#1. Defining the question

1.1 Specifying the data analytic objective

1. Perform clustering stating insights drawn from your analysis and visualizations.
2. Upon implementation, provide comparisons between the approaches learned this week i.e. K-Means clustering vs Hierarchical clustering highlighting the strengths and limitations of each approach in the context of your analysis.

1.2 Defining the metric of success

Clustering the data to maximize reaching the target audience

1.3 Understanding the context

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.

1.4 Recording the Experimental Design

1. Loading the data
2. Checking the data
3. Tidying the data
4. Univariate Analysis
5. Bivariate Analysis
6. Challenging the solution
7. Recommendations
8. Follow up questions

2. Loading the libraries

```
# Installing packages that we have not.

#library(devtools)
#install_github("vqv/ggbiplot", force = TRUE)
#install.packages("DataExplorer")
#install.packages("Hmisc")
#install.packages("pastecs")
#install.packages("psych")
#install.packages("corrplot")
#install.packages("factoextra")
#install.packages("Rtsne")
#install.packages("caret")
```

```r
# Loading Libraries necessary

library(tidyverse)
```

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

```
## -- Attaching packages --------------------------------------- tidyverse
1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(magrittr)
```

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

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

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

```r
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(RColorBrewer)
library(ggplot2)
library(lattice)
library(corrplot)

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

## corrplot 0.84 loaded

library(DataExplorer)

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

library(Hmisc)

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

## 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)

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

##
## 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

library(psych)

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

##
## 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)

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

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

library(Rtsne)
```

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

```
library(caret)
```

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

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

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

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

3. Loading Data

```
customers = read_csv('http://bit.ly/EcommerceCustomersDataset')
```

```
##
## -- Column specification --------------------------------------------------
------
## cols(
##   Administrative = col_double(),
##   Administrative_Duration = col_double(),
##   Informational = col_double(),
##   Informational_Duration = col_double(),
##   ProductRelated = col_double(),
##   ProductRelated_Duration = col_double(),
##   BounceRates = col_double(),
##   ExitRates = col_double(),
##   PageValues = col_double(),
##   SpecialDay = col_double(),
##   Month = col_character(),
##   OperatingSystems = col_double(),
##   Browser = col_double(),
##   Region = col_double(),
##   TrafficType = col_double(),
##   VisitorType = col_character(),
##   Weekend = col_logical(),
##   Revenue = col_logical()
## )
```

```
head(customers)
```

```
## # A tibble: 6 x 18
##   Administrative Administrative_D~ Informational Informational_D~
ProductRelated
##            <dbl>             <dbl>         <dbl>            <dbl>
```

```
<dbl>
## 1              0                 0             0                 0
1
## 2              0                 0             0                 0
2
## 3              0                -1             0                -1
1
## 4              0                 0             0                 0
2
## 5              0                 0             0                 0
10
## 6              0                 0             0                 0
19
## # ... with 13 more variables: ProductRelated_Duration <dbl>, BounceRates
<dbl>,
## #   ExitRates <dbl>, PageValues <dbl>, SpecialDay <dbl>, Month <chr>,
## #   OperatingSystems <dbl>, Browser <dbl>, Region <dbl>, TrafficType
<dbl>,
## #   VisitorType <chr>, Weekend <lgl>, Revenue <lgl>

tail(customers)

## # A tibble: 6 x 18
##   Administrative Administrative_D~ Informational Informational_D~
ProductRelated
##            <dbl>             <dbl>         <dbl>            <dbl>
<dbl>
## 1              0                 0             1                 0
16
## 2              3               145             0                 0
53
## 3              0                 0             0                 0
5
## 4              0                 0             0                 0
6
## 5              4                75             0                 0
15
## 6              0                 0             0                 0
3
## # ... with 13 more variables: ProductRelated_Duration <dbl>, BounceRates
<dbl>,
## #   ExitRates <dbl>, PageValues <dbl>, SpecialDay <dbl>, Month <chr>,
## #   OperatingSystems <dbl>, Browser <dbl>, Region <dbl>, TrafficType
<dbl>,
## #   VisitorType <chr>, Weekend <lgl>, Revenue <lgl>
```

structure of the data

```
str(customers)
```

```
## spec_tbl_df[,18] [12,330 x 18] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Administrative         : num [1:12330] 0 0 0 0 0 0 0 1 0 0 ...
##  $ Administrative_Duration: num [1:12330] 0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ Informational          : num [1:12330] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration : num [1:12330] 0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ ProductRelated         : num [1:12330] 1 2 1 2 10 19 1 1 2 3 ...
##  $ ProductRelated_Duration: num [1:12330] 0 64 -1 2.67 627.5 ...
##  $ BounceRates            : num [1:12330] 0.2 0 0.2 0.05 0.02 ...
##  $ ExitRates              : num [1:12330] 0.2 0.1 0.2 0.14 0.05 ...
##  $ PageValues             : num [1:12330] 0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialDay             : num [1:12330] 0 0 0 0 0 0 0.4 0 0.8 0.4 ...
##  $ Month                  : chr [1:12330] "Feb" "Feb" "Feb" "Feb" ...
##  $ OperatingSystems       : num [1:12330] 1 2 4 3 3 2 2 1 2 2 ...
##  $ Browser                : num [1:12330] 1 2 1 2 3 2 4 2 2 4 ...
##  $ Region                 : num [1:12330] 1 1 9 2 1 1 3 1 2 1 ...
##  $ TrafficType            : num [1:12330] 1 2 3 4 4 3 3 5 3 2 ...
##  $ VisitorType            : chr [1:12330] "Returning_Visitor"
"Returning_Visitor" "Returning_Visitor" "Returning_Visitor" "Returning_Visitor" ...
##  $ Weekend                : logi [1:12330] FALSE FALSE FALSE FALSE TRUE
FALSE ...
##  $ Revenue                : logi [1:12330] FALSE FALSE FALSE FALSE FALSE
FALSE ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Administrative = col_double(),
##   ..   Administrative_Duration = col_double(),
##   ..   Informational = col_double(),
##   ..   Informational_Duration = col_double(),
##   ..   ProductRelated = col_double(),
##   ..   ProductRelated_Duration = col_double(),
##   ..   BounceRates = col_double(),
##   ..   ExitRates = col_double(),
##   ..   PageValues = col_double(),
##   ..   SpecialDay = col_double(),
##   ..   Month = col_character(),
##   ..   OperatingSystems = col_double(),
##   ..   Browser = col_double(),
##   ..   Region = col_double(),
##   ..   TrafficType = col_double(),
##   ..   VisitorType = col_character(),
##   ..   Weekend = col_logical(),
##   ..   Revenue = col_logical()
##   .. )
```

