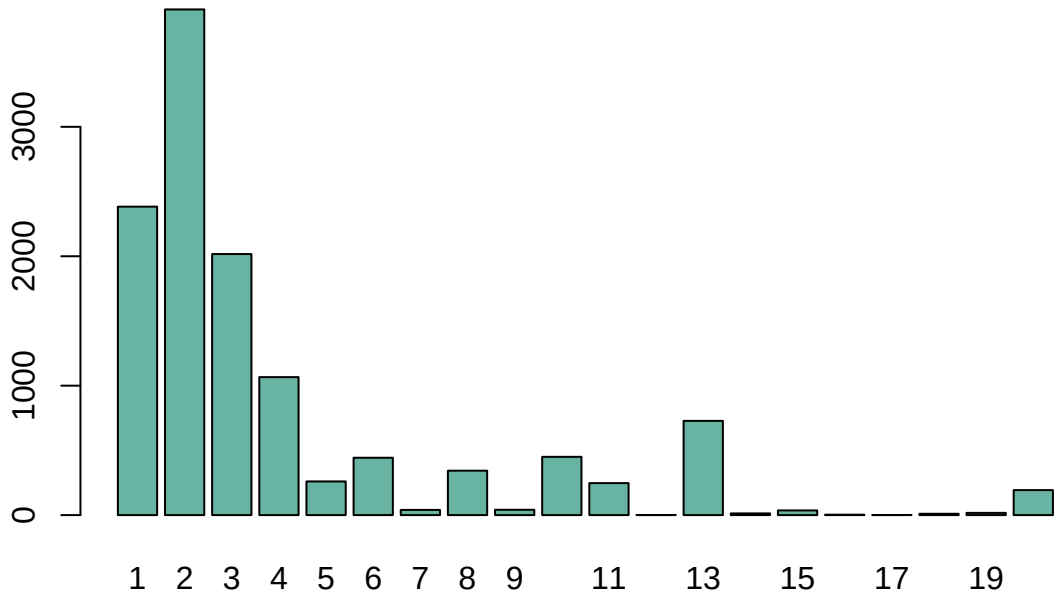
The browser with the most frequency was the browser number 2 followed by the browser number 1

```
##Region barplot
frequency_Region <- table(df1$Region)
barplot(height=frequency_Region, names = df1$name, col = "#69b3a2")
```



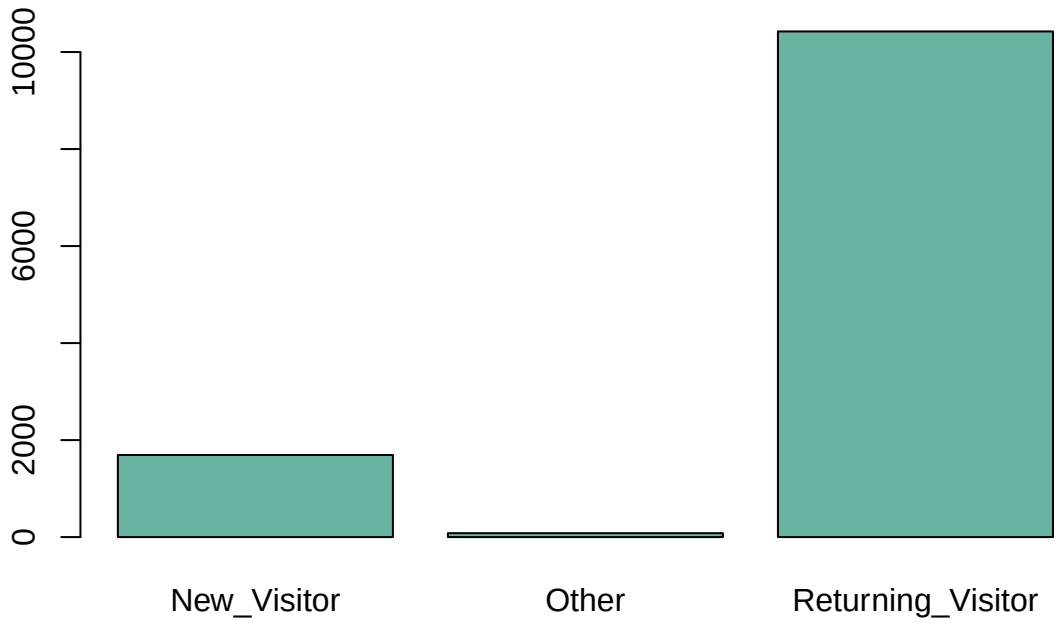
The operating system with the most frequency was operating system number 1 followed by number 3.

```
##TrafficType barplot
frequency_TrafficType <- table(df1$TrafficType)
barplot(height=frequency_TrafficType, names = df1$name, col = "#69b3a2")
```



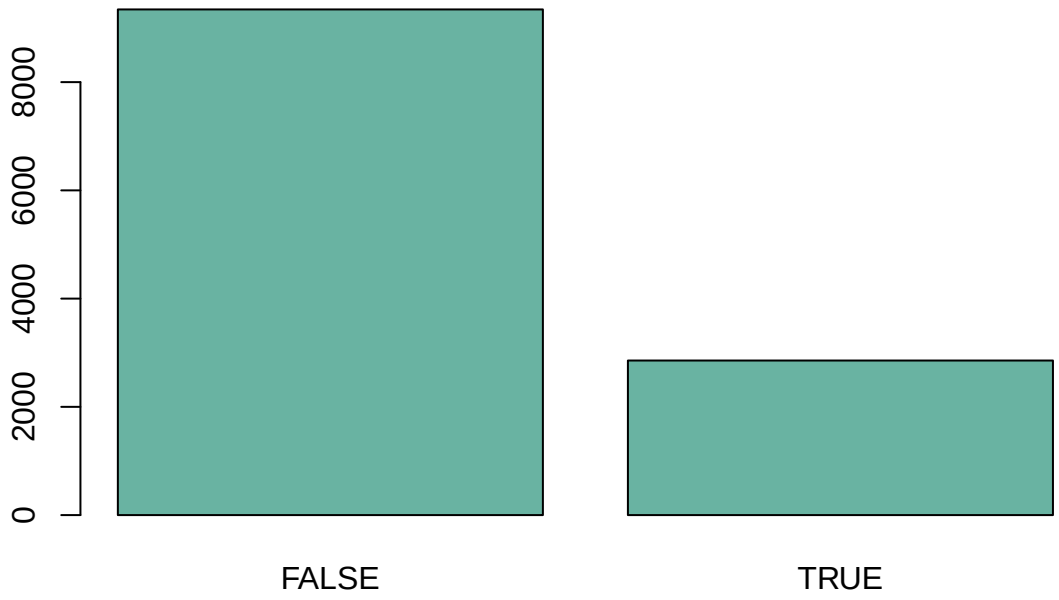
The traffic type with frequency is the traffic number

```
#VisitorType barplot
frequency_VisitorType <- table(df1$VisitorType)
barplot(height=frequency_VisitorType, names = df1$name, col = "#69b3a2")
```



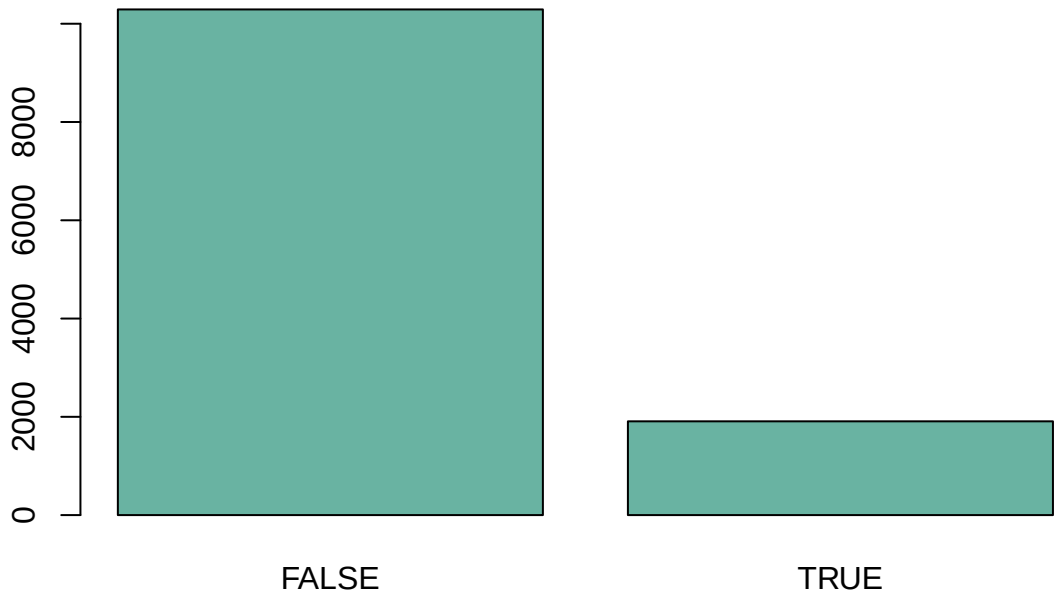
Returning customer has the most followed by new visitor

```
#Weekend barplot
frequency_Weekend <- table(df1$Weekend)
barplot(height=frequency_Weekend, names = df1$name, col = "#69b3a2")
```



False has the most.

