Clustering E-Commerce Customers

Exploratory Data Analysis

A. Defining the Question

A Kenyan entrepreneur has created an online cryptography course and would want to advertise it on her blog. She currently targets audiences originating from various countries. In the past, she ran ads to advertise a related course on the same blog and collected data in the process. My task as a Data Science Consultant is to to help her by creating a model that predicts which individuals are most likely to click on her ads

B. Metrics for Success 1. Plotting Clusters to visualise modelled clusters & noise points 2. Confusion Matrix to measure accuracy

C. Understanding the Context

It is imperative that all entrepreneurs identify all opportunities to advertise and boost their sales in whatever business they are engaged in. (See Problem Definition)

D. Recording the Experimental Design 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 using K-means & Hierarchical Clustering Algorithms 7.Challenge the Solution using DBSCAN Algorithm 8.Follow up Questions

1. Problem Definition

Kira Plastinina ("https://kiraplastinina.ru/") 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. The objectives of this project are as follows:

1. Learn the characteristics of customer groups
2. Perform clustering stating insights drawn from analysis and visualizations.
3. Upon implementation, provide comparisons between K-Means clustering vs Hierarchical clustering highlighting the strengths and limitations of each approach in the context of the analysis.

2. Data Sourcing

The dataset for this project can be found here : "http://bit.ly/EcommerceCustomersDataset"

3. Check the Data

```
# Loading the necessary libraries using pacman

pacman :: p_load(pacman, dplyr, tidyverse, GGally, ggplot2, ggthemes, ggvis, httr, lubridate,
  plotly, rio, rmarkdown, shiny, stringr, tidyr, psych, corrplot, caret, Amelia, mice)
```

```r
# Previewing first 10 records of our data

data <- import("~/R/K_Means & Hierarchichal Clustering of E-Commerce Customers/online_shoppers_

# converting data to a dataframe

data <- as.data.frame(data)
head(data,10)
```

```
##    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
## 7               0                      -1             0                     -1
## 8               1                      -1             0                     -1
## 9               0                       0             0                      0
## 10              0                       0             0                      0
##    ProductRelated ProductRelated_Duration BounceRates  ExitRates PageValues
## 1               1                0.000000  0.20000000 0.20000000          0
## 2               2               64.000000  0.00000000 0.10000000          0
## 3               1               -1.000000  0.20000000 0.20000000          0
## 4               2                2.666667  0.05000000 0.14000000          0
## 5              10              627.500000  0.02000000 0.05000000          0
## 6              19              154.216667  0.01578947 0.02456140          0
## 7               1               -1.000000  0.20000000 0.20000000          0
## 8               1               -1.000000  0.20000000 0.20000000          0
## 9               2               37.000000  0.00000000 0.10000000          0
## 10              3              738.000000  0.00000000 0.02222222          0
##    SpecialDay Month OperatingSystems Browser Region TrafficType
## 1         0.0   Feb                1       1      1           1
## 2         0.0   Feb                2       2      1           2
## 3         0.0   Feb                4       1      9           3
## 4         0.0   Feb                3       2      2           4
## 5         0.0   Feb                3       3      1           4
## 6         0.0   Feb                2       2      1           3
## 7         0.4   Feb                2       4      3           3
## 8         0.0   Feb                1       2      1           5
## 9         0.8   Feb                2       2      2           3
## 10        0.4   Feb                2       4      1           2
##          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
## 7  Returning_Visitor   FALSE   FALSE
## 8  Returning_Visitor    TRUE   FALSE
## 9  Returning_Visitor   FALSE   FALSE
## 10 Returning_Visitor   FALSE   FALSE
```

```r
# Checking the size and shape of data
dim(data)
```

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

```r
# Viewing data types using str().

str(data)
```

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

4. Perform Data Cleaning

```r
# Checking for missing data in our columns

colSums(is.na(data))
```

```
##          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 first 8 columns have 14 missing records. We’ll drop the 14 missing observations since they’re not many

```r
# Dropping missing observations, re-checking for missing records and checking new shape of dat

data <- na.omit(data)
colSums(is.na(data))
```

```
##          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
```

```
dim(data)
```

```
## [1] 12316    18
```

```
# Checking and dealing with duplicates by removing records that return TRUE
# for duplication and "returning visitors"
# This way, we'll only remove true duplicated records and not returning visitor records


no_duplicates <- data[!(duplicated(data) & data$VisitorType != "Returning_Visitor"),]
dim(no_duplicates)
```

```
## [1] 12311    18
```

```
# There were only 5 true duplicated records
```

```
# Changing all column names to lowercase
names(no_duplicates)[1:18] <- tolower(names(no_duplicates)[1:18])

# Renaming columns with long names
names(no_duplicates)[names(no_duplicates) == "administrative_duration"] <- "a_duration"
names(no_duplicates)[names(no_duplicates) == "informational_duration"] <- "i_duration"
names(no_duplicates)[names(no_duplicates) == "productrelated_duration"] <- "p_duration"

# new column names
colnames(no_duplicates)
```

```
##  [1] "administrative"   "a_duration"       "informational"    "i_duration"
##  [5] "productrelated"   "p_duration"       "bouncerates"      "exitrates"
##  [9] "pagevalues"       "specialday"       "month"            "operatingsystems"
## [13] "browser"          "region"           "traffictype"      "visitortype"
## [17] "weekend"          "revenue"
```

```r
# Converting integer columns to numeric

no_duplicates$administrative   <- as.numeric(no_duplicates$administrative)
no_duplicates$a_duration      <- as.numeric(no_duplicates$a_duration)

no_duplicates$informational   <- as.numeric(no_duplicates$informational)
no_duplicates$i_duration      <- as.numeric(no_duplicates$i_duration)

no_duplicates$productrelated   <- as.numeric(no_duplicates$productrelated)
no_duplicates$p_duration      <- as.numeric(no_duplicates$p_duration)

no_duplicates$specialday      <- as.numeric(no_duplicates$specialday)

no_duplicates$pagevalues      <- as.numeric(no_duplicates$pagevalues)
```

```r
# Factorising categorical columns

no_duplicates[,11:18] <- lapply(no_duplicates[,11:18], factor)  ## as.factor() could also be u
```

```r
# checking final data types by using colnames() function and creating a for loop for each colu

columns = colnames(no_duplicates)
for (column in seq(length(colnames(no_duplicates)))){
    print(columns[column])
    print(class(no_duplicates[, column]))
    cat('\n')
}
```

```
## [1] "administrative"
## [1] "numeric"
##
## [1] "a_duration"
## [1] "numeric"
##
## [1] "informational"
## [1] "numeric"
##
## [1] "i_duration"
## [1] "numeric"
##
## [1] "productrelated"
## [1] "numeric"
##
## [1] "p_duration"
## [1] "numeric"
##
## [1] "bouncerates"
## [1] "numeric"
##
## [1] "exitrates"
## [1] "numeric"
##
## [1] "pagevalues"
## [1] "numeric"
##
## [1] "specialday"
## [1] "numeric"
##
## [1] "month"
## [1] "factor"
##
## [1] "operatingsystems"
## [1] "factor"
##
## [1] "browser"
## [1] "factor"
##
## [1] "region"
## [1] "factor"
##
## [1] "traffictype"
## [1] "factor"
##
## [1] "visitortype"
## [1] "factor"
##
## [1] "weekend"
## [1] "factor"
##
## [1] "revenue"
## [1] "factor"
```

```
# Checking for Anomalies using levels() function
# checking unique values in month
levels(no_duplicates$month)
```

```
##  [1] "Aug"  "Dec"  "Feb"  "Jul"  "June" "Mar"  "May"  "Nov"  "Oct"  "Sep"
```

```r
# Checking Anomalies
# checking unique values in opertaingsystems
levels(no_duplicates$operatingsystems)
```

```
## [1] "1" "2" "3" "4" "5" "6" "7" "8"
```

```r
# Checking Anomalies
# checking unique values in browser
levels(no_duplicates$browser)
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13"
```

```r
# Checking Anomalies
# checking unique values in region
levels(no_duplicates$region)
```

```
## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9"
```

```r
# Checking Anomalies
# checking unique values in traffictype
levels(no_duplicates$traffictype)
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15"
## [16] "16" "17" "18" "19" "20"
```

```r
# Checking Anomalies
# checking unique values in visitortype
levels(no_duplicates$visitortype)
```