The data has 12330 obs. of 18 variables

Checking for the summary description of our data

```
# Checking for the summary description of our data

summary(customers)
```

```
##  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
##
```

4. DATA CLEANING

```
colSums(is.na(customers))
```

```
##          Administrative Administrative_Duration           Informational
##                      14                      14                      14
##  Informational_Duration          ProductRelated ProductRelated_Duration
##                      14                      14                      14
##             BounceRates               ExitRates              PageValues
```

```
##                       14                       14                        0
##              SpecialDay                    Month          OperatingSystems
##                        0                        0                        0
##                  Browser                   Region              TrafficType
##                        0                        0                        0
##              VisitorType                  Weekend                  Revenue
##                        0                        0                        0
```

We have missing values in 8 columns. Since we have quite a number of rows, we will go ahead and drop these missing values as we will be left with enough data for our analysis

```
# creating a new data frame that does not have missing values

customers1 <- na.omit(customers)
head(customers1)
```

```
## # A tibble: 6 x 18
##   Administrative Administrative_D~ Informational Informational_D~
ProductRelated
##            <dbl>             <dbl>         <dbl>            <dbl>
<dbl>
## 1              0                 0             0                0
1
## 2              0                 0             0                0
2
## 3              0                -1             0               -1
1
## 4              0                 0             0                0
2
## 5              0                 0             0                0
10
## 6              0                 0             0                0
19
## # ... with 13 more variables: ProductRelated_Duration <dbl>, BounceRates
<dbl>,
## #   ExitRates <dbl>, PageValues <dbl>, SpecialDay <dbl>, Month <chr>,
## #   OperatingSystems <dbl>, Browser <dbl>, Region <dbl>, TrafficType
<dbl>,
## #   VisitorType <chr>, Weekend <lgl>, Revenue <lgl>
```

```
# Confirming that we have no null values

sum(colSums(is.na(customers1)))
```

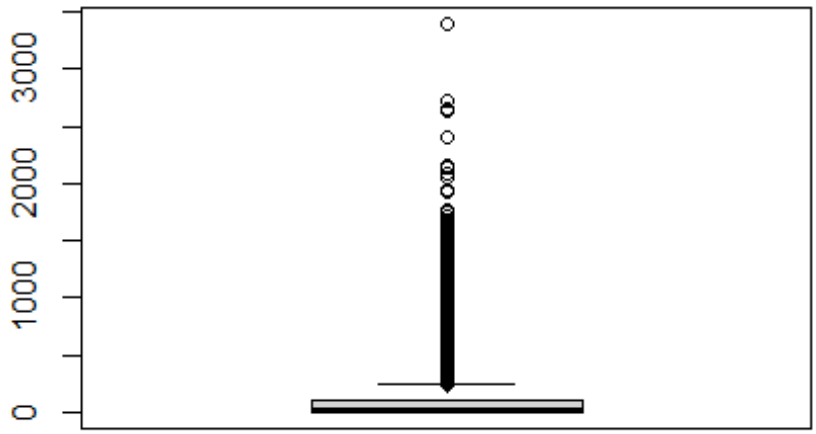
```
## [1] 0
```

```
# Checking for Duplicates

customer <- customers1[duplicated(customers1),]
dim(customer)
```

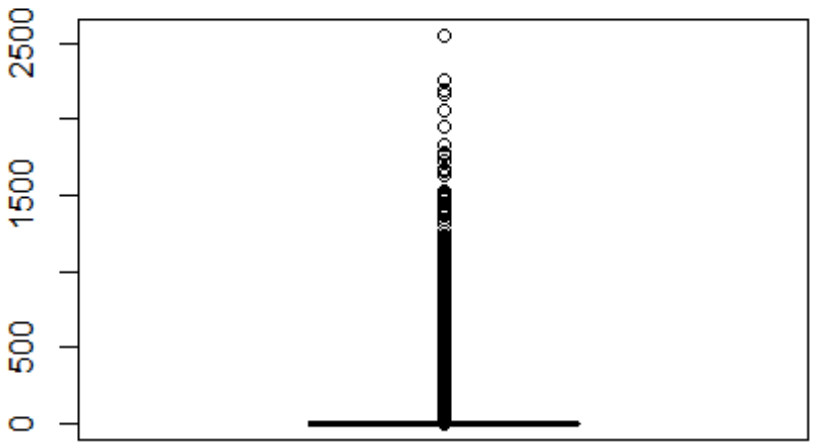
```
## [1] 117  18

# Removing these duplicated rows in the dataset
cust <- customers1[!duplicated(customers1), ]
dim(cust)

## [1] 12199    18
```