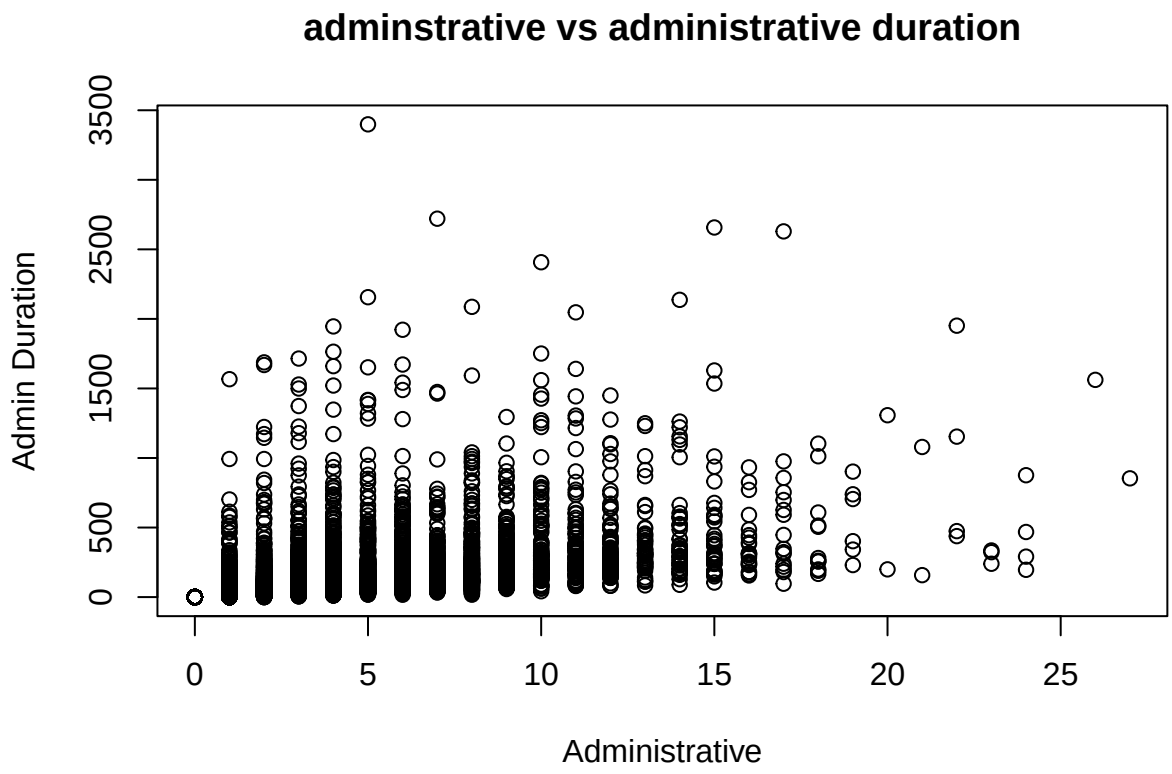
```
#Revenue barplot
frequency_Revenue <- table(df1$Revenue)
barplot(height=frequency_Revenue, names = df1$name, col = "#69b3a2")
```



2). Bivariate Analysis.

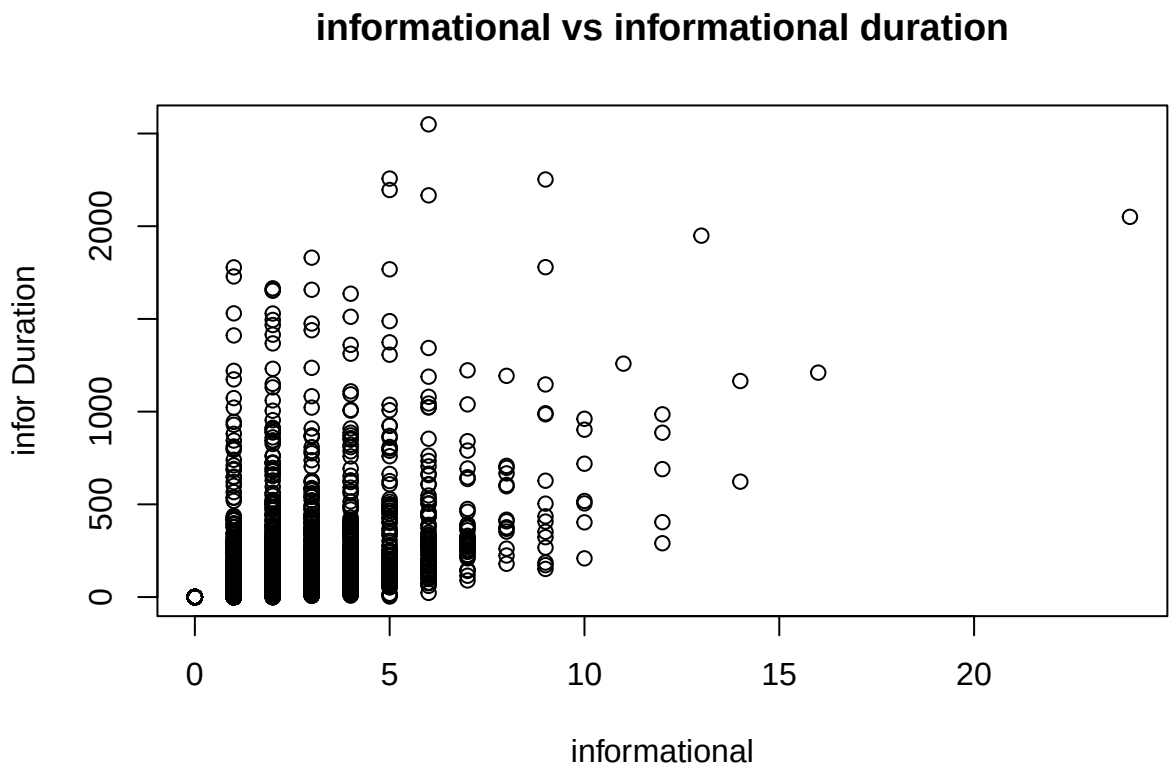
a). Numerical vs Numerical

```
#Scatter plot of administrative vs Administrative duration
plot(df1$Administrative, df1$Administrative_Duration, xlab = ("Administrative"),
     ylab = ("Admin Duration"), main = "adminstrative vs administrative duration")
```



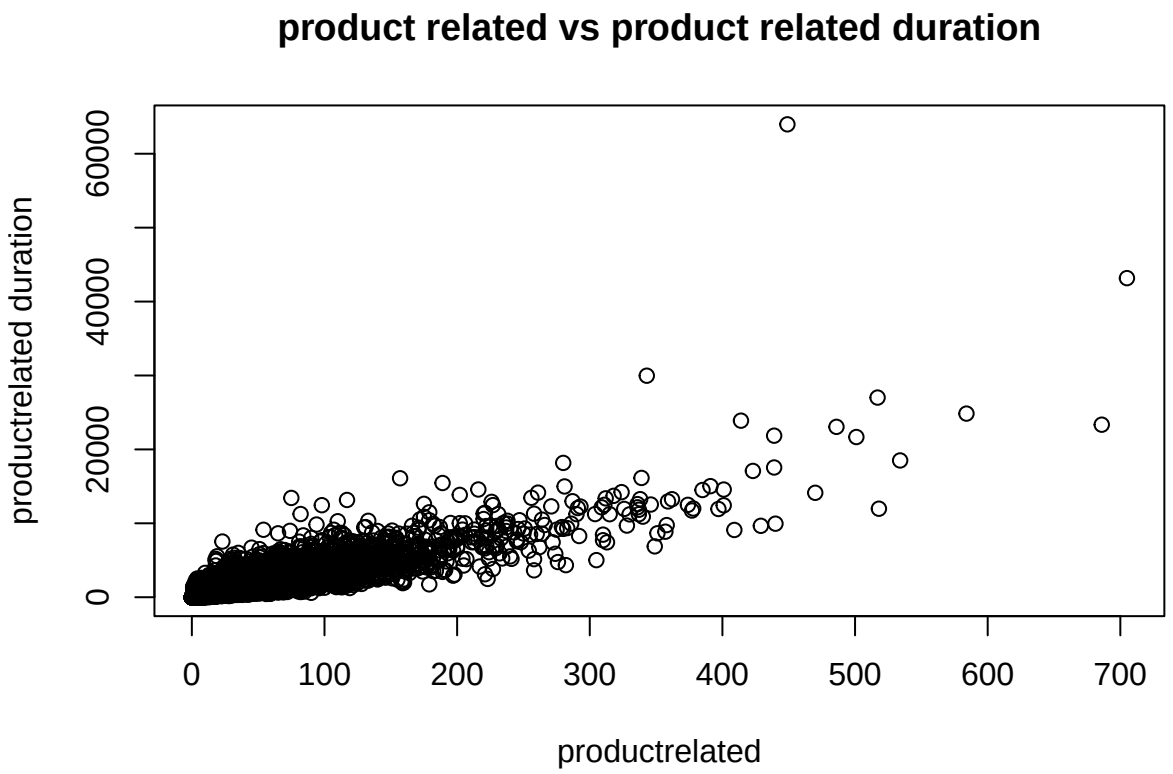
The the correlation between administrative and administrative duration is a weak correlation.

```
#Scatter plot of informational vs informataional duration
plot(df1$Informational, df1$Informational_Duration, xlab = ("informational"),
     ylab = ("infor Duration"), main = "informational vs informational duration")
```



