Customers behaviour analysis

Mildred Kulei

9/3/2021

CUSTOMER BEHAVIOUR ANALYSIS

1. Problem Definition

1.1 Specifying the Question

what is the characteristics of the customer groups.

1.2 Metric for success

Come up with an analysis that will make our client identify the behaviour and characteristics of it's customers.

1.3 Understanding the Context

Consumer/customer behaviour is the study of how individual customers, groups or organizations select, buy, use, and dispose ideas, goods, and services to satisfy their needs and wants. It refers to the actions of the consumers in the marketplace and the underlying motives for those actions. Marketers need to understand the buying behaviour of consumers for their products to do well. It is really important for marketers to understand what prompts a consumer to purchase a particular product and what stops him from buying, Thus the need to do customer behaviour analysis.

1.4 Experimental Design taken

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(Clustering)
7. Challenge the Solution
8. Follow up Questions

1.5 Data relevance

The data collected is relevant as it is sourced from Ecommerce customer

http://bit.ly/EcommerceCustomersDataset

2. Data Sourcing

Loading the data

Loading the necessary packages

```
library("data.table")
customer <- read.csv("online_shoppers_intention.csv")
```

```
#loading libraries
#library(ggplot2) # Data visualization

#install.packages("plotly")
library(plotly) # Interactive data visualizations
```

```
## Loading required package: ggplot2

##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(dplyr) # Data manipulation
```

```
##
## Attaching package: 'dplyr'

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

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(psych) # Will be used for correlation visualization
```

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

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

3. checking the data

```r
##Previewing the first 6 rows of dataset

head(customer)
```

```
##   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
```

```r
##Previewing the last 6 rows of dataset

tail(customer)
```

```
##       Administrative Administrative_Duration Informational
```

```
## 12325              0                      0             1
## 12326              3                    145             0
## 12327              0                      0             0
## 12328              0                      0             0
## 12329              4                     75             0
## 12330              0                      0             0
##       Informational_Duration ProductRelated ProductRelated_Duration BounceRates
## 12325                      0             16                 503.000 0.000000000
## 12326                      0             53                1783.792 0.007142857
## 12327                      0              5                 465.750 0.000000000
## 12328                      0              6                 184.250 0.083333333
## 12329                      0             15                 346.000 0.000000000
## 12330                      0              3                  21.250 0.000000000
##        ExitRates PageValues SpecialDay Month OperatingSystems Browser Region
## 12325 0.03764706    0.00000          0   Nov                2       2      1
## 12326 0.02903061   12.24172          0   Dec                4       6      1
## 12327 0.02133333    0.00000          0   Nov                3       2      1
## 12328 0.08666667    0.00000          0   Nov                3       2      1
## 12329 0.02105263    0.00000          0   Nov                2       2      3
## 12330 0.06666667    0.00000          0   Nov                3       2      1
##       TrafficType       VisitorType Weekend Revenue
## 12325           1 Returning_Visitor   FALSE   FALSE
## 12326           1 Returning_Visitor    TRUE   FALSE
## 12327           8 Returning_Visitor    TRUE   FALSE
## 12328          13 Returning_Visitor    TRUE   FALSE
## 12329          11 Returning_Visitor   FALSE   FALSE
## 12330           2       New_Visitor    TRUE   FALSE
```

```
##Basic structure of the data
str(customer)
```

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

```
# previewing the column names
colnames(customer)
```

```
##  [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"
```

```
# previewing the dataset
class(customer)
```

```
## [1] "data.frame"
```

```
# previewing the datatypes of the dataset
sapply(customer, 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"
```

```
# checking the shape of the data
dim(customer)
```

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

There are 12330 records of data and 18 columns.

4. Perform Data Cleaning

missing values

```
# checking for missing values
sum(is.na(customer))
```

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

There are 112 missing values

```r
# displaying all rows from the dataset that don't contain any missing values
customer1 <- na.omit(customer)
head(customer1)
```

```
##   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
```

Duplicates

```r
# Identifying duplicates
duplicates <- customer1[duplicated(customer1), ]
head(duplicates)
```

```
##     Administrative Administrative_Duration Informational Informational_Duration
## 159              0                       0             0                      0
## 179              0                       0             0                      0
## 419              0                       0             0                      0
## 457              0                       0             0                      0
## 484              0                       0             0                      0
## 513              0                       0             0                      0
##     ProductRelated ProductRelated_Duration BounceRates ExitRates PageValues
## 159              1                       0         0.2       0.2          0
## 179              1                       0         0.2       0.2          0
## 419              1                       0         0.2       0.2          0
## 457              1                       0         0.2       0.2          0
```

```
## 484              1                       0         0.2       0.2          0
## 513              1                       0         0.2       0.2          0
##     SpecialDay Month OperatingSystems Browser Region TrafficType
## 159          0   Feb                1       1      1           3
## 179          0   Feb                3       2      3           3
## 419          0   Mar                1       1      1           1
## 457          0   Mar                2       2      4           1
## 484          0   Mar                3       2      3           1
## 513          0   Mar                2       2      1           1
##           VisitorType Weekend Revenue
## 159 Returning_Visitor   FALSE   FALSE
## 179 Returning_Visitor   FALSE   FALSE
## 419 Returning_Visitor    TRUE   FALSE
## 457 Returning_Visitor   FALSE   FALSE
## 484 Returning_Visitor   FALSE   FALSE
## 513 Returning_Visitor   FALSE   FALSE
```

There are 119 duplicated rows

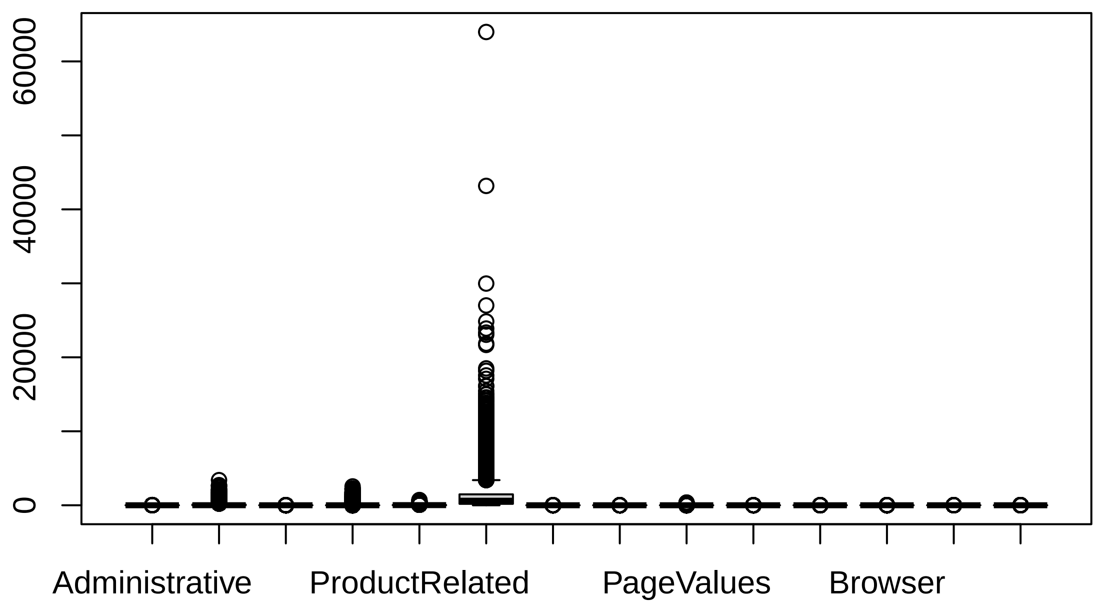
```r
#dealing with duplicates
# showing unique items fromthe dataset and assigning to a variable unique_items below

customer_unique <- unique(customer1)
head(customer_unique)
```

```
##   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
```

There are 12,199 unique rows in our dataset "customer_unique".

Outliers

```
numeric_df <- customer_unique %>% select_if(is.numeric)
boxplot(numeric_df)
```



Most of the column outliers and i will choose to work with them since they might be a true representaion of the data.

checking for anomalies

Anomalies are inconsistencies in the data

```
###Checking the number of unique values in each column
lengths(lapply(customer1, unique))
```

```
##          Administrative Administrative_Duration           Informational
##                      27                    3336                      17
##  Informational_Duration          ProductRelated ProductRelated_Duration
##                    1259                     311                    9552
##             BounceRates               ExitRates              PageValues
##                    1872                    4777                    2704
```

```
##              SpecialDay                      Month        OperatingSystems
##                       6                         10                       8
##                 Browser                     Region             TrafficType
##                      13                          9                      20
##             VisitorType                    Weekend                 Revenue
##                       3                          2                       2
```

```
str(customer1)
```

```
## 'data.frame':    12316 obs. of  18 variables:
##  $ Administrative         : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Administrative_Duration: num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ Informational          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration : num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ ProductRelated         : int  1 2 1 2 10 19 1 1 2 3 ...
##  $ ProductRelated_Duration: num  0 64 -1 2.67 627.5 ...
##  $ BounceRates            : num  0.2 0 0.2 0.05 0.02 ...
##  $ ExitRates              : num  0.2 0.1 0.2 0.14 0.05 ...
##  $ PageValues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialDay             : num  0 0 0 0 0 0 0.4 0 0.8 0.4 ...
##  $ Month                  : chr  "Feb" "Feb" "Feb" "Feb" ...
##  $ OperatingSystems       : int  1 2 4 3 3 2 2 1 2 2 ...
##  $ Browser                : int  1 2 1 2 3 2 4 2 2 4 ...
##  $ Region                 : int  1 1 9 2 1 1 3 1 2 1 ...
##  $ TrafficType            : int  1 2 3 4 4 3 3 5 3 2 ...
##  $ VisitorType            : chr  "Returning_Visitor" "Returning_Visitor" "Returning_Visitor" "Returni
##  $ Weekend                : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ Revenue                : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  - attr(*, "na.action")= 'omit' Named int [1:14] 1066 1133 1134 1135 1136 1137 1474 1475 1476 1477 .
##   ..- attr(*, "names")= chr [1:14] "1066" "1133" "1134" "1135" ...
```

Fromthe results of the anomalies, we can see that there are no anomalies detected, so i will retain the outliers since they might be as a results of the nature of the dataset.

5. Exploratory Data Analysis (Univariate, Bivariate & Multivariate)

5.1 Univariate analysis

```
#descriptive statistics
summary(customer_unique)
```

```
##  Administrative  Administrative_Duration Informational
##  Min.   : 0.00   Min.   :  -1.00         Min.   : 0.0000
##  1st Qu.: 0.00   1st Qu.:   0.00         1st Qu.: 0.0000
##  Median : 1.00   Median :   9.00         Median : 0.0000
##  Mean   : 2.34   Mean   :  81.68         Mean   : 0.5088
##  3rd Qu.: 4.00   3rd Qu.:  94.75         3rd Qu.: 0.0000
##  Max.   :27.00   Max.   :3398.75         Max.   :24.0000
```

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

From the above summeries, 1. more people visited the online site less during the weekedn as compared to weekdays. 2. Revenue collected was very little like about 20% of what was expected.