```
## [1] "New_Visitor"       "Other"             "Returning_Visitor"
```

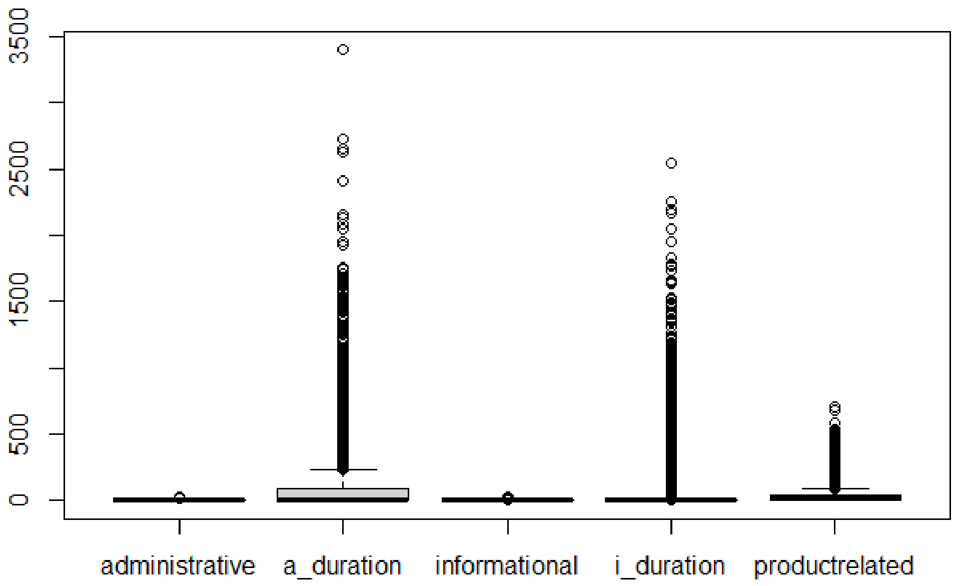
```r
# Checking Anomalies
# checking unique values in weekend
levels(no_duplicates$weekend)
```

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

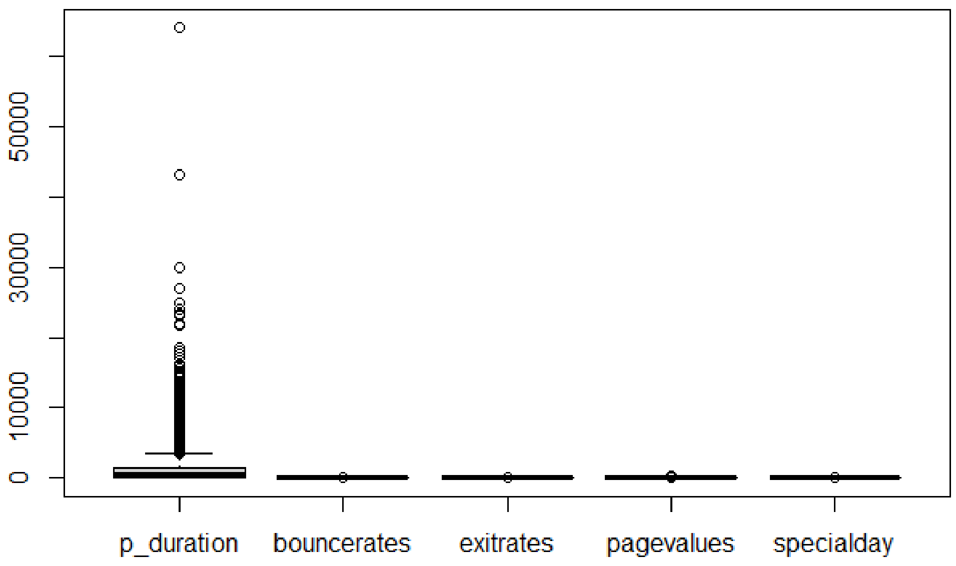
```r
# Checking Anomalies
# checking unique values in revenue
levels(no_duplicates$revenue)
```

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

```
# Checking For Outliers in Numerical Columns

# splitting numerical columns to half for better viewing of boxplots
num_cols1 <- no_duplicates[,1:5]
num_cols2 <- no_duplicates[,6:10]
boxplot(num_cols1)
```

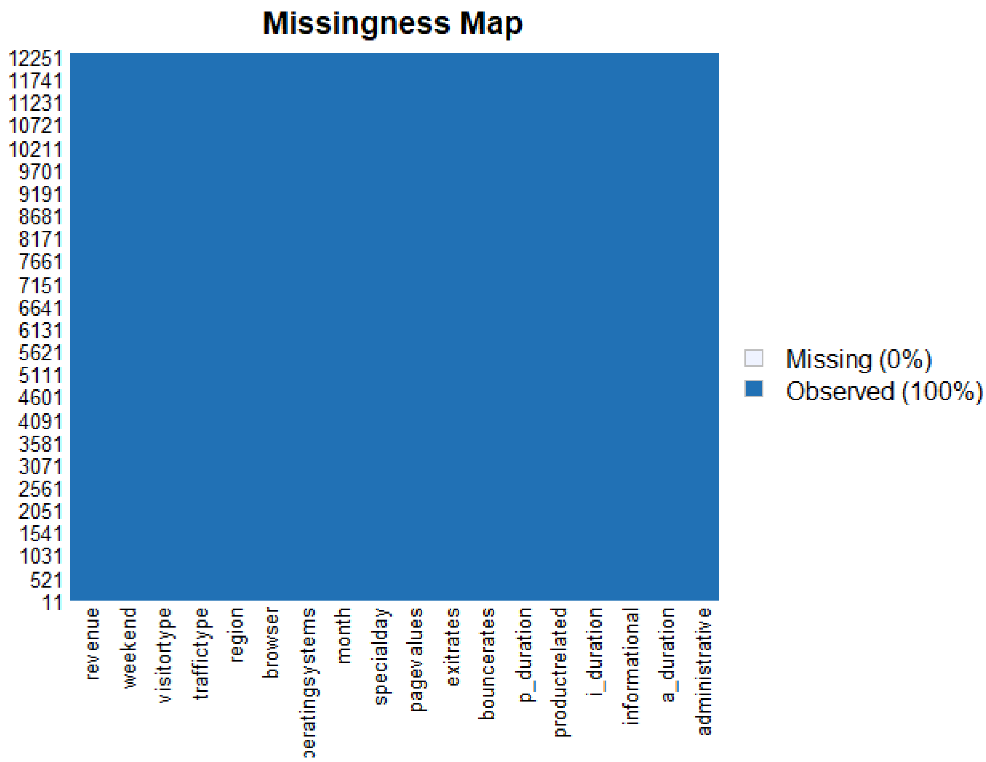



```
boxplot(num_cols2)
```



```
# From the boxplots, a_duration, i_duration, productrelated and p_duration have pronounced out

# The values of these features are derived from the URL information of the pages visited by th
```

```
# Checking for missing records
final <- no_duplicates
missmap(final)
```



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

Univariate

```
# Frequency tables

# Obtaining a table of product pages and time duration on each page

time_on_prod_page <- final %>%
        group_by(productrelated) %>% summarise_at(vars(p_duration,bouncerates,exitrates),fun
```

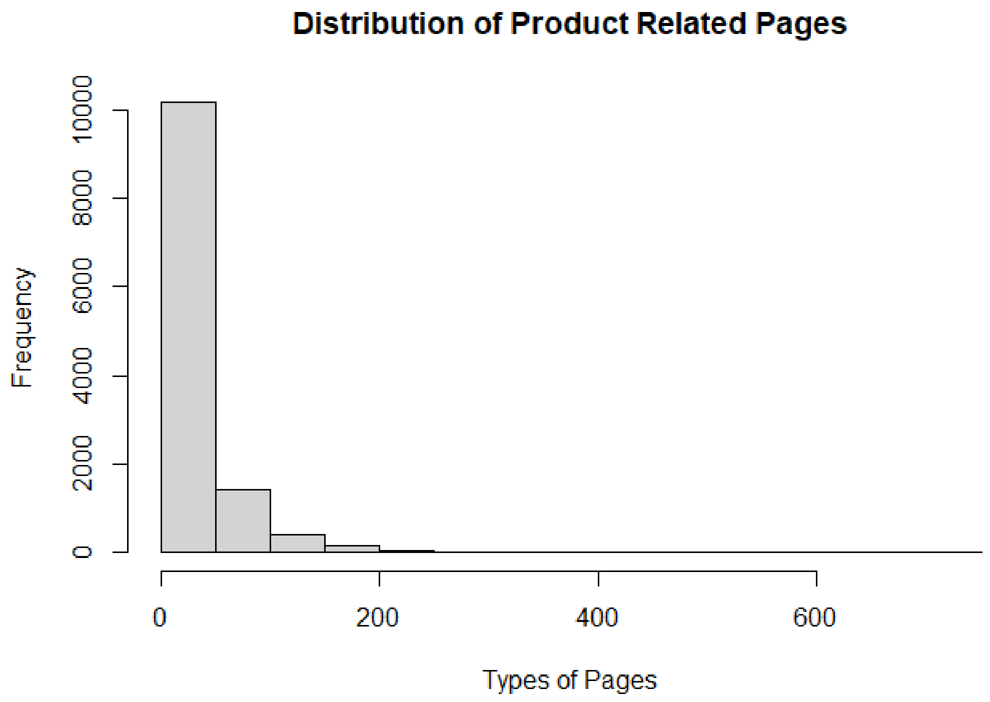
```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
time_on_prod_page
```

```
## # A tibble: 311 x 4
##    productrelated p_duration bouncerates exitrates
##             <dbl>      <dbl>       <dbl>     <dbl>
##  1              0          0        3.33      4.38
##  2              1     10330.      107.      114.
##  3              2     29092.       22.0      48.0
##  4              3     53882.       15.5      35.5
##  5              4     65578.       11.2      25.8
##  6              5     71559.        9.63     21.8
##  7              6     96837.        7.57     19.0
##  8              7    119062.        7.28     17.7
##  9              8    115396.        6.14     15.7
## 10              9    118569.        5.25     13.1
## # ... with 301 more rows
```

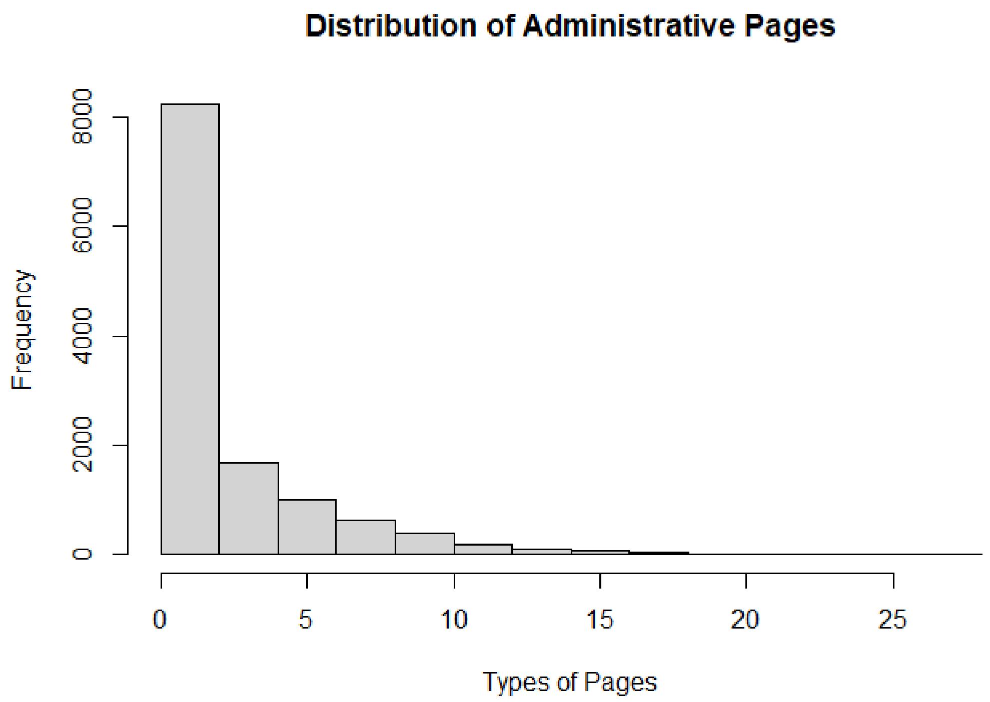
```
# From the above table we can see that product page 1 has a very high bounce and exit rate
# This is indicative that most users visit the page and subsequently exit immediately
# without visiting another page on the website or interacting with any of the elements on the

