Marketing and advertisement modelling in R

Angelo Sang

11/27/2021

Problem Statement

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. Perform clustering stating insights drawn from your analysis and visualizations. Upon implementation, provide comparisons between K-Means clustering vs Hierarchical clustering highlighting the strengths and limitations of each approach in the context of your analysis. Your findings should help inform the team in formulating the marketing and sales strategies of the brand.

Markdown Sections.

1.Problem Definition 2.Data Sourcing 3.Check the Data 4.Perform Data Cleaning 5.Perform Exploratory Data Analysis (Univariate, Bivariate & Multivariate) 6.Implement the Solution 7.Challenge the Solution 8.Follow up Questions

Data

The dataset consists of 10 numerical and 8 categorical attributes. The 'Revenue' attribute can be used as the class label.

Types of Pages: Administrative, Informational

Time spent on pages: Admin Duration and Info Duration

"Administrative", "Administrative Duration", "Informational", "Informational Duration", "Product Related" and "Product Related Duration" represents the number of different types of pages visited by the visitor in that session and total time spent in each of these page categories. The values of these features are derived from the URL information of the pages visited by the user and updated in real-time when a user takes an action, e.g. moving from one page to another.

Metrics: Bounce rate, Exit rate and Page Value

The "Bounce Rate", "Exit Rate" and "Page Value" features represent the metrics measured by "Google Analytics" for each page in the e-commerce site. The value of the "Bounce Rate" feature for a web page refers to the percentage of visitors who enter the site from that page and then leave ("bounce") without triggering any other requests to the analytics server during that session. The value of the "Exit Rate" feature for a specific web page is calculated as for all pageviews to the page, the percentage that was the last in the session. The "Page Value" feature represents the average value for a web page that a user visited before completing an e-commerce transaction.

Type of days: Speical or Ordinary

The "Special Day" feature indicates the closeness of the site visiting time to a specific special day (e.g. Mother's Day, Valentine's Day) in which the sessions are more likely to be finalized with the transaction. The value of this attribute is determined by considering the dynamics of e-commerce such as the duration between the order date and delivery date. For example, for Valentina's day, this value takes a nonzero value between February 2 and February 12, zero before and after this date unless it is close to another special day, and its maximum value of 1 on February 8.

Type of visit, Operating system, Browser and region(location)

The dataset also includes the operating system, browser, region, traffic type, visitor type as returning or new visitor, a Boolean value indicating whether the date of the visit is weekend, and month of the year.

Installing packages.

```
#install.packages("devtools")
#library(devtools)
#install_github("vqv/ggbiplot")
#install.packages("rtools")
#install.packages("DataExplorer")
#install.packages("Hmisc")
#install.packages("pastecs")
#install.packages("psych")
#install.packages("corrplot")
#install.packages("factoextra")
#install.packages("caret")
```

Loading the libraries

```
library("data.table")
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1

## -- 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(magrittr)
```

```
##
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':
##
##     set_names

## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(warn = -1)
library("ggbiplot")
```

```
## Loading required package: plyr

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:purrr':
##
##     compact
```

```
## Loading required package: scales

##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor

## Loading required package: grid
```

```
library(ggplot2)
library(lattice)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(DataExplorer)
library(Hmisc)
```

```
## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:plyr':
##
##     is.discrete, summarize

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(pastecs)
```

```
##
## Attaching package: 'pastecs'
```

```
## The following object is masked from 'package:magrittr':
##
##     extract

## The following objects are masked from 'package:dplyr':
##
##     first, last

## The following object is masked from 'package:tidyr':
##
##     extract

## The following objects are masked from 'package:data.table':
##
##     first, last
```

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:Hmisc':
##
##     describe

## The following objects are masked from 'package:scales':
##
##     alpha, rescale

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(factoextra)
```

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

```
library(caret)
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster

## The following object is masked from 'package:purrr':
##
##     lift
```

Loading the data

```
#specify the path where the file is located
library("data.table")
```

- obtaining the path to the working directrory

```
getwd()
```

```
## [1] "C:/Users/User/Desktop/MoringaExe/Advertisement-and-Marketing-models-in-R"
```

Loading the datasets

```
library(data.table)
df <- fread("http://bit.ly/EcommerceCustomersDataset")
head(df)
```

```
##    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
```

Previewing the top of the dataset

```
market_df <- data.frame(df)
head(market_df)
```

```
##   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
```

Previewing the summary of the dataset

```
summary(market_df)
```

```
##  Administrative   Administrative_Duration Informational
##  Min.   : 0.000   Min.   :  -1.00         Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.:   0.00         1st Qu.: 0.000
##  Median : 1.000   Median :   8.00         Median : 0.000
##  Mean   : 2.318   Mean   :  80.91         Mean   : 0.504
##  3rd Qu.: 4.000   3rd Qu.:  93.50         3rd Qu.: 0.000
##  Max.   :27.000   Max.   :3398.75         Max.   :24.000
##  NA's   :14       NA's   :14              NA's   :14
##  Informational_Duration ProductRelated   ProductRelated_Duration
##  Min.   :  -1.00        Min.   :  0.00   Min.   :   -1.0
##  1st Qu.:   0.00        1st Qu.:  7.00   1st Qu.:  185.0
```

```
##  Median :   0.00        Median : 18.00   Median :  599.8
##  Mean   :  34.51        Mean   : 31.76   Mean   : 1196.0
##  3rd Qu.:   0.00        3rd Qu.: 38.00   3rd Qu.: 1466.5
##  Max.   :2549.38        Max.   :705.00   Max.   :63973.5
##  NA's   :14             NA's   :14       NA's   :14
##   BounceRates         ExitRates         PageValues        SpecialDay
##  Min.   :0.000000   Min.   :0.00000   Min.   :  0.000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.01429   1st Qu.:  0.000   1st Qu.:0.00000
##  Median :0.003119   Median :0.02512   Median :  0.000   Median :0.00000
##  Mean   :0.022152   Mean   :0.04300   Mean   :  5.889   Mean   :0.06143
##  3rd Qu.:0.016684   3rd Qu.:0.05000   3rd Qu.:  0.000   3rd Qu.:0.00000
##  Max.   :0.200000   Max.   :0.20000   Max.   :361.764   Max.   :1.00000
##  NA's   :14         NA's   :14
##     Month           OperatingSystems    Browser           Region
##  Length:12330       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   : 2.357   Mean   :3.147
##                     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.   : 1.00   Length:12330       Mode :logical   Mode :logical
##  1st Qu.: 2.00   Class :character   FALSE:9462      FALSE:10422
##  Median : 2.00   Mode  :character   TRUE :2868      TRUE :1908
##  Mean   : 4.07
##  3rd Qu.: 4.00
##  Max.   :20.00
##
```

Properties of the dataset

Length

```
length(market_df)
```

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

Dimensions

```
dim(market_df)
```

```
## [1] 12330    18
```

Column Names

```
colnames(market_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"
```

Column data types

```
sapply(market_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

Missing Values

```
sum(is.na(market_df))
```

```
## [1] 112
```

Missing values per column

```
#Checking the sum of missing values per column
colSums(is.na(market_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 column with null values

```
# Return the column names containing missing observations
list_na <- colnames(market_df)[ apply(market_df, 2, anyNA) ]
list_na
```

```
## [1] "Administrative"          "Administrative_Duration"
## [3] "Informational"           "Informational_Duration"
## [5] "ProductRelated"          "ProductRelated_Duration"
## [7] "BounceRates"             "ExitRates"
```

Duplicates

```
duplicated_rows <- market_df[duplicated(market_df),]
dim(duplicated_rows)
```

```
## [1] 119  18
```

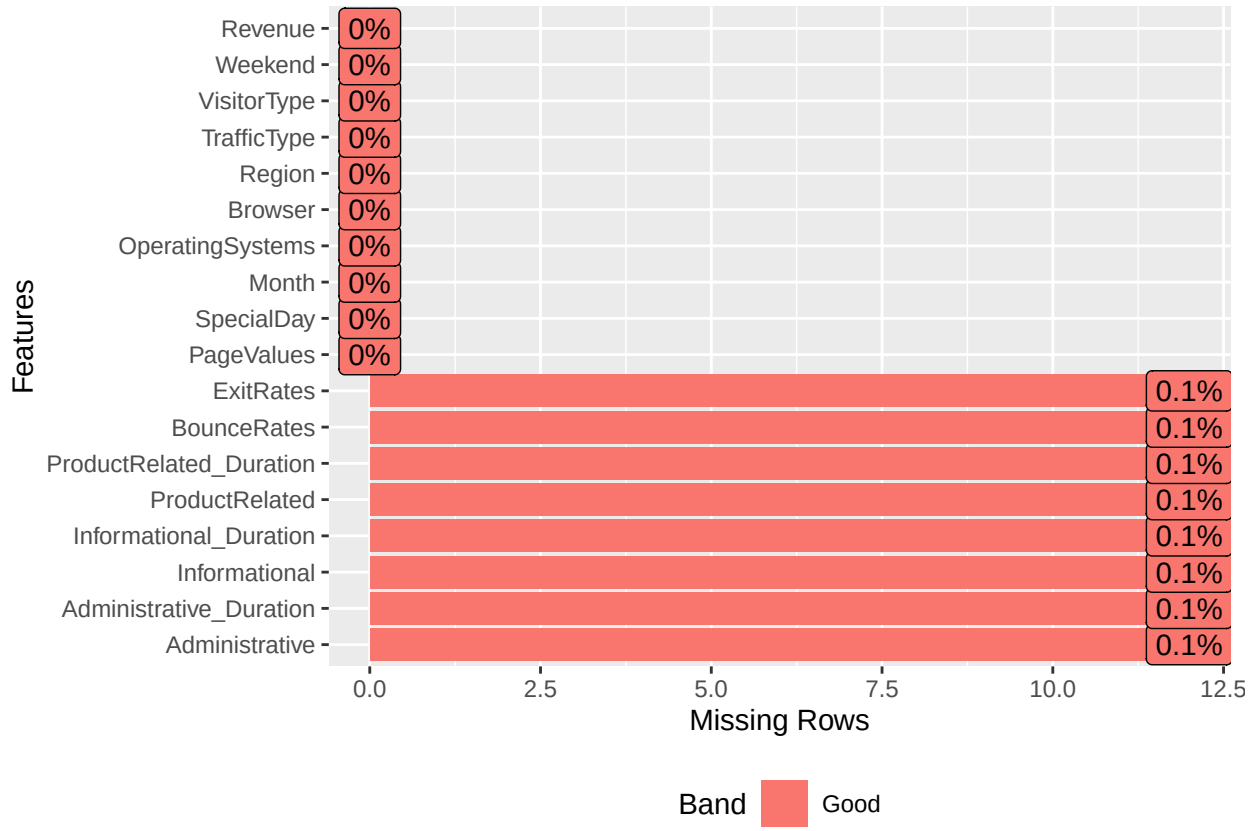
Removing duplicates

```
new_market_df <- market_df[-which(duplicated(market_df)), ]
dim(new_market_df)
```

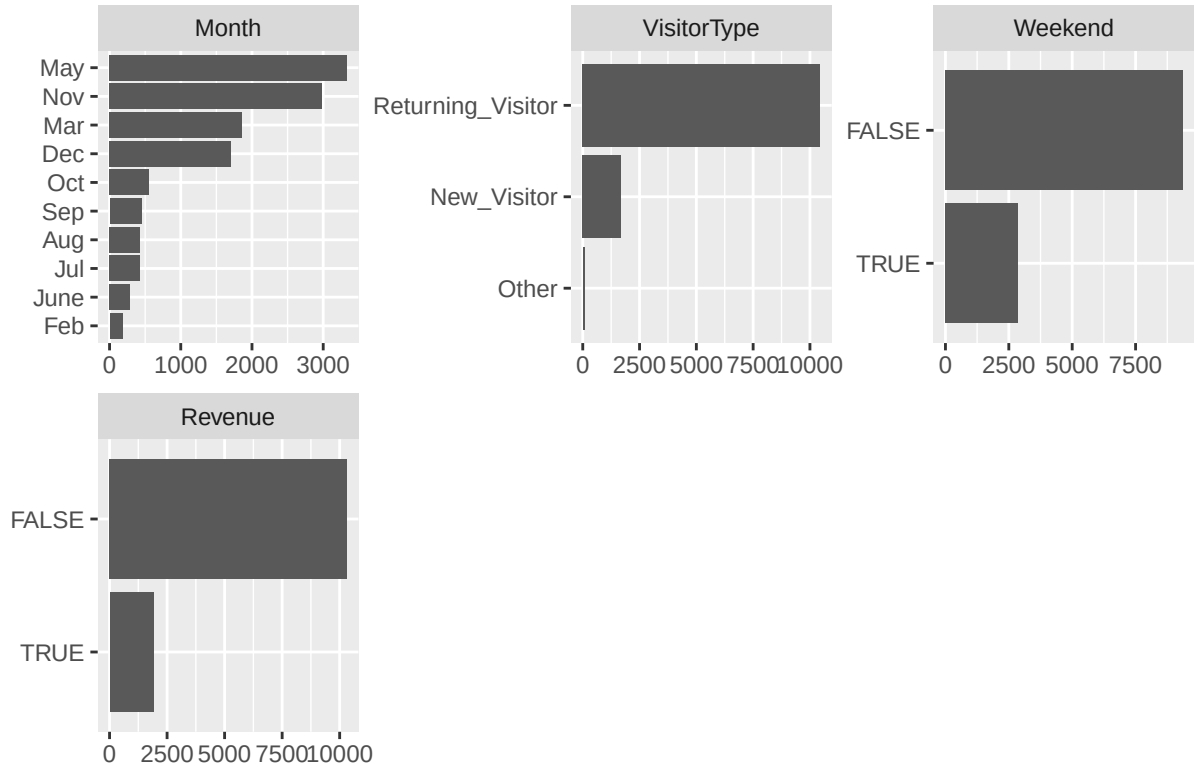
```
## [1] 12211    18
```

Exploring the data with Data Explorer

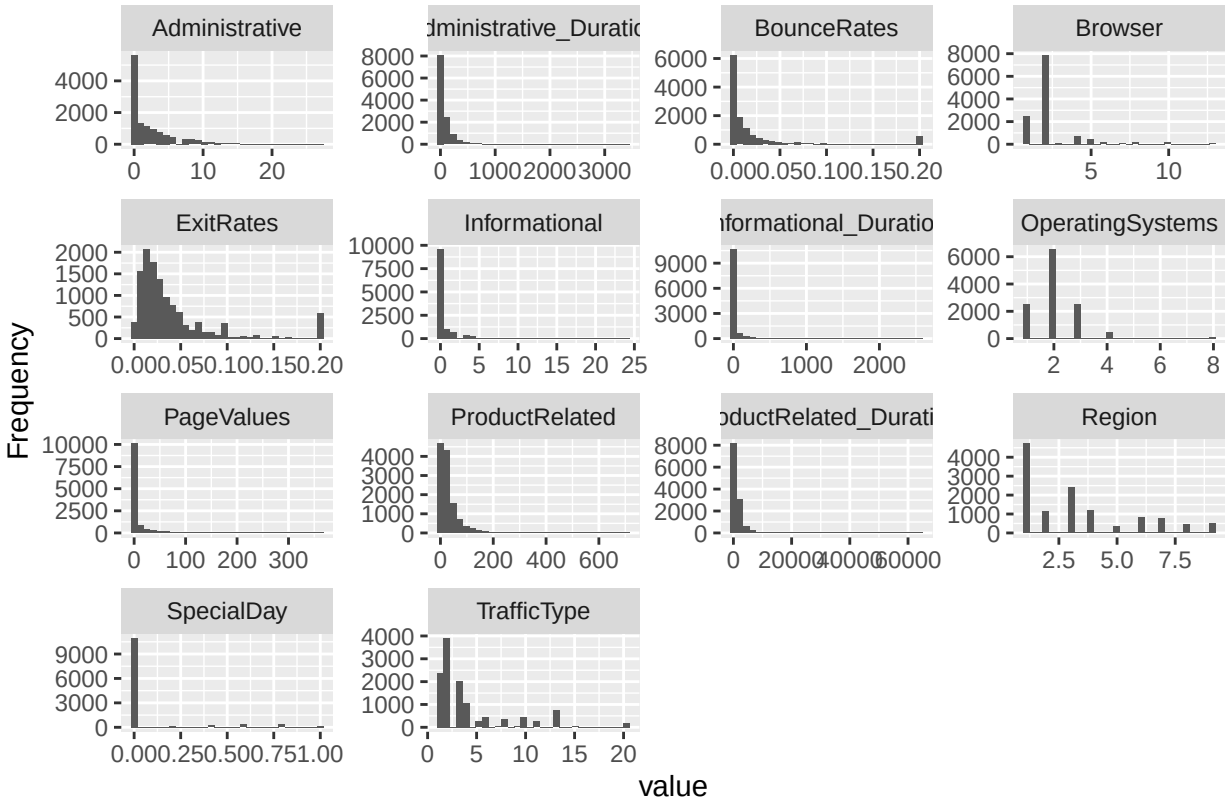
```
library(DataExplorer)
plot_missing(new_market_df) ## Are there missing values, and what is the missing data profile?
```




```
plot_bar(new_market_df) ## How does the categorical frequency for each discrete variable look like?
```




```
plot_histogram(new_market_df) ## What is the distribution of each continuous variable?
```




```
plot_str(new_market_df)
```

Data Types

```
sapply(new_market_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"
```

Perform Exploratory Data Analysis (Univariate, Bivariate &

Multivariate)

Univariate Analysis

###Administrative

```
unique(new_market_df$Administrative)
```

```
##  [1]  0  1  2  4 12  3 10  6  5  9  8 16 13 11  7 18 14 17 19 15 NA 24 22 21 20
## [26] 23 27 26
```

```
factor(unique(new_market_df$Administrative))
```

```
##  [1] 0    1    2    4    12   3    10   6    5    9    8    16   13   11   7
## [16] 18   14   17   19   15   <NA> 24   22   21   20   23   27   26
## 27 Levels: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 ... 27
```

- There are 14 missing values in this column thus we shall use the mean/mode to impute.
- Before performing any analysis on the column we have to drop the missing values.