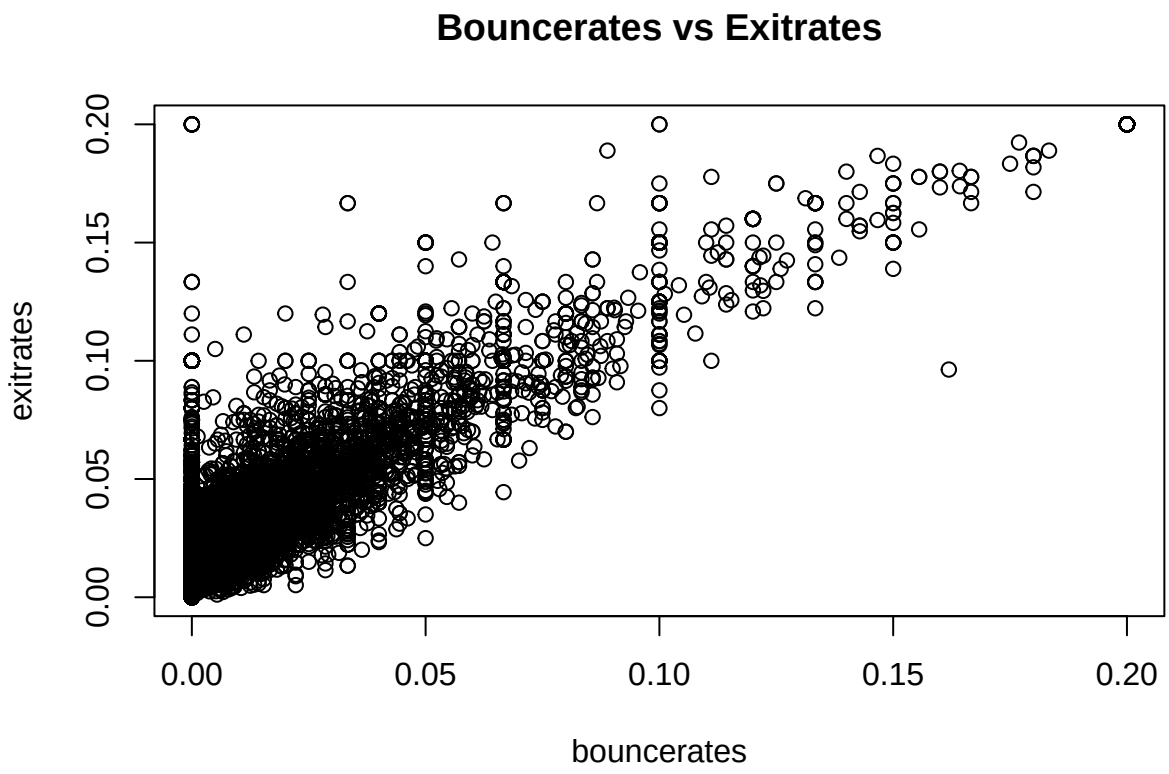
The the correlation between informational and informational duration is a negative correlation.

```
#Scatterplot of product related vs Product related information
plot(df1$ProductRelated, df1$ProductRelated_Duration, xlab = ("productrelated"),
     ylab = ("productrelated duration"), main = "product related vs product related duration")
```



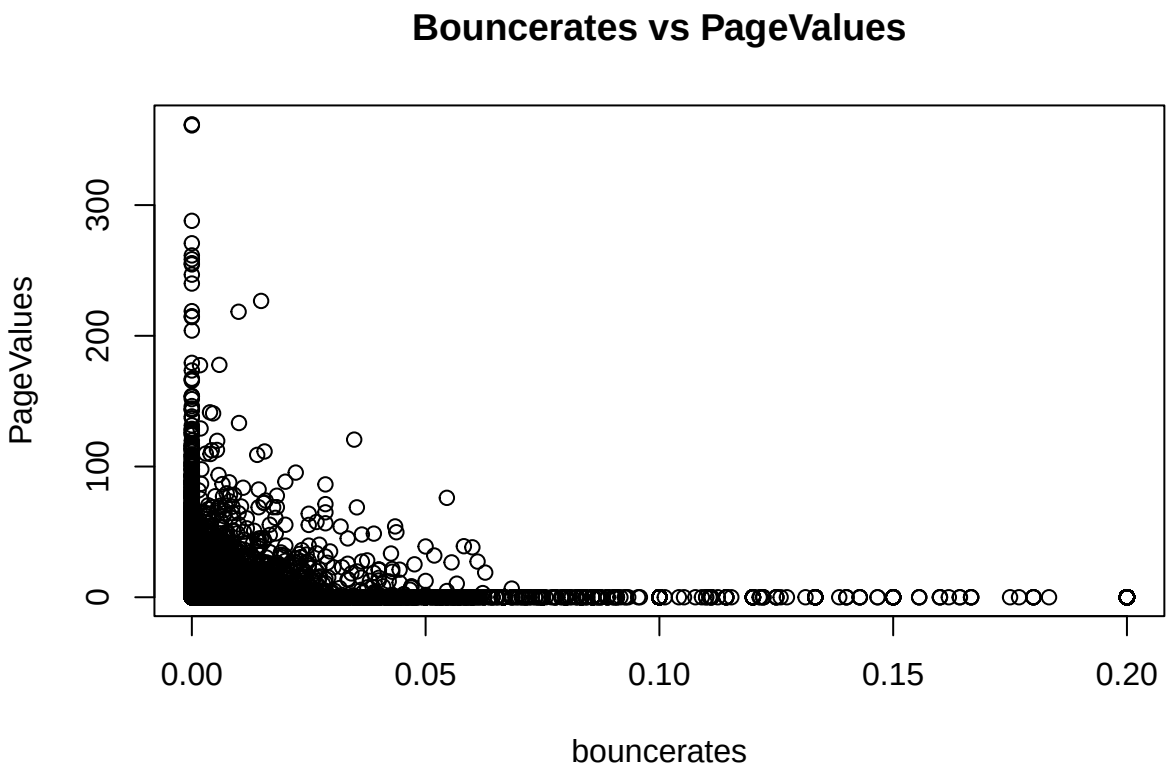
The correlation between the product related and product duration is positive correlation.

```
#Scatter plot between bounce rate vs exit rates
plot(df1$BounceRates, df1$ExitRates, xlab = ("bouncerates"), ylab = ("exitrates"),
     main = "Bouncerates vs Exitrates")
```



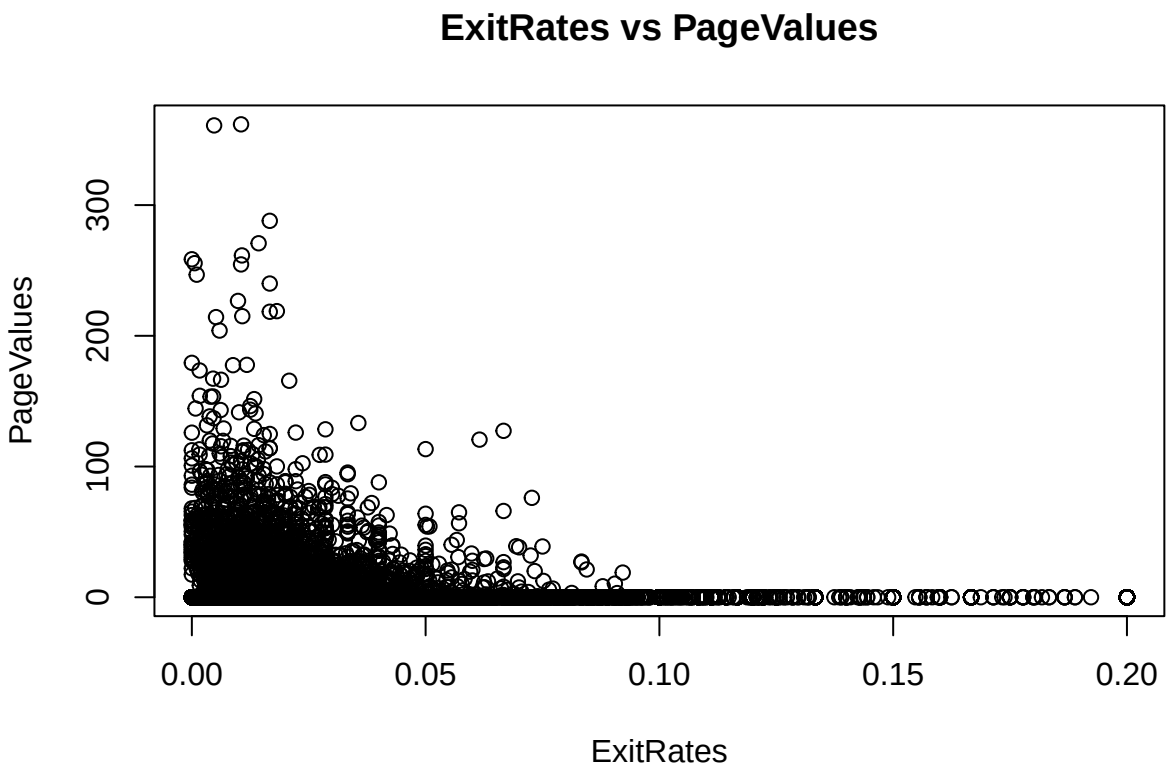
The correlation between bounce rates and exit rates is positive correlation.

```
#Scatter plot between bounce rate vs pagevalues
plot(df1$BounceRates, df1$PageValues, xlab = ("bouncerates"), ylab = ("PageValues"),
     main = "Bouncerates vs PageValues")
```



The correlation between bounce rates and page values is a positive correlation.

```
#Scatter plot between bounce rate vs Exit Rates
plot(df1$ExitRates, df1$PageValues, xlab = ("ExitRates"), ylab = ("PageValues"),
     main = "ExitRates vs PageValues")
```