# Kira Plastinina should consider changing the product being displayed on product page 1 to on
# captures visitors' attention

# i.e. Products with lowest bounce and exit rates should be displayed first
```

```
# Histogram of Product Related Pages

# plotting using hist()
hist(final$productrelated,
        main = "Distribution of Product Related Pages",
        xlab = "Types of Pages",
        ylab = "Frequency")
```




```
# Product related pages are highly skewed to the right indicating the first page types are vis
```

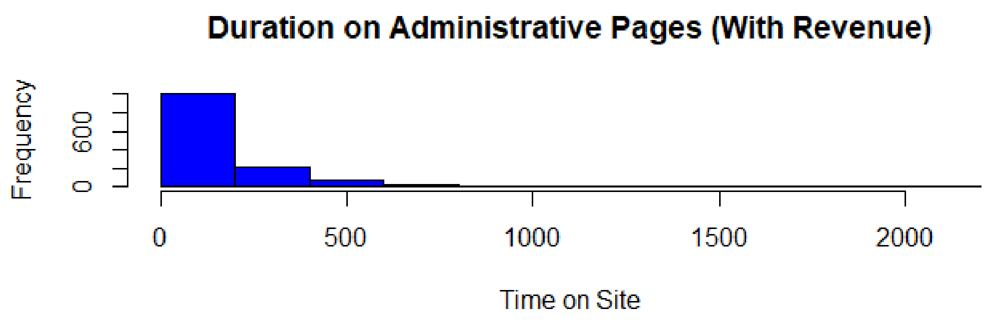
```
# Histogram of Administrative Pages

# plotting using hist()
hist(final$administrative,
        main = "Distribution of Administrative Pages",
        xlab = "Types of Pages",
        ylab = "Frequency")
```

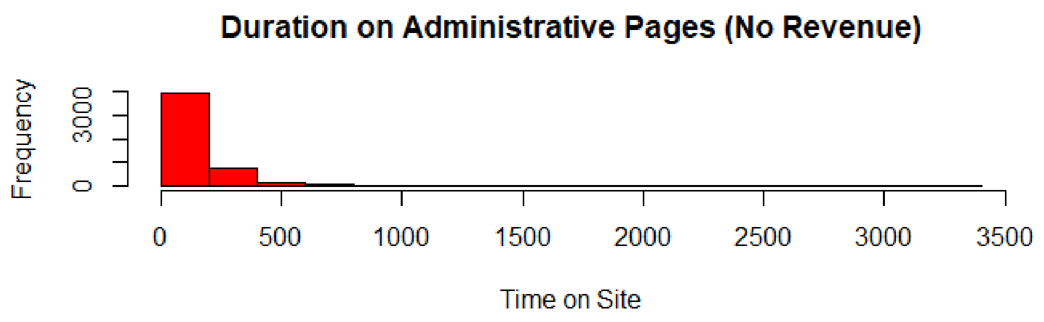


```
# Administrative pages are highly skewed to the right indicating the first few page types are
```

```
# Excluding records in data where a_duration is -1 value

final <- final %>% filter(final$a_duration > 0)
```

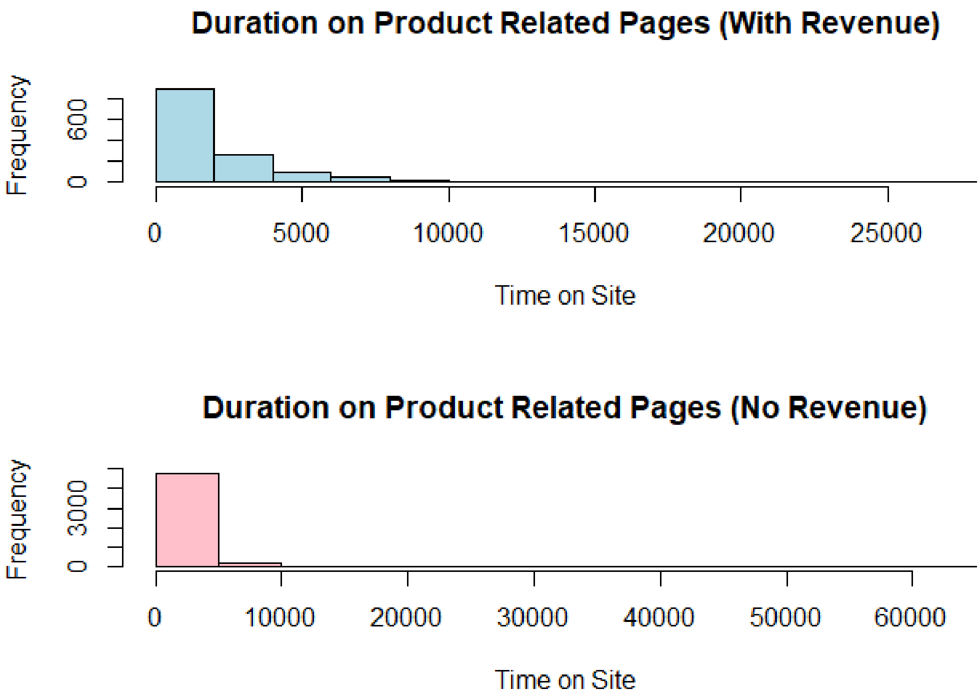
```
# First line puts graphs into 4 columns with 2 rows
par(mfrow=c(2,1))
# par(mar=c(2,2,2,2))

# Histograms of distribution of duration on administrative and product related pages
# and whether revenue was True

hist(final$a_duration [final$revenue == TRUE],
    main = "Duration on Administrative Pages (With Revenue)",
    xlab = "Time on Site",
    col = "blue")

hist(final$a_duration [final$revenue == FALSE],
    main = "Duration on Administrative Pages (No Revenue)",
    xlab = "Time on Site",
    col = "red")
```






```
# From above graphs, theres no significant chance in Revenue for persons who visit/linger on a
```

```
# First line puts graphs into 2 columns with 1 rows
par(mfrow=c(2,1))
# par(mar=c(2,2,2,2))

# Histograms of distribution of duration on administrative and product related pages
# and whether revenue was False

hist(final$p_duration [final$revenue == TRUE],
     main = "Duration on Product Related Pages (With Revenue)",
     xlab = "Time on Site",
     col = "lightblue")

hist(final$p_duration [final$revenue == FALSE],
     main = "Duration on Product Related Pages (No Revenue)",
     xlab = "Time on Site",
     col = "pink")
```

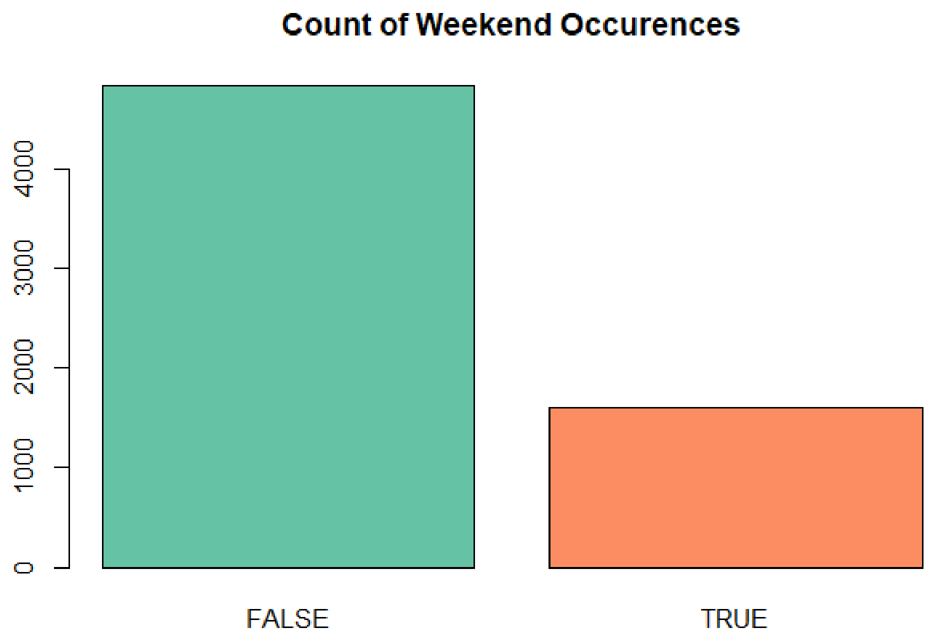



```
# From above graphs, we can see the graphs are highly skewed to the right. We will need to nor