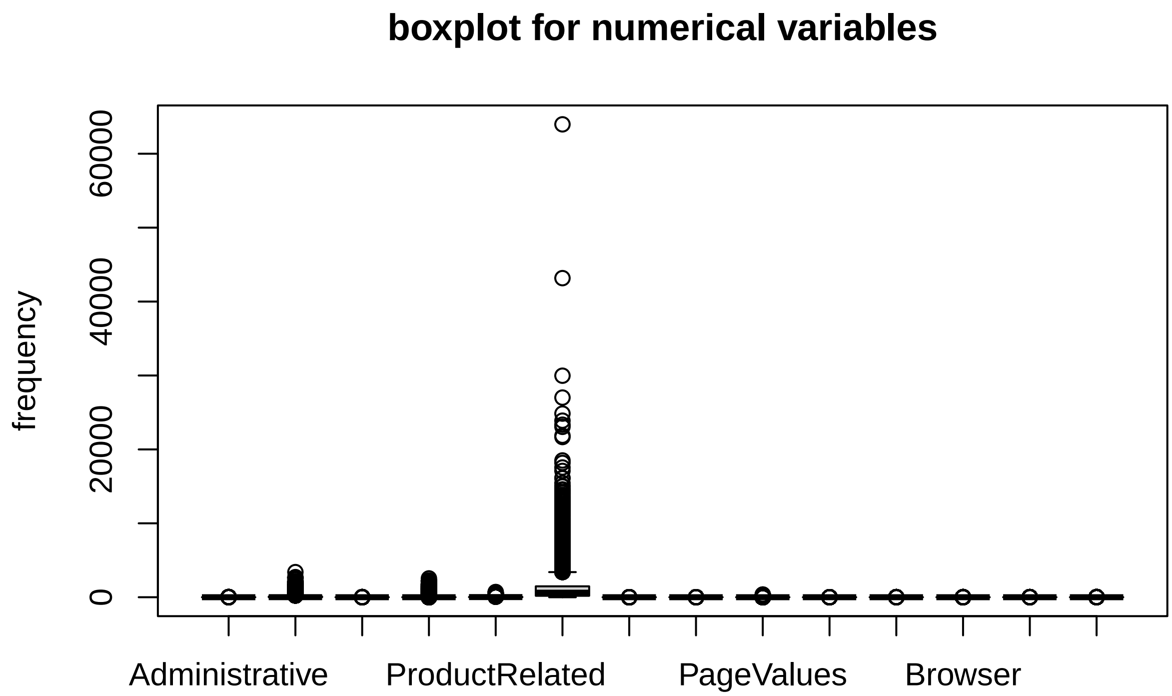
```
#this will show the measures of central tendancies and dispersion of the numerical column
describe(customer_unique)
```

```
## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf
```

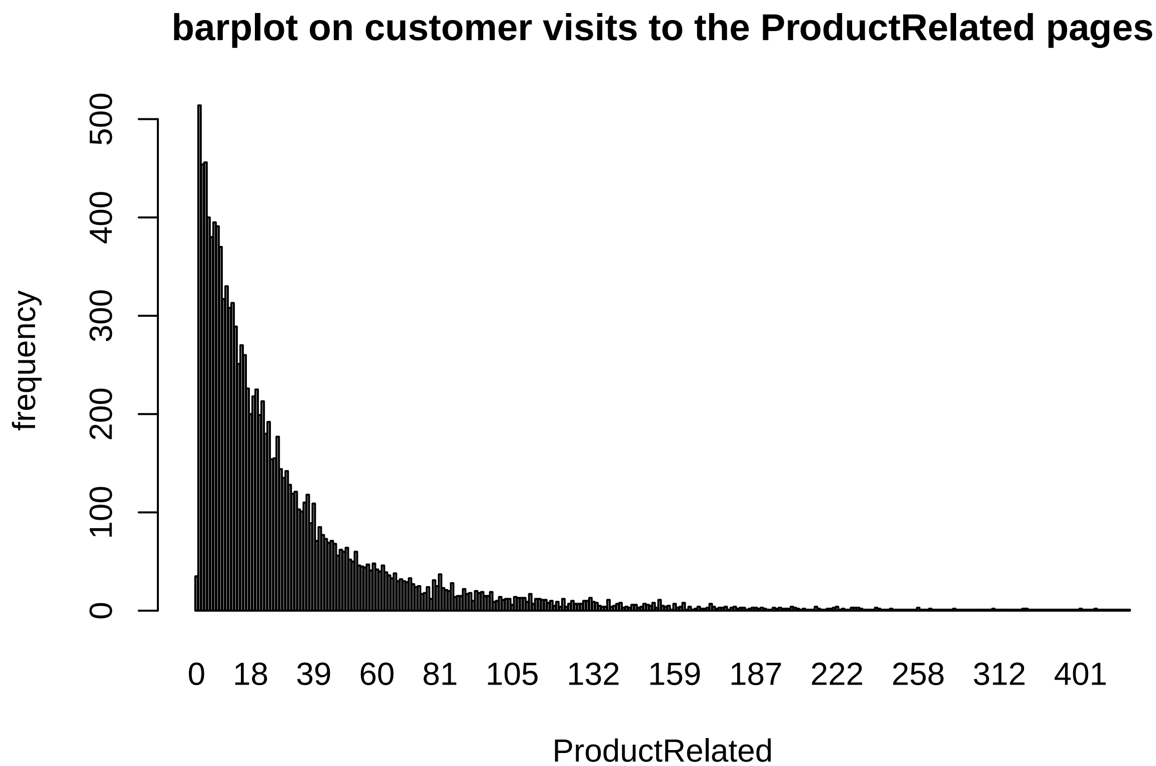
```
##                         vars     n    mean      sd median trimmed    mad min
## Administrative             1 12199    2.34    3.33   1.00    1.66   1.48   0
## Administrative_Duration    2 12199   81.68  177.53   9.00   42.87  13.34  -1
## Informational              3 12199    0.51    1.28   0.00    0.18   0.00   0
## Informational_Duration     4 12199   34.84  141.46   0.00    3.73   0.00  -1
## ProductRelated             5 12199   32.06   44.60  18.00   23.06  19.27   0
## ProductRelated_Duration    6 12199 1207.51 1919.93 609.54  832.36 745.12  -1
## BounceRates                7 12199    0.02    0.05   0.00    0.01   0.00   0
## ExitRates                  8 12199    0.04    0.05   0.03    0.03   0.02   0
## PageValues                 9 12199    5.95   18.66   0.00    1.33   0.00   0
```

```
## SpecialDay                10 12199    0.06    0.20    0.00    0.00    0.00    0
## Month*                    11 12199    6.17    2.37    7.00    6.36    1.48    1
## OperatingSystems          12 12199    2.12    0.91    2.00    2.06    0.00    1
## Browser                   13 12199    2.36    1.71    2.00    2.00    0.00    1
## Region                    14 12199    3.15    2.40    3.00    2.79    2.97    1
## TrafficType               15 12199    4.07    4.02    2.00    3.22    1.48    1
## VisitorType*              16 12199    2.72    0.69    3.00    2.89    0.00    1
## Weekend                   17 12199     NaN      NA      NA     NaN      NA Inf
## Revenue                   18 12199     NaN      NA      NA     NaN      NA Inf
##                             max    range  skew kurtosis    se
## Administrative            27.00    27.00  1.95     4.63  0.03
## Administrative_Duration 3398.75  3399.75  5.59    50.09  1.61
## Informational             24.00    24.00  4.01    26.64  0.01
## Informational_Duration  2549.38  2550.38  7.54    75.45  1.28
## ProductRelated           705.00   705.00  4.33    31.04  0.40
## ProductRelated_Duration 63973.52 63974.52  7.25   136.57 17.38
## BounceRates                0.20     0.20  3.15     9.25  0.00
## ExitRates                  0.20     0.20  2.23     4.62  0.00
## PageValues               361.76   361.76  6.35    64.93  0.17
## SpecialDay                 1.00     1.00  3.28     9.78  0.00
## Month*                    10.00     9.00 -0.83    -0.37  0.02
## OperatingSystems           8.00     7.00  2.03    10.27  0.01
## Browser                   13.00    12.00  3.22    12.53  0.02
## Region                     9.00     8.00  0.98    -0.16  0.02
## TrafficType               20.00    19.00  1.96     3.47  0.04
## VisitorType*               3.00     2.00 -2.05     2.23  0.01
## Weekend                    -Inf     -Inf    NA       NA    NA
## Revenue                    -Inf     -Inf    NA       NA    NA
```

```r
# creating a boxplot graph for all numerical variables
boxplot(numeric_df, ylab = 'frequency', main = 'boxplot for numerical variables')
```

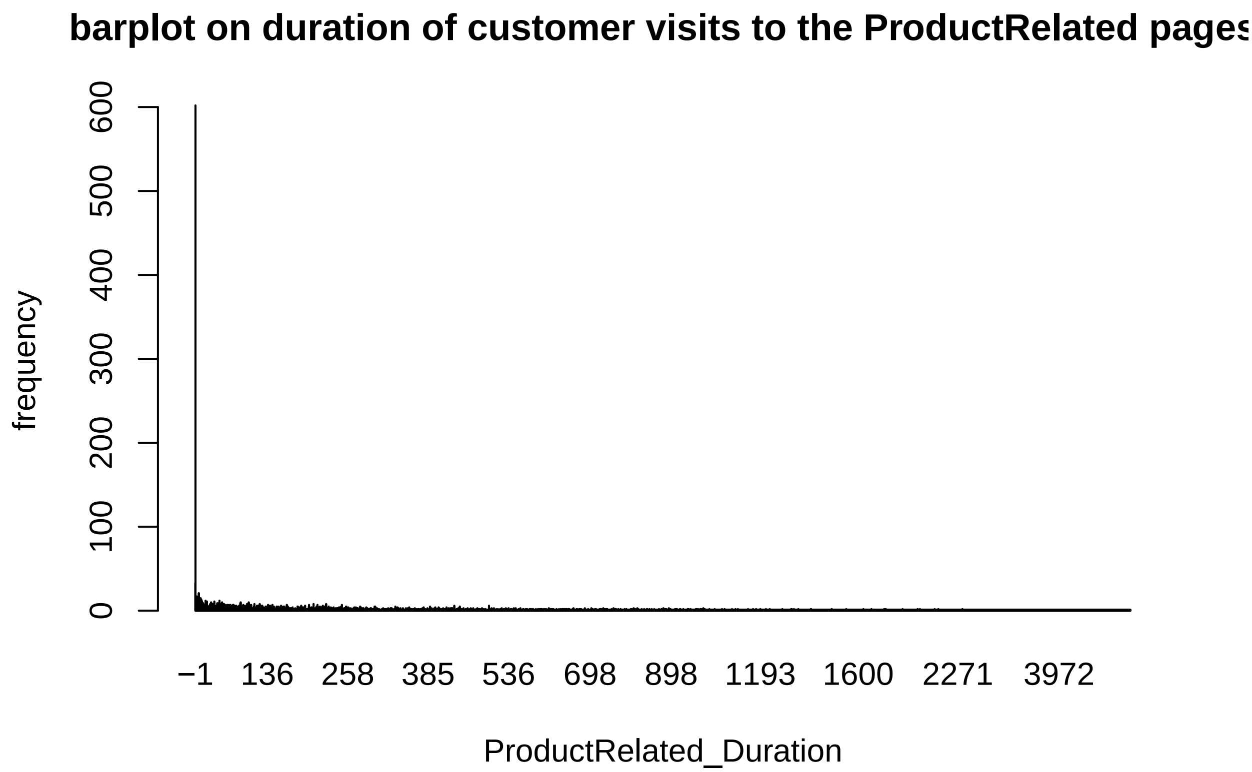



```
# fetching the columns
ProductRelated <- numeric_df$ProductRelated

# fetching the frequency distribution
ProductRelated_frequency <- table(ProductRelated)

# plotting the bargraph
barplot(ProductRelated_frequency,  xlab = 'ProductRelated', ylab = 'frequency',  main = 'barplot on cus
```