Checking for ourliers

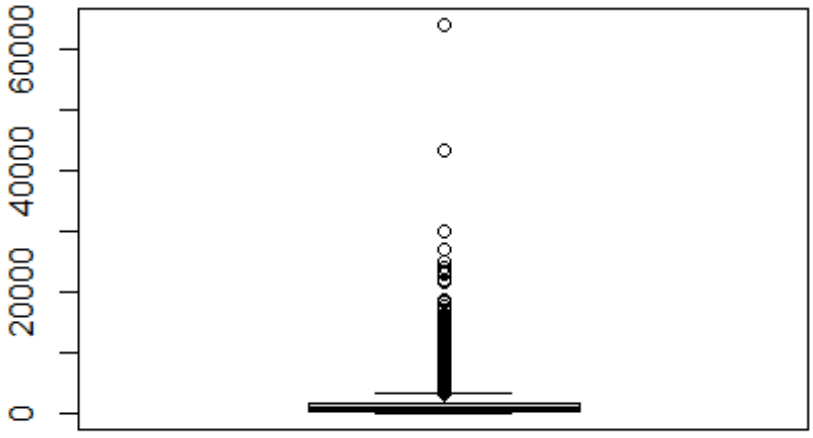
```
# Administrative_Duration
# Plot a boxplot to help us visualise any existing outliers
boxplot(cust$Administrative_Duration)
```



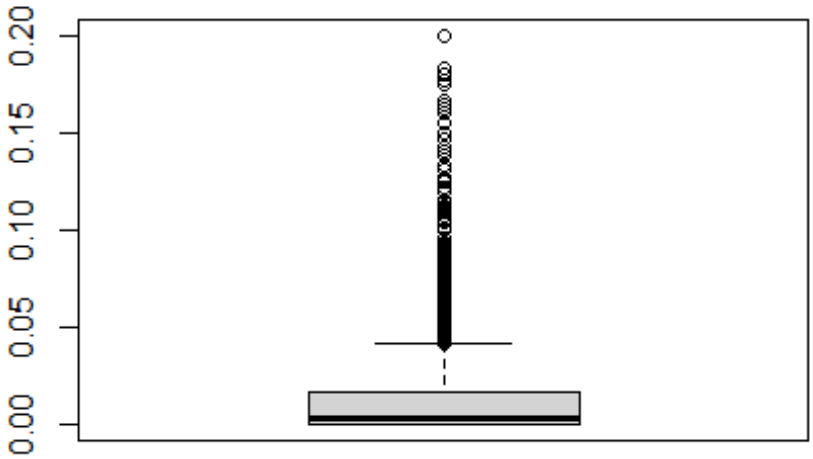

```
# Informational_Duration
boxplot(cust$Informational_Duration)
```



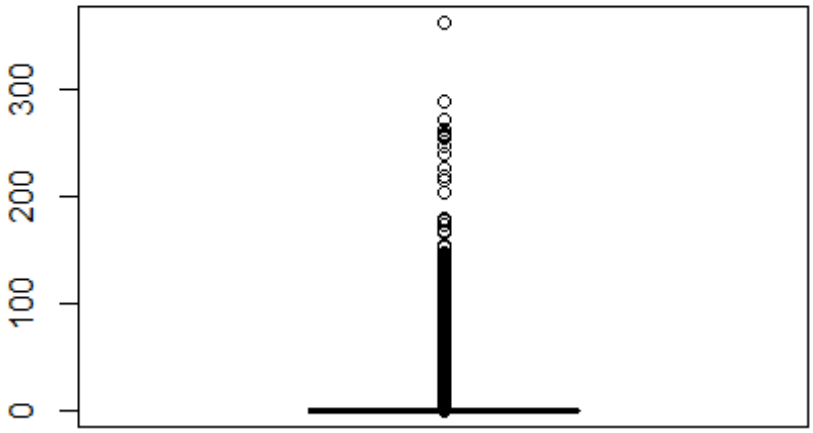

```
# ProductRelated_Duration
boxplot(cust$ProductRelated_Duration)
```



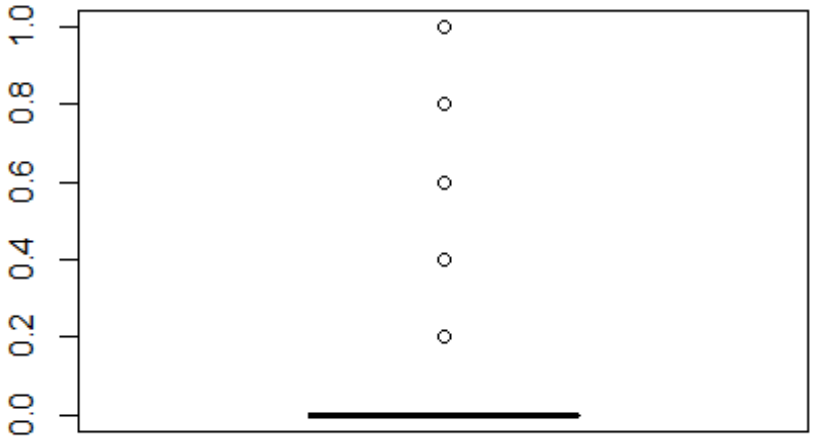
```r
# BounceRates
boxplot(cust$BounceRates)
```




```r
# PageValues
boxplot(cust$PageValues)
```




```r
# SpecialDay
boxplot(cust$SpecialDay)
```