```
length(new_market_df$Administrative)
```

```
## [1] 12211
```

```
dim(new_market_df)
```

```
## [1] 12211    18
```

```
sum(is.na(new_market_df))
```

```
## [1] 96
```

```
#there are 96 missing values in the new_market_df dataframe
markert_df2 <- new_market_df[-which(is.na(new_market_df)), ]
sum(is.na(markert_df2))
```

```
## [1] 0
```

```
dim(markert_df2)
```

```
## [1] 12199    18
```

```
colSums(is.na(markert_df2))
```

```
##          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
```

```
summary(markert_df2$Administrative)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    1.00    2.34    4.00   27.00
```

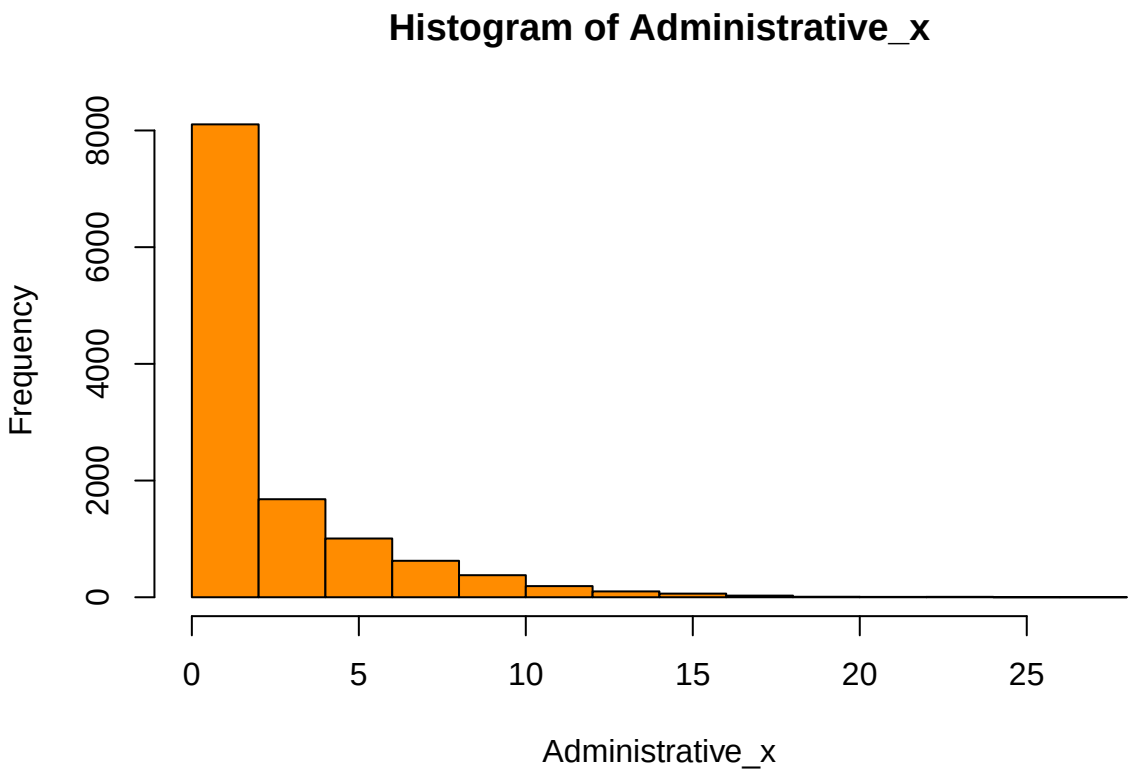
```
# median
median(markert_df2$Administrative)
```

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

```
# mode
Administrative_x <- markert_df2$Administrative
#sort(Daily.Internet.Usage_x)
names(table(Administrative_x))[table(Administrative_x)==max(table(Administrative_x))]
```

```
## [1] "0"
```

```
#each of the values printed below appear thrice in the dataset
#distribution
hist(Administrative_x, col=c("darkorange"))
```



- The adm distribution is right skewed.
- The highest value in the administrative column is 27
- The lowest value in the column is zero and it has the highest frequency.
- The mean is 2.34

Administrative_Duration

```
length(unique(markert_df2$Administrative_Duration))
```

```
## [1] 3336
```

```
summary(markert_df2$Administrative_Duration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -1.00    0.00    9.00   81.68   94.75 3398.75
```

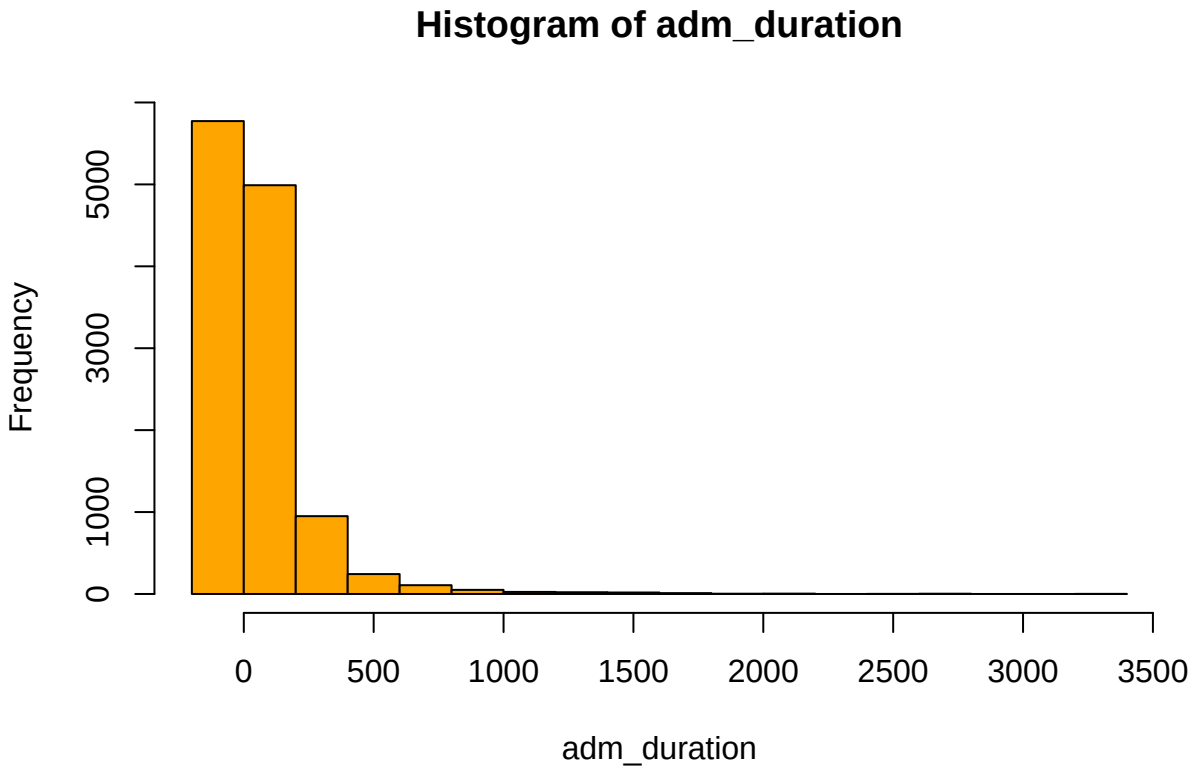
```
adm_duration <- markert_df2$Administrative_Duration
# median
median(adm_duration)
```

```
## [1] 9
```

```
# mode
#sort(adm_duration)
names(table(adm_duration))[table(adm_duration)==max(table(adm_duration ))]
```

```
## [1] "0"
```

```
#distribution
hist(adm_duration, col=c("orange"))
```



- The adm_duration distribution is right skewed.
- The highest value in the administrative column is 3398.75
- The lowest value in the column is 0 and it has the highest frequency.
- The mean is 81.68
- The median is 9

Information

```
length(unique(markert_df2$Informational))
```

```
## [1] 17
```

```r
#there are 17 unique elements in Informational column
summary(markert_df2$Informational)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.5088  0.0000 24.0000
```

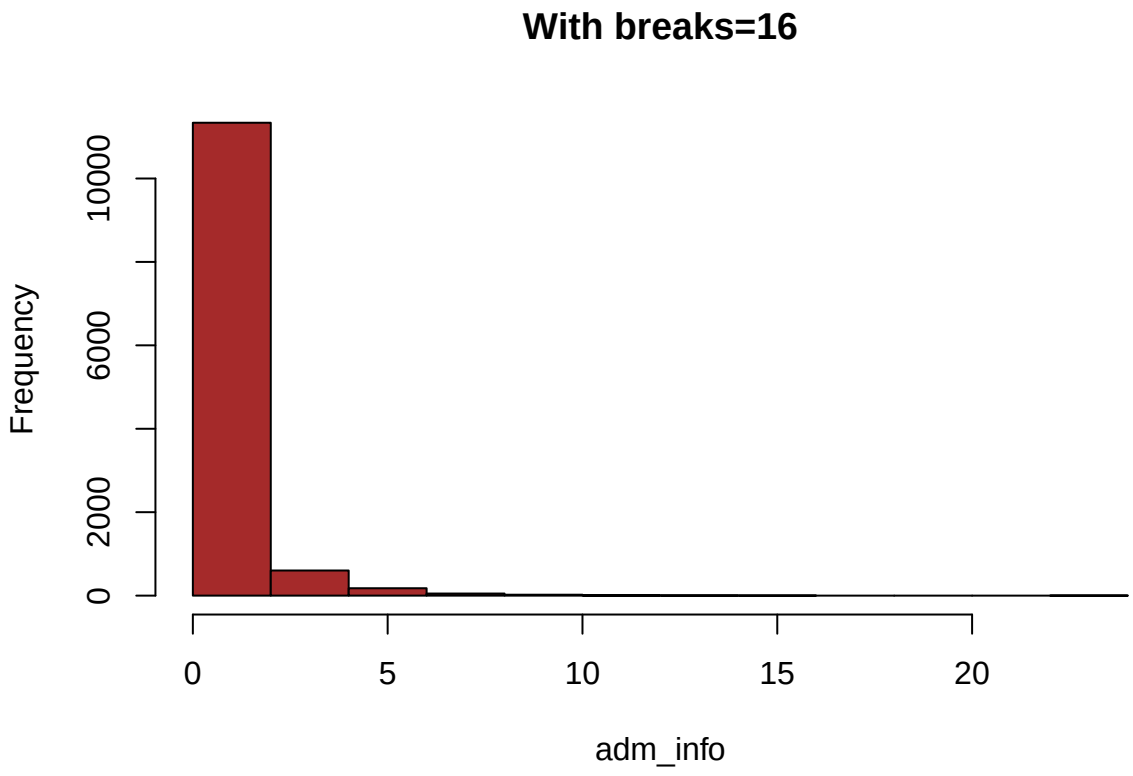
```r
adm_info <- markert_df2$Informational
# median
median(adm_info)
```

```
## [1] 0
```

```r
# mode
#sort(adm_duration)
names(table(adm_info))[table(adm_info)==max(table(adm_info ))]
```

```
## [1] "0"
```

```r
#The modal value in the information dataset is 0
#distribution
hist(adm_info,breaks = 16 , main="With breaks=16", col=c("brown"))
```



Informational_Duration

```
length(unique(markert_df2$Informational_Duration))
```

```
## [1] 1259
```

```
summary(markert_df2$Informational_Duration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -1.00    0.00    0.00   34.84    0.00 2549.38
```

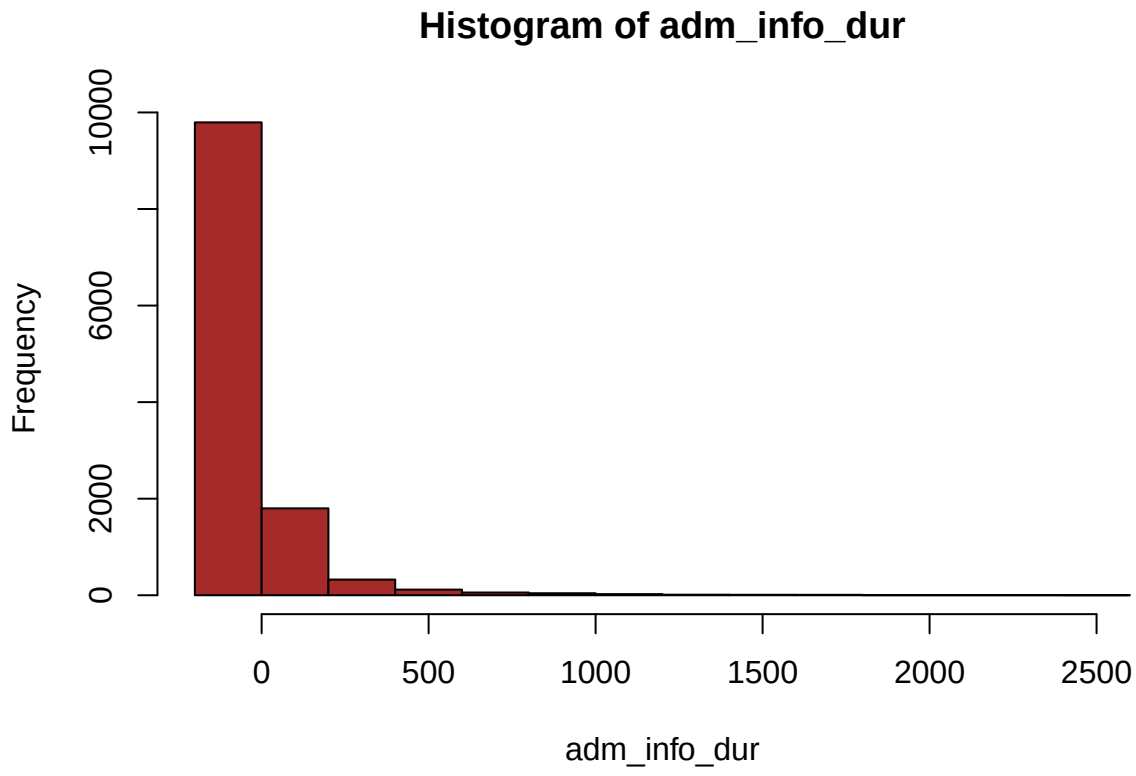
```
adm_info_dur <- markert_df2$Informational_Duration
# median
median(adm_info)
```

```
## [1] 0
```

```
# mode
#sort(adm_info_dur)
names(table(adm_info_dur))[table(adm_info_dur)==max(table(adm_info_dur ))]
```

```
## [1] "0"
```

```
#distribution
hist(adm_info_dur,col=c("brown"))
```



ProductRelated

```
length(unique(markert_df2$ProductRelated))
```

```
## [1] 311
```

```
summary(markert_df2$ProductRelated)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    8.00   18.00   32.06   38.00  705.00
```

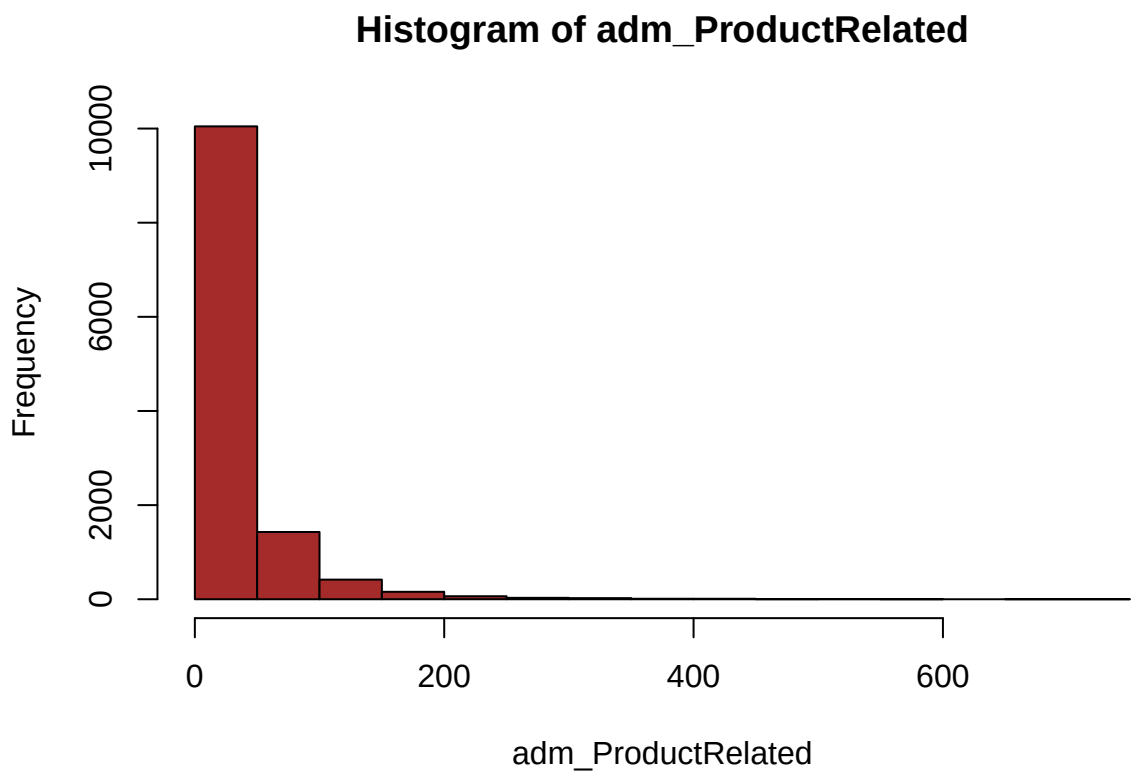
```
adm_ProductRelated <- markert_df2$ProductRelated
# median
median(adm_ProductRelated)
```

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

```
# mode
#sort(adm_info_dur)
names(table(adm_ProductRelated))[table(adm_ProductRelated)==max(table(adm_ProductRelated ))]
```

```
## [1] "1"
```

```
#distribution
hist(adm_ProductRelated,col=c("brown"))
```



ProductRelated_Duration