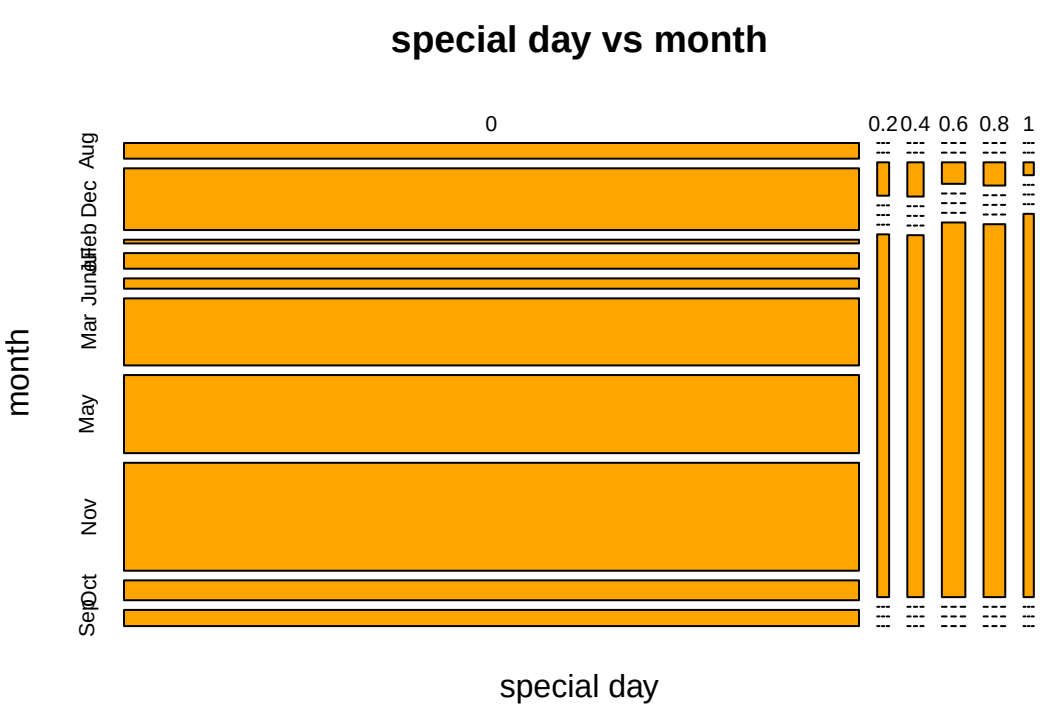


The correlation between the page values and exit rates is a positive correlation.

b). Categorical vs Categorical

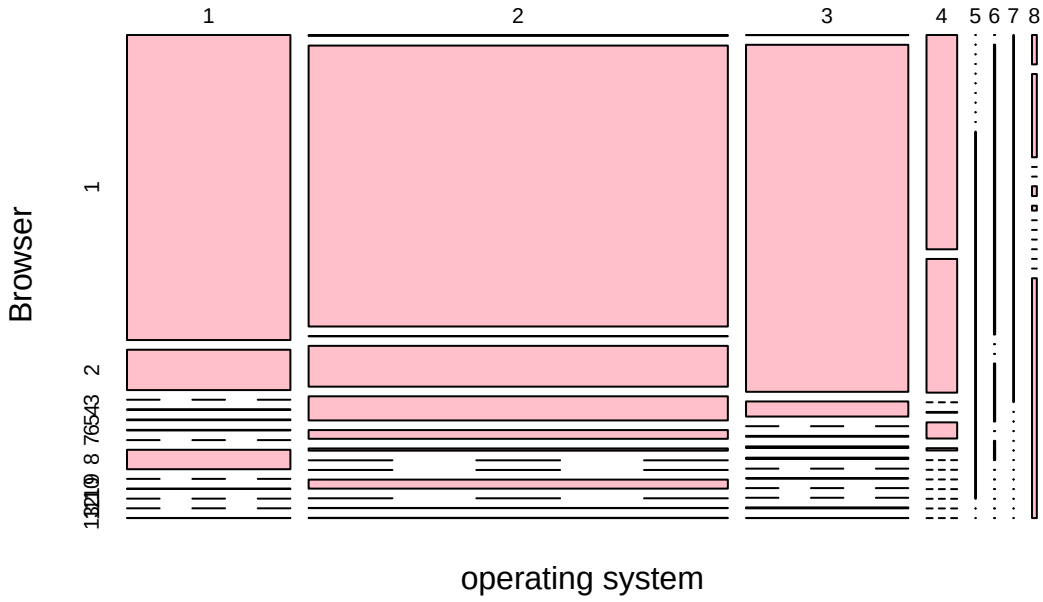
There is i apositive correlation between

```
#Mosaic plot of special day vs month
counts <- table(df1$SpecialDay, df1$Month)
#Create a mosaic plot
mosaicplot(counts, xlab="special day", ylab="month", main= "special day vs month",
           col ="orange")
```



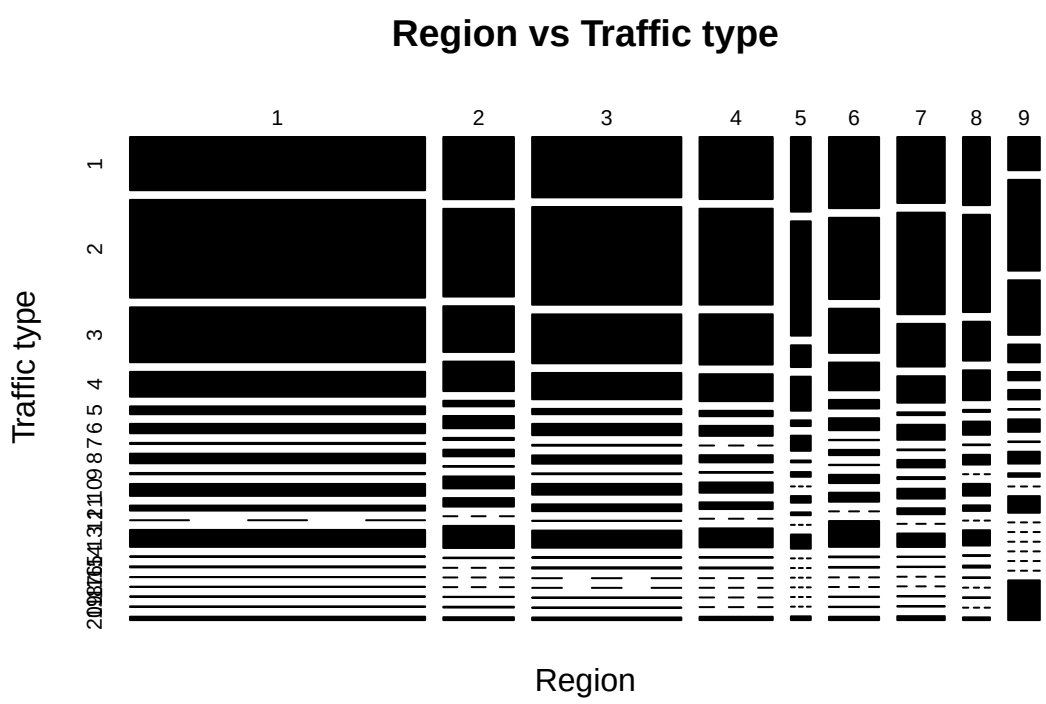
"0" to special day has the largest proportion in all the months.

```
#Mosaicplot of operating system vs browser
counts_1 <- table(df1$OperatingSystems, df1$Browser)
mosaicplot(counts_1, xlab="operating system", ylab="Browser",
           main="operating system vs browser", col ="pink")
```



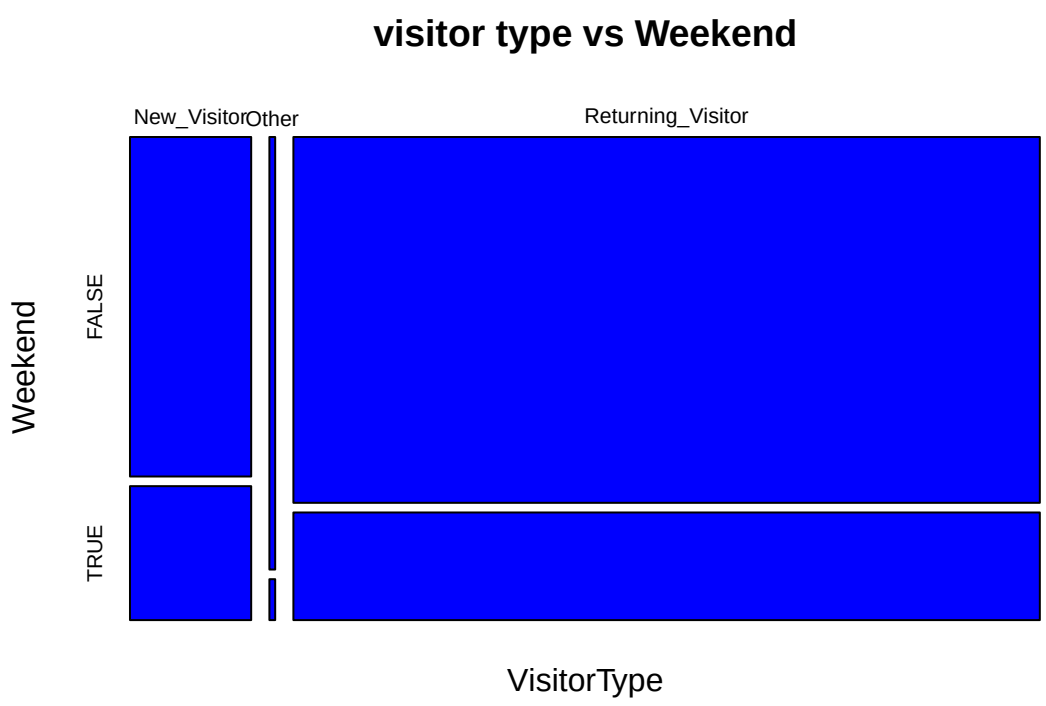
operating system number 2 and browser number 1 had the largest proportion

```
#Mosaic plot of Region vs Traffic type
counts <- table(df1$Region, df1$TrafficType)
#Create a mosaic plot
mosaicplot(counts, xlab="Region", ylab="Traffic type", main= "Region vs Traffic type",
           col ="black")
```