# However, visitors that visit more pages apart from the first product page end up giving our
```

```
# Barplot of weekend

# Giving colour
library(RColorBrewer)
```

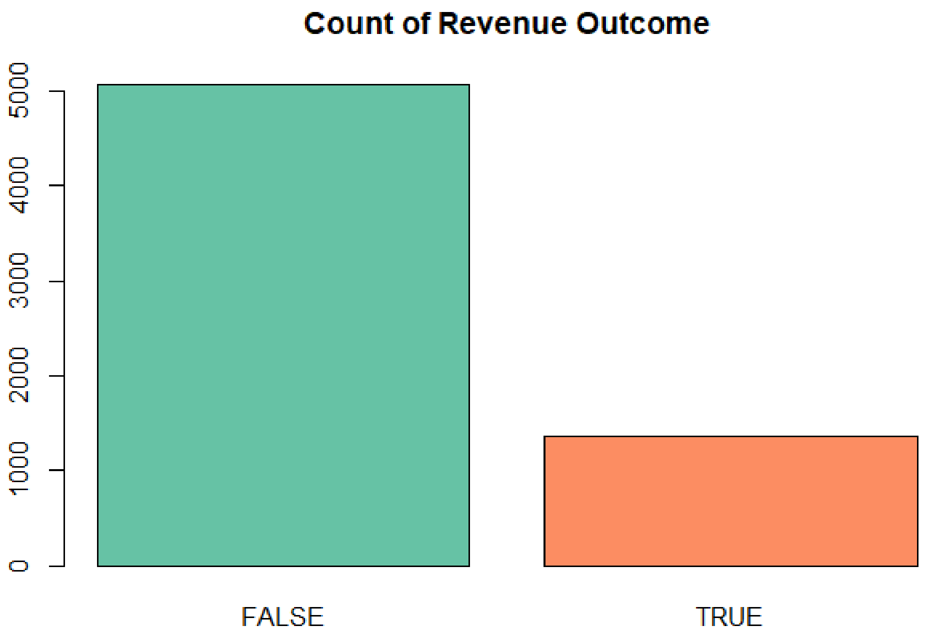
```
## Warning: package 'RColorBrewer' was built under R version 4.0.3
```

```r
coul <- brewer.pal(5, "Set2")

weekend <- table(final$weekend)

# plotting using barplot()
barplot(weekend,
        col=coul,
        main = "Count of Weekend Occurences")
```



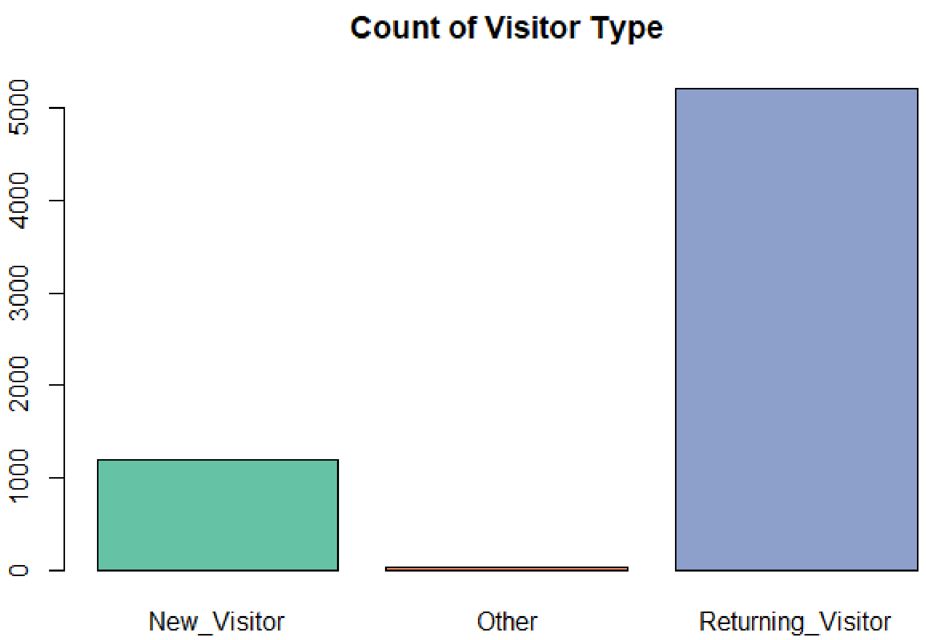

```r
# The barplot above reveals that most customers dont access the web page during the weekend
# judged by the occurrences of False in the weekend feature
```

```r
# Barplot of revenue

# creating revenue table
revenue <- table(final$revenue)

# plotting using barplot()
barplot(revenue,
        col=coul,
        main = "Count of Revenue Outcome")
```




```
# There are more outcomes of False than True in Revenue.
# This indicates our data is imbalanced since Revenue is our class variable
```

```
# Barplot of visitor type

# creating visitortype table
visitor <- table(final$visitortype)

# plotting using barplot()
barplot(visitor,
        col=coul,
        main = "Count of Visitor Type")
```



```
# Majority of records have returning visitors, therefore, Kira Plastinina should focus on retu
```

```
# Measures of Central Tendency using summary()
summary(final)
```

```
##  administrative    a_duration       informational       i_duration
##  Min.   : 1.00   Min.   :   1.333   Min.   : 0.0000   Min.   :   0.00
##  1st Qu.: 2.00   1st Qu.:  40.000   1st Qu.: 0.0000   1st Qu.:   0.00
##  Median : 3.00   Median :  88.000   Median : 0.0000   Median :   0.00
##  Mean   : 4.42   Mean   : 155.069   Mean   : 0.8196   Mean   :  56.87
##  3rd Qu.: 6.00   3rd Qu.: 183.000   3rd Qu.: 1.0000   3rd Qu.:  21.20
##  Max.   :27.00   Max.   :3398.750   Max.   :24.0000   Max.   :2549.38
##
##  productrelated     p_duration       bouncerates         exitrates
##  Min.   :  0.00   Min.   :    0.0   Min.   :0.000000   Min.   :0.00000
##  1st Qu.: 14.00   1st Qu.:  408.7   1st Qu.:0.000000   1st Qu.:0.01109
##  Median : 27.00   Median :  964.3   Median :0.001852   Median :0.01852
##  Mean   : 44.25   Mean   : 1668.6   Mean   :0.007029   Mean   :0.02266
##  3rd Qu.: 53.00   3rd Qu.: 2022.9   3rd Qu.:0.009608   3rd Qu.:0.02941
##  Max.   :705.00   Max.   :63973.5   Max.   :0.161905   Max.   :0.15000
##
##    pagevalues        specialday         month      operatingsystems
##  Min.   :  0.000   Min.   :0.0000   Nov    :1612   2      :3493
##  1st Qu.:  0.000   1st Qu.:0.0000   May    :1564   1      :1332
##  Median :  0.000   Median :0.0000   Mar    : 914   3      :1319
##  Mean   :  8.155   Mean   :0.0385   Dec    : 867   4      : 235
##  3rd Qu.:  7.058   3rd Qu.:0.0000   Oct    : 430   8      :  33
##  Max.   :361.764   Max.   :1.0000   Sep    : 330   6      :   8
##                                     (Other): 709   (Other):   6
##     browser         region      trafficetype              visitortype
##  2      :4280   1      :2442   2      :2468   New_Visitor      :1185
##  1      :1241   3      :1281   1      :1057   Other            :  33
##  4      : 344   4      : 626   3      : 857   Returning_Visitor:5208
##  5      : 234   2      : 596   4      : 596
##  10     :  90   7      : 401   13     : 286
##  6      :  76   6      : 397   10     : 228
##  (Other): 161   (Other): 683   (Other): 934
##   weekend      revenue
##  FALSE:4825   FALSE:5066
##  TRUE :1601   TRUE :1360
##
##
##
##
##
```

```
# Getting some Measures of Dispersion using describe()
describe(final)
```

```
##                    vars    n    mean      sd median trimmed     mad  min
## administrative        1 6426    4.42    3.45   3.00    3.89    2.97 1.00
## a_duration            2 6426  155.07  220.12  88.00  111.10   87.97 1.33
## informational         3 6426    0.82    1.57   0.00    0.45    0.00 0.00
## i_duration            4 6426   56.87  177.09   0.00   14.49    0.00 0.00
## productrelated        5 6426   44.25   53.84  27.00   33.60   23.72 0.00
## p_duration            6 6426 1668.57 2352.74 964.26 1222.40 1008.27 0.00
## bouncerates           7 6426    0.01    0.01   0.00    0.00    0.00 0.00
## exitrates             8 6426    0.02    0.02   0.02    0.02    0.01 0.00
## pagevalues            9 6426    8.15   20.26   0.00    3.26    0.00 0.00
## specialday           10 6426    0.04    0.16   0.00    0.00    0.00 0.00
## month*               11 6426    6.27    2.47   7.00    6.46    1.48 1.00
## operatingsystems*    12 6426    2.11    0.88   2.00    2.05    0.00 1.00
## browser*             13 6426    2.32    1.67   2.00    1.98    0.00 1.00
## region*              14 6426    3.17    2.40   3.00    2.82    2.97 1.00
## traffictype*         15 6426    3.91    3.77   2.00    3.10    1.48 1.00
## visitortype*         16 6426    2.63    0.78   3.00    2.78    0.00 1.00
## weekend*             17 6426    1.25    0.43   1.00    1.19    0.00 1.00
## revenue*             18 6426    1.21    0.41   1.00    1.14    0.00 1.00
##                         max    range  skew kurtosis    se
## administrative        27.00    26.00  1.57     3.31  0.04
## a_duration          3398.75  3397.42  4.58    32.84  2.75
## informational         24.00    24.00  3.06    16.19  0.02
## i_duration          2549.38  2549.38  5.88    46.61  2.21
## productrelated       705.00   705.00  3.73    22.28  0.67
## p_duration         63973.52 63973.52  6.69   108.12 29.35
## bouncerates            0.16     0.16  3.34    17.79  0.00
## exitrates              0.15     0.15  1.86     5.39  0.00
## pagevalues           361.76   361.76  5.38    48.27  0.25
## specialday             1.00     1.00  4.43    19.36  0.00
## month*                10.00     9.00 -0.82    -0.39  0.03
## operatingsystems*      8.00     7.00  1.94    10.08  0.01
## browser*              13.00    12.00  3.39    13.84  0.02
## region*                9.00     8.00  0.97    -0.18  0.03
## traffictype*          20.00    19.00  2.14     4.57  0.05
## visitortype*           3.00     2.00 -1.61     0.59  0.01
## weekend*               2.00     1.00  1.16    -0.66  0.01
## revenue*               2.00     1.00  1.41    -0.01  0.01
```

```
# Getting additional Measures of Dispersion for numerical variables

num_vars <- final[,1:10]