```
length(unique(markert_df2$ProductRelated_Duration))
```

```
## [1] 9552
```

```
summary(markert_df2$ProductRelated_Duration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    -1.0   193.6   609.5  1207.5  1477.6 63973.5
```

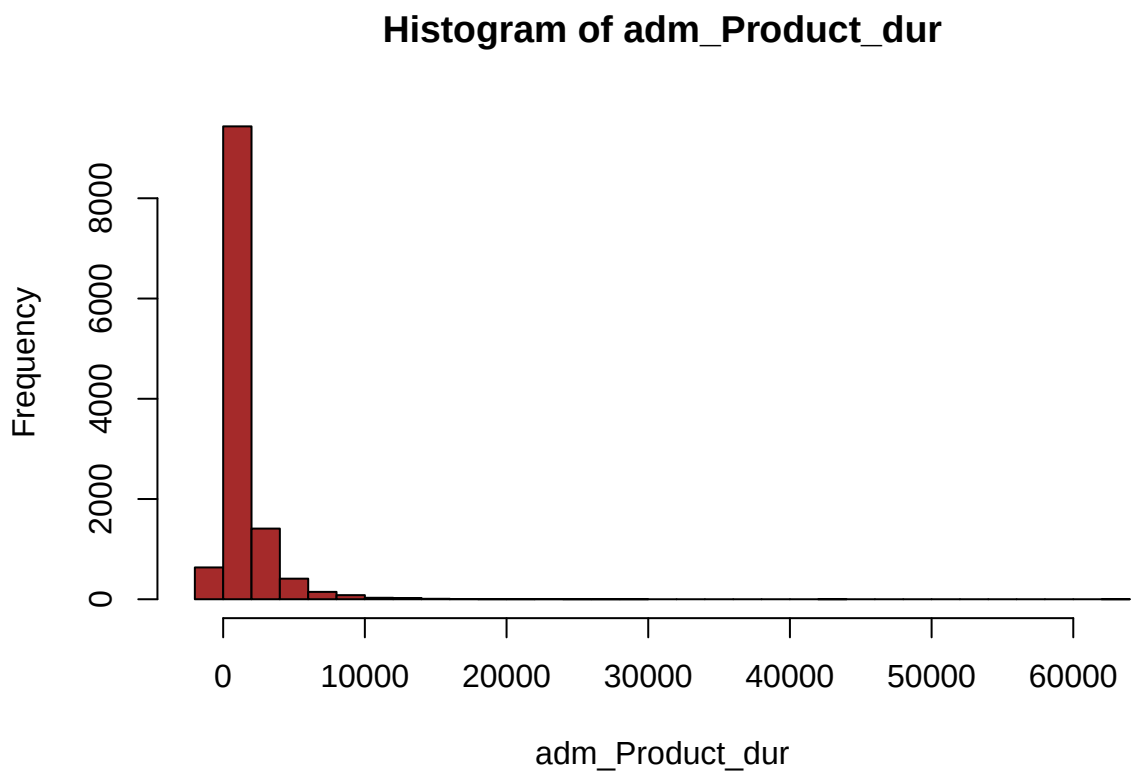
```
adm_Product_dur <- markert_df2$ProductRelated_Duration
# median
median(adm_Product_dur)
```

```
## [1] 609.5417
```

```
# mode
#sort(adm_info_dur)
names(table(adm_Product_dur))[table(adm_Product_dur)==max(table(adm_Product_dur ))]
```

```
## [1] "0"
```

```
#distribution
hist(adm_Product_dur,breaks=30,col=c("brown"))
```



BounceRates

```
length(unique(markert_df2$BounceRates))
```

```
## [1] 1872
```

```
summary(markert_df2$BounceRates)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00293 0.02045 0.01667 0.20000
```

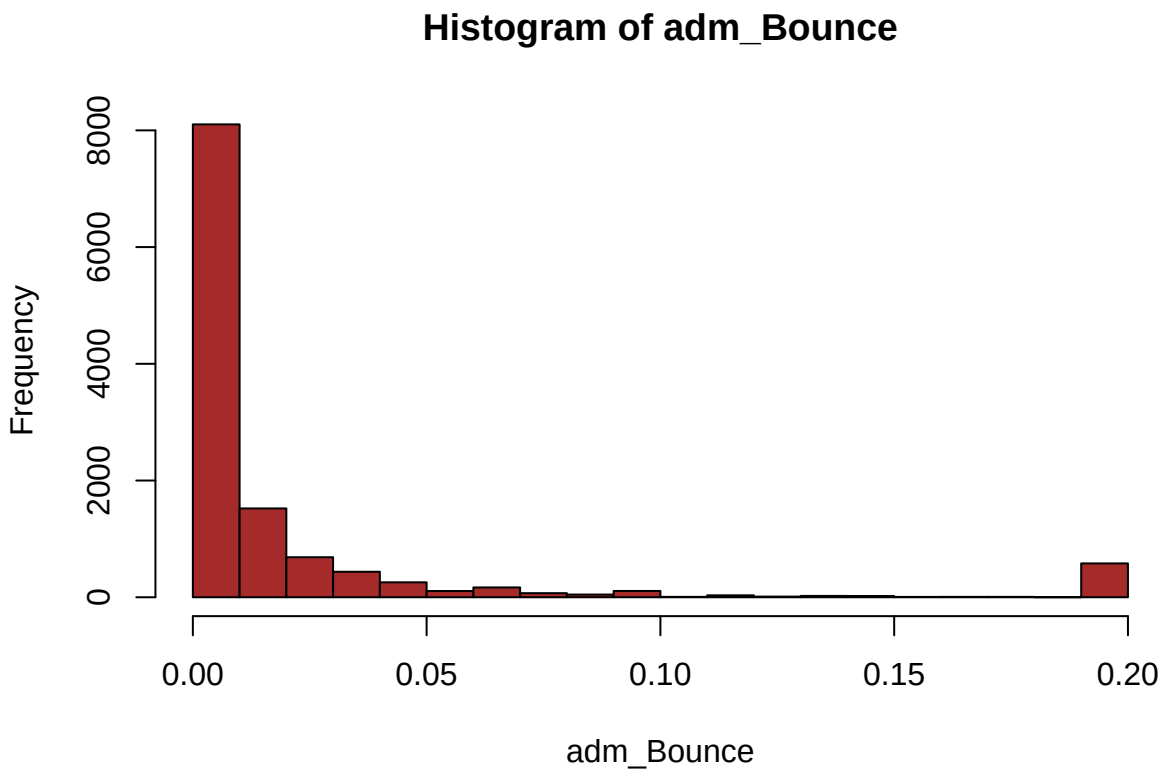
```
adm_Bounce <- markert_df2$BounceRates
# median
median(adm_Bounce)
```

```
## [1] 0.002930403
```

```
# mode
#sort(adm_info_dur)
names(table(adm_Bounce))[table(adm_Bounce)==max(table(adm_Bounce ))]
```

```
## [1] "0"
```

```
#distribution
hist(adm_Bounce,col=c("brown"))
```



ExitRates

```
length(unique(markert_df2$ExitRates))
```

```
## [1] 4777
```

```
summary(markert_df2$ExitRates)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.01422 0.02500 0.04150 0.04848 0.20000
```

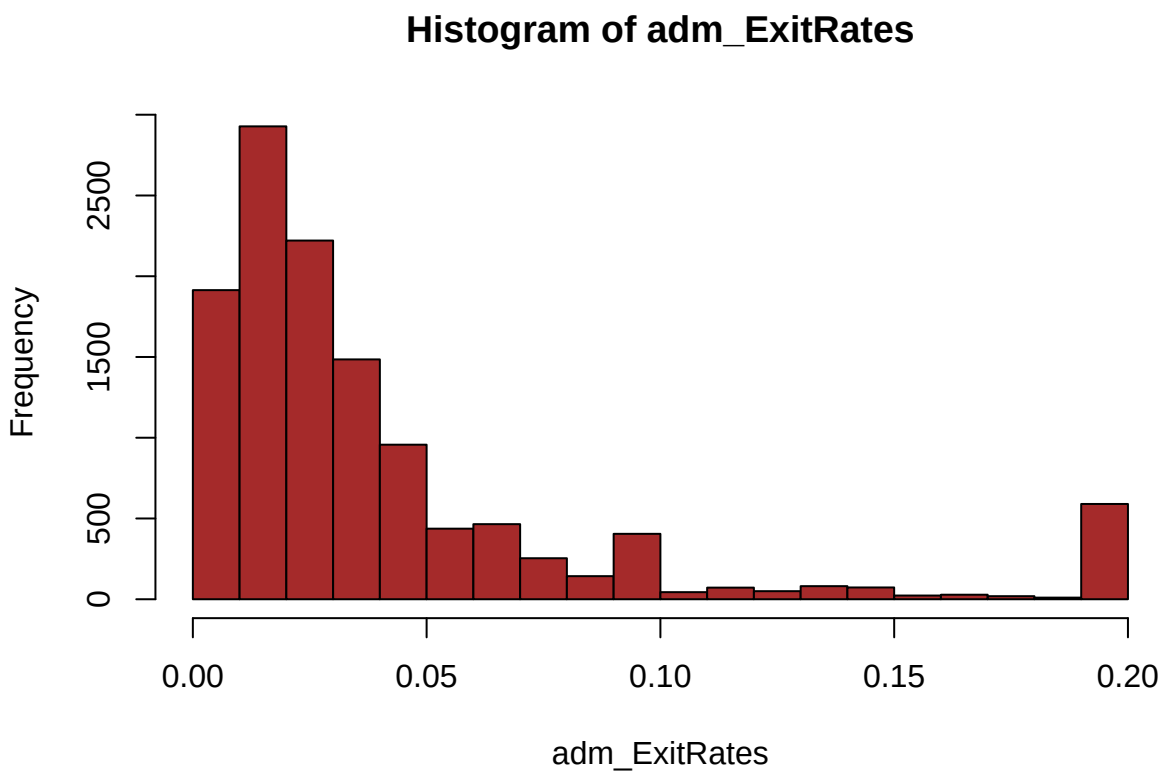
```
adm_ExitRates <- markert_df2$ExitRates
# median
median(adm_ExitRates)
```

```
## [1] 0.025
```

```
# mode
#sort(adm_info_dur)
names(table(adm_ExitRates))[table(adm_ExitRates)==max(table(adm_ExitRates ))]
```

```
## [1] "0.2"
```

```
#distribution
hist(adm_ExitRates,col=c("brown"))
```



Page Values

```
length(unique(markert_df2$PageValues))
```

```
## [1] 2704
```

```
summary(markert_df2$PageValues)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   5.952   0.000 361.764
```

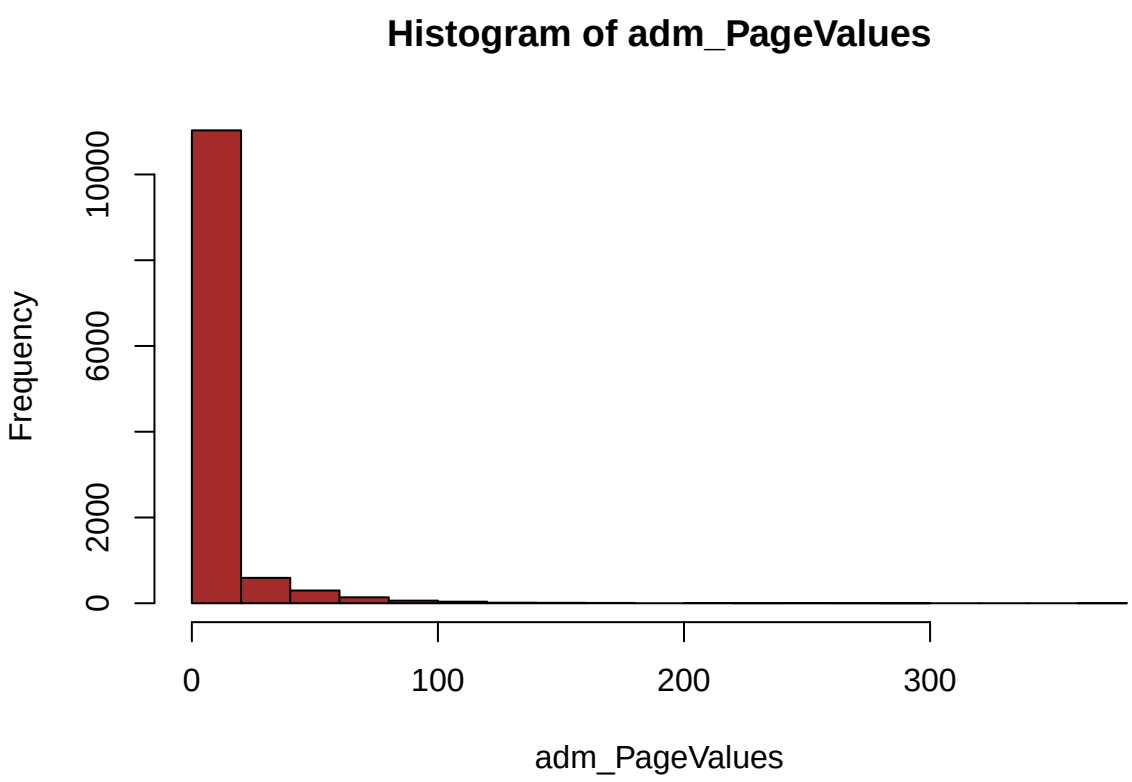
```
adm_PageValues <- markert_df2$PageValues
# median
median(adm_PageValues)
```

```
## [1] 0
```

```
# mode
#sort(adm_info_dur)
names(table(adm_PageValues))[table(adm_PageValues)==max(table(adm_PageValues
))]
```

```
## [1] "0"
```

```
#distribution
hist(adm_PageValues,col=c("brown"))
```



SpecialDay

```
length(unique(markert_df2$SpecialDay))
```

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

```
summary(markert_df2$SpecialDay)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.06197 0.00000 1.00000
```

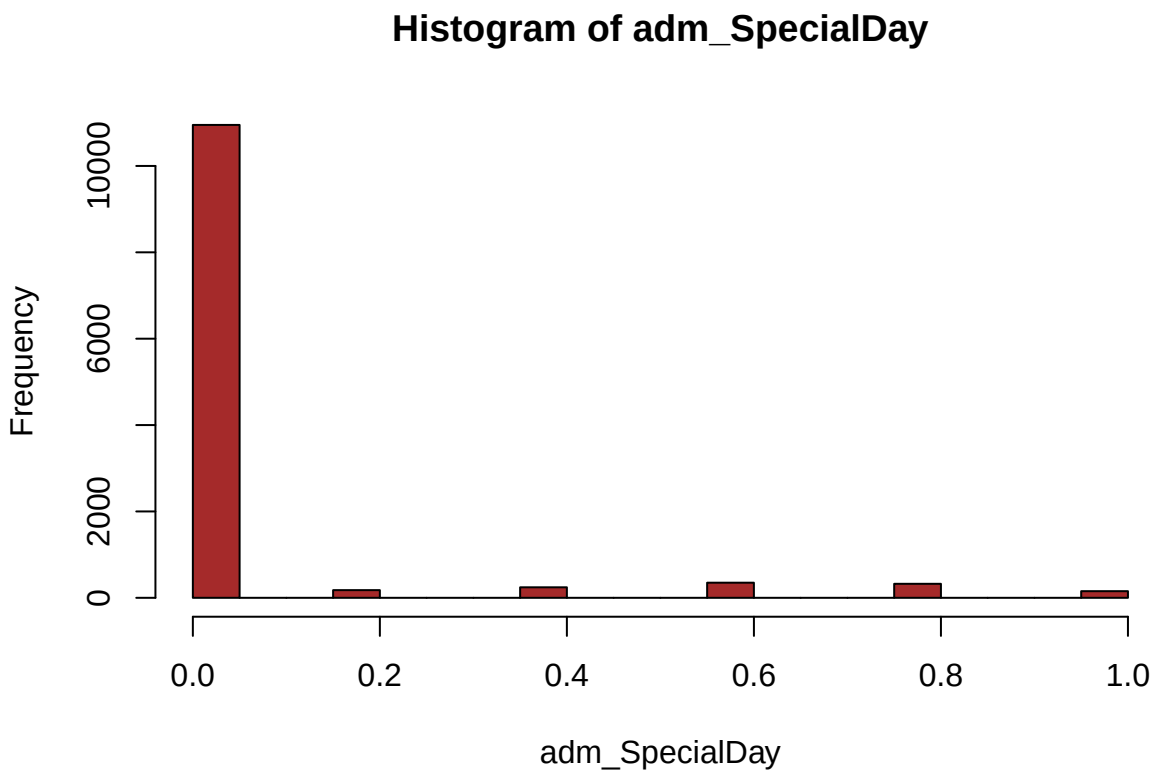
```
adm_SpecialDay <- markert_df2$SpecialDay
# median
median(adm_SpecialDay)
```

```
## [1] 0
```

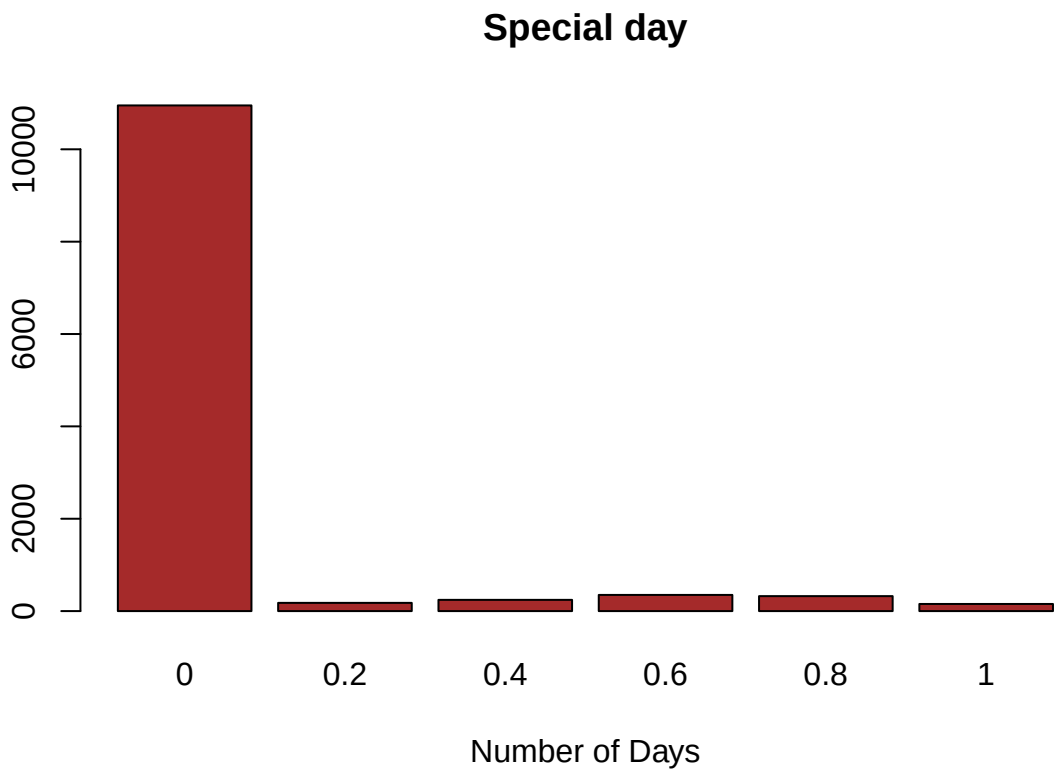
```
# mode
#sort(adm_info_dur)
names(table(adm_SpecialDay))[table(adm_SpecialDay)==max(table(adm_SpecialDay
))]
```