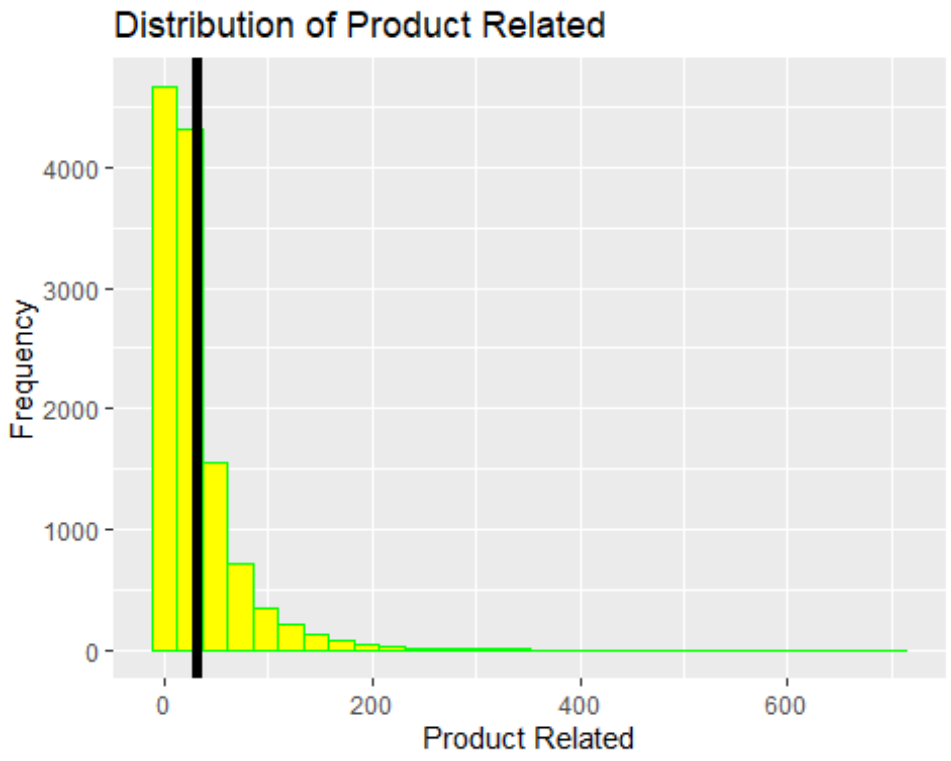
We have outliers in several of our numerical columns. We shall not delete the outliers as they will result in us losing so much customer data which could alter our analysis.

5. Exploratory Data Analysis

Univariate Analysis

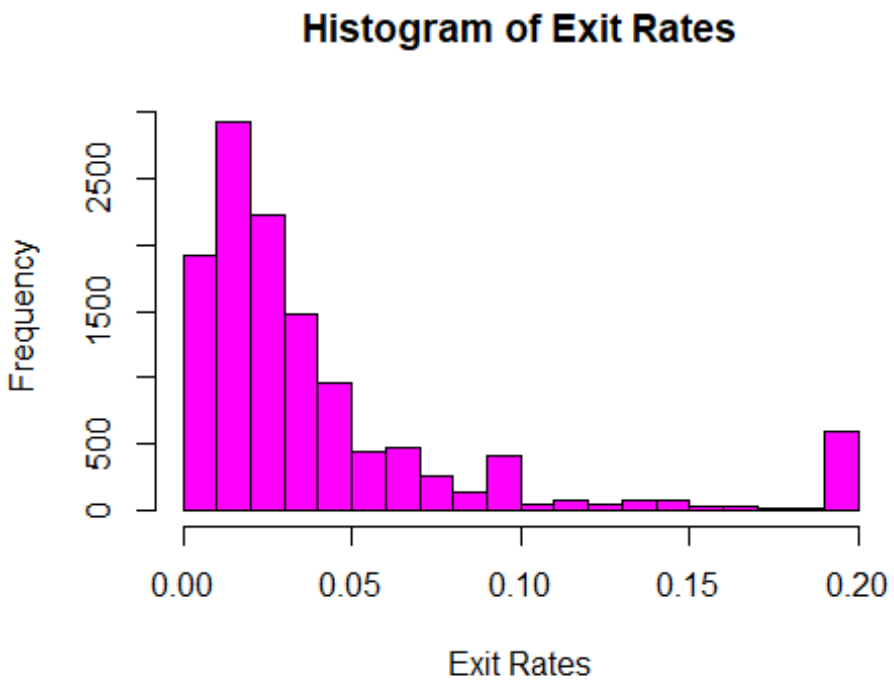
```
# Plotting a histogram using ggplots
#
#

cust %>%
    ggplot(aes(ProductRelated)) +
    geom_histogram(color = "Green",fill = "yellow") +
    geom_vline(xintercept = mean(cust$ProductRelated), lwd = 2) +
    labs(title = "Distribution of Product Related",
         x = "Product Related",
         y = "Frequency")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

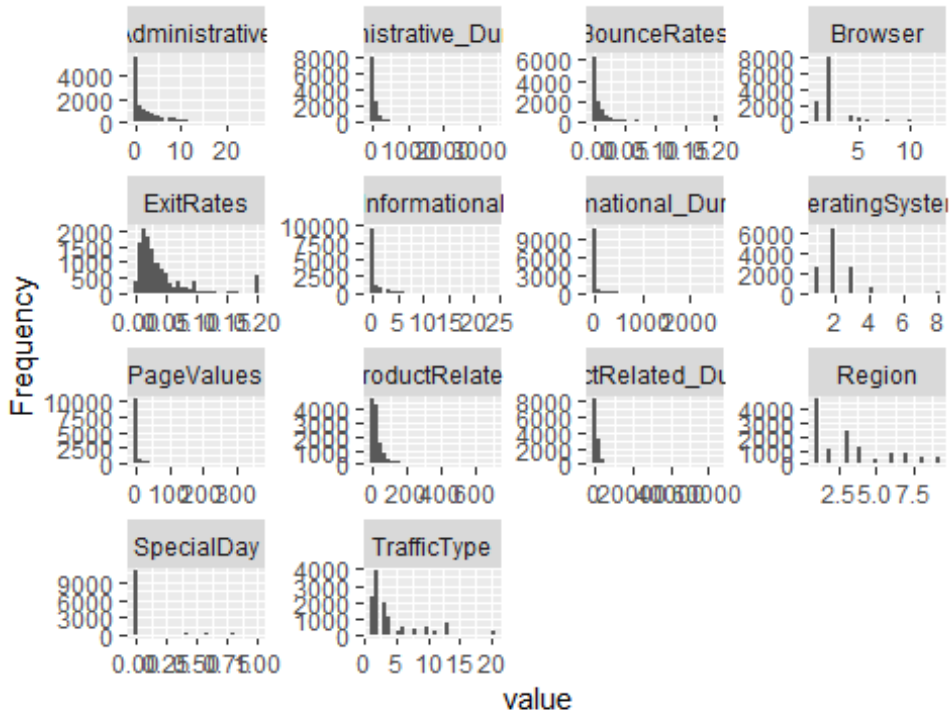