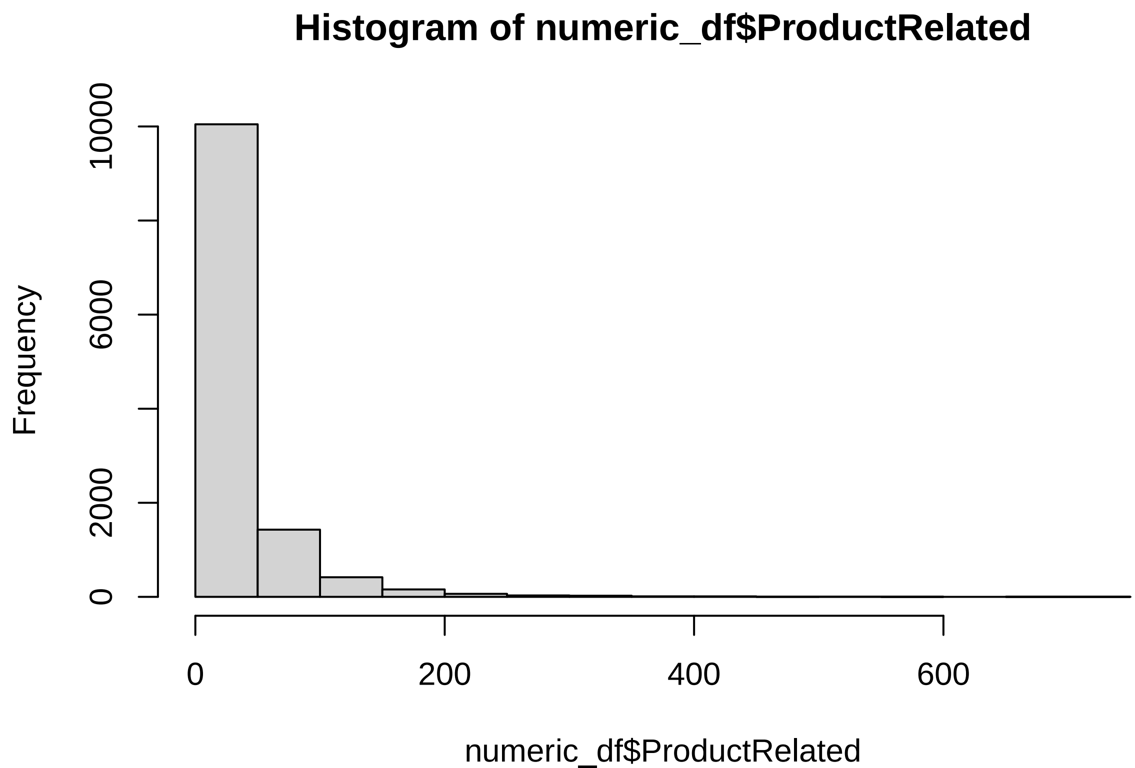
There are high frequency on few numbers of visited sites, the higher the number the lower the frequency.

```
# fetching the columns
ProductRelated_Duration <- numeric_df$ProductRelated_Duration

# fetching the frequency distribution
ProductRelated_Duration_frequency <- table(ProductRelated_Duration)

# plotting the bargraph
barplot(ProductRelated_Duration_frequency,  xlab = 'ProductRelated_Duration', ylab = 'frequency',  main
```

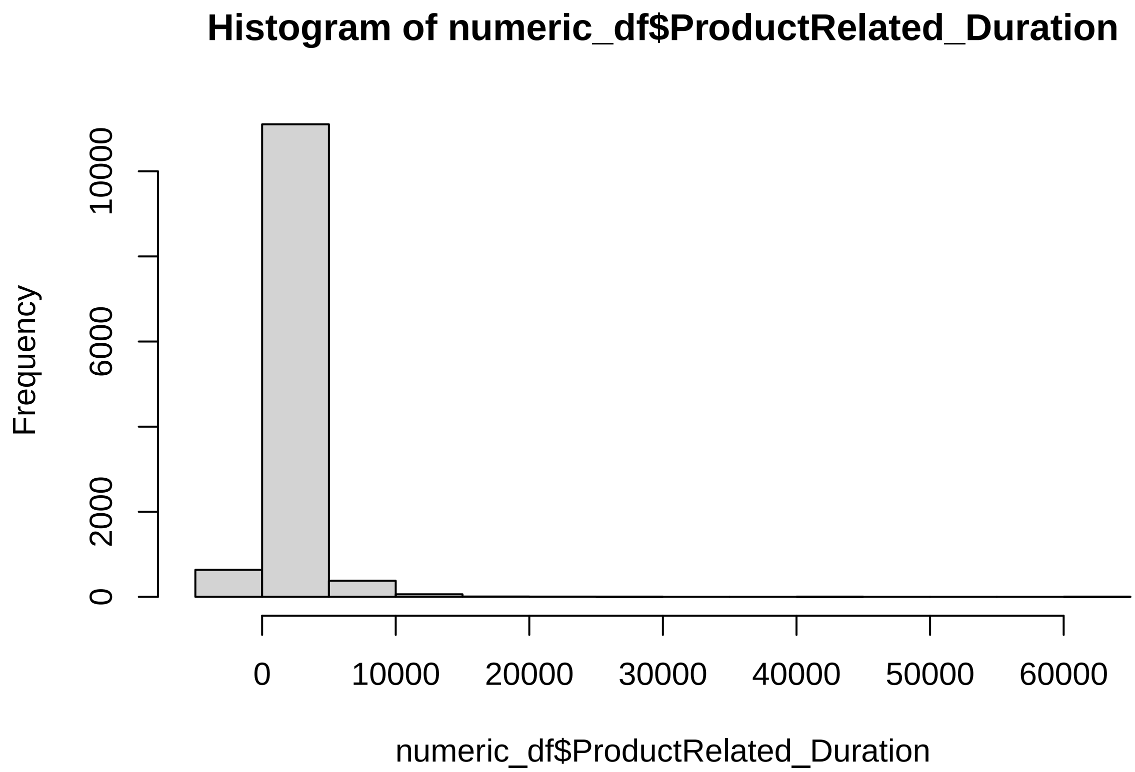


Most indivuals spend less time on product related sites.

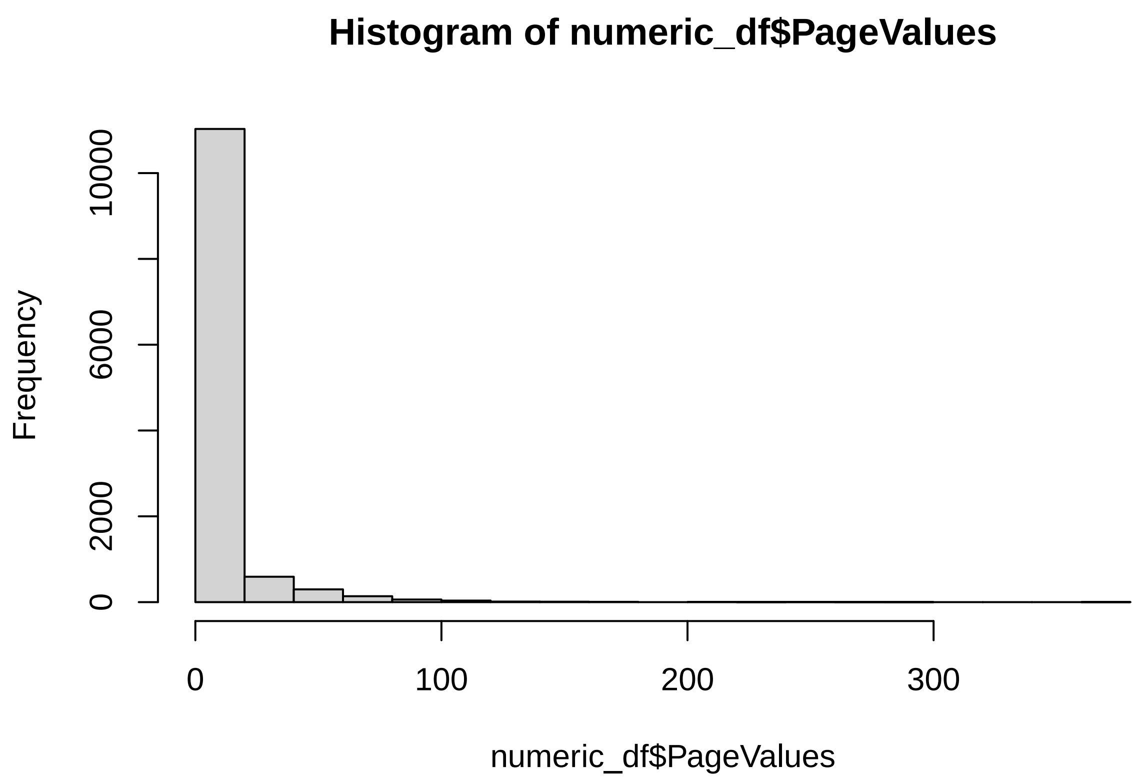
```r
# histogram of product related variable
hist(numeric_df$ProductRelated)
```




```
hist(numeric_df$ProductRelated_Duration)
```



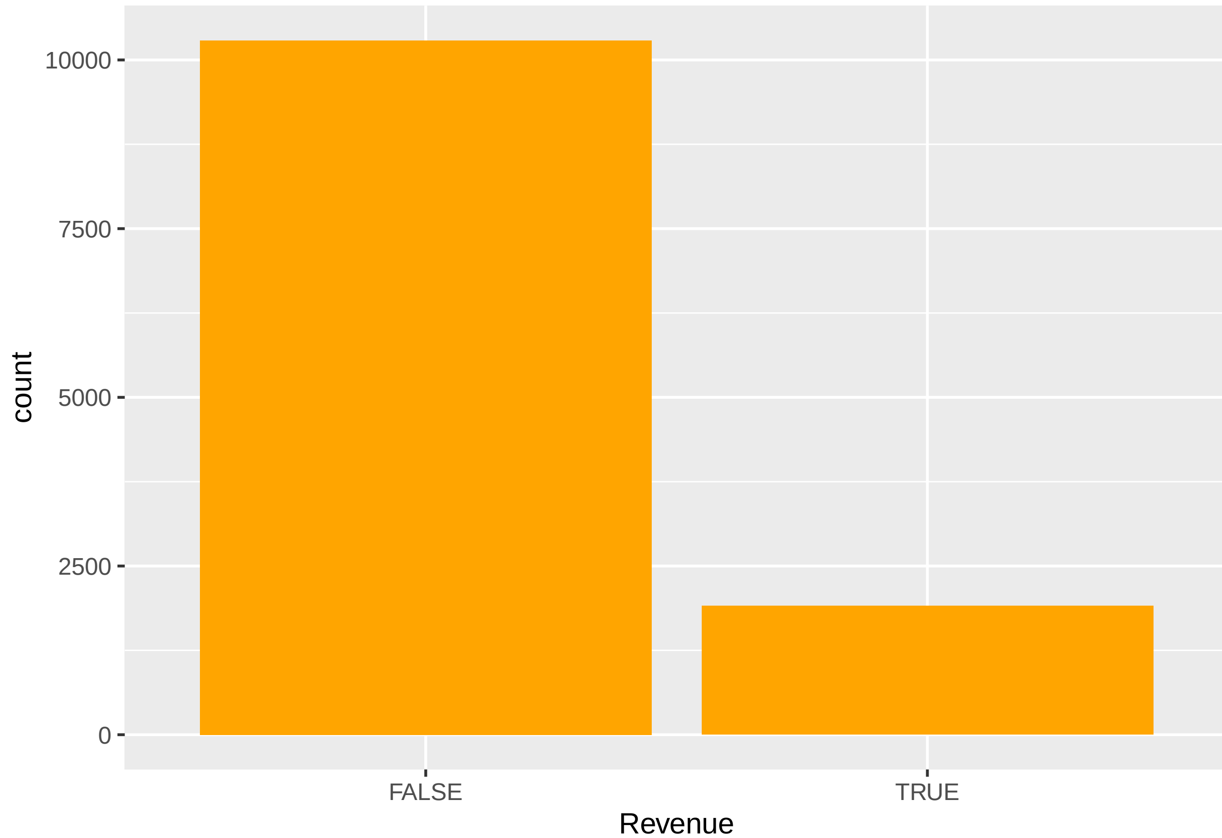

```r
# fetching the columns
hist(numeric_df$PageValues)
```



The lowest page value like value of 20 has verry high frequency compared to higher page values.