```
## [1] "0"
```

```
#distribution
hist(adm_SpecialDay,col=c("brown"))
```




```
# Simple Bar Plot
counts <- table(adm_SpecialDay)
barplot(counts, main="Special day",col=c("brown"),
 xlab="Number of Days")
```



Month

```
length(unique(markert_df2$Month))
```

```
## [1] 10
```

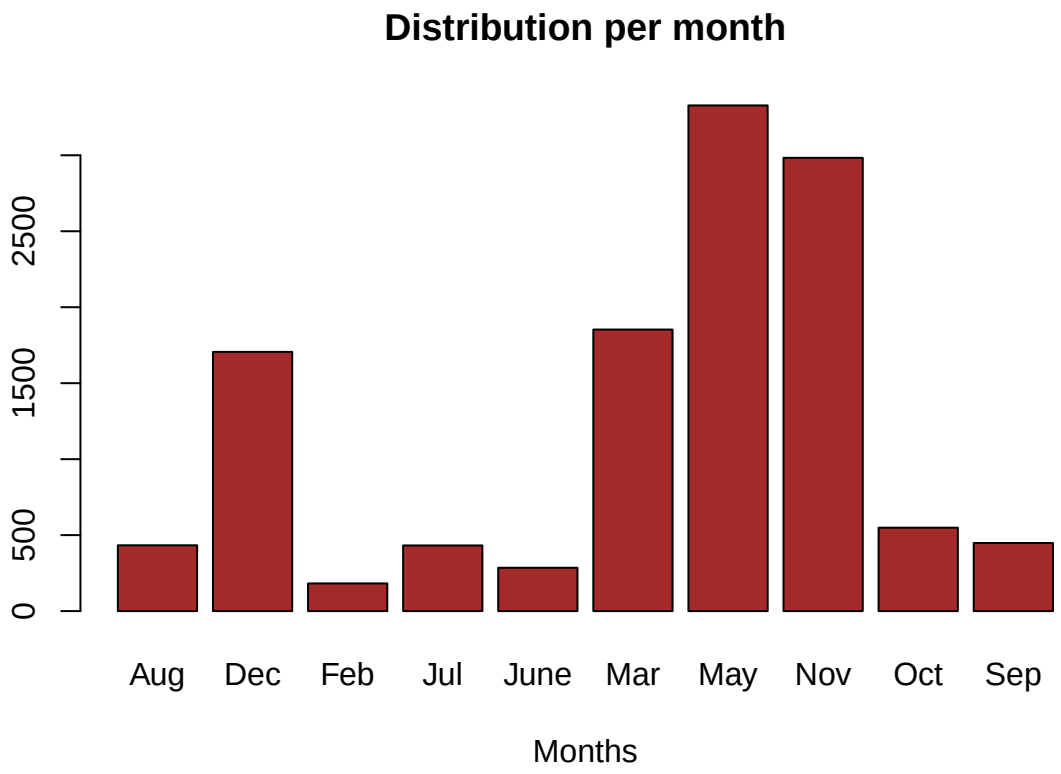
```
summary(markert_df2$Month)
```

```
##    Length     Class      Mode
##     12199 character character
```

```
adm_Month <- markert_df2$Month
# mode
#sort(adm_info_dur)
names(table(adm_Month))[table(adm_Month)==max(table(adm_Month ))]
```

```
## [1] "May"
```

```
#distribution
# Simple Bar Plot
counts <- table(adm_Month)
barplot(counts, main="Distribution per month",col=c("brown"),
 xlab="Months")
```



OperatingSystems

```
length(unique(markert_df2$OperatingSystems))
```

```
## [1] 8
```

```
summary(markert_df2$OperatingSystems)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   2.124   3.000   8.000
```

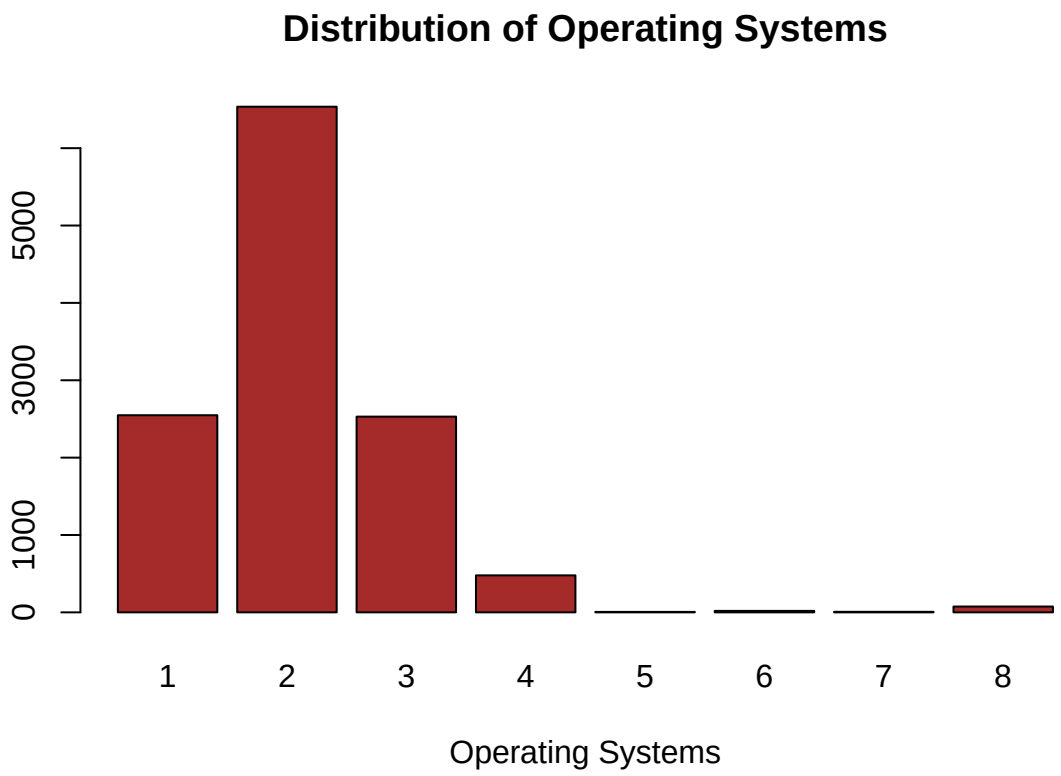
```
adm_OperatingSystems <- markert_df2$OperatingSystems
# median
median(adm_OperatingSystems)
```

```
## [1] 2
```

```
# mode
#sort(adm_info_dur)
names(table(adm_OperatingSystems))[table(adm_OperatingSystems)==max(table(adm_OperatingSystems ))]
```

```
## [1] "2"
```

```
#distribution
counts <- table(adm_OperatingSystems)
barplot(counts, main="Distribution of Operating Systems",col=c("brown"),
 xlab="Operating Systems")
```



#Browser

```
length(unique(markert_df2$Browser))
```

```
## [1] 13
```

```
summary(markert_df2$Browser)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   2.358   2.000  13.000
```

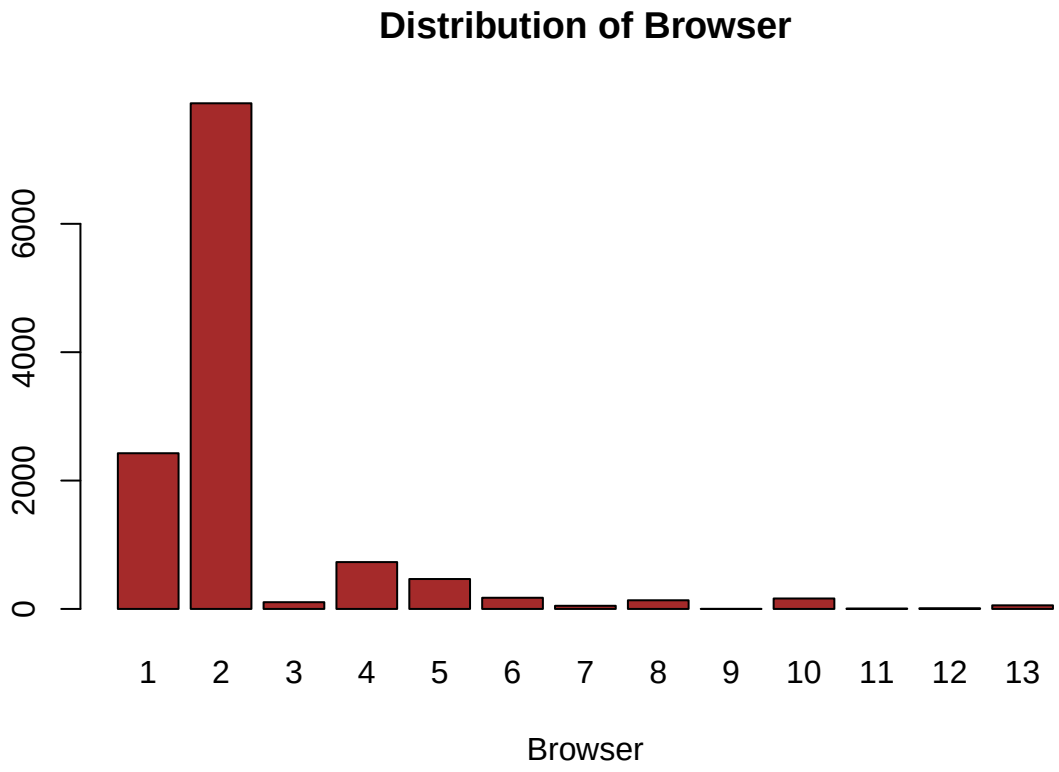
```
adm_Browser <- markert_df2$Browser
# median
median(adm_Browser)
```

```
## [1] 2
```

```
# mode
#sort(adm_info_dur)
names(table(adm_Browser))[table(adm_Browser)==max(table(adm_Browser ))]
```

```
## [1] "2"
```

```
#distribution
counts <- table(adm_Browser)
barplot(counts, main="Distribution of Browser",col=c("brown"),
 xlab="Browser")
```



Region

```
length(unique(markert_df2$Region))
```

```
## [1] 9
```

```
summary(markert_df2$Region)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   3.000   3.153   4.000   9.000
```

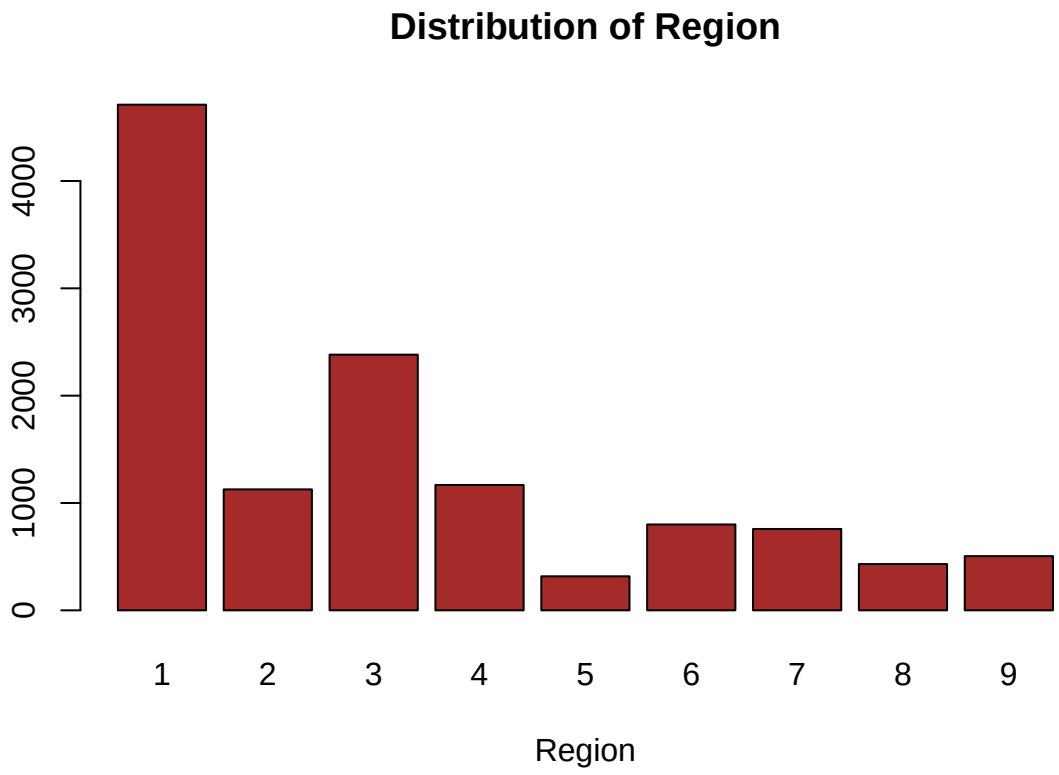
```r
adm_Region <- markert_df2$Region
# median
median(adm_Region)
```

```
## [1] 3
```

```r
# mode
#sort(adm_Region)
names(table(adm_Region))[table(adm_Region)==max(table(adm_Region ))]
```

```
## [1] "1"
```

```r
#distribution
counts <- table(adm_Region)
barplot(counts, main="Distribution of Region",col=c("brown"),
 xlab="Region")
```



TrafficType

```r
length(unique(markert_df2$TrafficType))
```

```
## [1] 20
```

```
summary(markert_df2$TrafficType)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   4.075   4.000  20.000
```

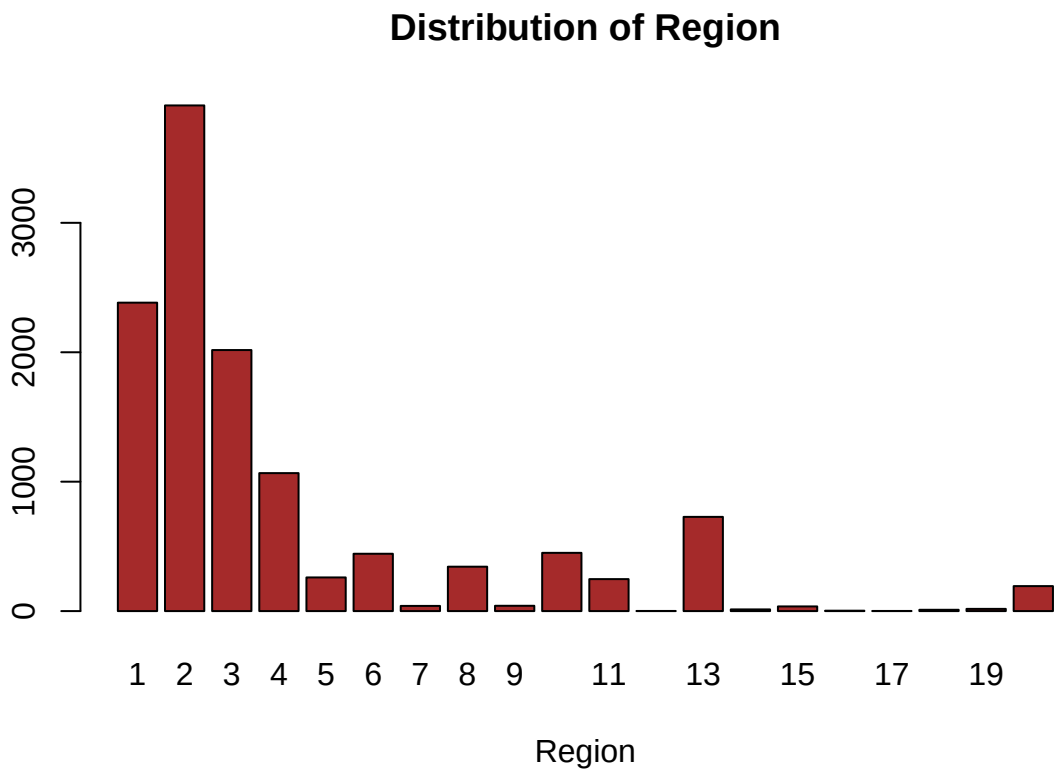
```
adm_TrafficType <- markert_df2$TrafficType
# median
median(adm_TrafficType)
```

```
## [1] 2
```

```
# mode
#sort(adm_info_dur)
names(table(adm_TrafficType))[table(adm_TrafficType)==max(table(adm_TrafficType ))]
```

```
## [1] "2"
```

```
#distribution
counts <- table(adm_TrafficType)
barplot(counts, main="Distribution of Region",col=c("brown"),
 xlab="Region")
```



VisitorType