Traffic type 2 and the region number 1 have the largest share of the proportions

```
#Mosaic plot of visitor type vs Weekend
counts <- table(df1$VisitorType, df1$Weekend)
#Create a mosaic plot
mosaicplot(counts, xlab="VisitorType", ylab="Weekend", main= "visitor type vs Weekend",
           col ="blue")
```

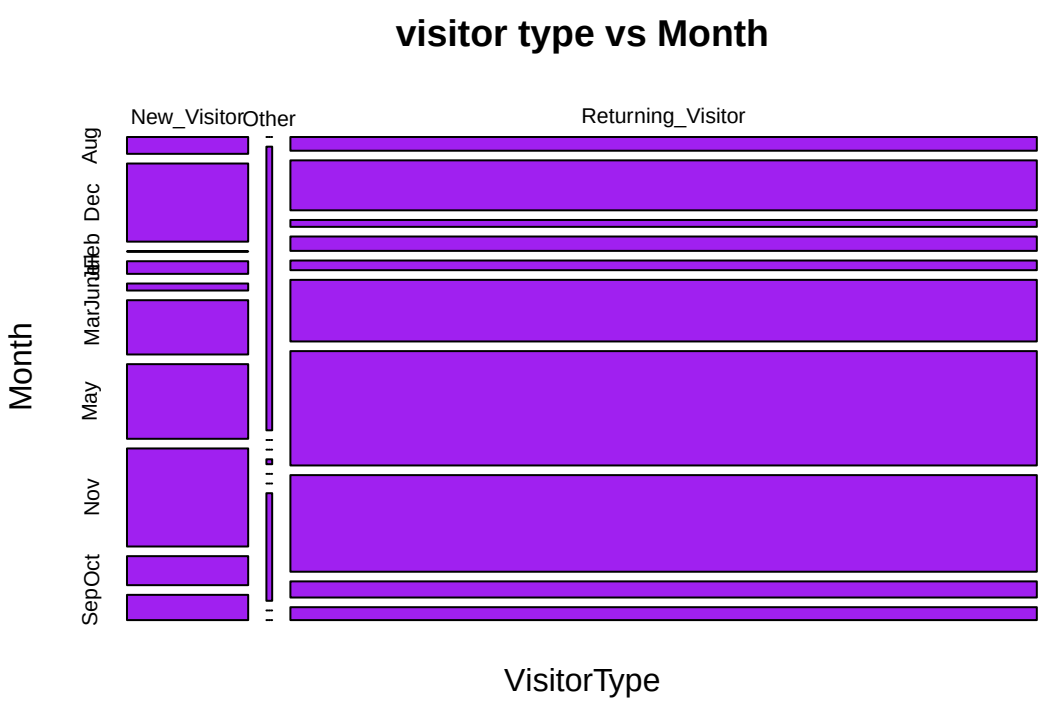


The returning visitors and weekend equal to false had the largest share of proportions

```
#Mosaic plot of special day vs visitor type
counts <- table(df1$SpecialDay, df1$VisitorType)
#Create a mosaic plot
mosaicplot(counts, xlab="Special day", ylab="Visitortype", main= "special day vs Visitor type",
           col ="green")
```

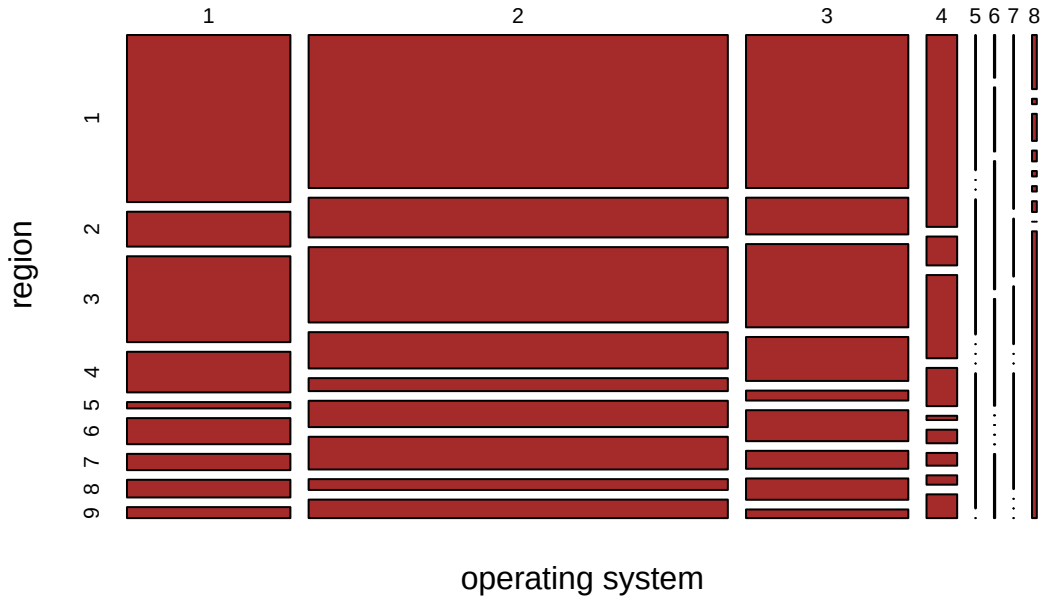
The zero "0" values of the special day and the returning visitor had the largest value of the proportion.

```
#Mosaic plot of visitor type vs month
counts <- table(df1$VisitorType, df1$Month)
#Create a mosaic plot
mosaicplot(counts, xlab="VisitorType", ylab="Month", main= "visitor type vs Month",
           col ="purple")
```



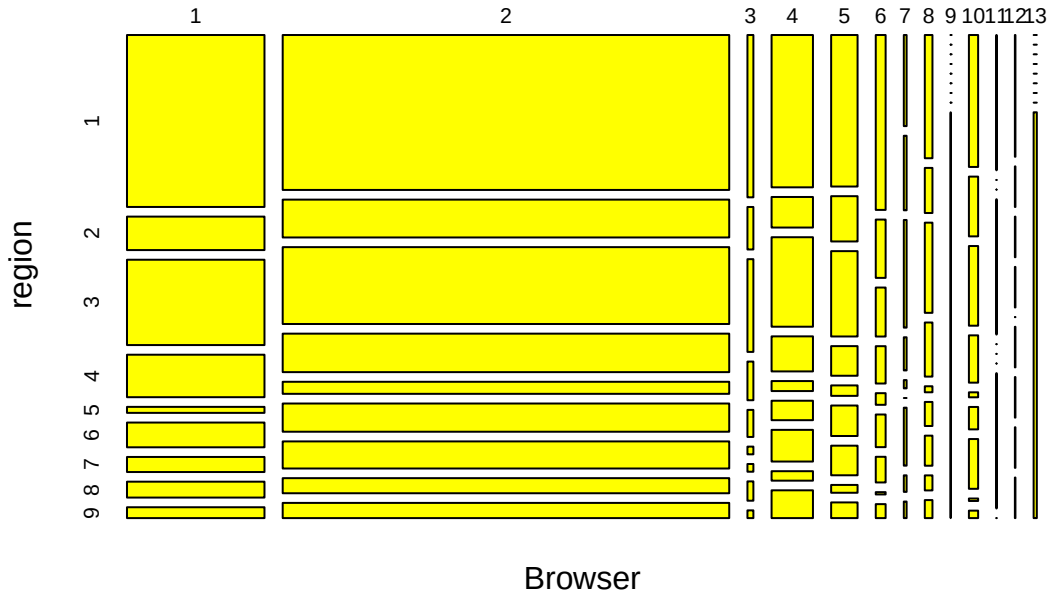
May and November has the largest proportion of both visitors

```
#Mosaic plot of operating system vs region
counts <- table(df1$OperatingSystems, df1$Region)
#Create a mosaic plot
mosaicplot(counts, xlab="operating system", ylab="region", main= "operating system vs region",
           col ="brown")
```

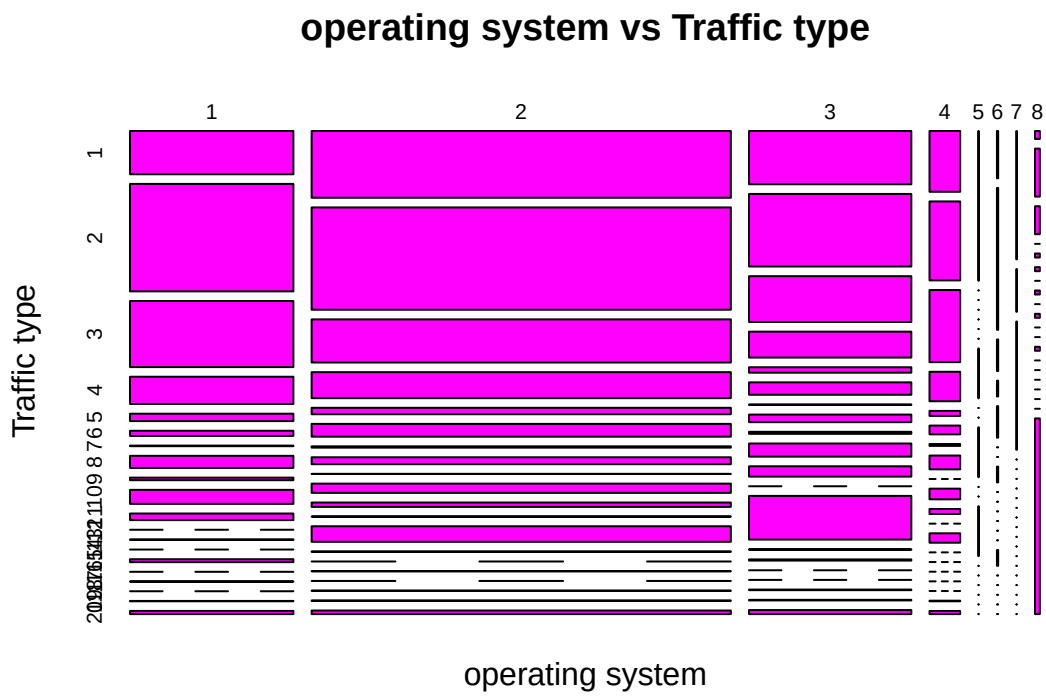


operating system number 2 has the largest proportions in all the regions followed by operating system 1.

```
#Mosaic plot of Browser vs region
counts <- table(df1$Browser, df1$Region)
#Create a mosaic plot
mosaicplot(counts, xlab="Browser", ylab="region", main= "Browser vs region",
           col ="yellow")
```