```
# plotting a histogram of Exit Rates

hist(cust$ExitRates,
     main = "Histogram of Exit Rates",
```

```
     xlab = "Exit Rates",
     col = "magenta")
```




```
# Plotting all histograms in the continuous variables in our data

plot_histogram(cust)
```



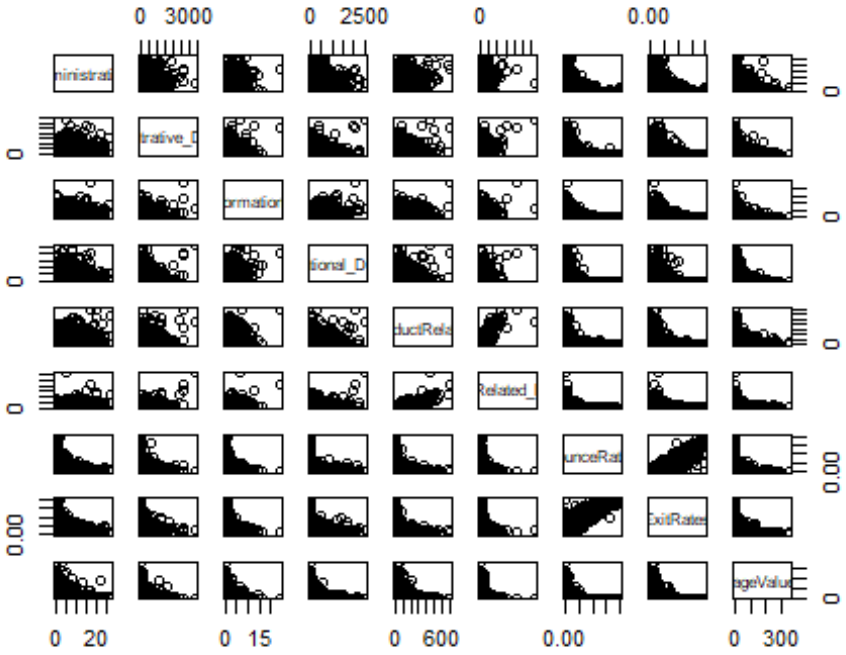
From the histograms, most of our variables are positively skewed.

```
# Bar plots of the categorical/factor modes variables

#par(mfrow=c(4,1))

#for(i in 11:16) {

#   counts <- table(cust[,i])

#   name <- names(cust)[i]
#   barplot(counts, main=name, col = heat.colors(20))}
```

May and November were busy months receiving high traffic, Feb received the least traffic of customers. Most vistors were returning type. Traffic mode number 2, 1 and 3 were heavily used in that order. Region number 1 had the most activity, region 5 was less active. Browser 2 and 1 were the most commonly used for browsing. Operating systems 2, 1 and 3 were mostly used by customers.

Bivariate Analysis

```
# Pair plots for the continous variables
pairs(cust[,1:9])
```

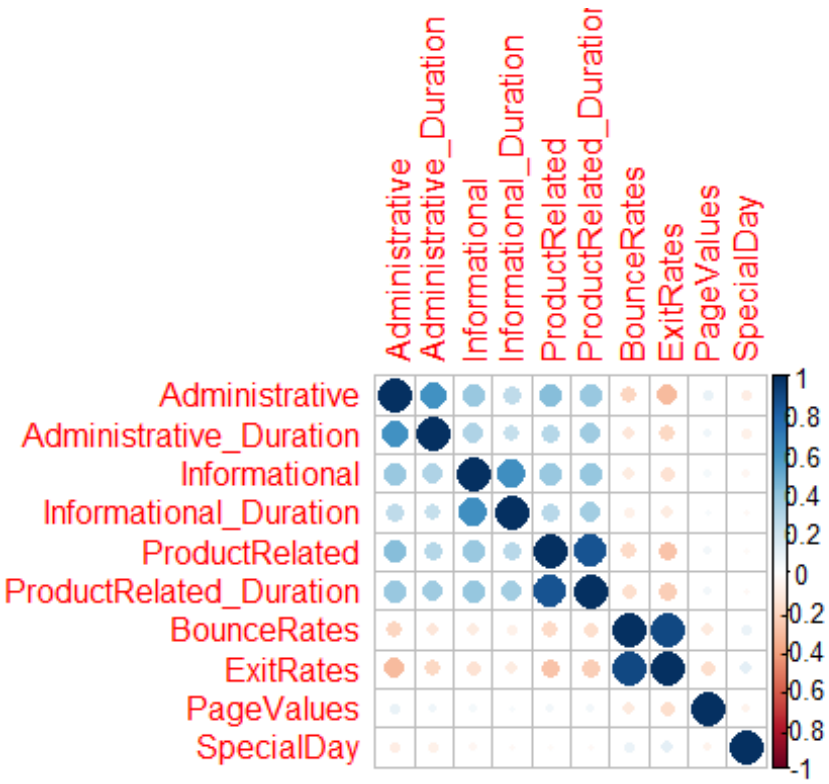