```
length(unique(markert_df2$VisitorType))
```

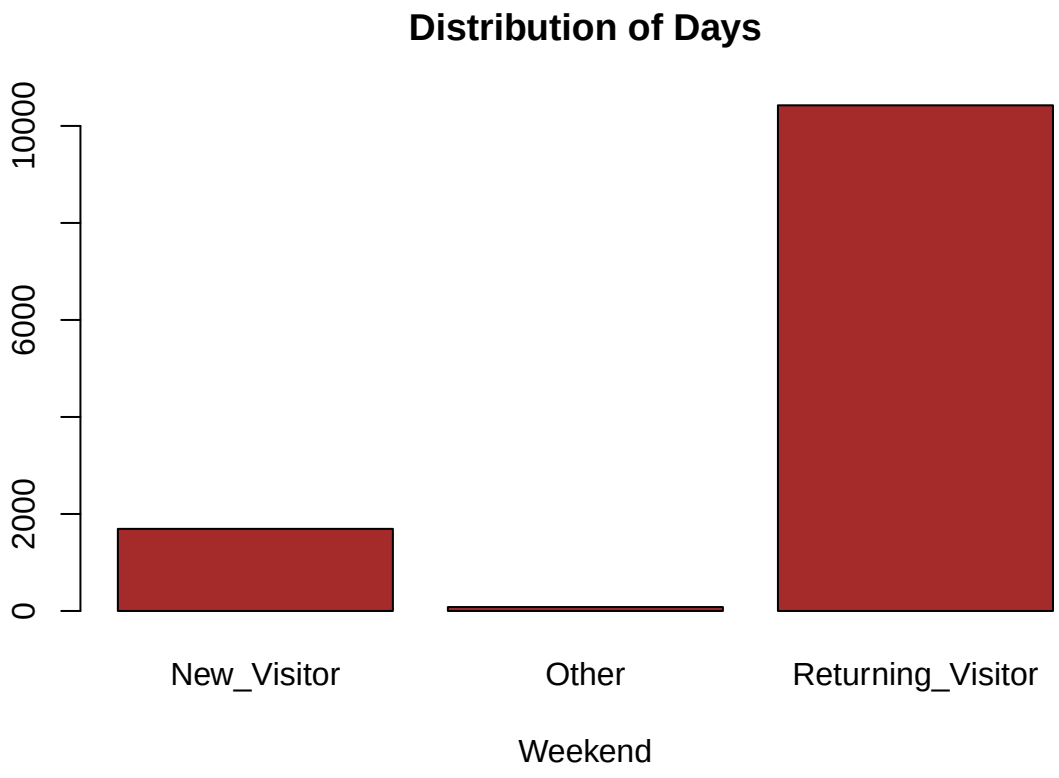
```
## [1] 3
```

```
adm_VisitorType <- markert_df2$VisitorType

#sort(adm_info_dur)
names(table(adm_VisitorType))[table(adm_VisitorType)==max(table(adm_VisitorType ))]
```

```
## [1] "Returning_Visitor"
```

```
#distribution
counts <- table(adm_VisitorType)
barplot(counts, main="Distribution of Days",col=c("brown"),
 xlab="Weekend")
```



#Weekend

```
length(unique(markert_df2$Weekend))
```

```
## [1] 2
```

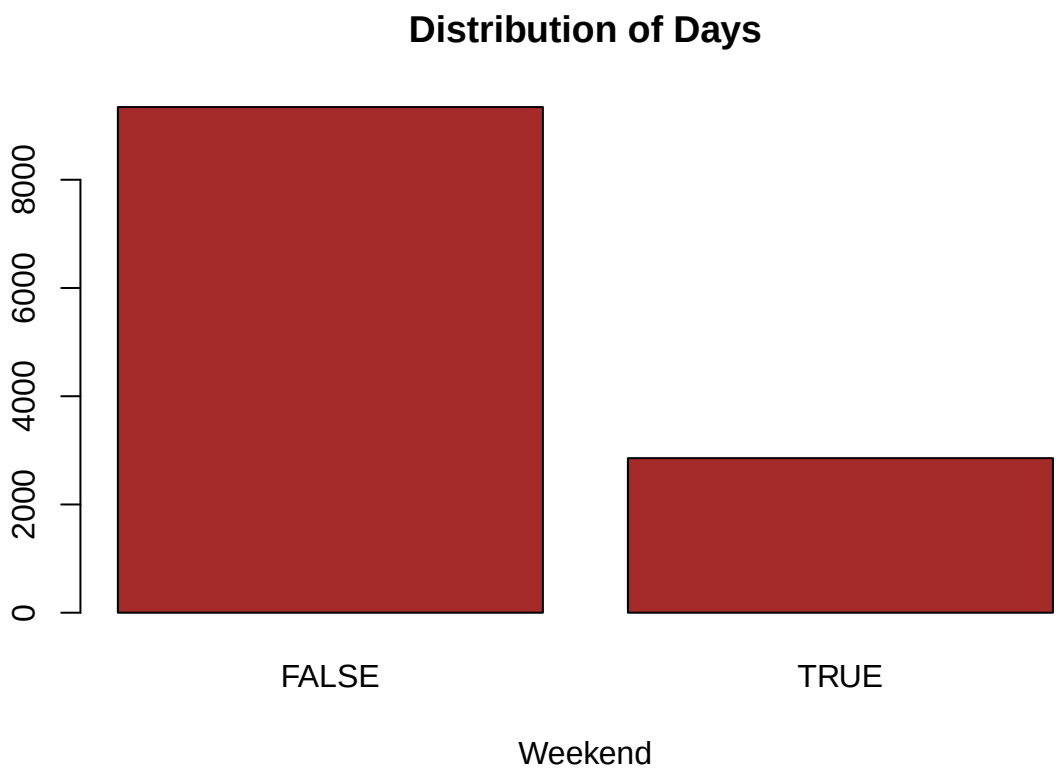
```r
adm_Weekend <- markert_df2$Weekend
# median
median(adm_Weekend)
```

```
## [1] FALSE
```

```r
# mode
#sort(adm_Weekend)
names(table(adm_Weekend))[table(adm_Weekend)==max(table(adm_Weekend ))]
```

```
## [1] "FALSE"
```

```r
#distribution
counts <- table(adm_Weekend)
barplot(counts, main="Distribution of Days",col=c("brown"),
 xlab="Weekend")
```



Revenue

```r
length(unique(markert_df2$Revenue))
```

```
## [1] 2
```

```
summary(markert_df2$Revenue)
```

```
##    Mode   FALSE    TRUE
## logical   10291    1908
```

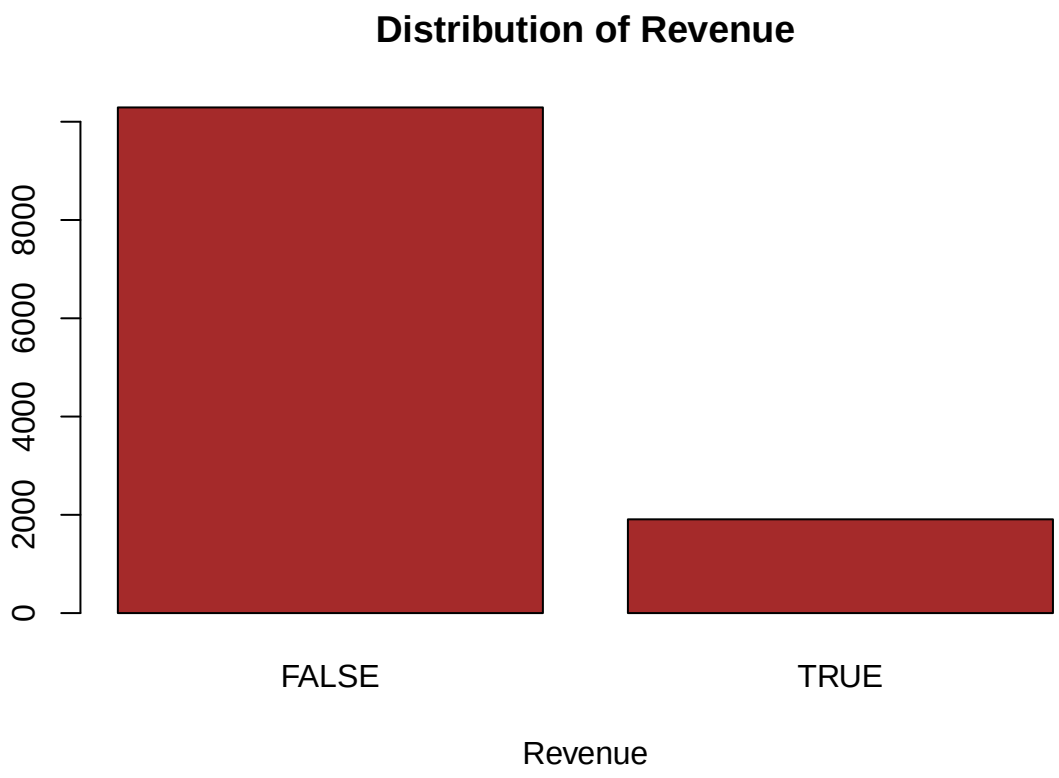
```
adm_Revenue <- markert_df2$Revenue
# median
median(adm_Revenue)
```

```
## [1] FALSE
```

```
# mode
#sort(adm_info_dur)
names(table(adm_Revenue))[table(adm_Revenue)==max(table(adm_Revenue ))]
```

```
## [1] "FALSE"
```

```
#distribution
counts <- table(adm_Revenue)
barplot(counts, main="Distribution of Revenue",col=c("brown"),
 xlab="Revenue")
```



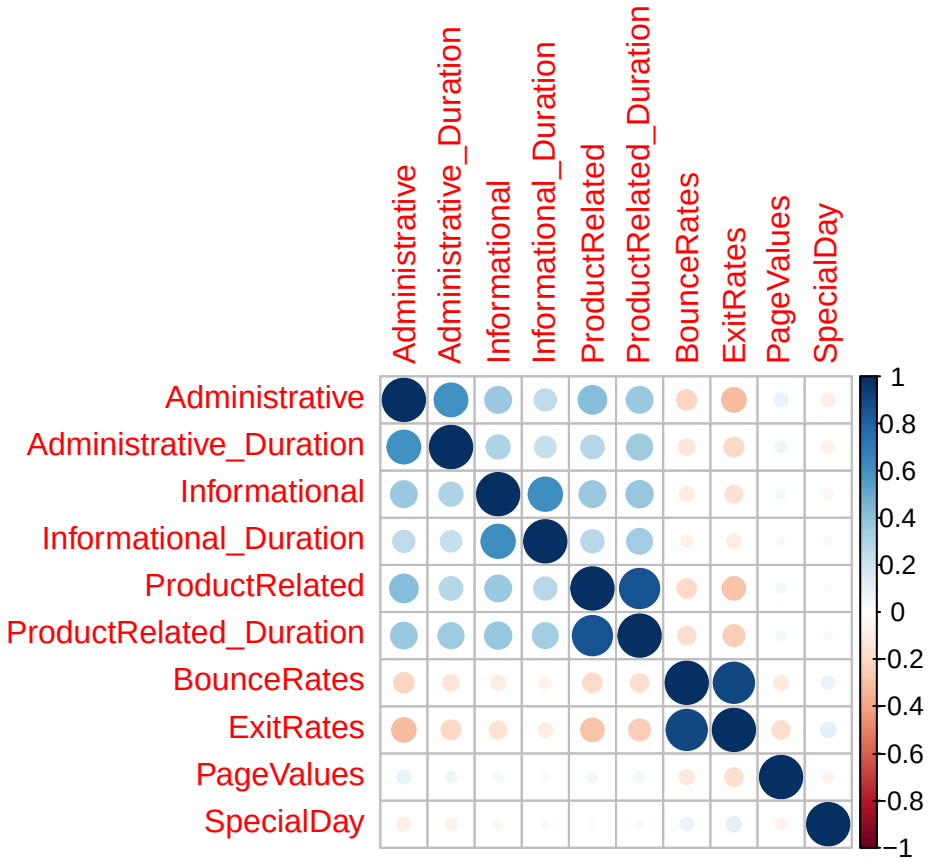
Bivariate Analysis

```
# calculate correlations
correlations <- cor(markert_df2[,1:10])
correlations
```

```
##                           Administrative Administrative_Duration Informational
## Administrative                1.00000000              0.60040965    0.37528761
## Administrative_Duration       0.60040965              1.00000000    0.30143630
## Informational                 0.37528761              0.30143630    1.00000000
## Informational_Duration        0.25478602              0.23718986    0.61867795
## ProductRelated                0.42819151              0.28678391    0.37260472
## ProductRelated_Duration       0.37102722              0.35351379    0.38608372
## BounceRates                  -0.21366664             -0.13733340   -0.10950530
## ExitRates                    -0.31127413             -0.20202445   -0.15956681
## PageValues                    0.09692097              0.06616837    0.04739015
## SpecialDay                   -0.09707210             -0.07473689   -0.04937677
##                           Informational_Duration ProductRelated
## Administrative                        0.25478602     0.42819151
## Administrative_Duration               0.23718986     0.28678391
## Informational                         0.61867795     0.37260472
## Informational_Duration                1.00000000     0.27906195
## ProductRelated                        0.27906195     1.00000000
## ProductRelated_Duration               0.34658069     0.86030819
## BounceRates                          -0.07015947    -0.19351577
## ExitRates                            -0.10293268    -0.28616321
## PageValues                            0.03006416     0.05411549
## SpecialDay                           -0.03129304    -0.02593062
##                           ProductRelated_Duration BounceRates  ExitRates
## Administrative                         0.37102722 -0.21366664 -0.3112741
## Administrative_Duration                0.35351379 -0.13733340 -0.2020245
## Informational                          0.38608372 -0.10950530 -0.1595668
## Informational_Duration                 0.34658069 -0.07015947 -0.1029327
## ProductRelated                         0.86030819 -0.19351577 -0.2861632
## ProductRelated_Duration                1.00000000 -0.17437550 -0.2453340
## BounceRates                           -0.17437550  1.00000000  0.9033582
## ExitRates                             -0.24533401  0.90335819  1.0000000
## PageValues                             0.05084062 -0.11599198 -0.1735715
## SpecialDay                            -0.03821065  0.08783999  0.1167838
##                            PageValues  SpecialDay
## Administrative             0.09692097 -0.09707210
## Administrative_Duration    0.06616837 -0.07473689
## Informational              0.04739015 -0.04937677
## Informational_Duration     0.03006416 -0.03129304
## ProductRelated             0.05411549 -0.02593062
## ProductRelated_Duration    0.05084062 -0.03821065
## BounceRates               -0.11599198  0.08783999
## ExitRates                 -0.17357154  0.11678376
## PageValues                 1.00000000 -0.06453271
## SpecialDay                -0.06453271  1.00000000
```

Correlation Plot

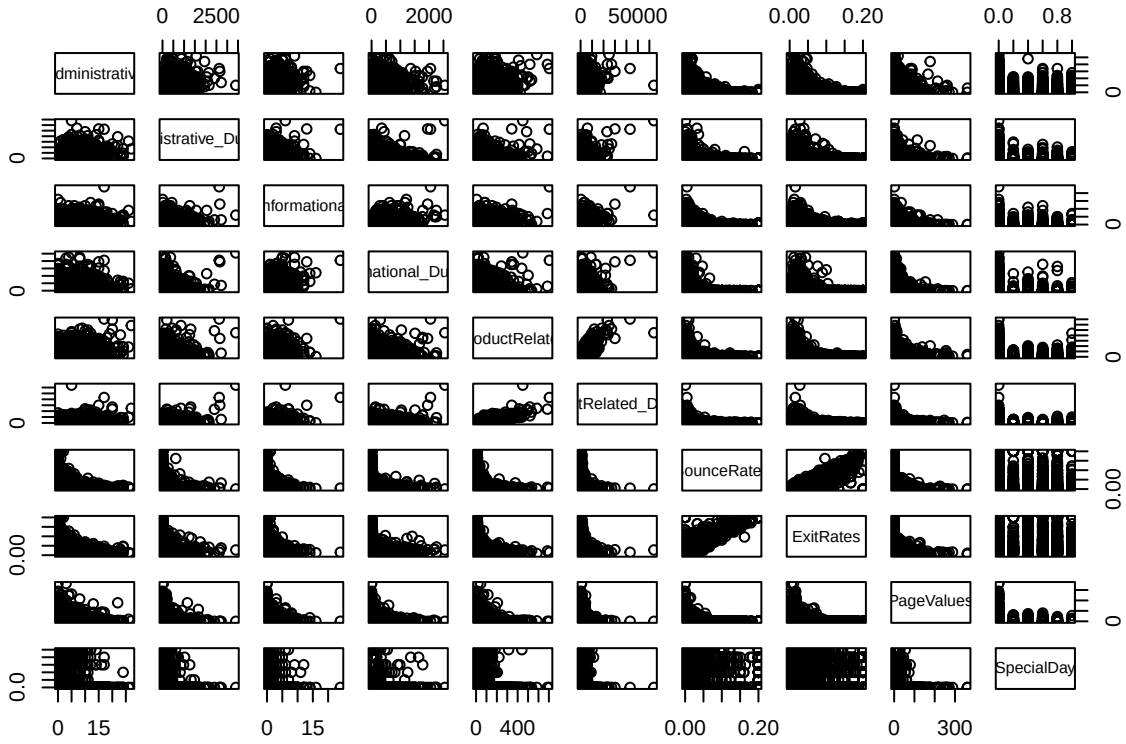
```
# create correlation plot
library(corrplot)
## corrplot 0.84 loaded
corrplot(correlations, method="circle")
```



- From the plot above, we can see that most of the variables have low Positive and Negative correlation

Pair Plots