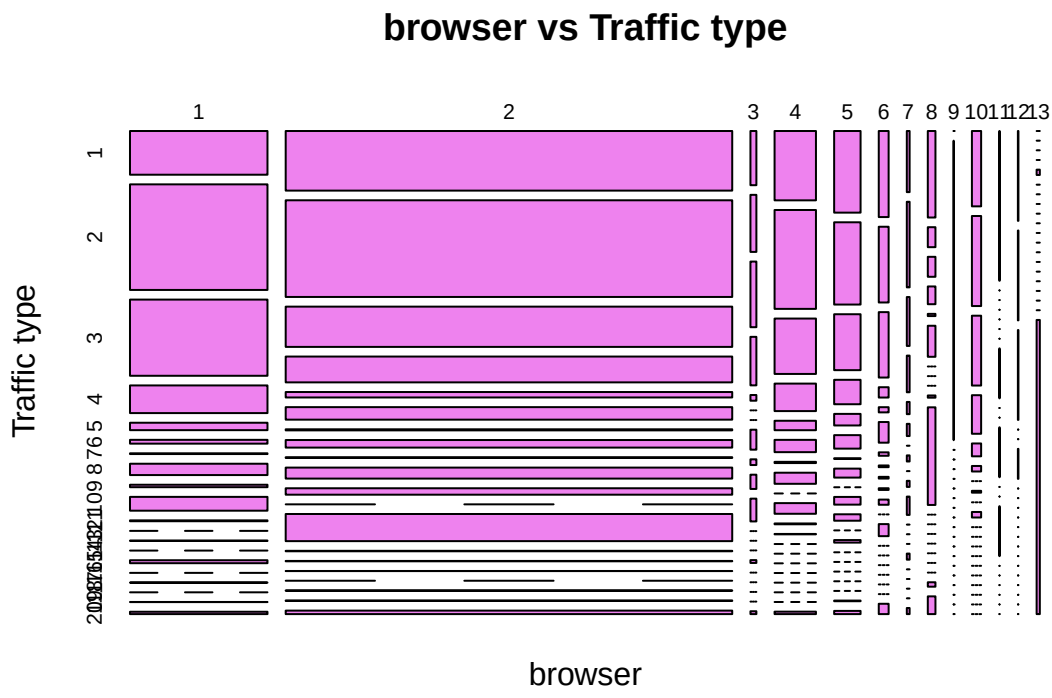



```
#Mosaic plot of operating system vs traffic type
counts <- table(df1$OperatingSystems, df1$TrafficType)
#Create a mosaic plot
mosaicplot(counts, xlab="operating system", ylab="Traffic type", main= "operating system vs Traffic type
           col ="magenta")
```



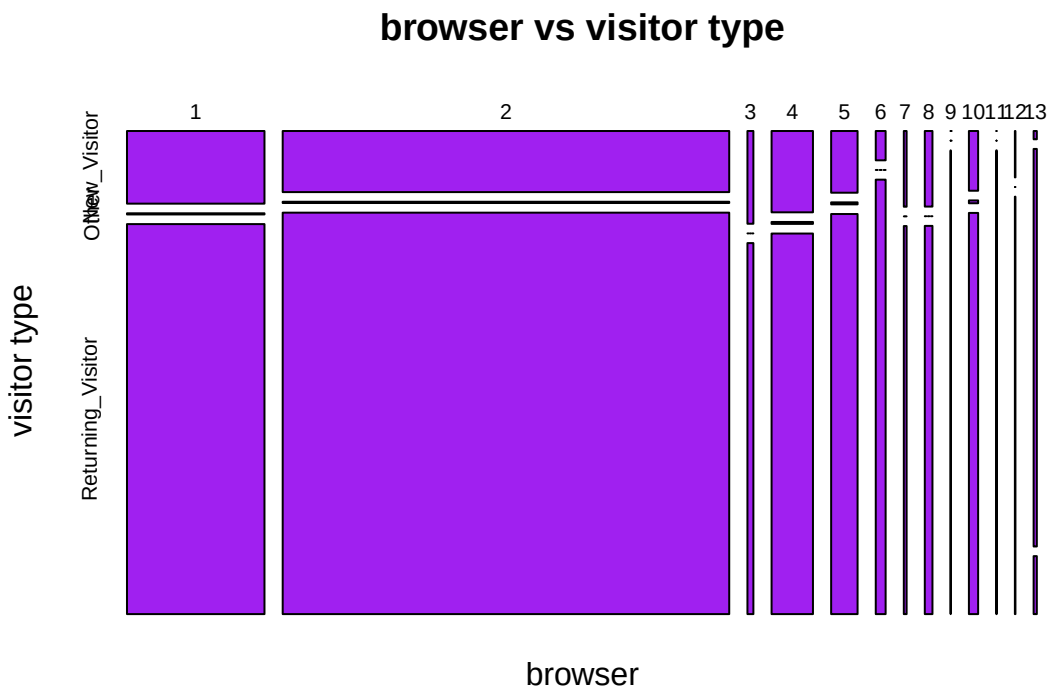
operating system type 2 has the largest proportion in all the traffics followed by operating system 1.

```
#Mosaic plot of browser vs traffic type
counts <- table(df1$Browser, df1$TrafficType)
#Create a mosaic plot
mosaicplot(counts, xlab="browser", ylab="Traffic type", main= "browser vs Traffic type",
           col ="violet")
```



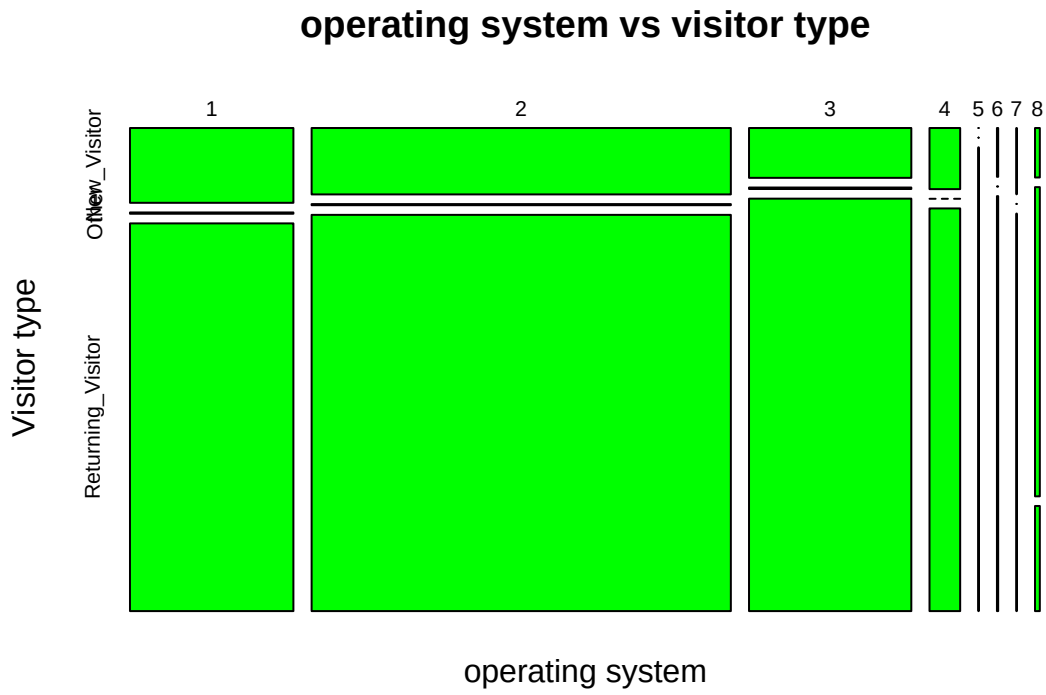
The browser2 has the largest proportions of all the traffic types followed by browser 1

```
#Mosaic plot of browser vs traffic type
counts <- table(df1$Browser, df1$VisitorType)
#Create a mosaic plot
mosaicplot(counts, xlab="browser", ylab="visitor type", main= "browser vs visitor type",
           col ="purple")
```



browser type 2 has the largest proportion of in all the visitors of the page.

```
#Mosaic plot of operating system vs Visitor type
counts <- table(df1$OperatingSystems, df1$VisitorType)
#Create a mosaic plot
mosaicplot(counts, xlab="operating system", ylab="Visitor type", main= "operating system vs visitor type
           col ="green")
```



operating system type2 has the largest proportion of among all the visitors.