```
# Correlations

# subseting our data frame to get the numeric variables
numerics <- cust[, c(1:10)]

# getting the correlation between these numeric variables
numerics.cor <- cor(numerics)
numerics.cor

##                          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
```

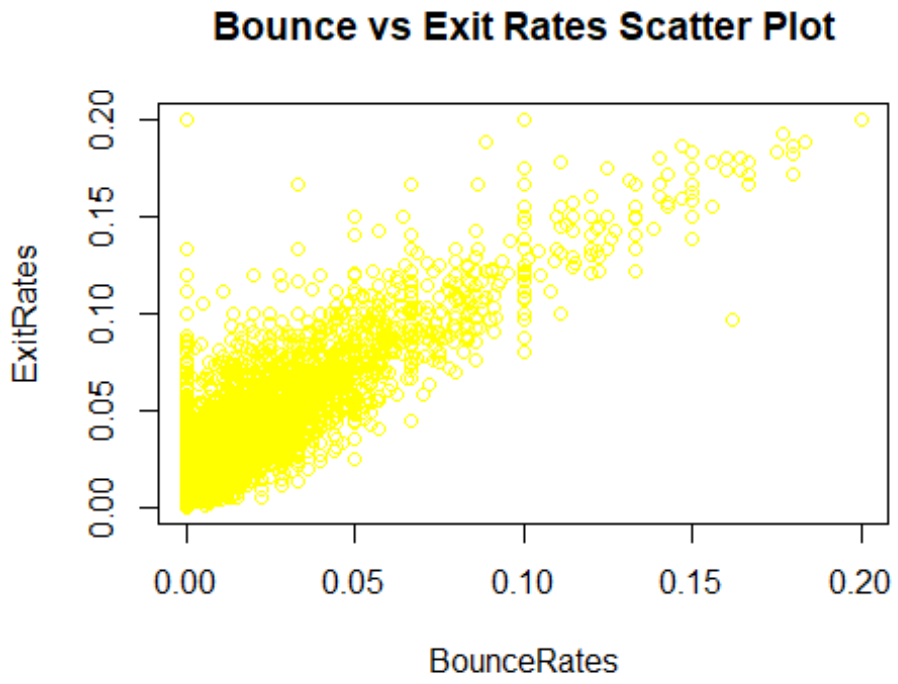
```
## PageValues                  0.09692097             0.06616837
0.04739015
## SpecialDay                -0.09707210            -0.07473689   -
0.04937677
##                        Informational_Duration ProductRelated
## Administrative                      0.25478602     0.42819151
## Administrative_Duration             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
```

```
# installing packages that we shall use to plot the correlation plots

install.packages("Hmisc")

## Warning: package 'Hmisc' is in use and will not be installed

library("Hmisc")

install.packages("corrplot")

## Warning: package 'corrplot' is in use and will not be installed
```

```
library(corrplot)

corrplot(numerics.cor)
```

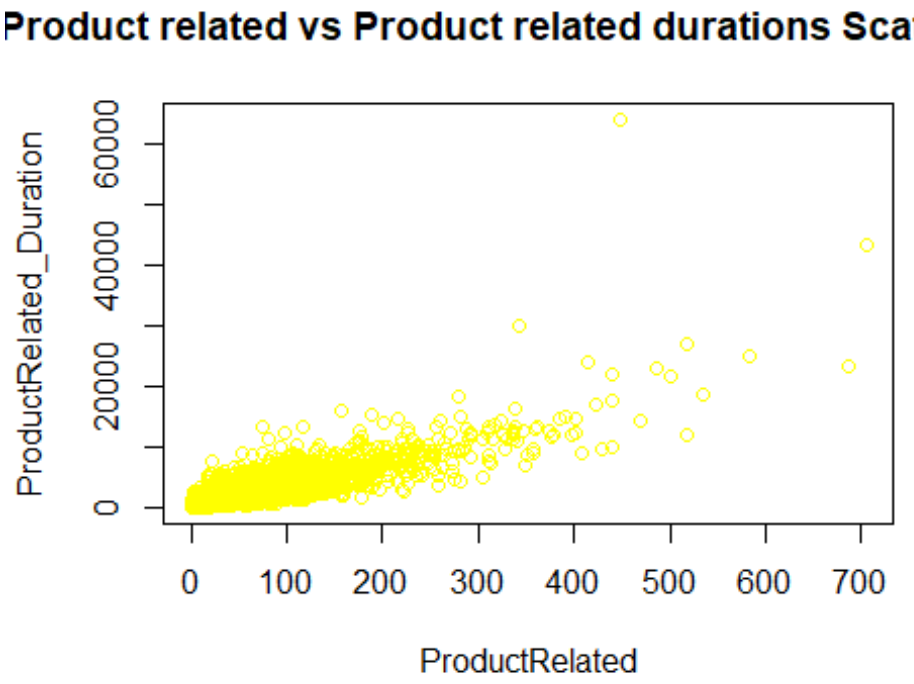



```
# Plotting a scatter plot

plot(ExitRates ~ BounceRates, data = cust,
      col = "yellow",
      main = "Bounce vs Exit Rates Scatter Plot")
```



There is a strong positive correlation between bounce rates and exit rates

```
plot(ProductRelated_Duration ~ ProductRelated, data = cust,
      col = "yellow",
      main = "Product related vs Product related durations Scatter Plot")
```



The is a positive correlation between product relation and product relation duration

K-MEAN CLUSTERING

```
# converting  some of our columns into numerical data types by one hot
encoding
dmy = dummyVars(" ~ .", data = cust)

df4 = data.frame(predict(dmy, newdata = cust))

# Checking the data types of each attribute
sapply(df4, 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"

# Confirming changes
glimpse(df4)

## 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,~
```

```r
# We are instructed to use Revenue as the class label,
# Hence we will remove it and store it in another variable

df4_copy <- df4[, -c(30:31)]
cust.class<- cust[, "Revenue"]

df4_copy_copy <- df4[, -c(30,31)]