```
pairs(markert_df2[,1:10])
```

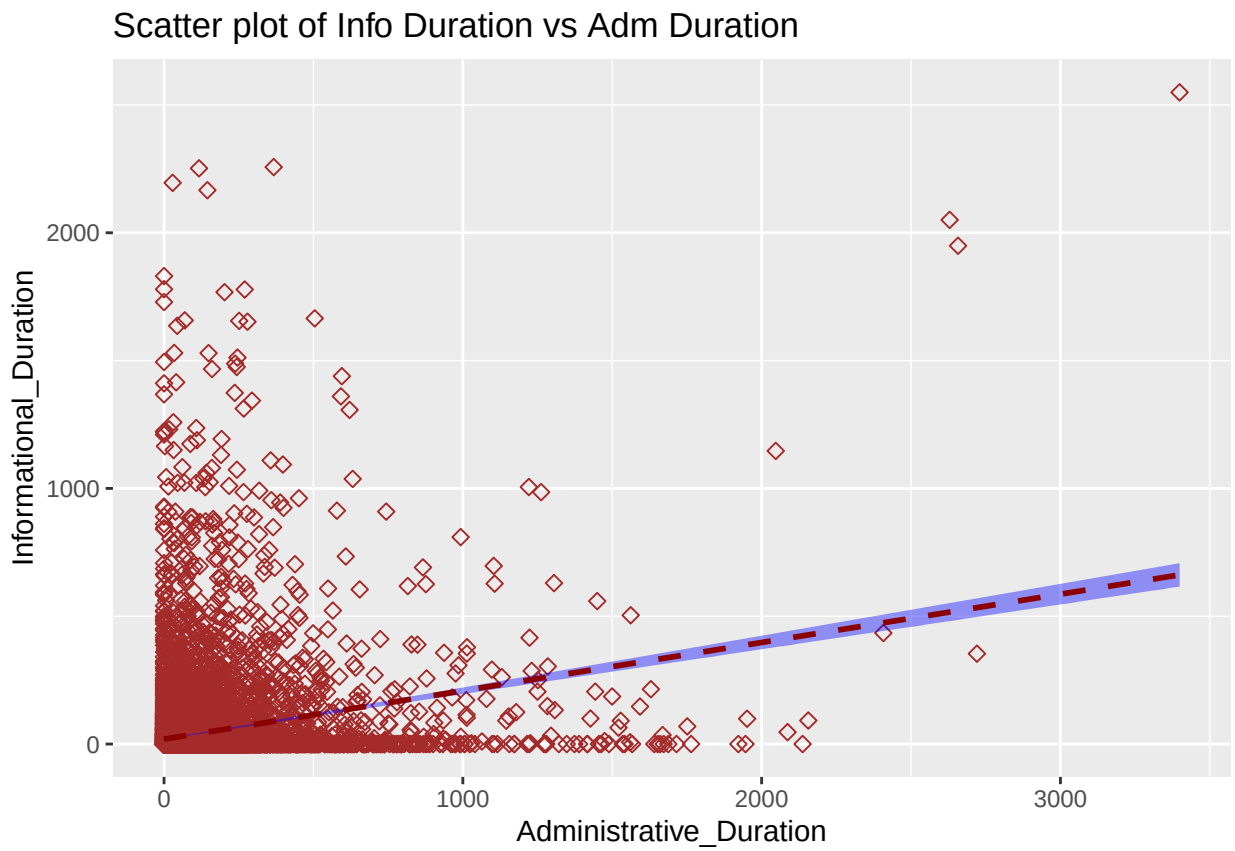


Sites Visited Duration

Scatter plot of Administrative_Duration vs Informational_Duration

```
library(ggplot2)
ggplot(markert_df2, aes(x = Administrative_Duration, y =
Informational_Duration)) +
 geom_point(size = 2, color= "brown", shape = 23)+
 geom_smooth(method=lm, linetype="dashed",color="darkred",
fill="blue")+
 labs(title = "Scatter plot of Info Duration vs Adm Duration")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



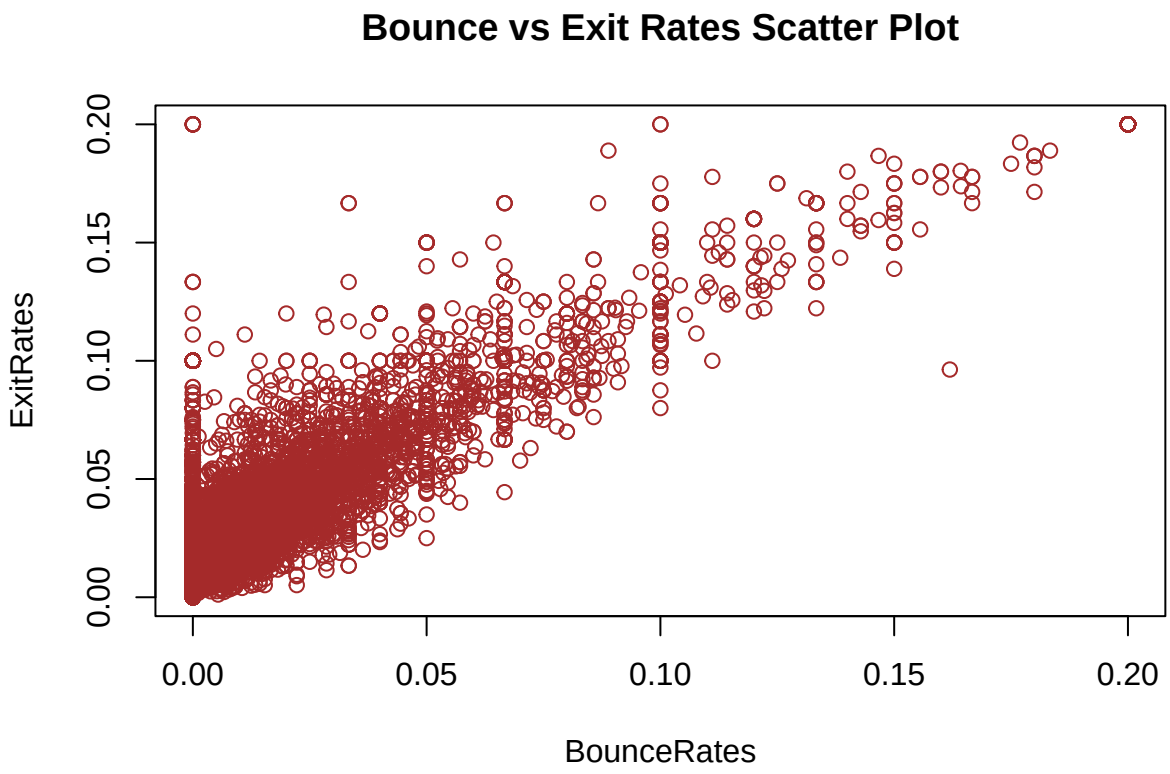

```
# `geom_smooth()` using formula 'y ~ x'
```

- There is a positive non-linear correlation between the time spent on the Administrative site and the Informational site

Metrics

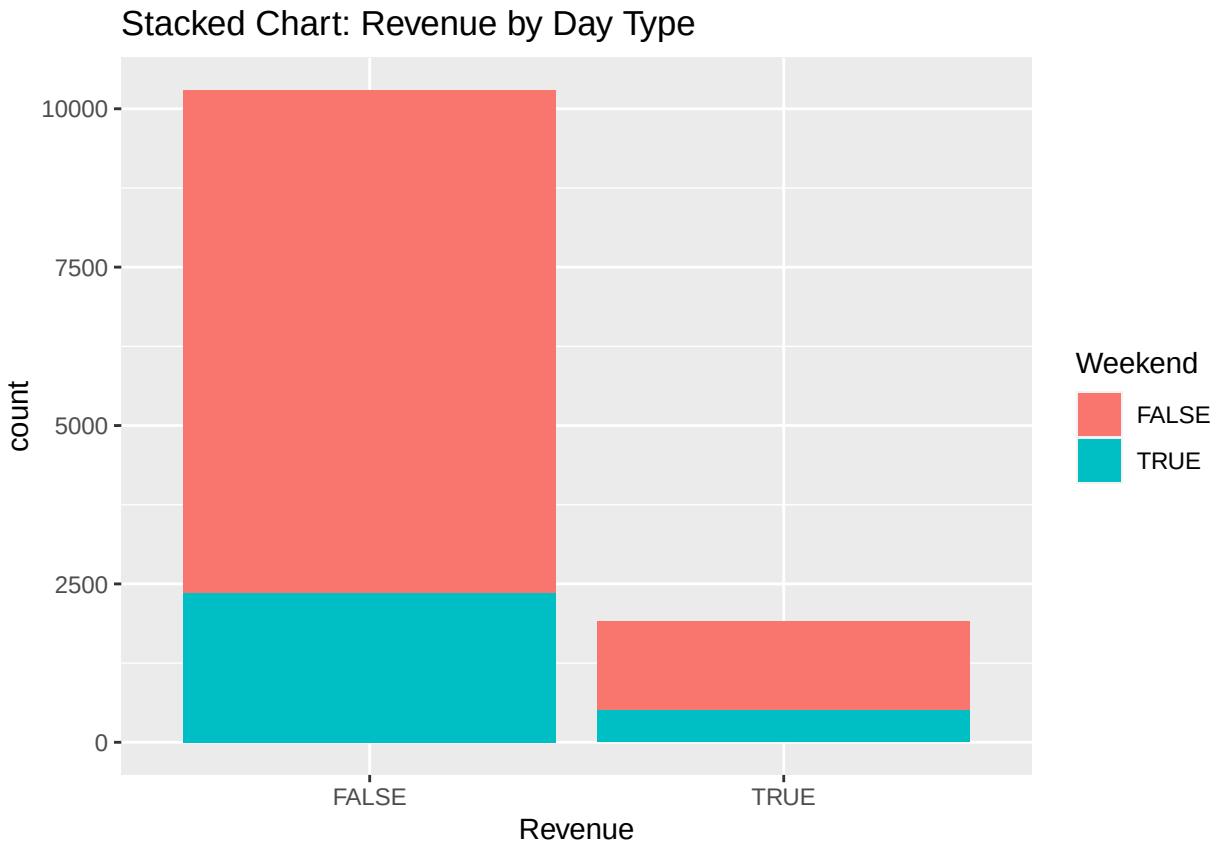
Scatter plot Bounce vs Exit Rates Scatter Plot

```
plot(ExitRates ~ BounceRates, dat = markert_df2,
 col = "brown",
 main = "Bounce vs Exit Rates Scatter Plot")
```



Stacked bar chart:Revenue vs Day Type

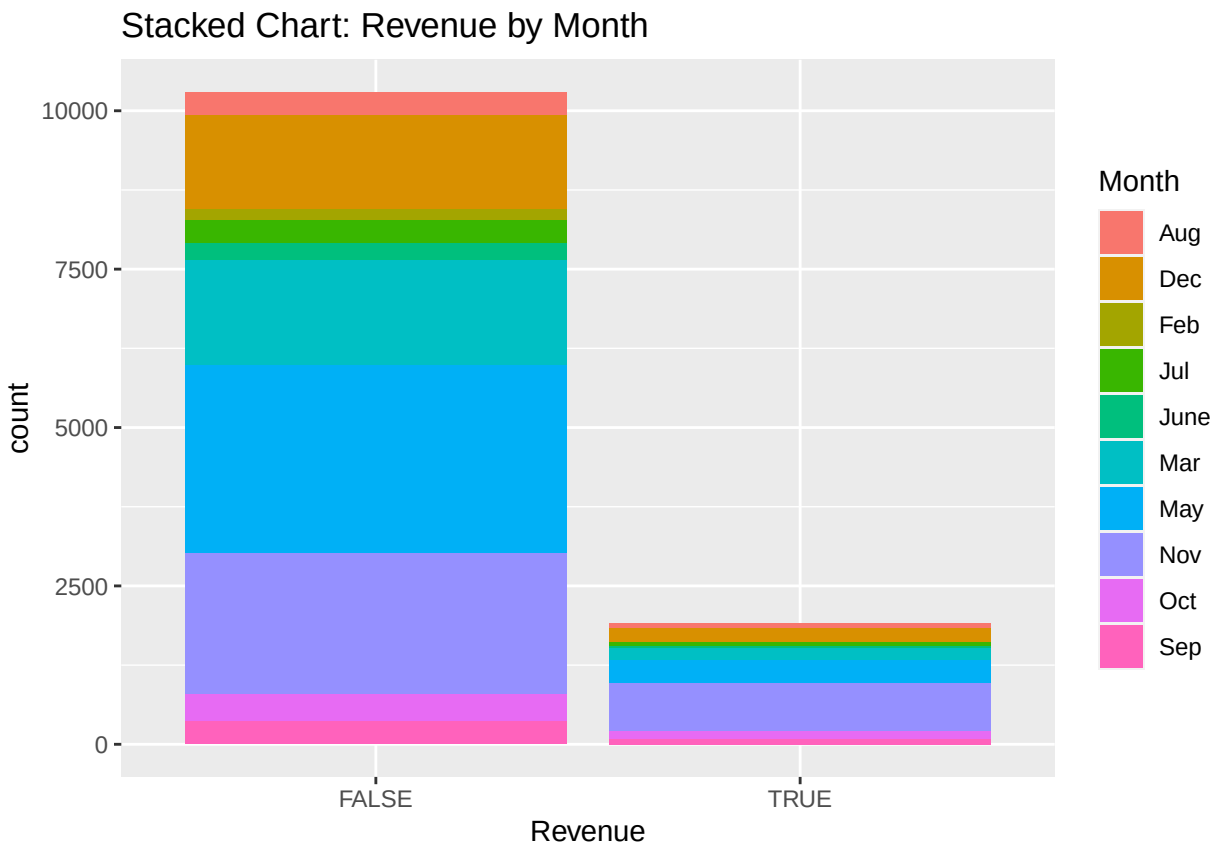
```
library(magrittr)
markert_df2 %>%
 ggplot(aes(Revenue)) +
 geom_bar(aes(fill = Weekend))+
 labs(title = "Stacked Chart: Revenue by Day Type")
```



- From the stacked chart, we can see that most of the revenue is generated during the week and not over the weekend.

Revenue vs Month

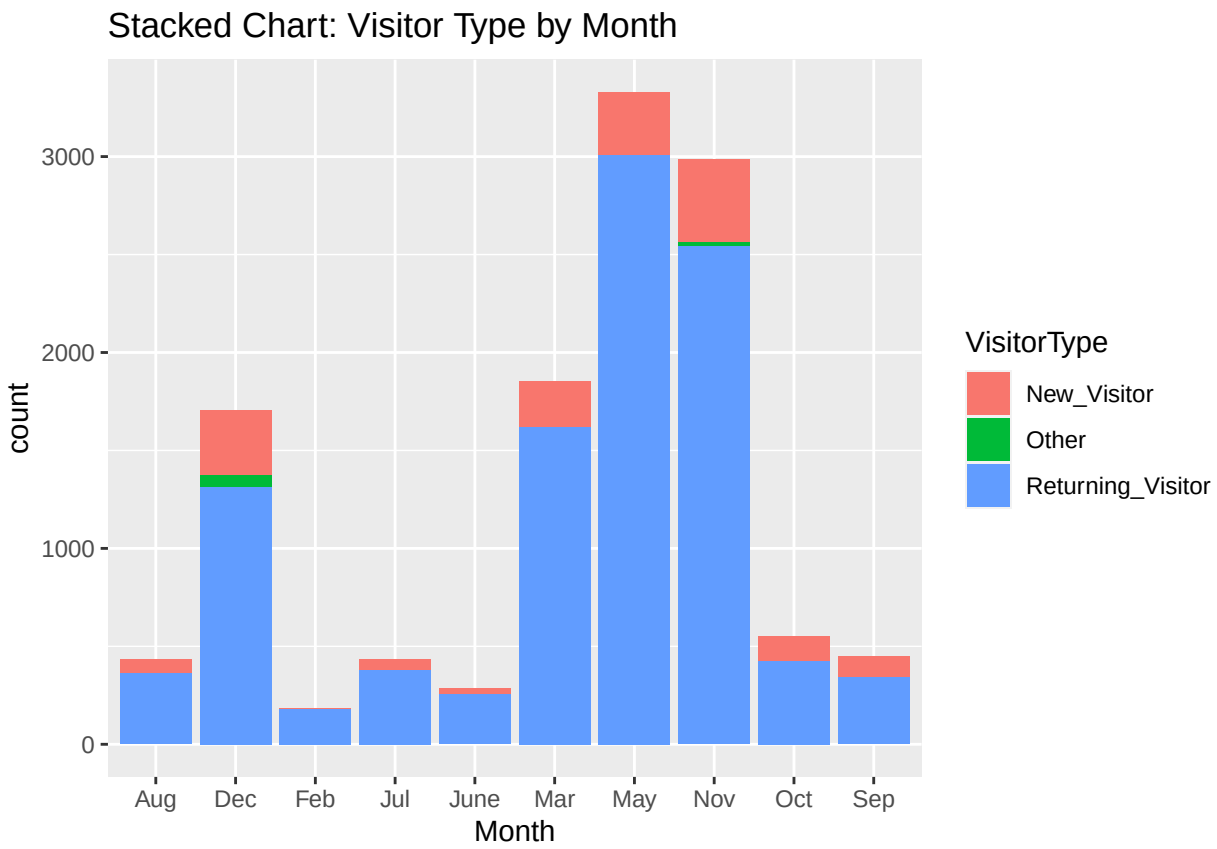
```
# Stacked bar chart: Revenue vs Month
markert_df2 %>%
 ggplot(aes(Revenue)) +
 geom_bar(aes(fill = Month))+
 labs(title = "Stacked Chart: Revenue by Month")
```



Type of visitor

###Stacked bar chart: Visitor Type vs Month

```
markert_df2 %>%
 ggplot(aes(Month)) +
 geom_bar(aes(fill = VisitorType))+
 labs(title = "Stacked Chart: Visitor Type by Month")
```



Multivariate Analysis

```
# A glimpse of the data
library(dplyr)
glimpse(markert_df2)
```

```
## Rows: 12,199
## Columns: 18
## $ Administrative          <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2~
## $ Administrative_Duration <dbl> 0, 0, -1, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0, 0, 0~
## $ Informational           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ Informational_Duration  <dbl> 0, 0, -1, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0, 0, 0~
## $ ProductRelated          <int> 1, 2, 1, 2, 10, 19, 1, 1, 2, 3, 3, 16, 7, 6, 2~
## $ ProductRelated_Duration <dbl> 0.000000, 64.000000, -1.000000, 2.666667, 627.~
## $ BounceRates             <dbl> 0.200000000, 0.000000000, 0.200000000, 0.05000~
## $ ExitRates               <dbl> 0.200000000, 0.100000000, 0.200000000, 0.14000~
## $ PageValues              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ SpecialDay              <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.4, 0.0, 0.8, 0~
## $ Month                   <chr> "Feb", "Feb", "Feb", "Feb", "Feb", "Feb", "Feb~
## $ OperatingSystems        <int> 1, 2, 4, 3, 3, 2, 2, 1, 2, 2, 1, 1, 1, 2, 3, 1~
## $ Browser                 <int> 1, 2, 1, 2, 3, 2, 4, 2, 2, 4, 1, 1, 1, 5, 2, 1~
## $ Region                  <int> 1, 1, 9, 2, 1, 1, 3, 1, 2, 1, 3, 4, 1, 1, 3, 9~
## $ TrafficType             <int> 1, 2, 3, 4, 4, 3, 3, 5, 3, 2, 3, 3, 3, 3, 3, 3~
```

```
## $ VisitorType             <chr> "Returning_Visitor", "Returning_Visitor", "Ret~
## $ Weekend                 <lgl> FALSE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE~
## $ Revenue                 <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALS~
```