6. Implement the solution.

#a) Kmeans clustering

```
#Check the stracture of the dataset
str(df1)
```

```
## 'data.frame':    12199 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" "Returni
```

```
##  $ Weekend                : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ Revenue                : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  - attr(*, "na.action")= 'omit' Named int [1:14] 1066 1133 1134 1135 1136 1137 1474 1475 1476 1477 .
##   ..- attr(*, "names")= chr [1:14] "1066" "1133" "1134" "1135" ...
```

```
#Label encoding non numeric character
#Convert the logic to string datatype

#Load the library
library(superml)
```

```
## Loading required package: R6
```

```
label <- LabelEncoder$new()
df1$Month <- label$fit_transform(df1$Month)
df1$VisitorType <- label$fit_transform(df1$VisitorType)
```

```
#Remove the label from the dataset since kmeans clustering is un supervised
#Learning.
df1_new <- df1[, c(1:16)]
head(df1_new, 2)
```

```
##   Administrative Administrative_Duration Informational Informational_Duration
## 1              0                       0             0                      0
## 2              0                       0             0                      0
##   ProductRelated ProductRelated_Duration BounceRates ExitRates PageValues
## 1              1                       0         0.2       0.2          0
## 2              2                      64         0.0       0.1          0
##   SpecialDay Month OperatingSystems Browser Region TrafficType VisitorType
## 1          0     0                1       1      1           1           0
## 2          0     0                2       2      1           2           0
```

```
#Normalize the dataset
#Normalization function
normalize <- function(x){
  return ((x-min(x)) / (max(x)-min(x)))}
#Apply the normalization to the dataset
df1_new_norm <- normalize(df1_new)
#preview two rows of Normarlized dataset
head(df1_new_norm, 2)
```

```
##   Administrative Administrative_Duration Informational Informational_Duration
## 1   1.563122e-05            1.563122e-05  1.563122e-05           1.563122e-05
## 2   1.563122e-05            1.563122e-05  1.563122e-05           1.563122e-05
##   ProductRelated ProductRelated_Duration  BounceRates    ExitRates   PageValues
## 1   3.126245e-05            1.563122e-05 1.875747e-05 1.875747e-05 1.563122e-05
## 2   4.689367e-05            1.016029e-03 1.563122e-05 1.719434e-05 1.563122e-05
##     SpecialDay        Month OperatingSystems      Browser       Region
## 1 1.563122e-05 1.563122e-05     3.126245e-05 3.126245e-05 3.126245e-05
## 2 1.563122e-05 1.563122e-05     4.689367e-05 4.689367e-05 3.126245e-05
##    TrafficType  VisitorType
## 1 3.126245e-05 1.563122e-05
## 2 4.689367e-05 1.563122e-05
```

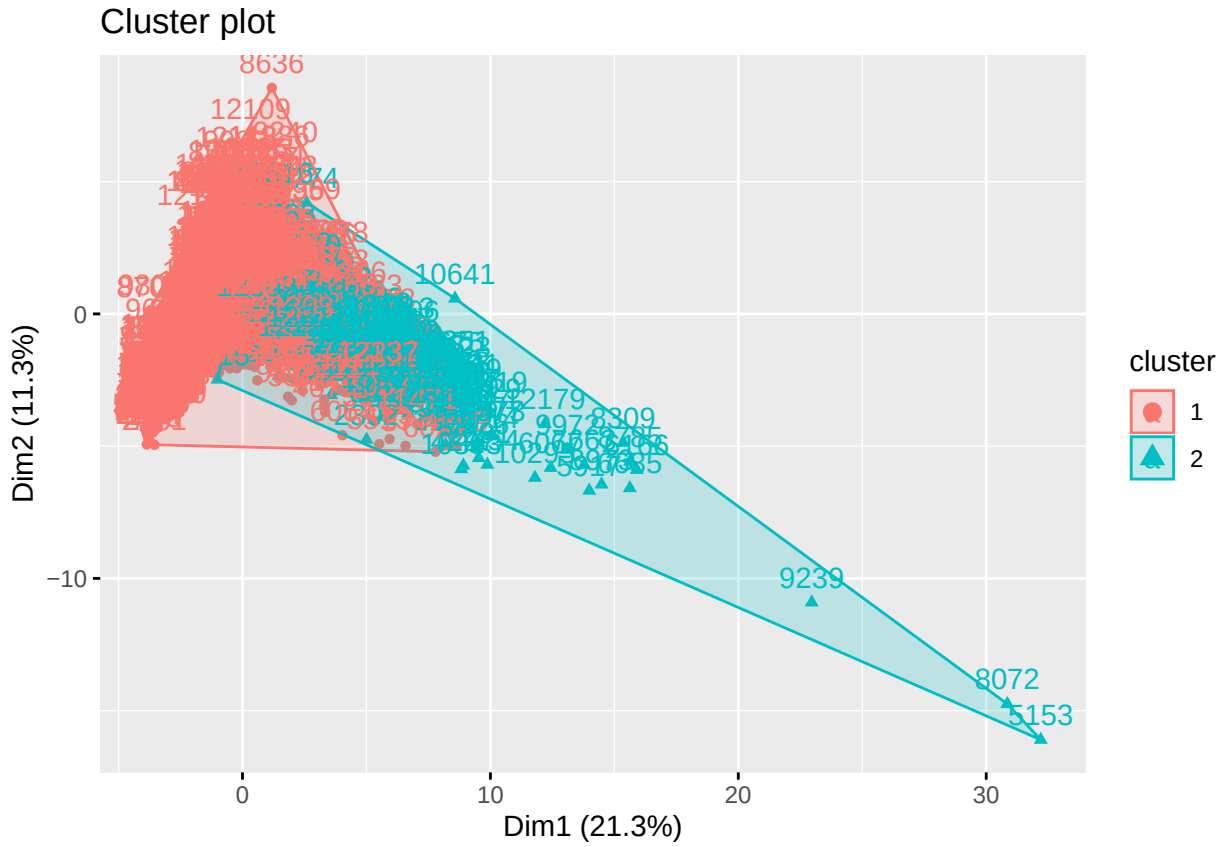
```
#Apply Kmeans clustering
k_cluster <- kmeans(df1_new_norm, centers = 2, nstart = 20)
```

```
#Visualization
library(cluster)
library(factoextra)
```

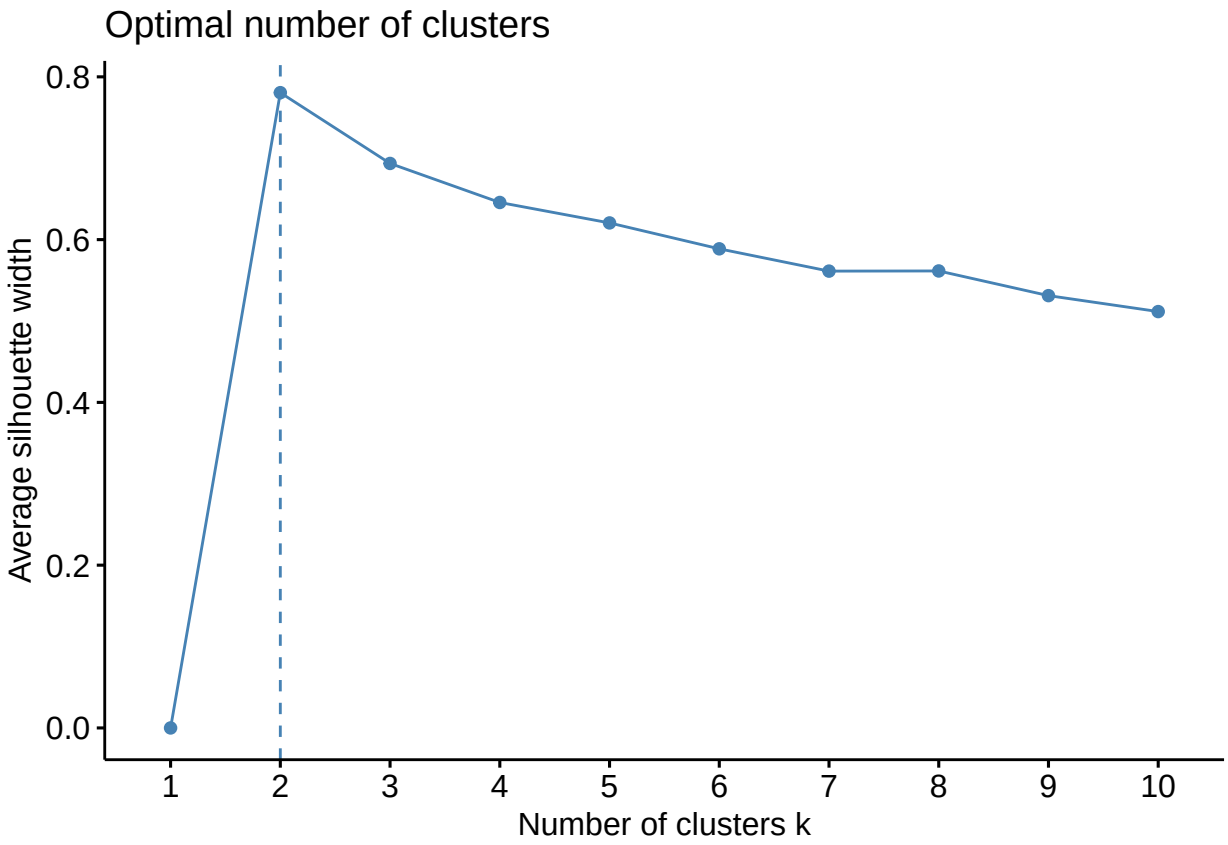
```
## Loading required package: ggplot2
```

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

```
fviz_cluster(k_cluster, data = df1_new_norm)
```




```
#Determining the optimal of value of k using silhouette methond
fviz_nbclust(x = df1_new_norm, FUNcluster = kmeans, method = 'silhouette')
```