5.2 Bivariate analysis

```
#Plotting the number of customers who brought in revenues.
ggplot(customer_unique, aes(Revenue)) +
  geom_bar(fill = "orange")
```

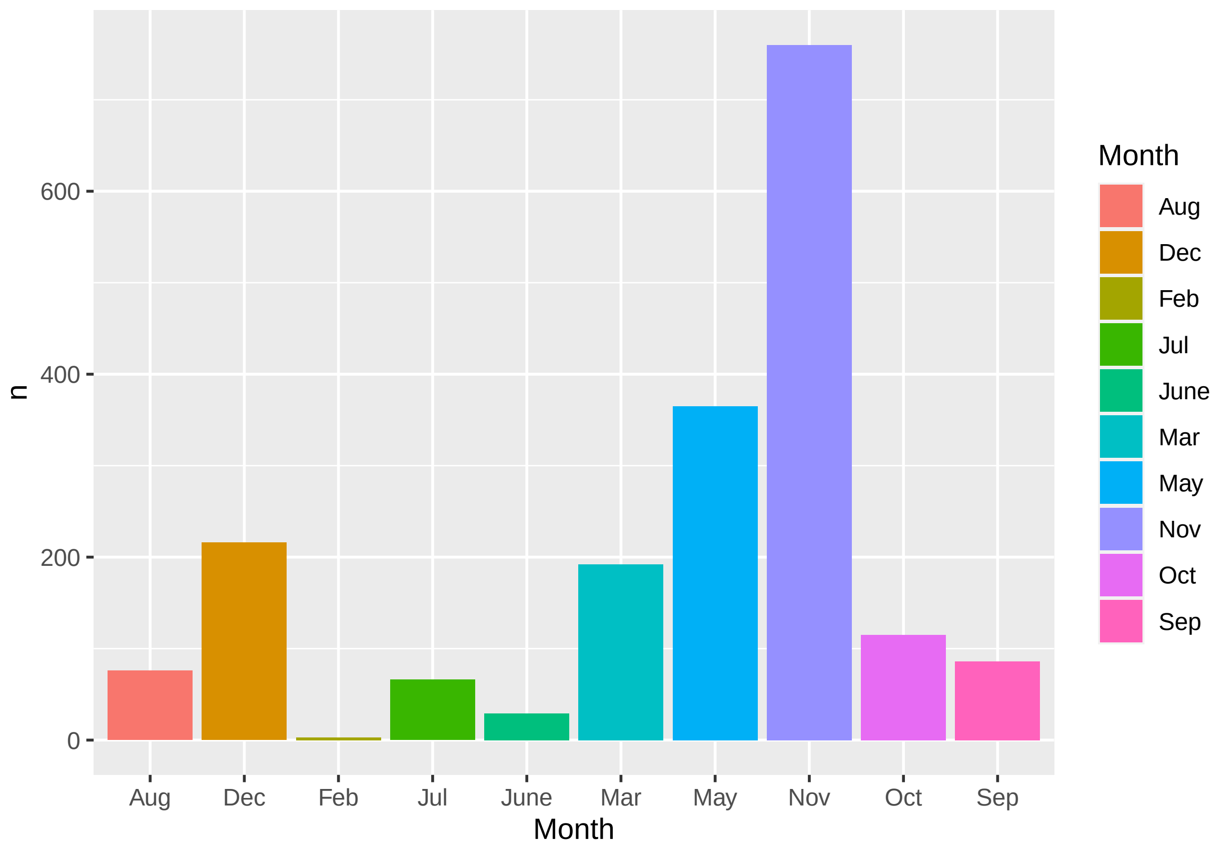


Very few customers brought in revenue

```
#changing the datatype of revenue to numeric
customer_unique$Revenue = as.character(customer_unique$Revenue)
customer_unique$Revenue <- recode(customer_unique$Revenue , 'TRUE' = 1, 'FALSE' = 0 )
```

```
#Grouping the month with the total number of persons who had revenue
month <- customer_unique %>%
  group_by(Month) %>%
  summarise(n=sum(Revenue, na.rm=TRUE)) %>%
  arrange(desc(n))%>%
  head(10)
```

```
#now ploting the months
m <- ggplot(month, aes(x = `Month`, y = n))

m + geom_col(aes(fill = `Month`))
```



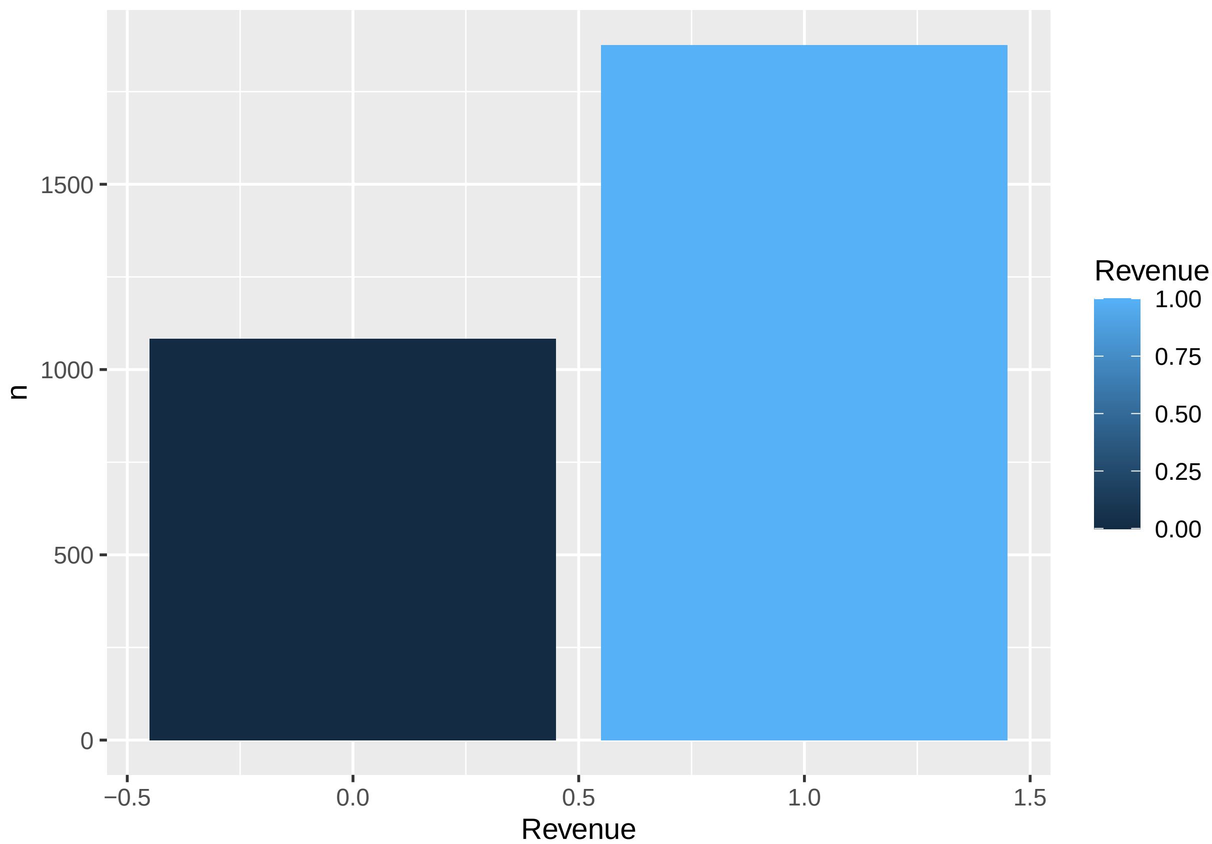
The following months had the most revenues:

1.November 2.December 3.May 4.March

The month of november has the most revenue collected, it might be there are alot of offers during that month.

```
#Grouping the mean number of product related duration by whether one brought in revenue or not.
product_related <- customer_unique %>%
  group_by(Revenue) %>%
  summarise(n=mean(ProductRelated_Duration, na.rm=TRUE)) %>%
  arrange(desc(n))%>%
  head(10)
```

```
#Viewing the results.
p <- ggplot(product_related, aes(x = `Revenue`, y = n))

p + geom_col(aes(fill = `Revenue`))
```

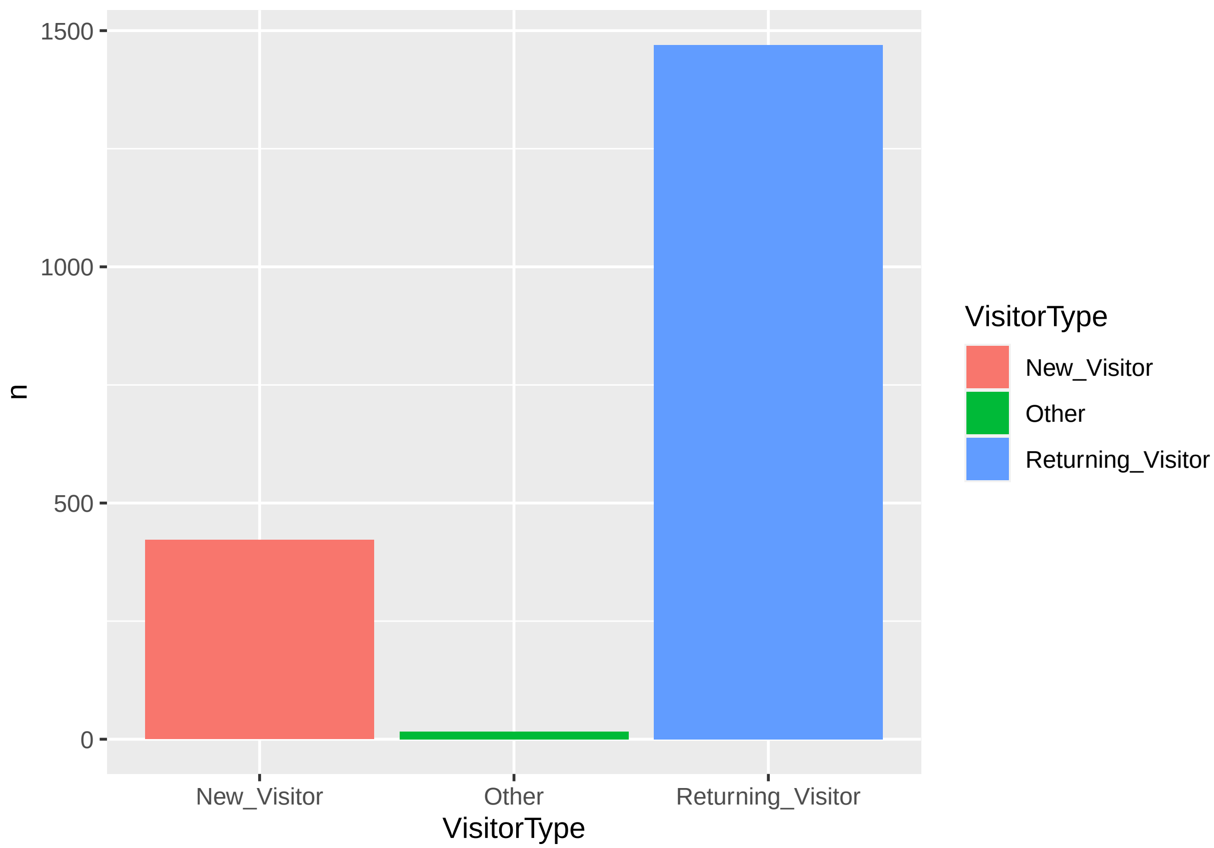



```
#  scale_fill_manual(values = c('yellow', 'Red'))
```

The more time spent on the product related pages the more likely that they will bring revenue.

```
#Grouping the visitor type by the revenues
visitor <- customer_unique %>%
  group_by(VisitorType) %>%
  summarise(n=sum(Revenue, na.rm=TRUE)) %>%
  arrange(desc(n))%>%
  head(10)
```