Dummify the data

```
# One hot encoding of the factor variables.
library(caret)
dmy <- dummyVars(" ~ .", data = markert_df2)
dummy_df <- data.frame(predict(dmy, newdata = markert_df2))
#print(dummy_df)
glimpse(dummy_df)
```

```
## Rows: 12,199
## Columns: 31
## $ Administrative               <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,~
## $ Administrative_Duration      <dbl> 0, 0, -1, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0,~
## $ Informational                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ Informational_Duration       <dbl> 0, 0, -1, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0,~
## $ ProductRelated               <dbl> 1, 2, 1, 2, 10, 19, 1, 1, 2, 3, 3, 16, 7,~
## $ ProductRelated_Duration      <dbl> 0.000000, 64.000000, -1.000000, 2.666667,~
## $ BounceRates                  <dbl> 0.200000000, 0.000000000, 0.200000000, 0.~
## $ ExitRates                    <dbl> 0.200000000, 0.100000000, 0.200000000, 0.~
## $ PageValues                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ SpecialDay                   <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.4, 0.0, 0~
## $ MonthAug                     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ MonthDec                     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ MonthFeb                     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ MonthJul                     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ MonthJune                    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ MonthMar                     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ MonthMay                     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ MonthNov                     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ MonthOct                     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ MonthSep                     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ OperatingSystems             <dbl> 1, 2, 4, 3, 3, 2, 2, 1, 2, 2, 1, 1, 1, 2,~
## $ Browser                      <dbl> 1, 2, 1, 2, 3, 2, 4, 2, 2, 4, 1, 1, 1, 5,~
## $ Region                       <dbl> 1, 1, 9, 2, 1, 1, 3, 1, 2, 1, 3, 4, 1, 1,~
## $ TrafficType                  <dbl> 1, 2, 3, 4, 4, 3, 3, 5, 3, 2, 3, 3, 3, 3,~
## $ VisitorTypeNew_Visitor       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ VisitorTypeOther             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ VisitorTypeReturning_Visitor <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ WeekendFALSE                 <dbl> 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1,~
## $ WeekendTRUE                  <dbl> 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,~
## $ RevenueFALSE                 <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ RevenueTRUE                  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
```

Checking the resultant datatypes

```
sapply(dummy_df, class)
```

```
##                Administrative      Administrative_Duration
##                     "numeric"                   "numeric"
##                Informational       Informational_Duration
##                     "numeric"                   "numeric"
##               ProductRelated      ProductRelated_Duration
##                     "numeric"                   "numeric"
##                  BounceRates                   ExitRates
##                     "numeric"                   "numeric"
##                   PageValues                  SpecialDay
##                     "numeric"                   "numeric"
##                     MonthAug                    MonthDec
##                     "numeric"                   "numeric"
##                     MonthFeb                    MonthJul
##                     "numeric"                   "numeric"
##                    MonthJune                    MonthMar
##                     "numeric"                   "numeric"
##                     MonthMay                    MonthNov
##                     "numeric"                   "numeric"
##                     MonthOct                    MonthSep
##                     "numeric"                   "numeric"
##             OperatingSystems                     Browser
##                     "numeric"                   "numeric"
##                       Region                 TrafficType
##                     "numeric"                   "numeric"
##       VisitorTypeNew_Visitor             VisitorTypeOther
##                     "numeric"                   "numeric"
## VisitorTypeReturning_Visitor                 WeekendFALSE
##                     "numeric"                   "numeric"
##                   WeekendTRUE                 RevenueFALSE
##                     "numeric"                   "numeric"
##                   RevenueTRUE
##                     "numeric"
```

Seperating the dependent and independent variables

```
#removing the revenue column from the data
#we select all the column indexes before 30
dummy_df2 <- dummy_df[, -c(30:31)]
dim(dummy_df2)
```

```
## [1] 12199    29
```

```
dummy_df.class<- markert_df2[, "Revenue"]
```

SCALING VS NORMALIZATION

Scaling

- In this step the data is transformed to fit withing the range between 0 and 1

```
dummy_df2_scaled <- scale(dummy_df2)
summary(dummy_df2_scaled)
```

```
##  Administrative    Administrative_Duration Informational
##  Min.   :-0.7025   Min.   :-0.46574        Min.   :-0.3988
##  1st Qu.:-0.7025   1st Qu.:-0.46011        1st Qu.:-0.3988
##  Median :-0.4023   Median :-0.40941        Median :-0.3988
##  Mean   : 0.0000   Mean   : 0.00000        Mean   : 0.0000
##  3rd Qu.: 0.4984   3rd Qu.: 0.07361        3rd Qu.:-0.3988
##  Max.   : 7.4035   Max.   :18.68474        Max.   :18.4127
##  Informational_Duration ProductRelated    ProductRelated_Duration
##  Min.   :-0.2533        Min.   :-0.7188   Min.   :-0.6295
##  1st Qu.:-0.2463        1st Qu.:-0.5394   1st Qu.:-0.5281
##  Median :-0.2463        Median :-0.3152   Median :-0.3115
##  Mean   : 0.0000        Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:-0.2463        3rd Qu.: 0.1332   3rd Qu.: 0.1407
##  Max.   :17.7758        Max.   :15.0881   Max.   :32.6919
##   BounceRates          ExitRates         PageValues       SpecialDay
##  Min.   :-0.45034   Min.   :-0.8973   Min.   :-0.319   Min.   :-0.3103
##  1st Qu.:-0.45034   1st Qu.:-0.5897   1st Qu.:-0.319   1st Qu.:-0.3103
##  Median :-0.38580   Median :-0.3567   Median :-0.319   Median :-0.3103
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.000   Mean   : 0.0000
##  3rd Qu.:-0.08326   3rd Qu.: 0.1511   3rd Qu.:-0.319   3rd Qu.:-0.3103
##  Max.   : 3.95470   Max.   : 3.4273   Max.   :19.070   Max.   : 4.6969
##     MonthAug          MonthDec          MonthFeb          MonthJul
##  Min.   :-0.1918   Min.   :-0.4032   Min.   :-0.1231   Min.   :-0.1916
##  1st Qu.:-0.1918   1st Qu.:-0.4032   1st Qu.:-0.1231   1st Qu.:-0.1916
##  Median :-0.1918   Median :-0.4032   Median :-0.1231   Median :-0.1916
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:-0.1918   3rd Qu.:-0.4032   3rd Qu.:-0.1231   3rd Qu.:-0.1916
##  Max.   : 5.2126   Max.   : 2.4799   Max.   : 8.1254   Max.   : 5.2188
##    MonthJune         MonthMar          MonthMay          MonthNov
##  Min.   :-0.1547   Min.   :-0.4232   Min.   :-0.6125   Min.   :-0.5689
##  1st Qu.:-0.1547   1st Qu.:-0.4232   1st Qu.:-0.6125   1st Qu.:-0.5689
##  Median :-0.1547   Median :-0.4232   Median :-0.6125   Median :-0.5689
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:-0.1547   3rd Qu.:-0.4232   3rd Qu.: 1.6326   3rd Qu.:-0.5689
##  Max.   : 6.4653   Max.   : 2.3628   Max.   : 1.6326   Max.   : 1.7576
##     MonthOct          MonthSep       OperatingSystems     Browser
##  Min.   :-0.2171   Min.   :-0.1952   Min.   :-1.2397   Min.   :-0.7940
##  1st Qu.:-0.2171   1st Qu.:-0.1952   1st Qu.:-0.1371   1st Qu.:-0.2094
##  Median :-0.2171   Median :-0.1952   Median :-0.1371   Median :-0.2094
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:-0.2171   3rd Qu.:-0.1952   3rd Qu.: 0.9654   3rd Qu.:-0.2094
##  Max.   : 4.6064   Max.   : 5.1213   Max.   : 6.4782   Max.   : 6.2212
##      Region          TrafficType       VisitorTypeNew_Visitor
##  Min.   :-0.89629   Min.   :-0.76562   Min.   :-0.4014
```

```
##  1st Qu.:-0.89629   1st Qu.:-0.51661   1st Qu.:-0.4014
##  Median :-0.06381   Median :-0.51661   Median :-0.4014
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.35244   3rd Qu.:-0.01858   3rd Qu.:-0.4014
##  Max.   : 2.43366   Max.   : 3.96567   Max.   : 2.4910
##  VisitorTypeOther   VisitorTypeReturning_Visitor  WeekendFALSE
##  Min.   :-0.08175   Min.   :-2.4241              Min.   :-1.8086
##  1st Qu.:-0.08175   1st Qu.: 0.4125              1st Qu.: 0.5529
##  Median :-0.08175   Median : 0.4125              Median : 0.5529
##  Mean   : 0.00000   Mean   : 0.0000              Mean   : 0.0000
##  3rd Qu.:-0.08175   3rd Qu.: 0.4125              3rd Qu.: 0.5529
##  Max.   :12.23081   Max.   : 0.4125              Max.   : 0.5529
##   WeekendTRUE
##  Min.   :-0.5529
##  1st Qu.:-0.5529
##  Median :-0.5529
##  Mean   : 0.0000
##  3rd Qu.:-0.5529
##  Max.   : 1.8086
```

Normalizing

- Normalization is a technique often applied to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values.

```
dummy_df2_norm <- as.data.frame(apply(dummy_df2, 2, function(x) (x -
min(x))/(max(x)-min(x))))
summary(dummy_df2_norm)
```

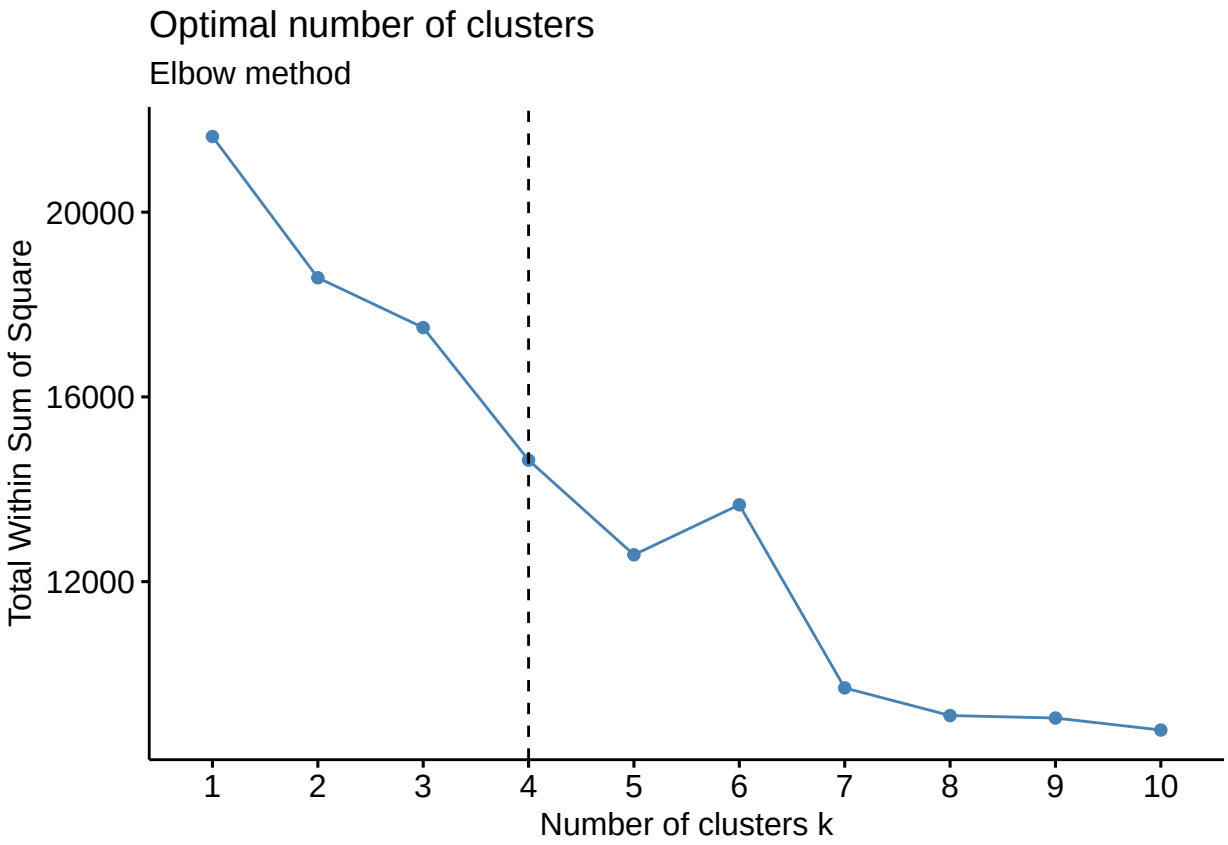
```
##  Administrative    Administrative_Duration Informational
##  Min.   :0.00000   Min.   :0.0000000       Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0002941       1st Qu.:0.0000
##  Median :0.03704   Median :0.0029414       Median :0.0000
##  Mean   :0.08667   Mean   :0.0243201       Mean   :0.0212
##  3rd Qu.:0.14815   3rd Qu.:0.0281638       3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :1.0000000       Max.   :1.0000
##  Informational_Duration ProductRelated    ProductRelated_Duration
##  Min.   :0.0000000      Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.0003921      1st Qu.:0.01135   1st Qu.:0.003042
##  Median :0.0003921      Median :0.02553   Median :0.009543
##  Mean   :0.0140518      Mean   :0.04547   Mean   :0.018891
##  3rd Qu.:0.0003921      3rd Qu.:0.05390   3rd Qu.:0.023112
##  Max.   :1.0000000      Max.   :1.00000   Max.   :1.000000
##   BounceRates        ExitRates         PageValues        SpecialDay
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.07111   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.01465   Median :0.12500   Median :0.00000   Median :0.00000
##  Mean   :0.10223   Mean   :0.20748   Mean   :0.01645   Mean   :0.06197
##  3rd Qu.:0.08333   3rd Qu.:0.24242   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##     MonthAug          MonthDec         MonthFeb          MonthJul
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000
```

```
##  Median :0.00000   Median :0.0000   Median :0.00000   Median :0.00000
##  Mean   :0.03549   Mean   :0.1398   Mean   :0.01492   Mean   :0.03541
##  3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.00000   Max.   :1.00000
##    MonthJune          MonthMar         MonthMay         MonthNov
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.00000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.02336   Mean   :0.1519   Mean   :0.2728   Mean   :0.2445
##  3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##     MonthOct         MonthSep       OperatingSystems    Browser
##  Min.   :0.000   Min.   :0.00000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.000   1st Qu.:0.00000   1st Qu.:0.1429   1st Qu.:0.08333
##  Median :0.000   Median :0.00000   Median :0.1429   Median :0.08333
##  Mean   :0.045   Mean   :0.03672   Mean   :0.1606   Mean   :0.11318
##  3rd Qu.:0.000   3rd Qu.:0.00000   3rd Qu.:0.2857   3rd Qu.:0.08333
##  Max.   :1.000   Max.   :1.00000   Max.   :1.0000   Max.   :1.00000
##      Region        TrafficType      VisitorTypeNew_Visitor VisitorTypeOther
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000         Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.05263   1st Qu.:0.0000         1st Qu.:0.00000
##  Median :0.2500   Median :0.05263   Median :0.0000         Median :0.00000
##  Mean   :0.2692   Mean   :0.16182   Mean   :0.1388         Mean   :0.00664
##  3rd Qu.:0.3750   3rd Qu.:0.15789   3rd Qu.:0.0000         3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.0000         Max.   :1.00000
##  VisitorTypeReturning_Visitor  WeekendFALSE     WeekendTRUE
##  Min.   :0.0000                Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.0000                1st Qu.:1.0000   1st Qu.:0.0000
##  Median :1.0000                Median :1.0000   Median :0.0000
##  Mean   :0.8546                Mean   :0.7659   Mean   :0.2341
##  3rd Qu.:1.0000                3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :1.0000                Max.   :1.0000   Max.   :1.0000
```

Finding the Optimal number of clusters

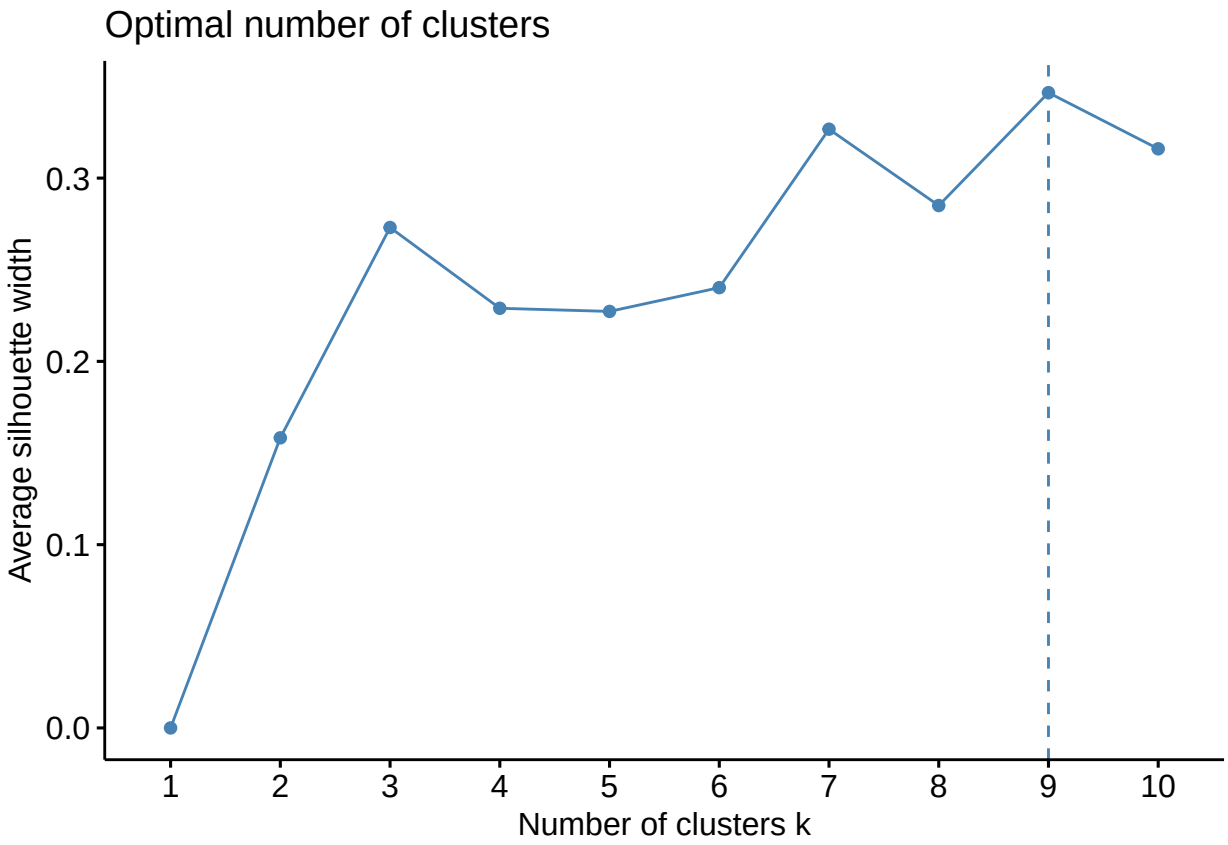
Method 1: Elbow method

```
# Searching for the optimal number of clusters
# # Elbow method
# Searching for the optimal number of clusters
# # Elbow method
library(factoextra)
library(ggplot2)
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
fviz_nbclust(dummy_df2_norm, kmeans, method = "wss") +
 geom_vline(xintercept = 4, linetype = 2)+
 labs(subtitle = "Elbow method")
```