# Skewness
skew(num_vars)
```

```
##  [1] 1.574668 4.575444 3.062271 5.884487 3.733576 6.689829 3.344587 1.860296
##  [9] 5.379180 4.432755
```

```
# We can see all numerical columns are highly skewed hence not a normal distribution.
# We will have to normalise our dataset
```

```r
# computing the interquartile ranges for numerical variables

administrative_iqr        <- IQR(num_vars$administrative)
a_duration_iqr            <- IQR(num_vars$a_duration)
informational_iqr         <- IQR(num_vars$informational)
i_duration_iqr            <- IQR(num_vars$i_duration)
productrelated_iqr        <- IQR(num_vars$productrelated)
p_duration                <- IQR(num_vars$p_duration)
bouncerates_iqr           <- IQR(num_vars$bouncerates)
exitrates_iqr             <- IQR(num_vars$exitrates)
pagevalues_iqr            <- IQR(num_vars$pagevalues)
specialdaya_iqr           <- IQR(num_vars$specialday)


print("administrative_iqr :",quote=TRUE)
```

```
## [1] "administrative_iqr :"
```

```r
administrative_iqr
```

```
## [1] 4
```

```r
print("a_duration_iqr :",quote=TRUE)
```

```
## [1] "a_duration_iqr :"
```

```r
a_duration_iqr
```

```
## [1] 143
```

```r
print("informational_iqr :",quote=TRUE)
```

```
## [1] "informational_iqr :"
```

```r
informational_iqr
```

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

```r
print("i_duration_iqr :",quote=TRUE)
```

```
## [1] "i_duration_iqr :"
```

```r
i_duration_iqr
```

```
## [1] 21.2
```

```
print("productrelated_iqr :",quote=TRUE)
```

```
## [1] "productrelated_iqr :"
```

```
productrelated_iqr
```

```
## [1] 39
```

```
print("p_duration :",quote=TRUE)
```

```
## [1] "p_duration :"
```

```
p_duration
```

```
## [1] 1614.188
```

```
print("bouncerates_iqr :",quote=TRUE)
```

```
## [1] "bouncerates_iqr :"
```

```
bouncerates_iqr
```

```
## [1] 0.009608144
```

```
print("exitrates_iqr :",quote=TRUE)
```

```
## [1] "exitrates_iqr :"
```

```
exitrates_iqr
```

```
## [1] 0.01832565
```

```
print("pagevalues_iqr :",quote=TRUE)
```

```
## [1] "pagevalues_iqr :"
```

```
pagevalues_iqr
```

```
## [1] 7.058281
```

```
print("specialdaya_iqr :",quote=TRUE)
```

```
## [1] "specialdaya_iqr :"
```

```
specialdaya_iqr
```

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

```
# Product related pages (p_duration) range is very high indicating people spend a lot of time
```

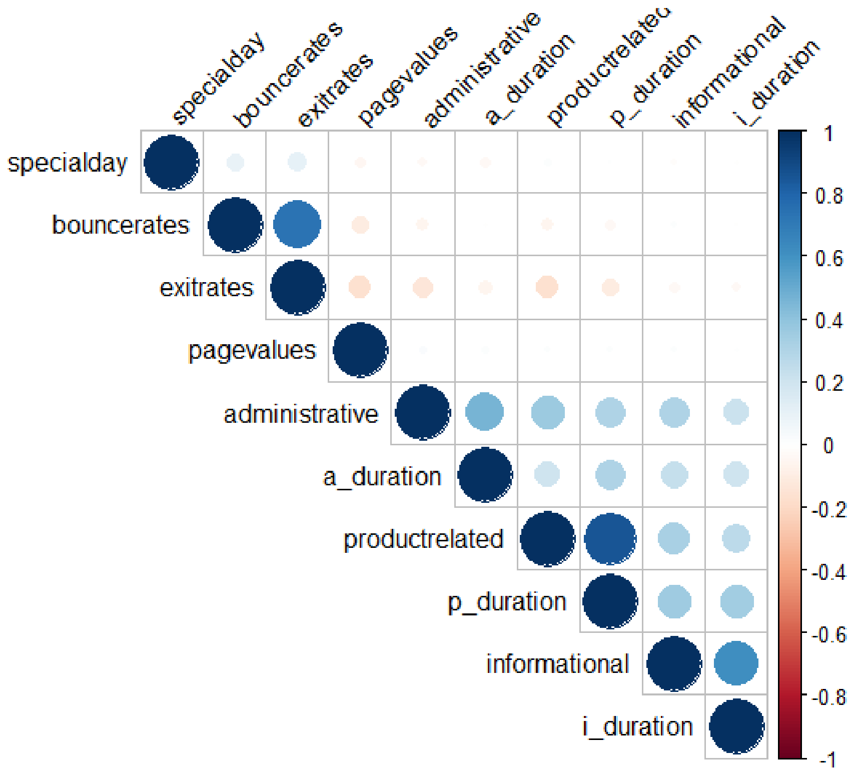
Bivariate & Multivariate

```
# Correlations
# Computing a correlation matrix between all numerical variables using pearson method and roun

correlations <- cor(num_vars, method = "pearson")
round(correlations, 2)
```

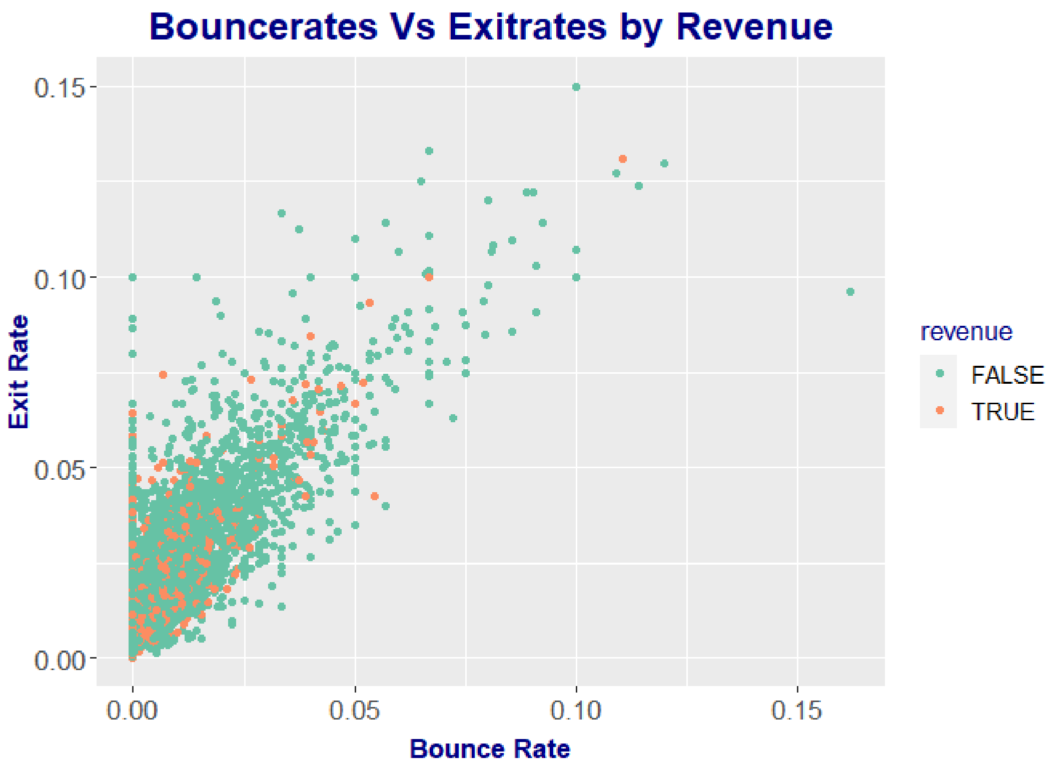
```
##                 administrative a_duration informational i_duration
## administrative            1.00       0.46          0.30       0.21
## a_duration                0.46       1.00          0.23       0.20
## informational             0.30       0.23          1.00       0.61
## i_duration                0.21       0.20          0.61       1.00
## productrelated            0.36       0.20          0.33       0.26
## p_duration                0.30       0.30          0.36       0.35
## bouncerates              -0.06      -0.01          0.01       0.00
## exitrates                -0.14      -0.06         -0.04      -0.03
## pagevalues                0.02       0.02          0.02       0.01
## specialday               -0.03      -0.04         -0.02      -0.01
##                 productrelated p_duration bouncerates exitrates pagevalues
## administrative            0.36       0.30       -0.06     -0.14       0.02
## a_duration                0.20       0.30       -0.01     -0.06       0.02
## informational             0.33       0.36        0.01     -0.04       0.02
## i_duration                0.26       0.35        0.00     -0.03       0.01
## productrelated            1.00       0.86       -0.06     -0.16       0.02
## p_duration                0.86       1.00       -0.04     -0.10       0.01
## bouncerates              -0.06      -0.04        1.00      0.73      -0.10
## exitrates                -0.16      -0.10        0.73      1.00      -0.16
## pagevalues                0.02       0.01       -0.10     -0.16       1.00
## specialday                0.02       0.00        0.10      0.11      -0.04
##                 specialday
## administrative      -0.03
## a_duration          -0.04
## informational       -0.02
## i_duration          -0.01
## productrelated       0.02
## p_duration           0.00
## bouncerates          0.10
## exitrates            0.11
## pagevalues          -0.04
## specialday           1.00
```

```
# informational & i_duration, productrelated & p_duration, bouncerates & exitrates are strongl
# 0.61, 0.86 & 0.73 respectively.
```

```
# Viewing the correlations better to support the above notions

corrplot(correlations, type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
```