# Previewing the class column

head (cust.class)
```

```
## # A tibble: 6 x 1
##   Revenue
##   <lgl>
## 1 FALSE
## 2 FALSE
## 3 FALSE
## 4 FALSE
## 5 FALSE
## 6 FALSE
```

```
# Previewing the copy dataset with dummies
head(df4_copy)
```

```
##   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 MonthAug MonthDec MonthFeb MonthJul MonthJune MonthMar
MonthMay
## 1          0        0        0        1        0         0        0
0
## 2          0        0        0        1        0         0        0
0
## 3          0        0        0        1        0         0        0
0
## 4          0        0        0        1        0         0        0
0
## 5          0        0        0        1        0         0        0
0
## 6          0        0        0        1        0         0        0
0
##   MonthNov MonthOct MonthSep OperatingSystems Browser Region TrafficType
## 1        0        0        0                1       1      1           1
```

```
## 2        0        0        0                2       2      1           2
## 3        0        0        0                4       1      9           3
## 4        0        0        0                3       2      2           4
## 5        0        0        0                3       3      1           4
## 6        0        0        0                2       2      1           3
##   VisitorTypeNew_Visitor VisitorTypeOther VisitorTypeReturning_Visitor
## 1                      0                0                           1
## 2                      0                0                           1
## 3                      0                0                           1
## 4                      0                0                           1
## 5                      0                0                           1
## 6                      0                0                           1
##   WeekendFALSE WeekendTRUE
## 1            1           0
## 2            1           0
## 3            1           0
## 4            1           0
## 5            0           1
## 6            1           0
```

```
# Normalizing the a copy of the original data

df_norm <- as.data.frame(apply(df4_copy, 2, function(x) (x - min(x))/(max(x)-
min(x))))

# Applying K-Means  Clustering algorithm
# Using 3 centroids as K=3

result <- kmeans(df_norm, 10)

# Previewing the number of records in each cluster

result$size
```

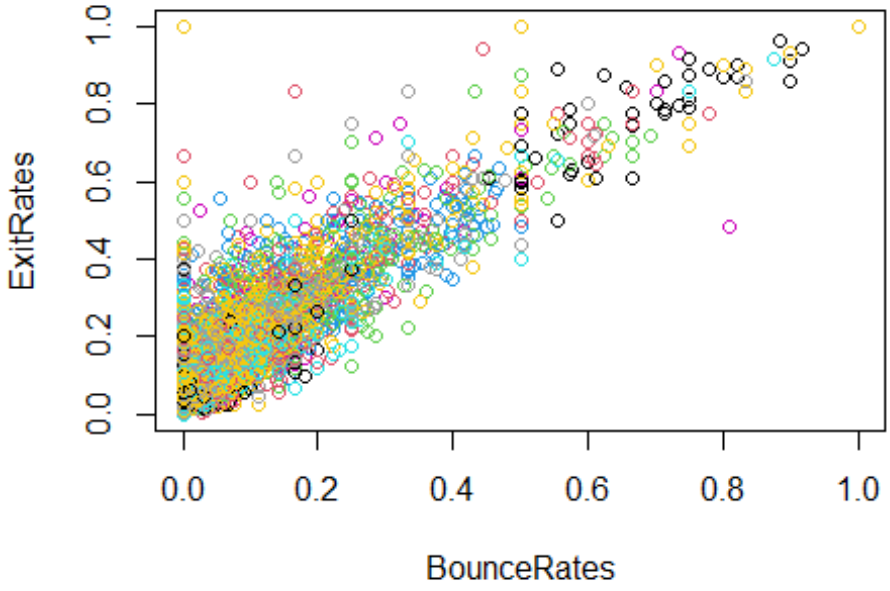
```
##  [1] 1559   56 1447 2154  791 1237 1894 1036  319 1706
```

```
# Viewing the cluster center data points by each attribute

result$centers
```

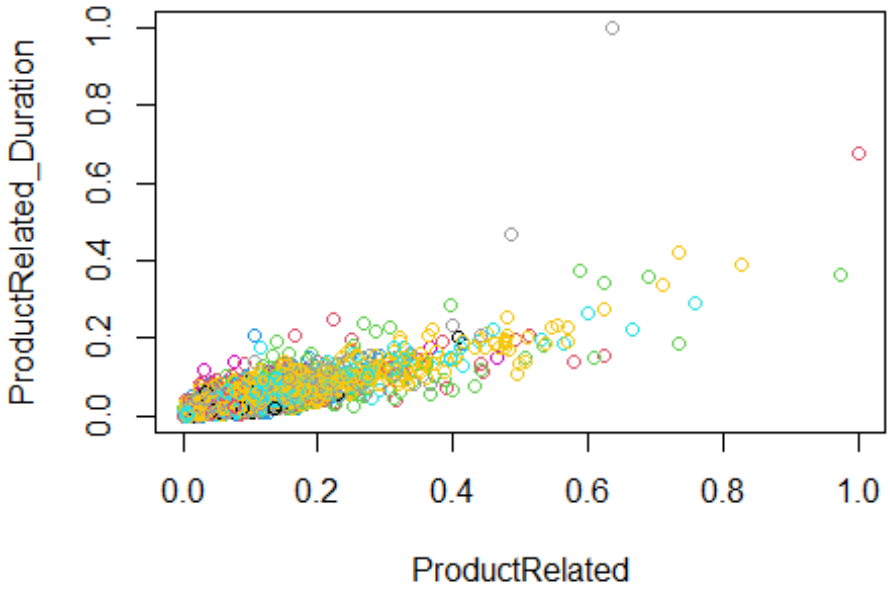
```
##    Administrative Administrative_Duration Informational
Informational_Duration
## 1     0.096405578            0.0280056403  0.0142185161
0.0081648673
## 2     0.072089947            0.0233755371  0.0089285714
0.0023579632
## 3     0.108628324            0.0285327300  0.0215100207
0.0144937644
## 4     0.074589910            0.0212518408  0.0185507583
0.0114560675
## 5     0.098422063            0.0274243768  0.0325537295
```