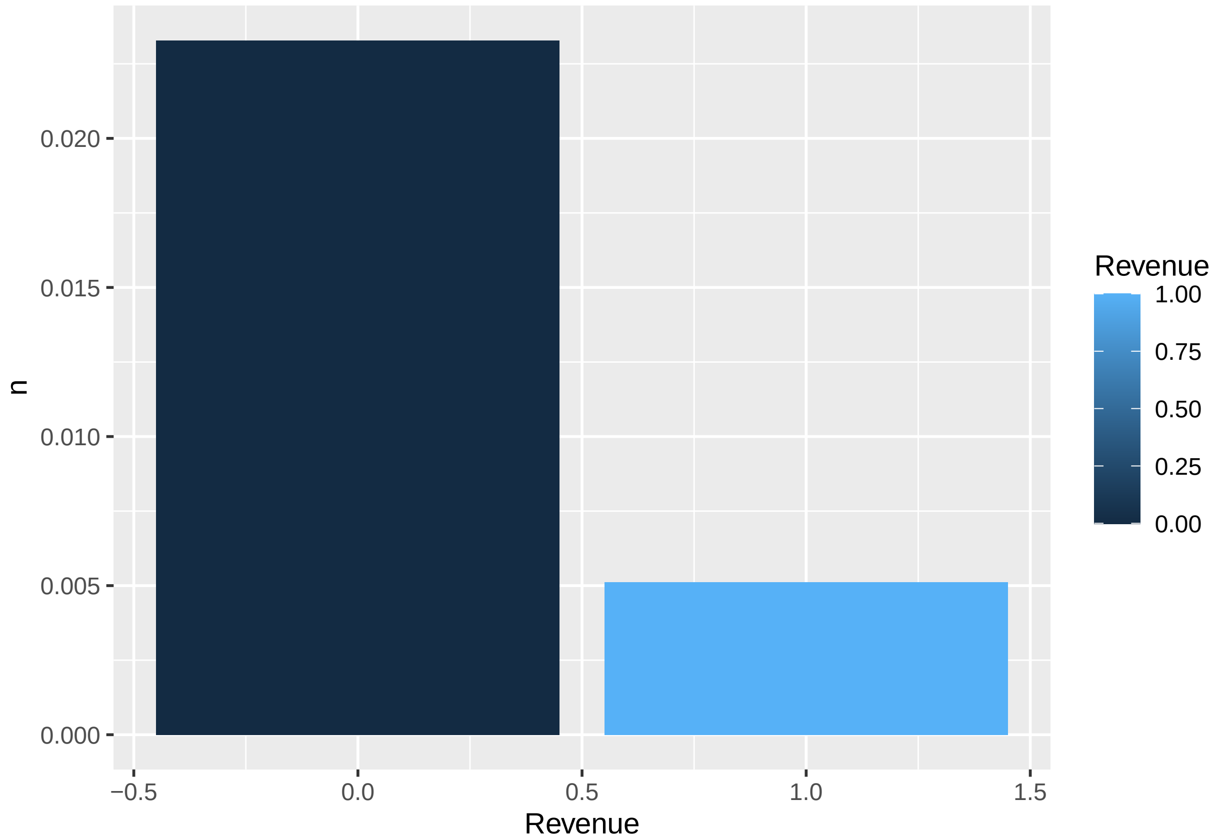
```
#Viewing the results of the visitor type
V <- ggplot(visitor, aes(x = `VisitorType`, y = n))

V + geom_col(aes(fill = `VisitorType`))
```



A returning visitor is more likely to purchase the product

```
#Grouping the mean bounce rate by the earning of revenue
bounce_rate <- customer_unique %>%
  group_by(Revenue) %>%
  summarise(n=mean(BounceRates, na.rm=TRUE)) %>%
  arrange(desc(n))%>%
  head(10)
```

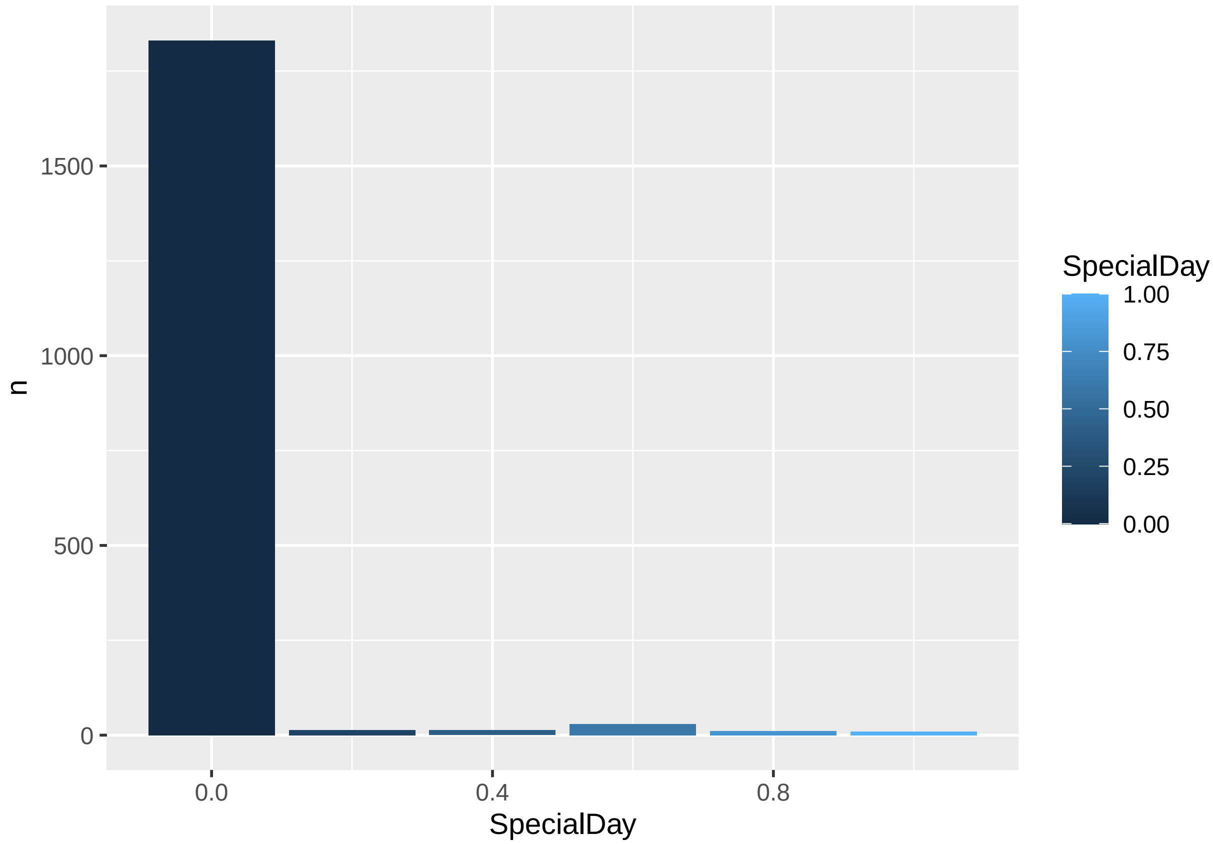
```
#Viewing the results.
c <- ggplot(bounce_rate, aes(x = `Revenue`, y = n))

c + geom_col(aes(fill = `Revenue`))
```



The mean bouncing rate for an individual who does not bring in revenue is higher compared to the one who brings in revenue.

```
#Grouping the special days by the number of generated revenues
special_day <- customer_unique %>%
  group_by(SpecialDay) %>%
  summarise(n=sum(Revenue, na.rm=TRUE)) %>%
  arrange(desc(n))%>%
  head(6)
```

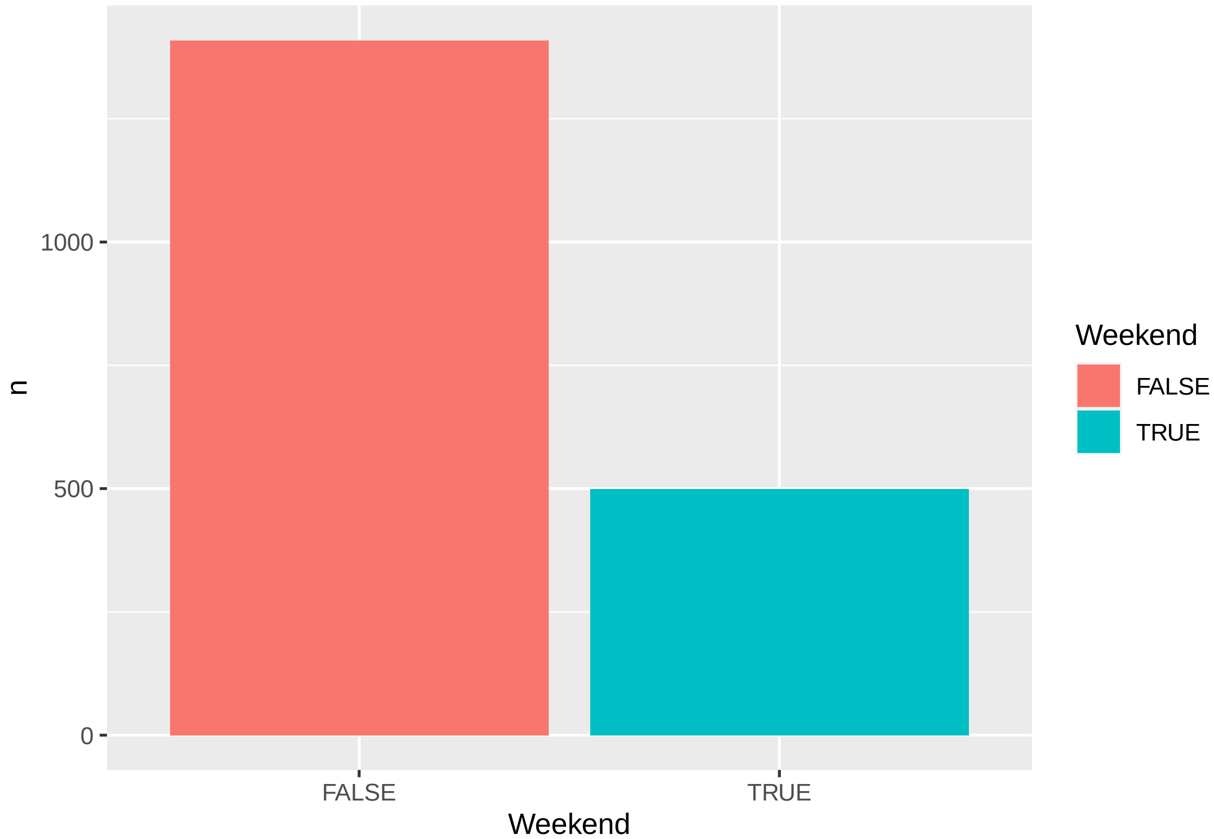
```
#Viewing the results.
c <- ggplot(special_day, aes(x = `SpecialDay`, y = n))

c + geom_col(aes(fill = `SpecialDay`))
```



There is no relationship between the special days and the amount of revenue generated.

```
#Grouping the weekends by the number of Revenues generated
weekend <- customer_unique %>%
  group_by(Weekend) %>%
  summarise(n=sum(Revenue, na.rm=TRUE))
```

```
#Viewing the results.
w <- ggplot(weekend, aes(x = `Weekend`, y = n))

w + geom_col(aes(fill = `Weekend`))
```



The most number of revenues was generated during weekdays.

```
#Printing out correlations in our dataset
cols <-cor(numeric_df)
cols
```

```
##                         Administrative Administrative_Duration Informational
## Administrative             1.000000000             0.600409653    0.37528761
## Administrative_Duration    0.600409653             1.000000000    0.30143630
## Informational              0.375287611             0.301436296    1.00000000
## Informational_Duration     0.254786021             0.237189860    0.61867795
## ProductRelated             0.428191515             0.286783914    0.37260472
## ProductRelated_Duration    0.371027224             0.353513793    0.38608372
## BounceRates               -0.213666635            -0.137333397   -0.10950530
## ExitRates                 -0.311274132            -0.202024452   -0.15956681
## PageValues                 0.096920968             0.066168365    0.04739015
## SpecialDay                -0.097072098            -0.074736885   -0.04937677
## OperatingSystems          -0.006697922            -0.007610715   -0.00962587
## Browser                   -0.025763658            -0.015833675   -0.03876681
## Region                    -0.007262053            -0.006723711   -0.03047732
## TrafficType               -0.034784126            -0.015075015   -0.03518669
##                         Informational_Duration ProductRelated
## Administrative                     0.254786021    0.428191515
## Administrative_Duration            0.237189860    0.286783914
## Informational                      0.618677947    0.372604721
## Informational_Duration             1.000000000    0.279061948
## ProductRelated                     0.279061948    1.000000000
```