```
# Scatter Plots

# Setting graph dimensions
options(repr.plot.width = 13, repr.plot.height = 7)

# Plotting using ggplot() and using theme() for theme of the plot

bouncexit_rates = ggplot(data = final, aes(x = bouncerates, y = exitrates , col = revenue)) +
    geom_point() +
    labs(title = 'Bouncerates Vs Exitrates by Revenue', x = 'Bounce Rate', y = 'Exit Rate') +
    scale_color_brewer(palette = 'Set2') +
    theme(plot.title=element_text(size=18, face="bold", color="navyblue",hjust=0.5, lineheight=
          plot.subtitle=element_text(size=15, face="bold", hjust=0.5),
         axis.title.x = element_text(color = 'navyblue', size = 13, face = 'bold', vjust = -0.5
         axis.title.y = element_text(color = 'navyblue', size = 13, face = 'bold', vjust = 0.5)
         axis.text.y = element_text(size = 13),
         axis.text.x = element_text(size = 13),
         legend.title = element_text(size = 13, color = 'navyblue'),
        legend.text = element_text(size = 11))

plot(bouncexit_rates)
```

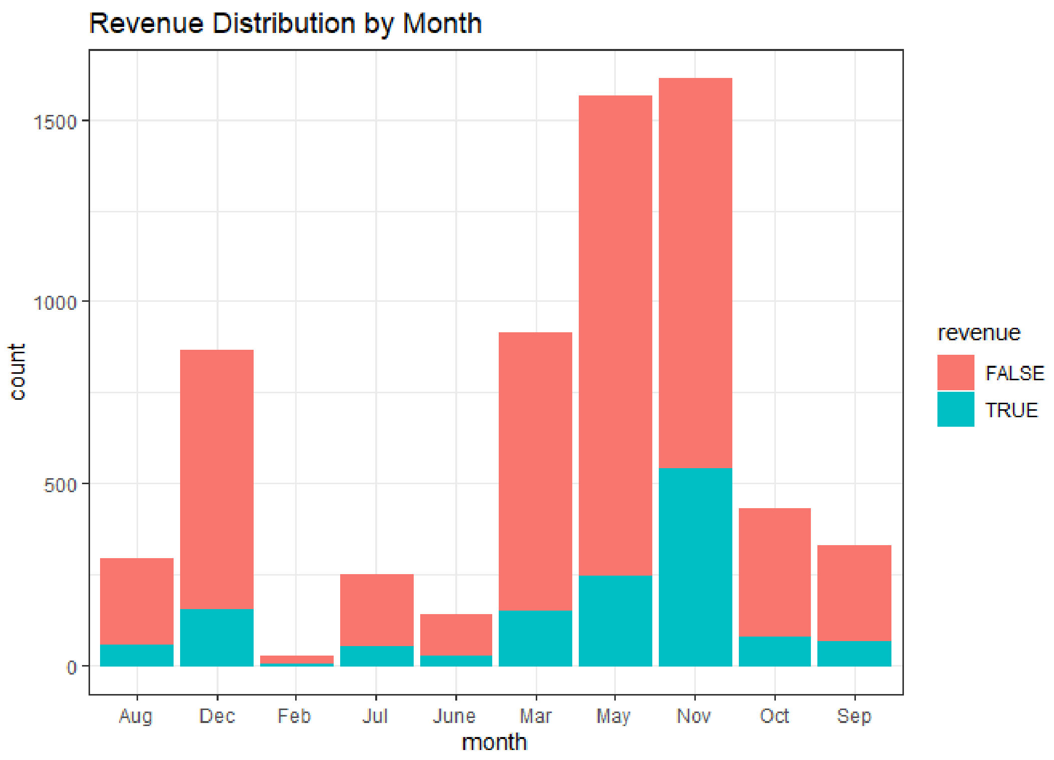



```
# There appears to be somekind of linear relationship between the two variables
# When bouncerates and exitrates are lower on product related pages, there will be revenue as
# Lower bouncerates and exitrates can also signify disinterest in the products on the page as
```

```
# Combined Bar Charts

# Using geom_bar to plot a combined bar chart with the count function for discrete variables s

c <- ggplot(final, aes(x=month, fill=revenue, color=revenue)) +
geom_bar(binwidth = 1) + labs(title="Revenue Distribution by Month")
```

```
## Warning: Ignoring unknown parameters: binwidth
```

```
c + theme_bw()
```

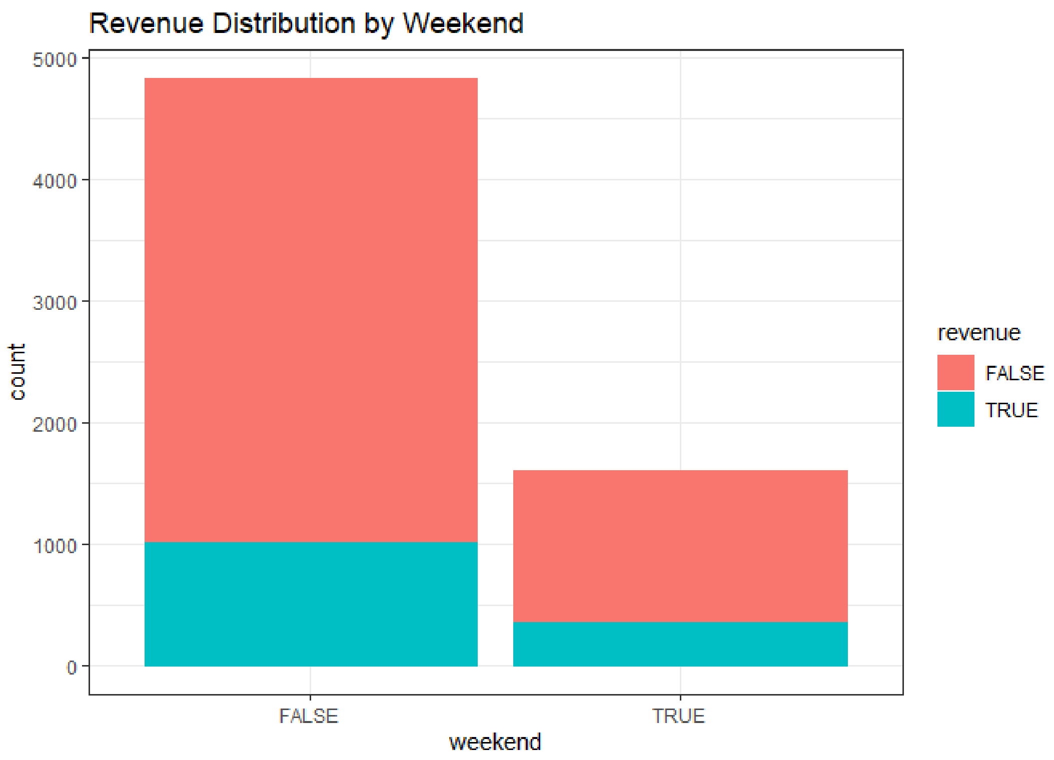



```
# The months of Mar, May and November generate more activity hence more revenue. These months
```

```
# Combined Bar Charts

# Using geom_bar to plot a combined bar chart for weekend and revenue

d <- ggplot(final, aes(x=weekend, fill=revenue, color=revenue)) +
geom_bar(binwidth = 1) + labs(title="Revenue Distribution by Weekend")
```

```
## Warning: Ignoring unknown parameters: binwidth
```

```
d + theme_bw()
```

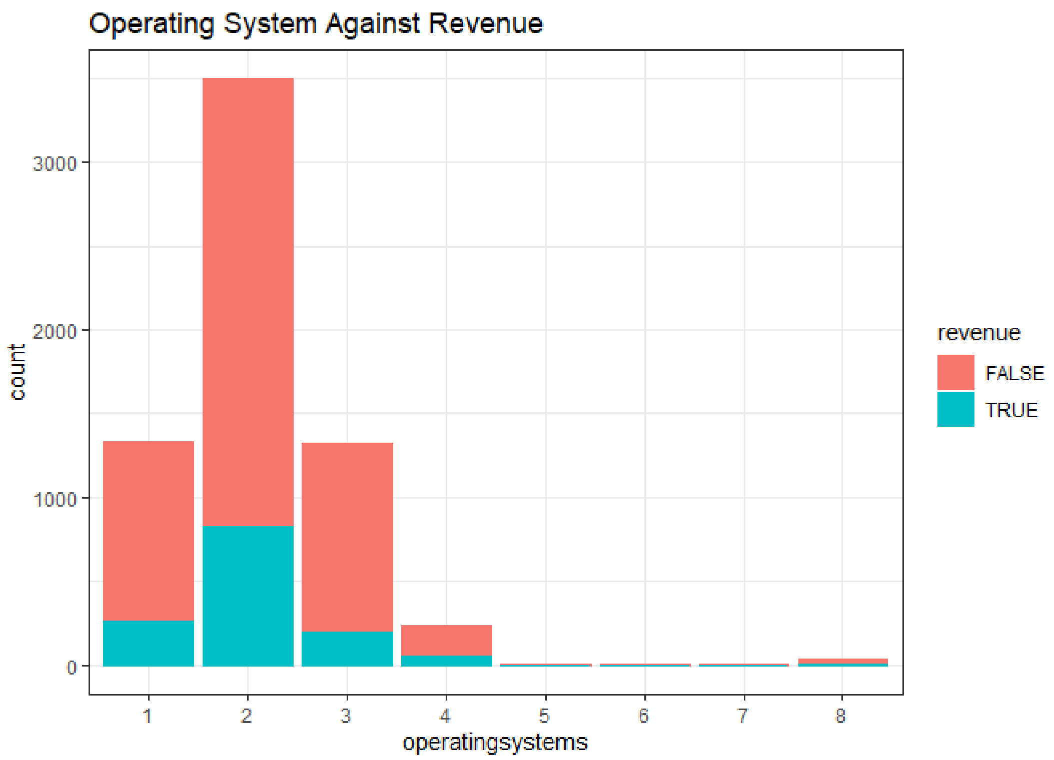



```
# The weekend does not bring in much revenue since activity is higher during weekdays. Therefo
```

```
# Combined Bar Charts

# Using geom_bar to plot a combined bar chart for operating system and revenue

e <- ggplot(final, aes(x=operatingsystems, fill=revenue, color=revenue)) +
geom_bar(binwidth = 1) + labs(title="Operating System Against Revenue")
```