```
0.0227246296
## 6     0.064463008            0.0198800762  0.0170439235
0.0124017834
## 7     0.099417263            0.0275356194  0.0275651179
0.0173811148
## 8     0.086622337            0.0244943839  0.0260215573
0.0200849968
## 9     0.000812725            0.0003024379  0.0003918495
0.0003724328
## 10    0.087447354            0.0238791760  0.0227139508
0.0150908529
##    ProductRelated ProductRelated_Duration BounceRates  ExitRates
PageValues
## 1     0.025599243            0.0098944636  0.02225585 0.09909113
0.030272573
## 2     0.022821682            0.0115910854  0.09537235 0.20018753
0.075888455
## 3     0.056231773            0.0218966547  0.06953111 0.18188255
0.014532005
## 4     0.041236821            0.0170270487  0.06730060 0.19058831
0.013081924
## 5     0.068816742            0.0287368632  0.07122683 0.15986081
0.020557889
## 6     0.028560289            0.0132562215  0.10695635 0.23317584
0.008564211
## 7     0.069898223            0.0297024571  0.10749265 0.20799382
0.017129666
## 8     0.046978285            0.0209540853  0.10072091 0.21453222
0.015647022
## 9     0.004388715            0.0006083992  0.89078548 0.93237532
0.000000000
## 10    0.041691818            0.0168356777  0.10597300 0.21288373
0.014398351
##    SpecialDay   MonthAug  MonthDec     MonthFeb   MonthJul  MonthJune
MonthMar
## 1  0.02155228 0.04618345 0.2135985 0.0006414368 0.03463759 0.01924310
0.1488133
## 2  0.00000000 0.00000000 0.8750000 0.0000000000 0.00000000 0.01785714
0.0000000
## 3  0.01935038 0.19281272 0.0000000 0.0877677954 0.18866621 0.13199724
0.0000000
## 4  0.21903435 0.00000000 0.0000000 0.0000000000 0.00000000 0.00000000
0.0000000
## 5  0.00000000 0.00000000 0.0000000 0.0000000000 0.00000000 0.00000000
0.0000000
## 6  0.00000000 0.00000000 0.0000000 0.0000000000 0.00000000 0.00000000
1.0000000
## 7  0.00000000 0.00000000 0.0000000 0.0000000000 0.00000000 0.00000000
0.0000000
## 8  0.00000000 0.00000000 1.0000000 0.0000000000 0.00000000 0.00000000
```

```
0.0000000
## 9  0.24075235 0.03134796 0.0000000 0.0815047022 0.04388715 0.06896552
0.0000000
## 10 0.08546307 0.04220399 0.1688159 0.0164126612 0.05334115 0.02403283
0.2250879
##     MonthMay  MonthNov   MonthOct   MonthSep OperatingSystems   Browser
## 1  0.2007697 0.1872996 0.07953817 0.06927518        0.1472556 0.1081356
## 2  0.0000000 0.1071429 0.00000000 0.00000000        0.9311224 0.9047619
## 3  0.0000000 0.0000000 0.21907395 0.17968210        0.1547043 0.1142594
## 4  1.0000000 0.0000000 0.00000000 0.00000000        0.1630190 0.1173785
## 5  0.0000000 1.0000000 0.00000000 0.00000000        0.1638071 0.1038769
## 6  0.0000000 0.0000000 0.00000000 0.00000000        0.1540594 0.1138507
## 7  0.0000000 1.0000000 0.00000000 0.00000000        0.1579424 0.1042767
## 8  0.0000000 0.0000000 0.00000000 0.00000000        0.1530612 0.1109234
## 9  0.7429467 0.0000000 0.01253918 0.01880878        0.1670399 0.1076280
## 10 0.3657679 0.0000000 0.06096131 0.04337632        0.1591861 0.1016999
##       Region TrafficType VisitorTypeNew_Visitor VisitorTypeOther
## 1  0.2829538   0.1472604             1.00000000      0.000000000
## 2  0.9508929   0.9069549             0.01785714      0.964285714
## 3  0.2753974   0.1438912             0.00000000      0.000000000
## 4  0.2618384   0.1827933             0.00000000      0.000000000
## 5  0.2586915   0.2041387             0.16055626      0.002528445
## 6  0.2649555   0.1039867             0.00000000      0.000000000
## 7  0.2455781   0.1750959             0.00000000      0.007391763
## 8  0.2764237   0.1433144             0.00000000      0.007722008
## 9  0.2699843   0.2390695             0.01880878      0.000000000
## 10 0.2676583   0.1437650             0.00000000      0.001758499
##    VisitorTypeReturning_Visitor WeekendFALSE WeekendTRUE
## 1                    0.00000000    0.7742142  0.22578576
## 2                    0.01785714    0.9642857  0.03571429
## 3                    1.00000000    1.0000000  0.00000000
## 4                    1.00000000    1.0000000  0.00000000
## 5                    0.83691530    0.0000000  1.00000000
## 6                    1.00000000    1.0000000  0.00000000
## 7                    0.99260824    1.0000000  0.00000000
## 8                    0.99227799    1.0000000  0.00000000
## 9                    0.98119122    0.9843260  0.01567398
## 10                   0.99824150    0.0000000  1.00000000

# Plotting to see how exit rates and bounce rates data points have been
distributed in clusters

plot(df_norm[c(7,8)], col = result$cluster)
```




```
# Product Related, vs Product Related Duration

plot(df_norm[, 5:6], col = result$cluster)
```



Hierachical Clustering

```
# We use R function hclust()
# For hierarchical clustering
# First we use the dist() to compute the Euclidean distance btwn obs
# d will be the first argument in the hclust() dissimilairty matrix
#

d <- dist(df_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)
```