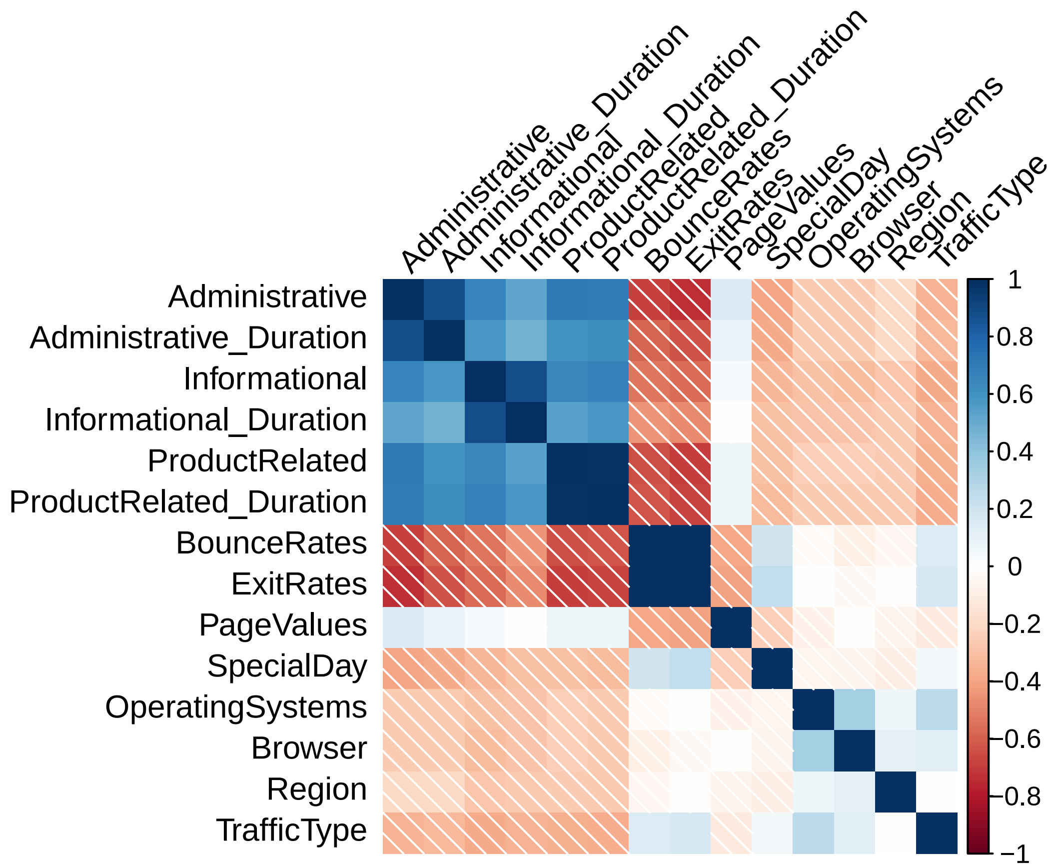
```
## ProductRelated_Duration           0.346580691    0.860308186
## BounceRates                      -0.070159472   -0.193515772
## ExitRates                        -0.102932678   -0.286163211
## PageValues                        0.030064160    0.054115494
## SpecialDay                       -0.031293040   -0.025930622
## OperatingSystems                 -0.009749983    0.004090351
## Browser                          -0.019609349   -0.013706213
## Region                           -0.027920098   -0.040106501
## TrafficType                      -0.025163571   -0.044344333
##                         ProductRelated_Duration  BounceRates    ExitRates
## Administrative                      0.371027224 -0.213666635 -0.311274132
## Administrative_Duration             0.353513793 -0.137333397 -0.202024452
## Informational                       0.386083717 -0.109505298 -0.159566815
## Informational_Duration              0.346580691 -0.070159472 -0.102932678
## ProductRelated                      0.860308186 -0.193515772 -0.286163211
## ProductRelated_Duration             1.000000000 -0.174375499 -0.245334012
## BounceRates                        -0.174375499  1.000000000  0.903358192
## ExitRates                          -0.245334012  0.903358192  1.000000000
## PageValues                          0.050840624 -0.115991977 -0.173571542
## SpecialDay                         -0.038210652  0.087839995  0.116783762
## OperatingSystems                    0.002775788  0.026839839  0.016482012
## Browser                            -0.007838332 -0.016018380 -0.003565541
## Region                             -0.034862498  0.001432015 -0.001837556
## TrafficType                        -0.037506944  0.089199039  0.087386232
##                          PageValues   SpecialDay OperatingSystems      Browser
## Administrative           0.09692097 -0.097072098     -0.006697922 -0.025763658
## Administrative_Duration  0.06616837 -0.074736885     -0.007610715 -0.015833675
## Informational            0.04739015 -0.049376774     -0.009625870 -0.038766808
## Informational_Duration   0.03006416 -0.031293040     -0.009749983 -0.019609349
## ProductRelated           0.05411549 -0.025930622      0.004090351 -0.013706213
## ProductRelated_Duration  0.05084062 -0.038210652      0.002775788 -0.007838332
## BounceRates             -0.11599198  0.087839995      0.026839839 -0.016018380
## ExitRates               -0.17357154  0.116783762      0.016482012 -0.003565541
## PageValues               1.00000000 -0.064532709      0.018583782  0.045845065
## SpecialDay              -0.06453271  1.000000000      0.012757766  0.003465984
## OperatingSystems         0.01858378  0.012757766      1.000000000  0.212244823
## Browser                  0.04584506  0.003465984      0.212244823  1.000000000
## Region                   0.01059087 -0.016452464      0.071953240  0.091889464
## TrafficType              0.01223694  0.052827944      0.182874100  0.102886237
##                               Region TrafficType
## Administrative          -0.007262053 -0.03478413
## Administrative_Duration -0.006723711 -0.01507502
## Informational           -0.030477323 -0.03518669
## Informational_Duration  -0.027920098 -0.02516357
## ProductRelated          -0.040106501 -0.04434433
## ProductRelated_Duration -0.034862498 -0.03750694
## BounceRates              0.001432015  0.08919904
## ExitRates               -0.001837556  0.08738623
## PageValues               0.010590868  0.01223694
## SpecialDay              -0.016452464  0.05282794
## OperatingSystems         0.071953240  0.18287410
## Browser                  0.091889464  0.10288624
## Region                   1.000000000  0.04252523
## TrafficType              0.042525234  1.00000000
```

```r
#Importing the library to do the correlation plot
#install.packages("corrplot",dependencies=TRUE)
```

```r
#Loading the corrplot in our google colab
library("corrplot")
```

```
## corrplot 0.90 loaded
```

```r
#Printing out the correlation plot
corrplot(cor(cols), method="shade", tl.col="black", tl.srt=45)
```



There is an evident of positive correlation between the following columns:

Administrative Administrative Duration Informational Informational Duration ProductRelated ProductRelated_Duration

The following columns are negatively linear:

BounceRates ExitRates.

5.3 Multivariate analysis

```r
#Factorizing categorical variables in our dataset.
customer_unique$VisitorType <- as.integer(as.factor(customer_unique$VisitorType))
customer_unique$Month <- as.integer(as.factor(customer_unique$Month))
customer_unique$Weekend <- as.integer(as.factor(customer_unique$Weekend))
```

```r
# previewing the datatypes of the dataset and check if the data types have changed.
sapply(customer_unique, class)
```

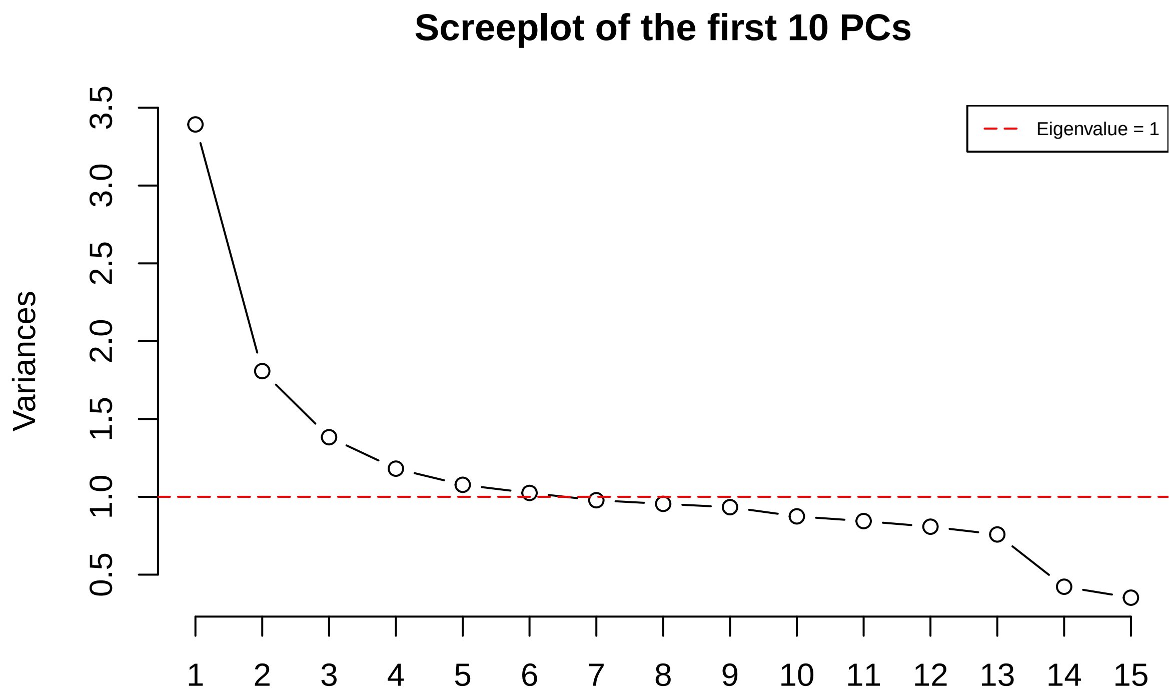
```
##          Administrative Administrative_Duration           Informational
##               "integer"               "numeric"               "integer"
##  Informational_Duration          ProductRelated ProductRelated_Duration
##               "numeric"               "integer"               "numeric"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

```r
#Using the principal component analysis to check for component variance.
customer.pca <- prcomp(customer_unique[,c(1:17)], center = TRUE, scale. = TRUE)
summary(customer.pca)
```

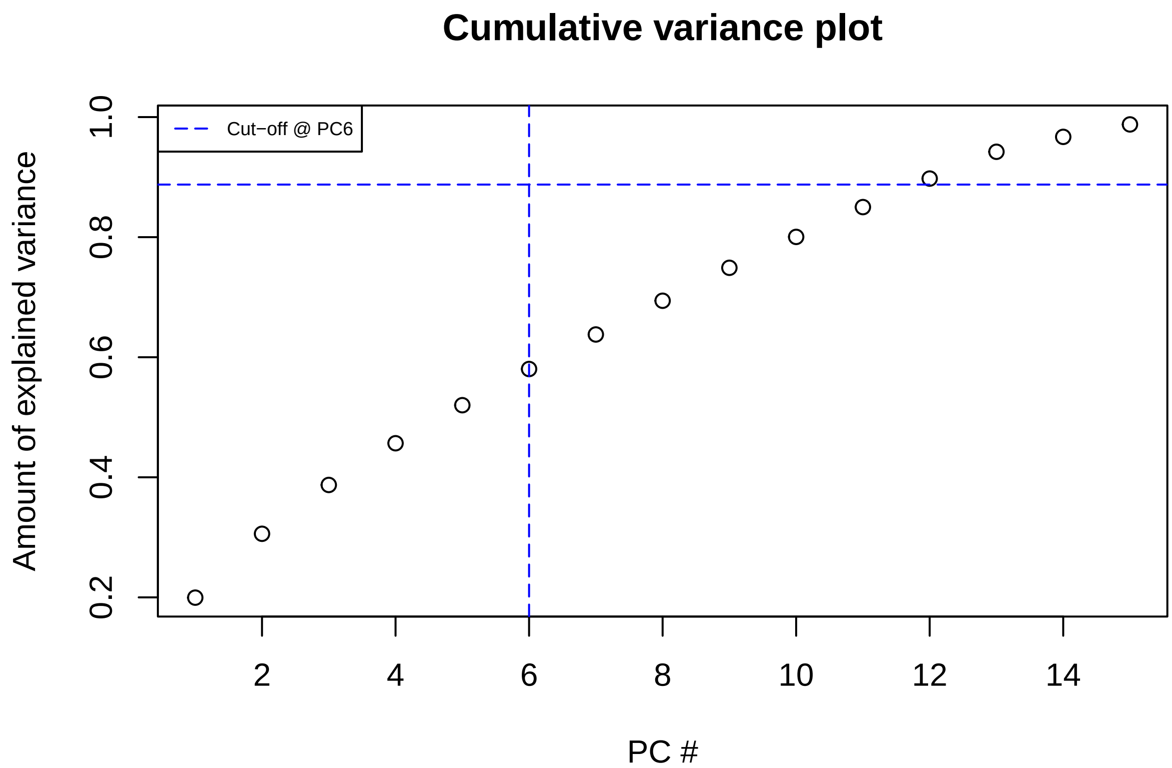
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.8419 1.3445 1.17602 1.08676 1.03789 1.01238 0.98900
## Proportion of Variance 0.1996 0.1063 0.08135 0.06947 0.06337 0.06029 0.05754
## Cumulative Proportion  0.1996 0.3059 0.38725 0.45672 0.52009 0.58038 0.63791
##                            PC8     PC9    PC10    PC11    PC12   PC13    PC14
## Standard deviation     0.97717 0.96615 0.93509 0.91878 0.89899 0.8707 0.64989
## Proportion of Variance 0.05617 0.05491 0.05143 0.04966 0.04754 0.0446 0.02484
## Cumulative Proportion  0.69408 0.74899 0.80042 0.85008 0.89762 0.9422 0.96706
##                           PC15    PC16    PC17
## Standard deviation     0.59337 0.35182 0.28991
## Proportion of Variance 0.02071 0.00728 0.00494
## Cumulative Proportion  0.98778 0.99506 1.00000
```