```
## Warning: Ignoring unknown parameters: binwidth
```

```
e + theme_bw()
```

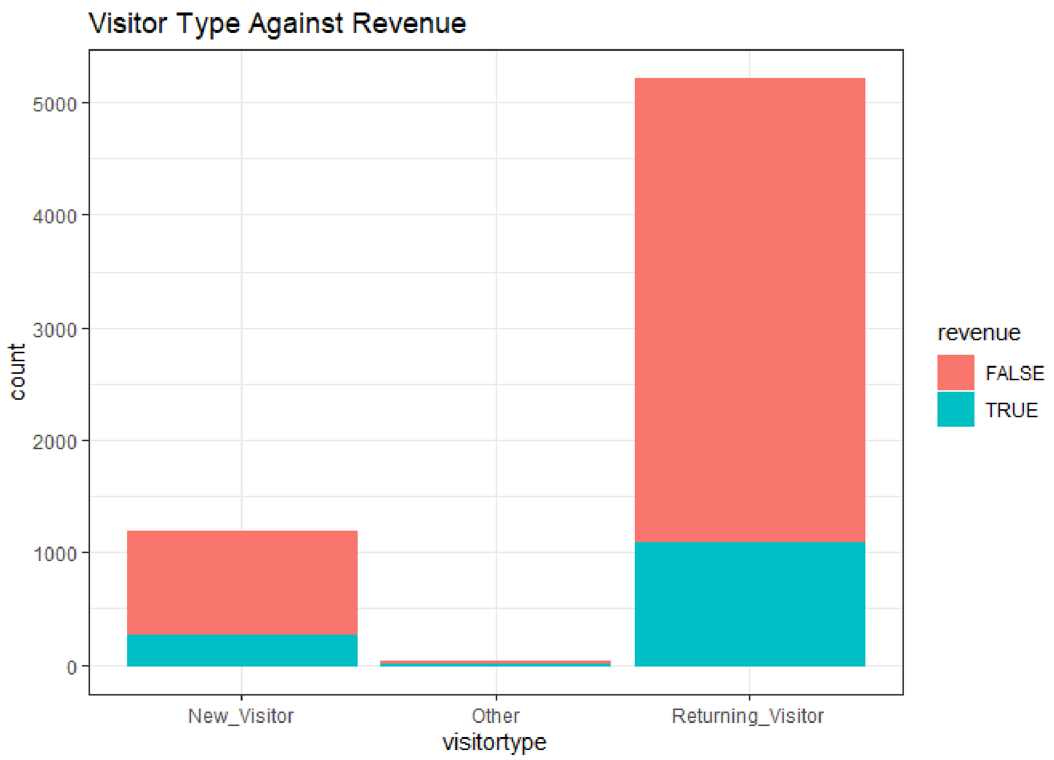



```
# Clients using the 1st, 2nd and 3rd operating systems are more active and bring in more reven
```

```
# Combined Bar Charts

# Using geom_bar to plot a combined bar chart for visitor type and revenue

e <- ggplot(final, aes(x=visitortype, fill=revenue, color=revenue)) +
geom_bar(binwidth = 1) + labs(title="Visitor Type Against Revenue")
```

```
## Warning: Ignoring unknown parameters: binwidth
```

```
e + theme_bw()
```

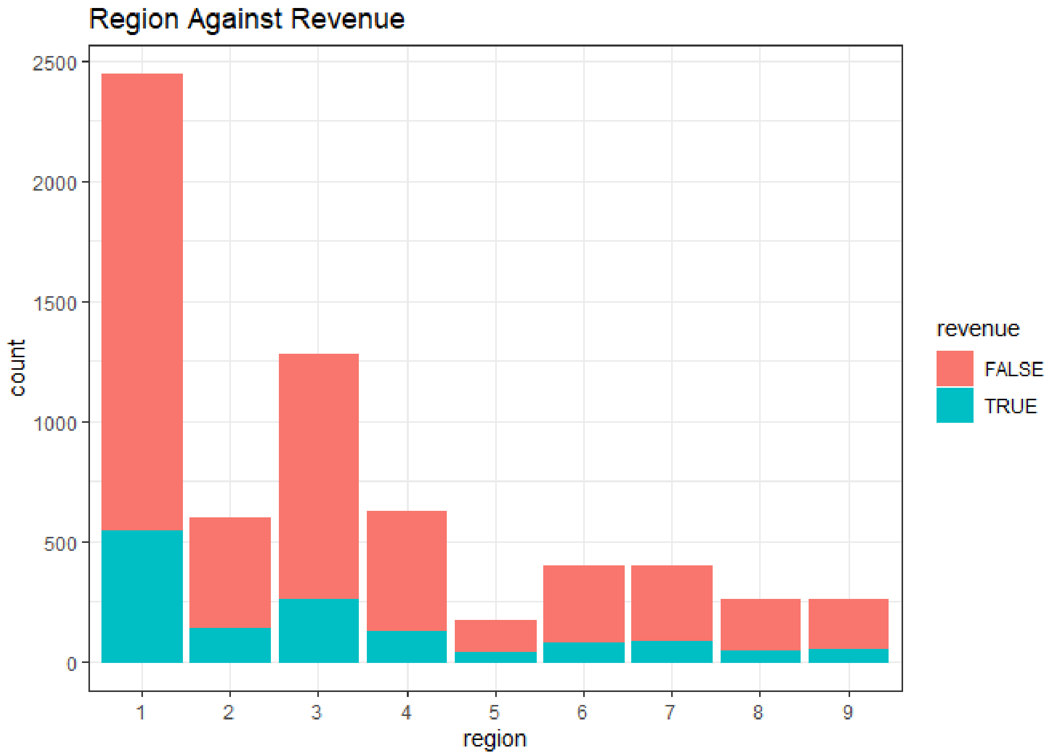



```
# Returning visitors should be targeted first followed by new visitors since they bring more r
```

```
# Combined Bar Charts

# Using geom_bar to plot a combined bar chart for region and revenue

e <- ggplot(final, aes(x=region, fill=revenue, color=revenue)) +
geom_bar(binwidth = 1) + labs(title="Region Against Revenue")
```

```
## Warning: Ignoring unknown parameters: binwidth
```

```
e + theme_bw()
```

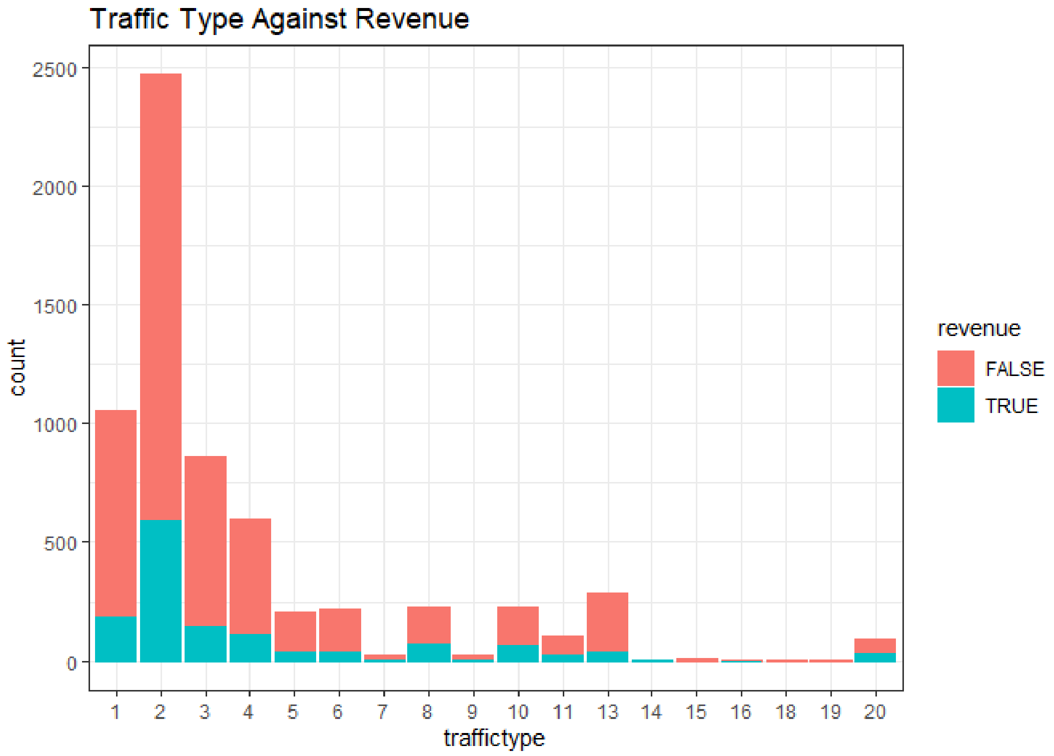



```
# The first three regions can be target marketed for more returns since theyre more active
```

```
# Combined Bar Charts

# Using geom_bar to plot a combined bar chart for Traffic Type and revenue

e <- ggplot(final, aes(x=traffictype, fill=revenue, color=revenue)) +
geom_bar(binwidth = 1) + labs(title="Traffic Type Against Revenue")
```