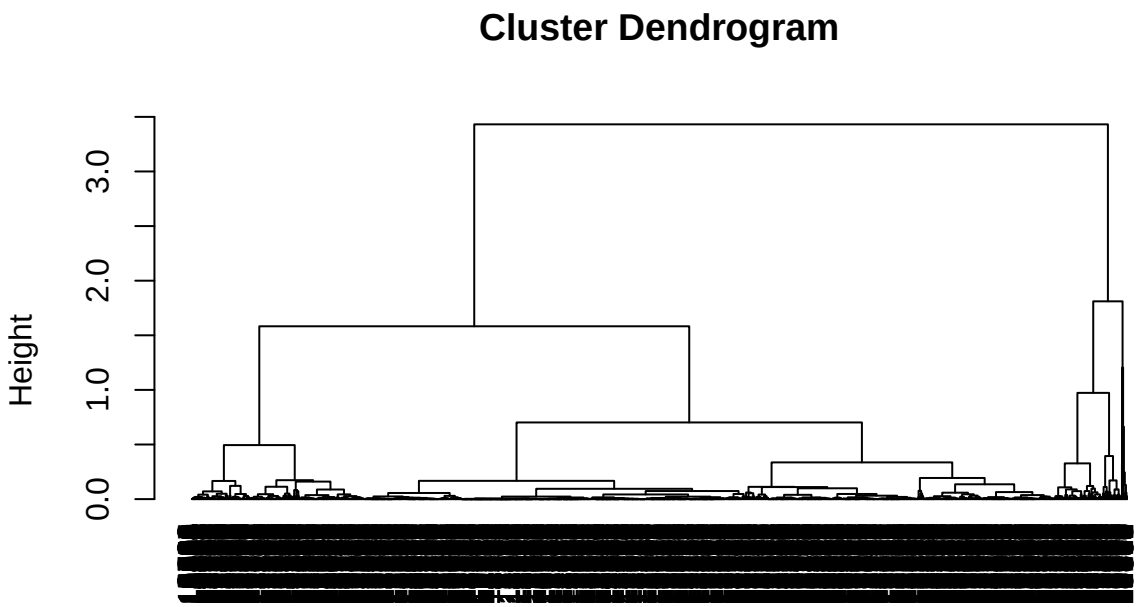


b) Hierarchical clustering

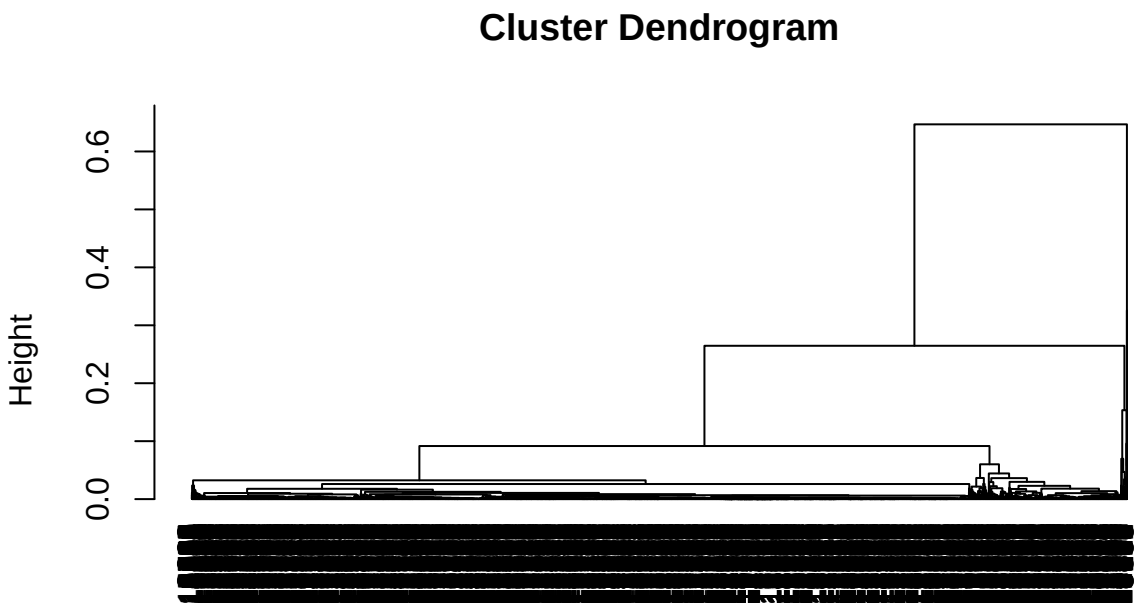
```
#Define distance D
D <- dist(df1_new_norm, method = "euclidean")
```

```
#hierarchical clustering using the Ward.D2's method
res.hc <- hclust(D, method = "ward.D2" )
```

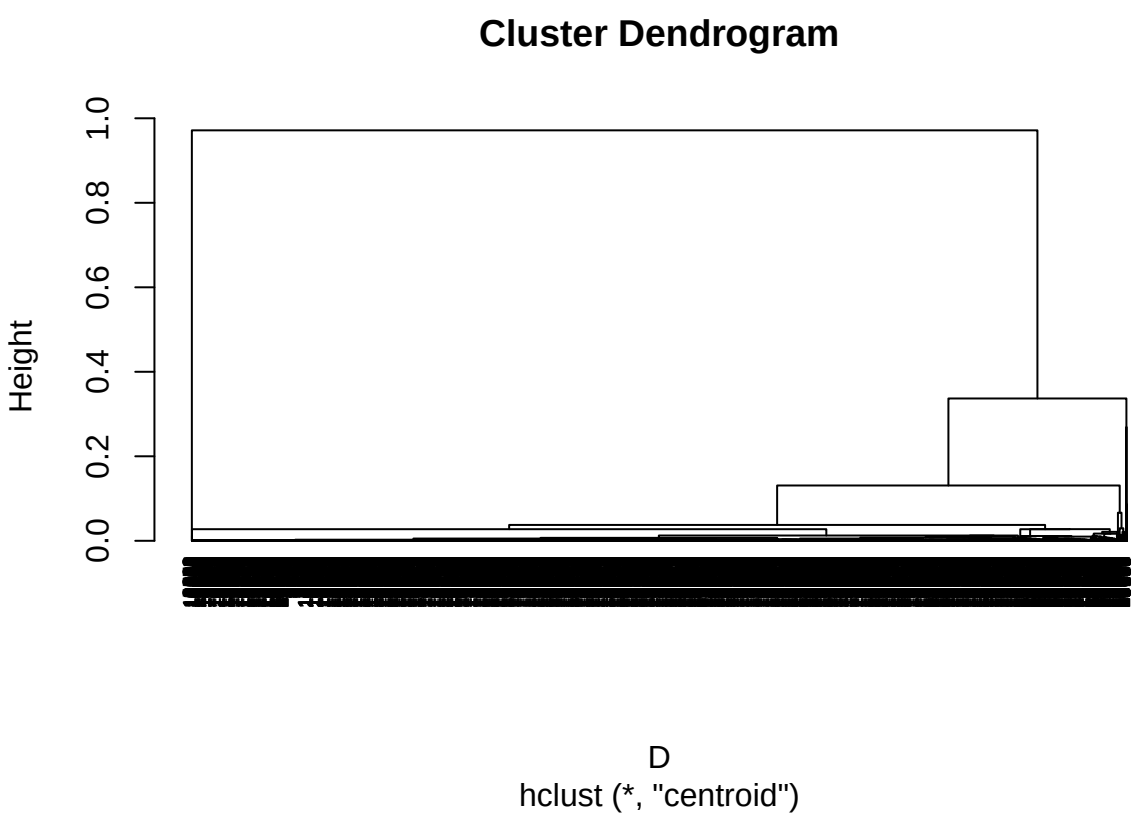
```
#Plot to come up with dendrogram
plot(res.hc, cex = 0.9, hang = -3)
```




```
#Plot using mcquitty
res.hc <- hclust(D, method = "mcquitty" )
plot(res.hc, cex = 0.9, hang = -3)
```




```r
#Use the centroid methond
res.hc <- hclust(D, method = "centroid" )
#Plot
plot(res.hc, cex = 0.6, hang = -1)
```



Conclusions

1. Visitors of the webpage prefer to visit the pages during the special day
2. The Visitors of the webpage, they visit the webpage on high number during the month of November and may.
3. The operating system preferred by the visitors of the webpage is the operating system number 2.
4. The browser preferred by the visitors of the webpage is the browser type 2.
5. The region with high visitors is the region type 1.
6. The visitors who visits the page on frequent basis is the returning visitors.
7. The visitors of the page prefer to visit the page on special day during the month of November.
8. Visitors with operating system type 2 prefer to use browser type 2 while accessing the webpages.
9. Visitors from region type 1 prefer the operating system type 2 and browser type 2
10. Users from the operating system type 2 prefer to use traffic type 2.
11. Visitors with the browser type 2 prefer the traffic type 2

12.The visitors of the webpage spend more time on the product related page compared to other pages.

Recomendations

1. The introduction of new products and brands on the webpage should be done on the special day.
2. The operating system type2, browser type 2 and traffic type 2 needs to be well maintained since they are highly preferred by the visitors.
3. The product related page should be well maintained since most of the visitors spend most there time on that particular page.

4. The product displayed in the web pages should target mostly visitors from region type1.
5. The returning customers should be offered with some incentives in order to encourage them to revisit the page frequently.

7. Challenge the solution.

```
Even though the kmeans clustering have managed  to cluster the customer
into two clusters, some other point of the data are still out of these two
clusters and it haven't been catered for.
```

8. Follow up question.

a). Do we have the right data?

```
Yes, the data was appropriate
```

b). Do we need another data?

No, the data was appropriate.

c).Do we have the right question?

Yes, the question is clear and straight forward.

9.Comparison and difference of kmeans clustering and Hierarchical clustering

1. The difference between the two arises in the number of clusters, for example in the kmeans clustering, the number of clusters are predefined while in Hierarchical its either agglomerative or divisive
2. The similarity between the two arises from the all of them being the clustering models.