Proportion of Variance: This is the amount of variance the component accounts for in the data, ie PC1 accounts for 19% of total variance in the data alone! Cumulative Proportion: This is simply the accumulated amount of explained variance, ie. if we used the first 10 components we would be able to account for 80% of total variance in the data.

```r
screeplot(customer.pca, type = "l", npcs = 15, main = "Screeplot of the first 10 PCs")
abline(h = 1, col="red", lty=5)
legend("topright", legend=c("Eigenvalue = 1"),
       col=c("red"), lty=5, cex=0.6)
```




```
cumpro <- cumsum(customer.pca$sdev^2 / sum(customer.pca$sdev^2))
plot(cumpro[0:15], xlab = "PC #", ylab = "Amount of explained variance", main = "Cumulative variance pl
abline(v = 6, col="blue", lty=5)
abline(h = 0.88759, col="blue", lty=5)
legend("topleft", legend=c("Cut-off @ PC6"),
       col=c("blue"), lty=5, cex=0.6)
```



We notice that the first 6 components has an Eigenvalue >1 and explains almost 60% of variance. so we will use the first 6 variables in our analysis.

6. Implement the Solution

6.1 K-means Clustering

```
#Separating the response variables and the class variable.
customer.new<- customer_unique[, c(1:6)]
customer.class<- customer_unique[, "Revenue"]
head(customer.new)
```

```
##   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
## 1              1                0.000000
## 2              2               64.000000
## 3              1               -1.000000
```

```
## 4              2                2.666667
## 5             10              627.500000
## 6             19              154.216667
```

```
head(customer.class)
```

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

```
#Normalizing our continuous variables.
normalize <- function(x){
  return ((x-min(x)) / (max(x)-min(x)))
}
customer.new$Administrative<- normalize(customer.new$Administrative)
customer.new$Administrative_Duration<- normalize(customer.new$Administrative_Duration)
customer.new$ProductRelated<- normalize(customer.new$ProductRelated)
customer.new$ProductRelated_Duration<- normalize(customer.new$ProductRelated_Duration)
customer.new$Informational<- normalize(customer.new$Informational)
customer.new$Informational_Duration<- normalize(customer.new$Informational_Duration)
head(customer.new)
```

```
##   Administrative Administrative_Duration Informational Informational_Duration
## 1              0            0.0002941393             0           0.0003920992
## 2              0            0.0002941393             0           0.0003920992
## 3              0            0.0000000000             0           0.0000000000
## 4              0            0.0002941393             0           0.0003920992
## 5              0            0.0002941393             0           0.0003920992
## 6              0            0.0002941393             0           0.0003920992
##   ProductRelated ProductRelated_Duration
## 1    0.001418440            1.563122e-05
## 2    0.002836879            1.016029e-03
## 3    0.001418440            0.000000e+00
## 4    0.002836879            5.731448e-05
## 5    0.014184397            9.824223e-03
## 6    0.026950355            2.426226e-03
```

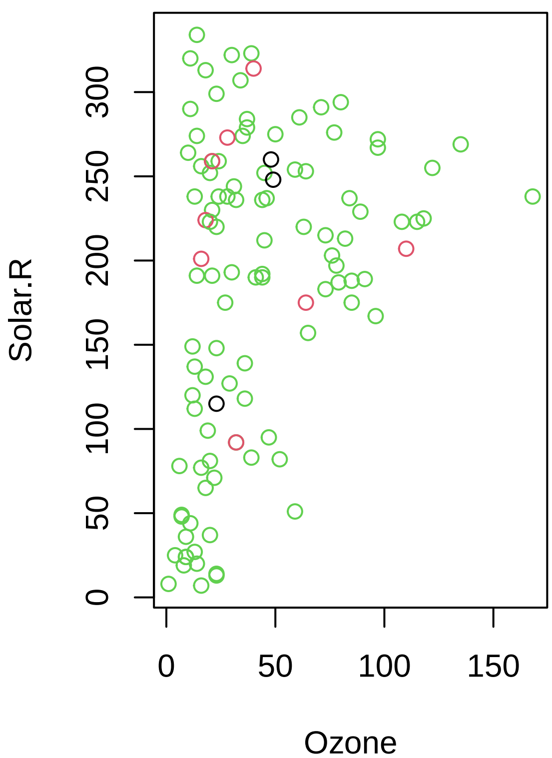
```
# Applying the K-means clustering algorithm with no. of centroids(k)=3
# ---
#
result<- kmeans(customer.new,3)

# Previewing the no. of records in each cluster
#
result$size
```

```
## [1] 1001 3258 7940
```

```
# Getting the value of cluster center datapoint value(3 centers for k=3)
# ---
#
result$centers
```

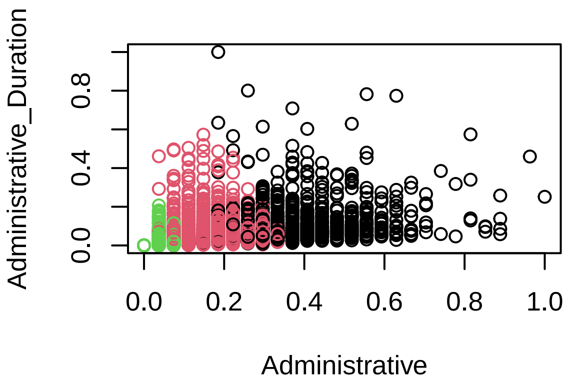
```
##   Administrative Administrative_Duration Informational Informational_Duration
## 1     0.38805639             0.108557639    0.09565435            0.077862078
## 2     0.16803083             0.046259705    0.02889042            0.016117333
## 3     0.01528594             0.004697754    0.00865869            0.005159646
##   ProductRelated ProductRelated_Duration
## 1     0.13769635              0.05828508
## 2     0.05635316              0.02301451
## 3     0.02938189              0.01223175
```

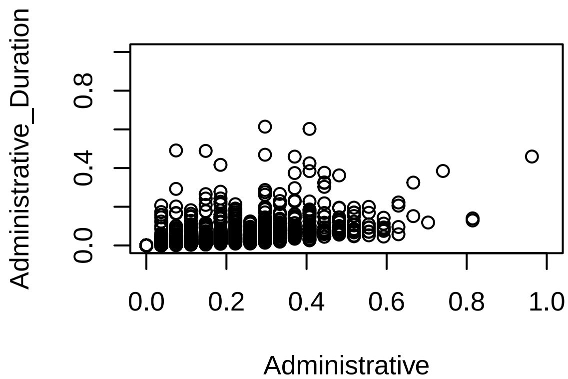
```
# Visualizing the  clustering results
# ---
#
par(mfrow = c(1,2), mar = c(5,4,2,2))

# Plotting to see how Ozone and Solar.R data points have been distributed in clusters
# ---
#
plot(airquality[,1:2], col = result$cluster)
```



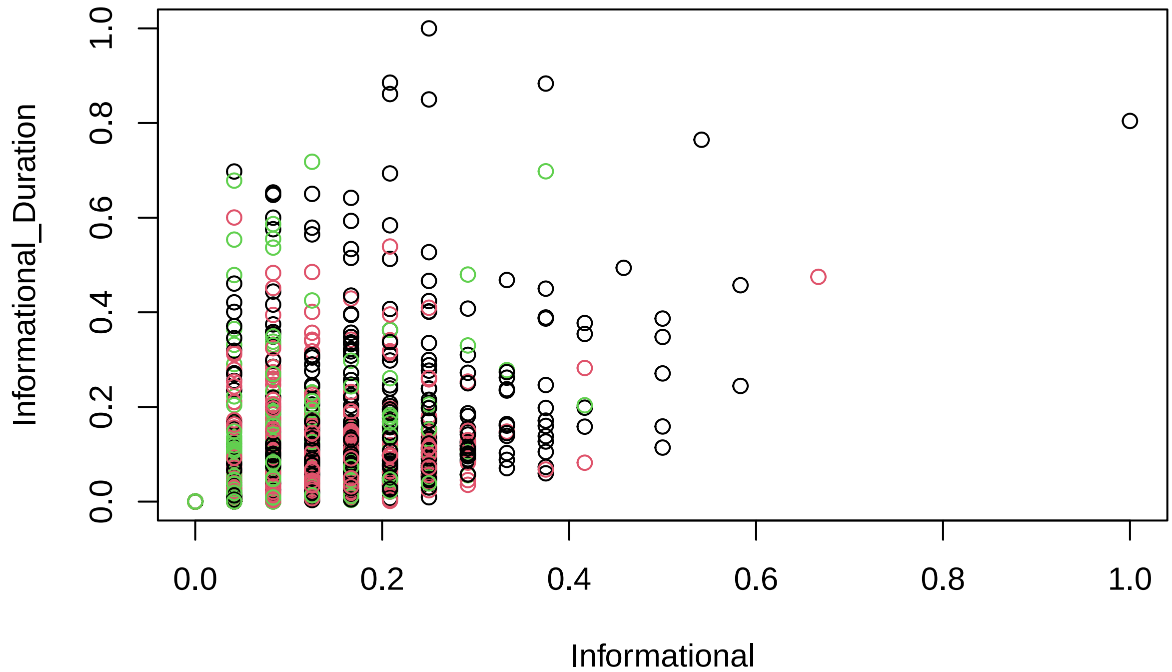

```
# Verifying the results of clustering
# ---
#
par(mfrow = c(2,2), mar = c(5,4,2,2))

# Plotting to see how administrative and administrative_duration data points have been distributed in c
```

```
plot(customer.new[c(1,2)], col = result$cluster)

# Plotting to see how administrative and administrative_duration  data points have been distributed
# originally as per "class" attribute in dataset
# ---
#
plot(customer.new[c(1,2)], col = customer.class)
```