Method 2: Silhouette

```
library(cluster)
fviz_nbclust(dummy_df2_norm, kmeans, method = "silhouette")
```



Implementing the Solution

K-MEANS CLUSTERING

```
outputk <- kmeans(dummy_df2_norm, 4)
```

```
# Previewing the number of records in each cluster
outputk$size
```

```
## [1] 2447  585 6311 2856
```

The cluster center datapoints Per attribute

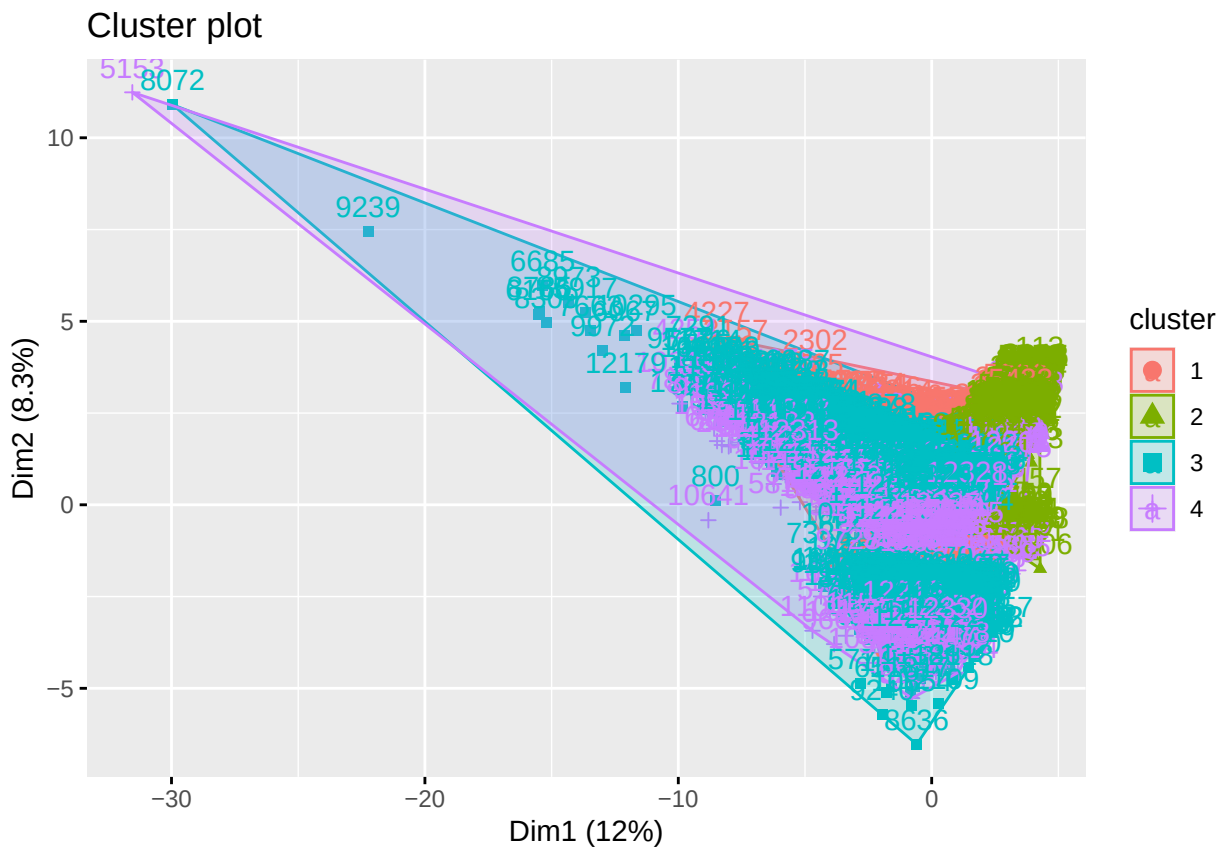
```
outputk$centers
```

```
##   Administrative Administrative_Duration Informational Informational_Duration
## 1    0.074921673             0.0214748616  0.0178960632             0.011090342
## 2    0.001139601             0.0007227781  0.0005698006             0.000396925
## 3    0.096721186             0.0270371335  0.0229031321             0.015410136
```

```
## 4    0.092034962            0.0255872972  0.0244952148           0.016384511
##   ProductRelated ProductRelated_Duration BounceRates ExitRates PageValues
## 1    0.039009121            0.0160651203  0.07501658 0.1950075 0.01525303
## 2    0.003290295            0.0006655947  0.91710417 0.9511681 0.00000000
## 3    0.051178680            0.0215498449  0.04401766 0.1536894 0.01798473
## 4    0.047043427            0.0191676942  0.08728349 0.1847150 0.01747125
##    SpecialDay   MonthAug  MonthDec    MonthFeb   MonthJul  MonthJune  MonthMar
## 1 0.215856150 0.00000000 0.0000000 0.000000000 0.00000000 0.00000000 0.0000000
## 2 0.071111111 0.02735043 0.1145299 0.051282051 0.04102564 0.05470085 0.1623932
## 3 0.004341626 0.05086357 0.2025036 0.019648233 0.04816986 0.03264142 0.2032958
## 4 0.055602241 0.03361345 0.1264006 0.009803922 0.03641457 0.01645658 0.1663165
##    MonthMay  MonthNov   MonthOct   MonthSep OperatingSystems   Browser
## 1 1.0000000 0.0000000 0.00000000 0.00000000        0.1623562 0.1179335
## 2 0.2854701 0.2410256 0.01025641 0.01196581        0.1748474 0.1146724
## 3 0.0000000 0.3249881 0.06322294 0.05466646        0.1584309 0.1158163
## 4 0.2500000 0.2769608 0.05042017 0.03361345        0.1610644 0.1029704
##      Region TrafficType VisitorTypeNew_Visitor VisitorTypeOther
## 1 0.2632816   0.1816618             0.09317532       0.00000000
## 2 0.2717949   0.2168241             0.03760684       0.01709402
## 3 0.2716091   0.1494550             0.15274917       0.01014102
## 4 0.2682511   0.1608801             0.16771709       0.00245098
##   VisitorTypeReturning_Visitor WeekendFALSE WeekendTRUE
## 1                    0.9068247            1           0
## 2                    0.9452991            1           0
## 3                    0.8371098            1           0
## 4                    0.8298319            0           1
```

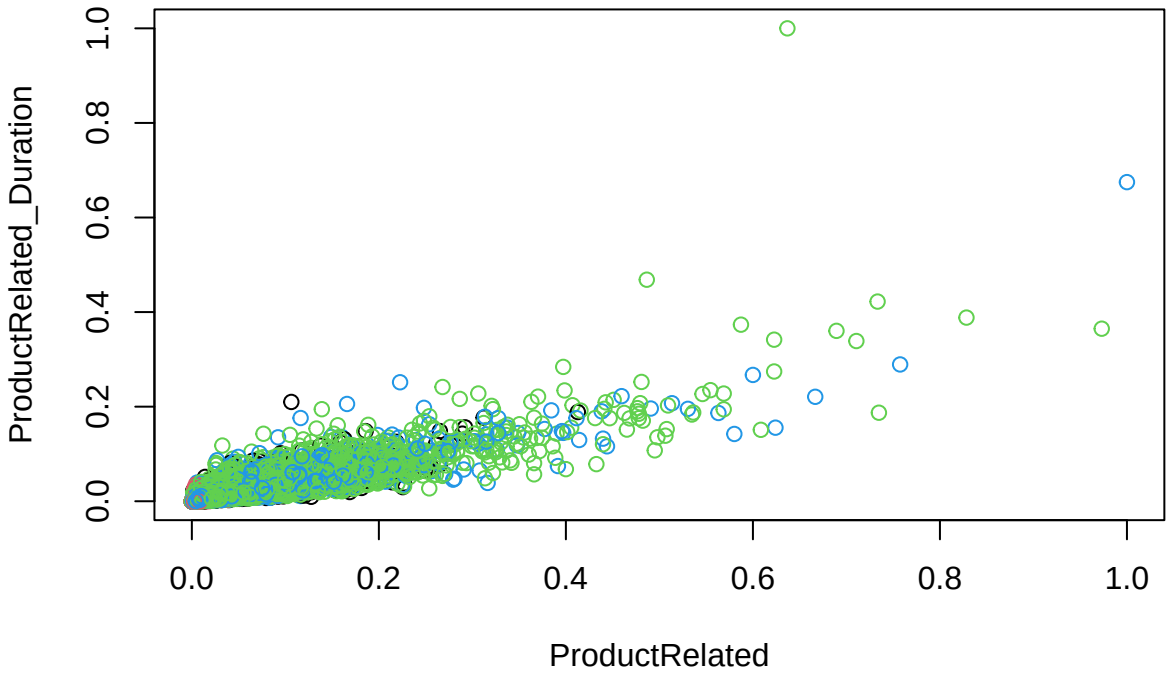
Visualizing the clusters of the whole dataset

```
options(repr.plot.width = 11, repr.plot.height = 6)
fviz_cluster(outputk, dummy_df2_norm)
```



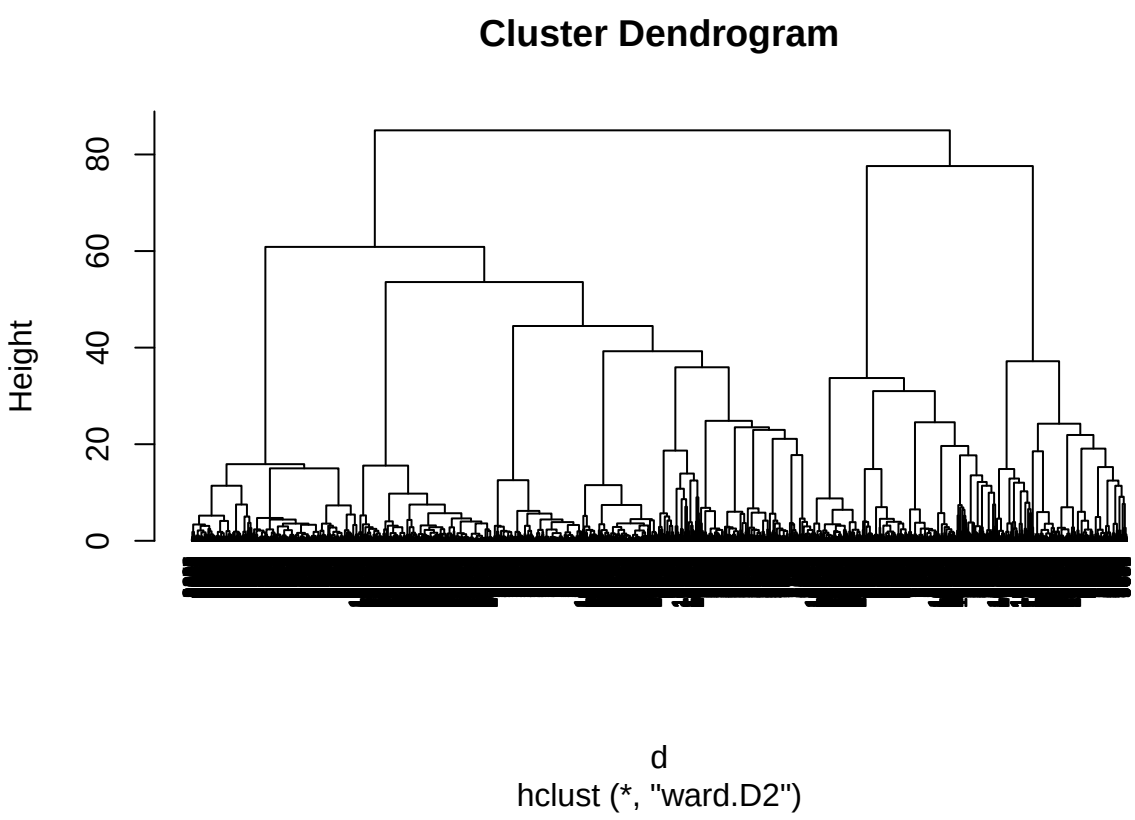
Visualizing variable datatypes on a scatter plot

```
# Plotting two variables to see how their data points
# have been distributed in the cluster
# Product Related, vs Product Related Duration
plot(dummy_df2_norm[, 5:6], col = outputk$cluster)
```



HIERACHICAL CLUSTERING

```
d <- dist(dummy_df2_norm, method = "euclidean")
# We then apply hierarchical clustering using the Ward's method
res.hc <- hclust(d, method = "ward.D2")
# Lastly we plot the obtained dendrogram
#--
plot(res.hc, cex = 0.6, hang = -1)
```



Challenging the Solution

- Using a different number of clusters 9 clusters using the silhouette method

K-MEANS CLUSTERING

```
outputs <- kmeans(dummy_df2_norm, 9)
```

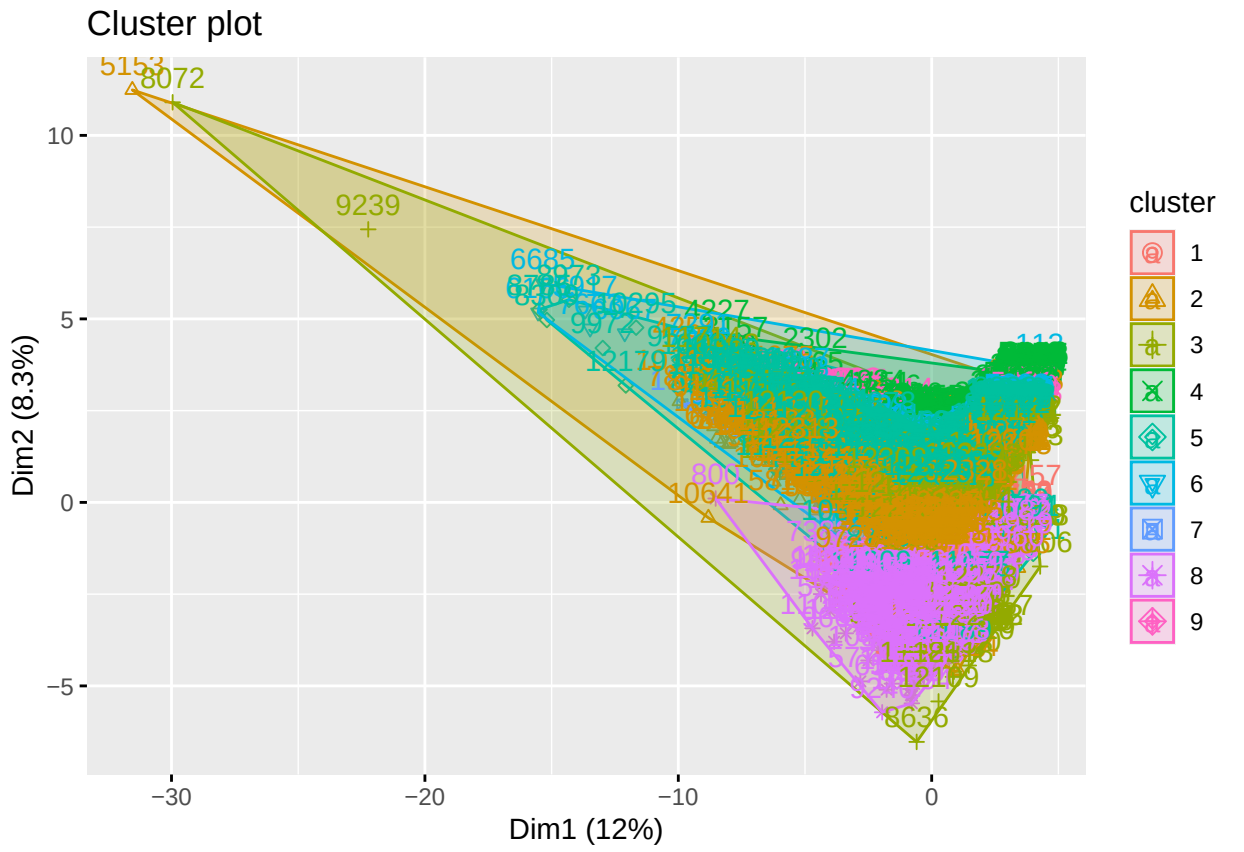
Results

```
# Previewing the number of records in each cluster
outputs$size
```

```
## [1]  319 2377 1082  828 1900 1530 1237 1374 1552
```

Visualizing the clusters of the whole dataset

```
options(repr.plot.width = 11, repr.plot.height = 6)
fviz_cluster(outputs, dummy_df2_norm)
```



Summary

Compasiron Between K-MEANS and HIERACHICAL clustering From the Analysis, we can identify that: 1. K-means Cluster Analysis performs much better in identyfing patterns as compared to Hierrachical clustering.

2. Since the dataset is large, visualizing hierrachical clusters is abit cumbersome as compared to K-means clustering.

3. K-means clustering yields better results using the optimal number of clusters which can be determined by Elbow and Silhouette Methods