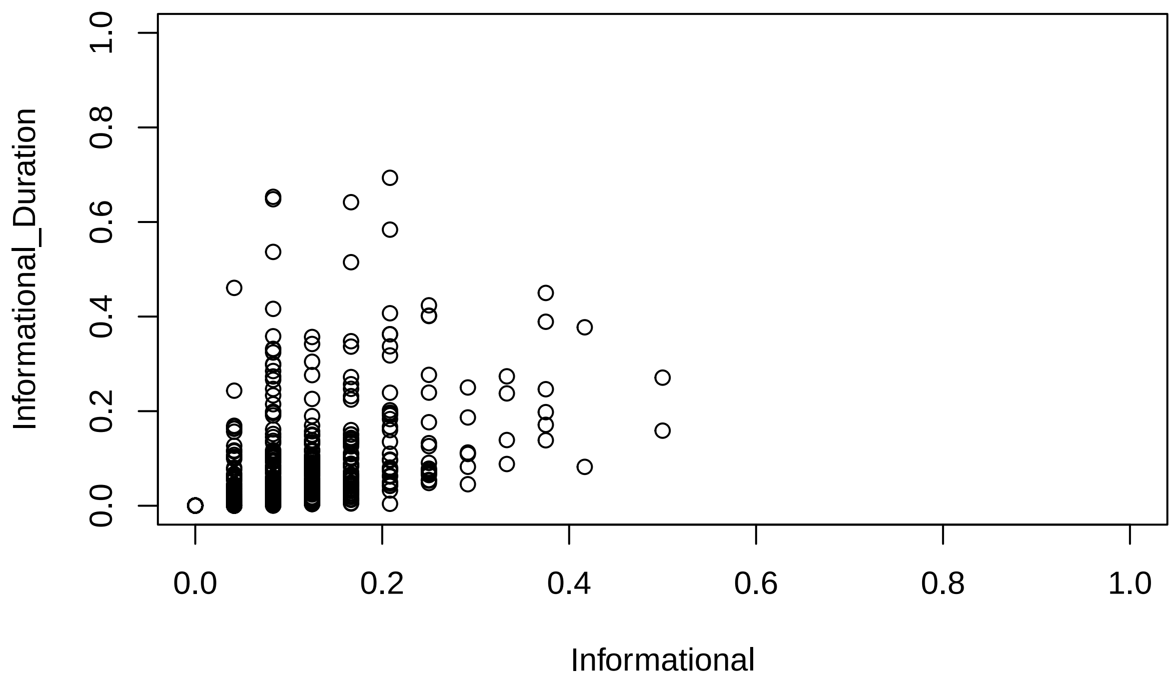




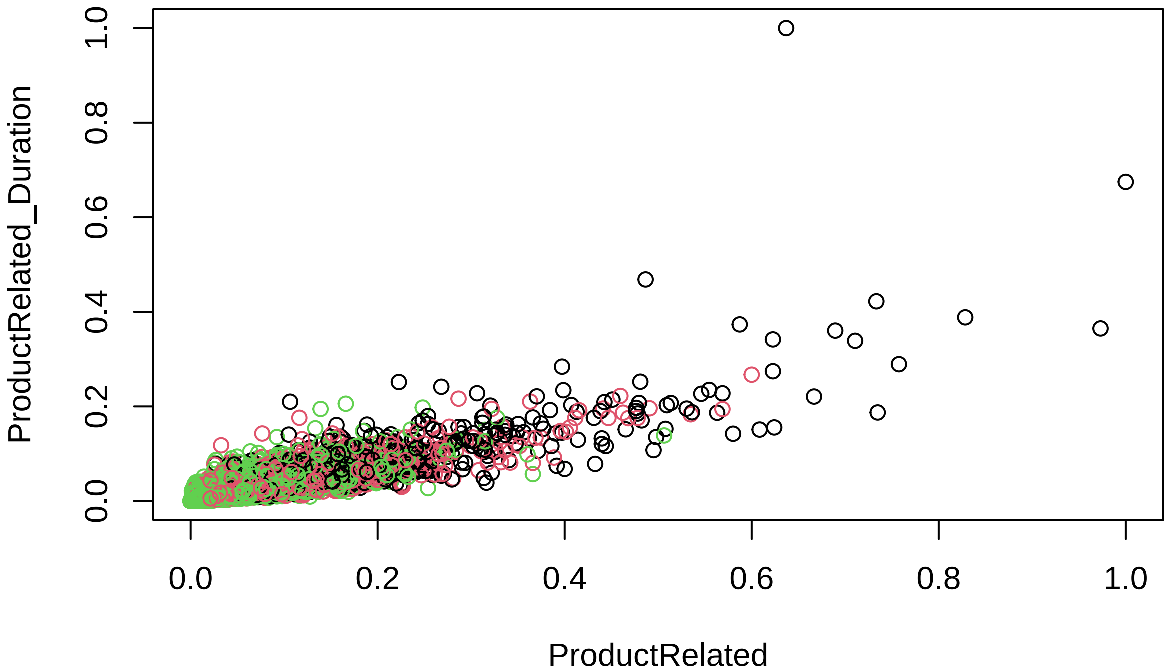

```
# Plotting to see how informational and informational_duration data points have been distributed in clu
# ---
#
plot(customer.new[c(3,4)], col = result$cluster)
```



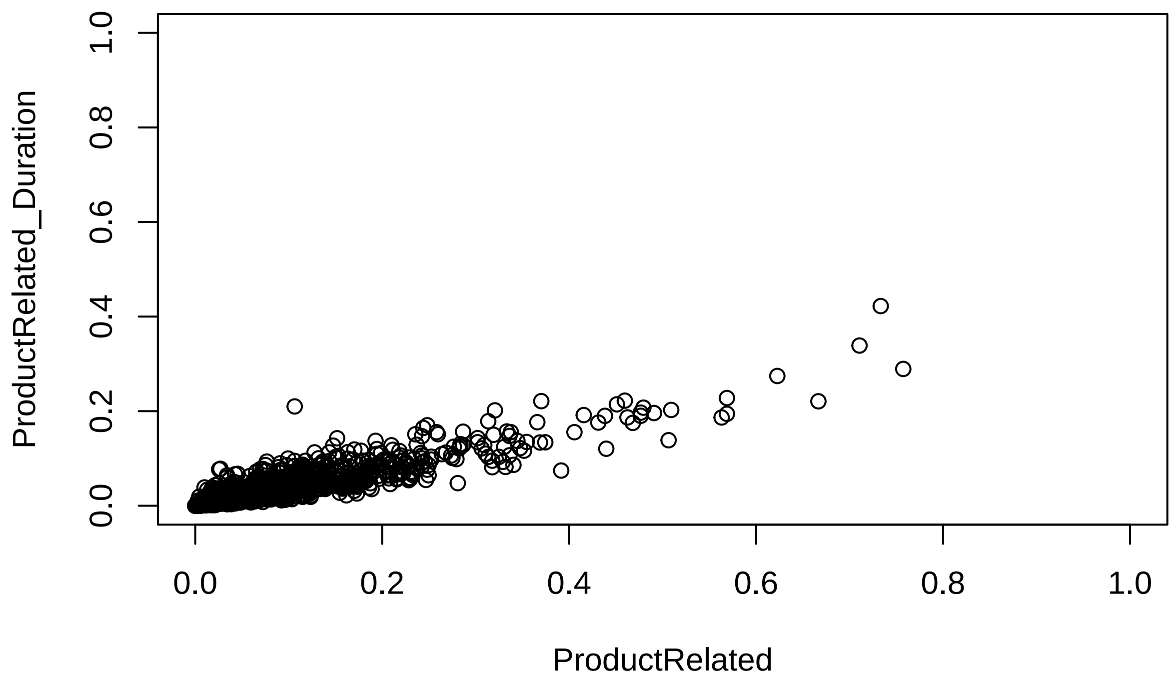

```
plot(customer.new[c(3,4)], col = customer.class)
```




```
# Plotting to see how product reated and product related duration data points have been distributed in
# ---
#
plot(customer.new[c(5,6)], col = result$cluster)
```




```
plot(customer.new[c(5,6)], col = customer.class)
```




```
table(result$cluster, customer.class)
```

```
##    customer.class
##        0    1
##   1  727  274
##   2 2590  668
##   3 6974  966
```

The first cluster correctly classified 6974 values correctly and 966 incorrectly. The second cluster correctly classified 727 values correctly and 274 values incorrectly. The third cluster correctly classified 2590 values correctly and 668 values incorrectly.

6.2 Hierachical Clustering

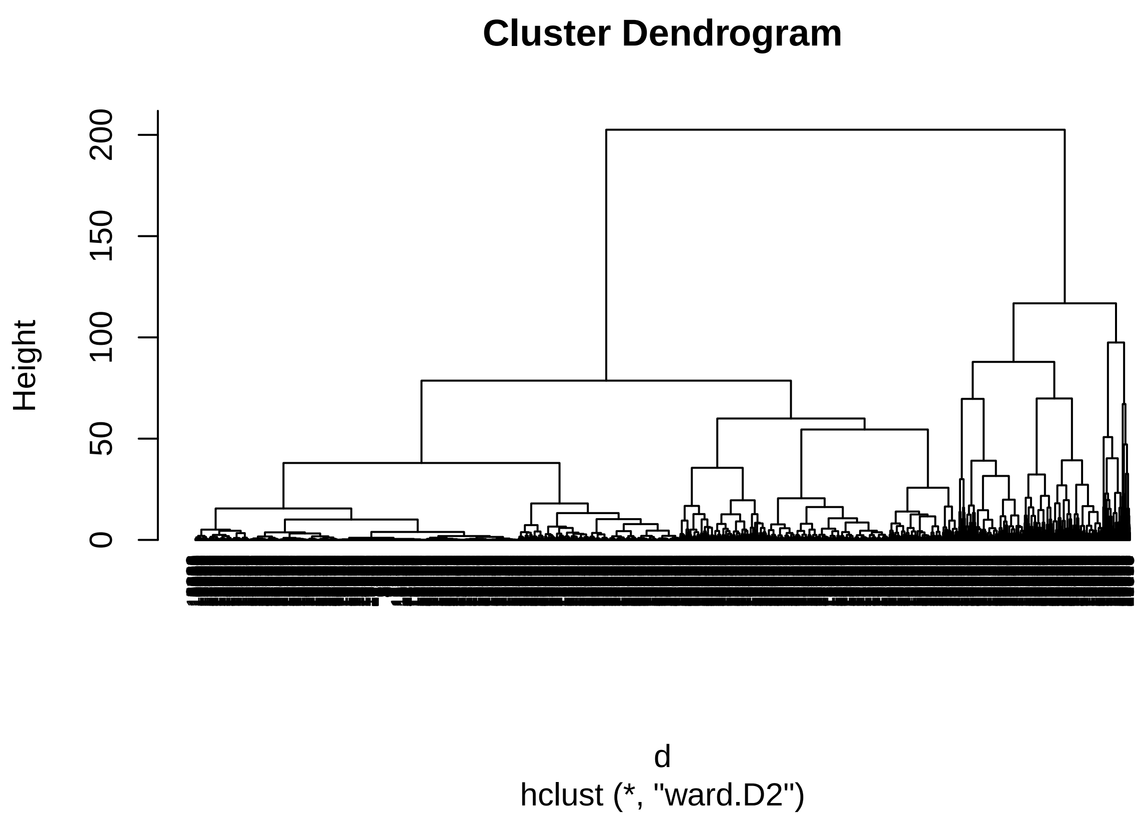
```
# we start by scaling the data using the R function scale() as follows

customer_h <- scale(customer_unique[, c(1:6)])
head(customer_h)
```

```
##   Administrative Administrative_Duration Informational Informational_Duration
## 1     -0.7025315              -0.4601081    -0.3988128             -0.2462725
## 2     -0.7025315              -0.4601081    -0.3988128             -0.2462725
## 3     -0.7025315              -0.4657410    -0.3988128             -0.2533417
```

```
## 4      -0.7025315                 -0.4601081    -0.3988128                -0.2462725
## 5      -0.7025315                 -0.4601081    -0.3988128                -0.2462725
## 6      -0.7025315                 -0.4601081    -0.3988128                -0.2462725
##   ProductRelated ProductRelated_Duration
## 1      -0.6963635              -0.6289343
## 2      -0.6739424              -0.5955997
## 3      -0.6963635              -0.6294551
## 4      -0.6739424              -0.6275453
## 5      -0.4945739              -0.3020990
## 6      -0.2927843              -0.5486101
```

```r
# We now use the R function hclust() for hierarchical clustering

d <- dist(customer_h, method = "euclidean")

# We then hierarchical clustering using the Ward's method

res.hc <- hclust(d, method = "ward.D2" )
```

```r
# Lastly, we plot the obtained dendrogram
# ---
#
plot(res.hc, cex = 0.6, hang = -1)
```



We were not really able to draw insights from the dendogram above.

7. Challenging the Solution

Our Hierachical Clustering Method did not perform as well even after performing feature reduction using PCA. This might have been caused by the high number of records that was in our dataset.