```
## Warning: Ignoring unknown parameters: binwidth
```

```
e + theme_bw()
```



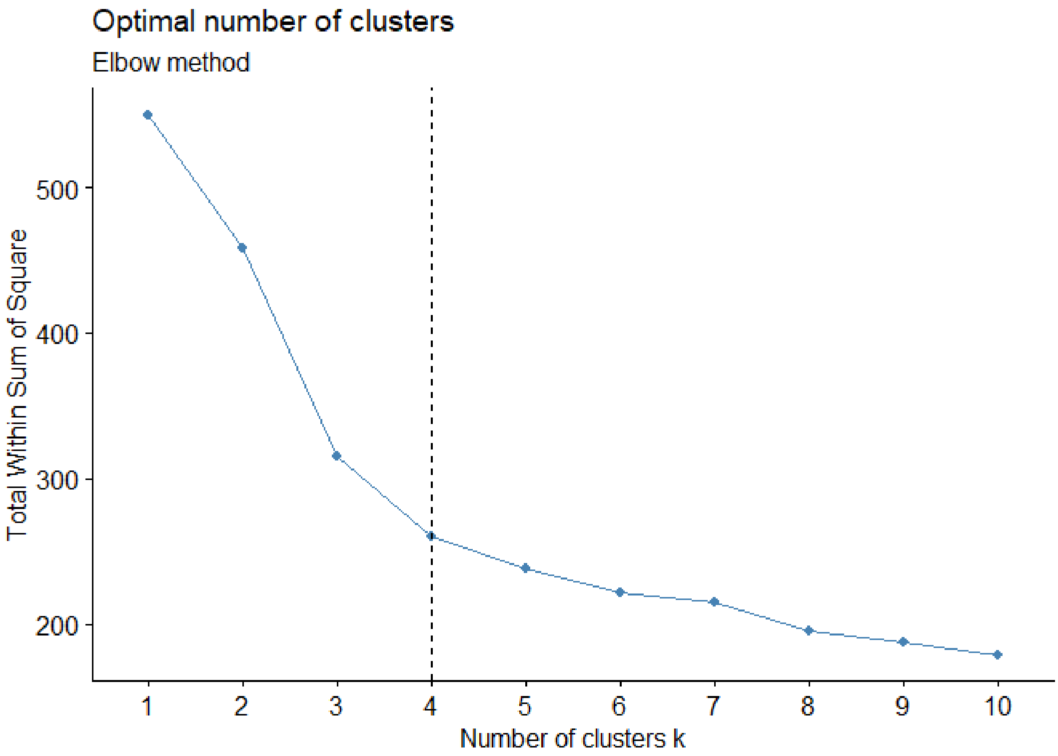

```
# Traffic types 1 to 4 bring clients who are more active and generate more revenue, Especially
```

6. Implement the Solution

Using K-Means

```
# Normalizing the dataset so that no particular attribute
# has more impact on clustering algorithm than others.
# ---
#
normalize <- function(x){
  return ((x-min(x)) / (max(x)-min(x)))
}
final2 <- final

# normalising first 10 numerical columns

final2$administrative <- normalize(final2$administrative)
final2$a_duration <- normalize(final2$a_duration)
final2$informational <- normalize(final2$informational)
final2$i_duration <- normalize(final2$i_duration)
final2$productrelated <- normalize(final2$productrelated)
final2$p_duration <- normalize(final2$p_duration)
final2$bouncerates <- normalize(final2$bouncerates)
final2$exitrates <- normalize(final2$exitrates)
final2$pagevalues <- normalize(final2$pagevalues)
final2$specialday <- normalize(final2$specialday)
```

```r
# Obtaining optimal nearest neighbours using elbow method

pacman :: p_load(factoextra) # loading necessary library


fviz_nbclust(final2[,1:10], kmeans, method = "wss") +
    geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
```




```r
# However, we already know our class has two clusters ( True or False for Revenue)
# Applying the K-means clustering algorithm with no. of centroids(k)=2 and removing the label

result <- kmeans(final2[,1:10],2)
```

```r
# Previewing the no. of records in each cluster

result$size
```

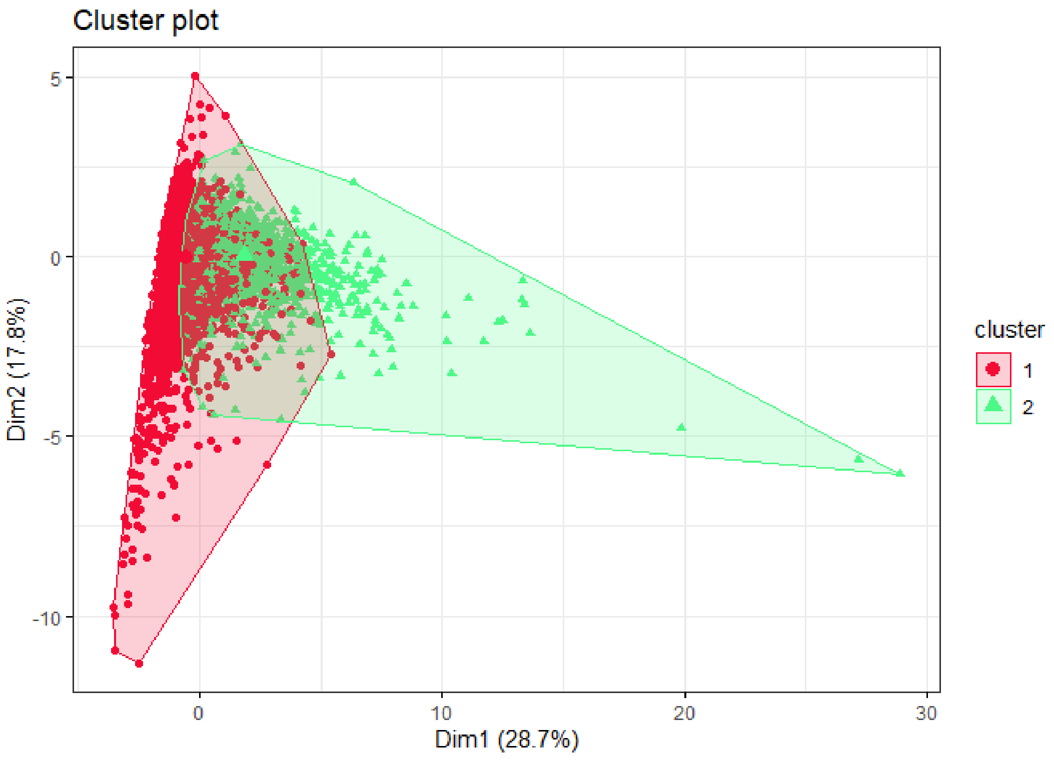
```
## [1] 4843 1583
```

```r
# Getting the value of cluster center datapoint value(2 centers for k=2)

result$centers
```

```
##   administrative a_duration informational i_duration productrelated p_duration
## 1     0.07192776 0.02934319    0.02100970 0.01141884     0.04553095 0.01886562
## 2     0.31398513 0.09391806    0.07435776 0.05562205     0.11550741 0.04816058
##   bouncerates exitrates pagevalues specialday
## 1  0.04691164 0.1622612 0.02124630 0.04526120
## 2  0.03271248 0.1168250 0.02650383 0.01781428
```

```r
# Getting the class

final.class <- final[,18]
```

```r
# Setting plot options
# par(mfrow = c(2,2), mar = c(5,4,2,2))

# Plotting to see how data points have been distributed in clusters using fviz_cluster

fviz_cluster(result, data = final2[, 1:10],
             palette = c("#f20b34", "#4cf886"),
             geom = "point",
             ellipse.type = "convex",
             ggtheme = theme_bw()
             )
```




```r
# Getting the accuracy using a confusion matrix by comparing result cluster with our class

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

```
##    final.class
##     FALSE TRUE
##   1  3896  947
##   2  1170  413
```

Strengths and Limitations

```
# K means is preferred or performs well when we have an idea of the number of clusters/centroi

# Limitations
# this algorithm only accepts numerical variables/features hence some important categorical fe
# This algorithm cannot work with NA values, noise or outliers in the dataset. It requires a m
```

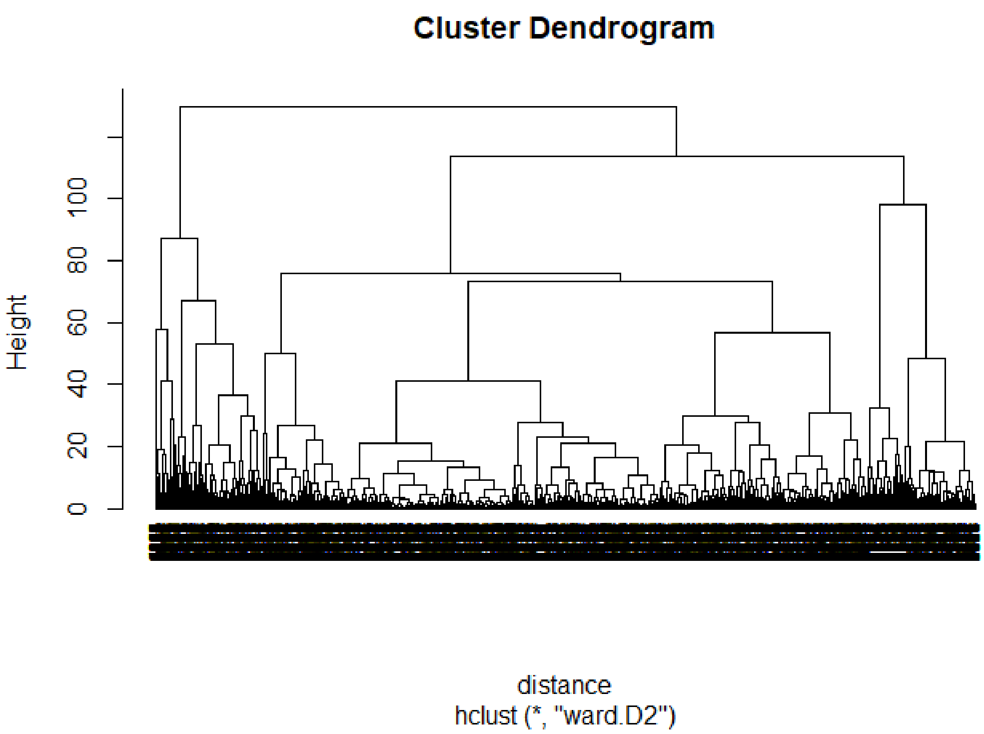
Using Hierarchichal Clustering

```
# We note that the variables have a large different means and variances.
# This is explained by the fact that the variables are measured in different
# units

# They must be standardized (scaled) to make them comparable such that
# they have mean zero and standard deviation one.

final3 <- scale(final[,1:10])
```

```
# Using the dist() function to compute the Euclidean distance between observations,
# and saving it in variable distance which will be the first argument in the following hclust(

distance <- dist(final3, method = "euclidean")
```

```
# We then use hierarchical clustering using the Ward's method

result.hc <- hclust(distance, method = "ward.D2" )
```

```
# Plotting the obtained dendrogram

plot(result.hc, cex = 0.6, hang = -1)
```



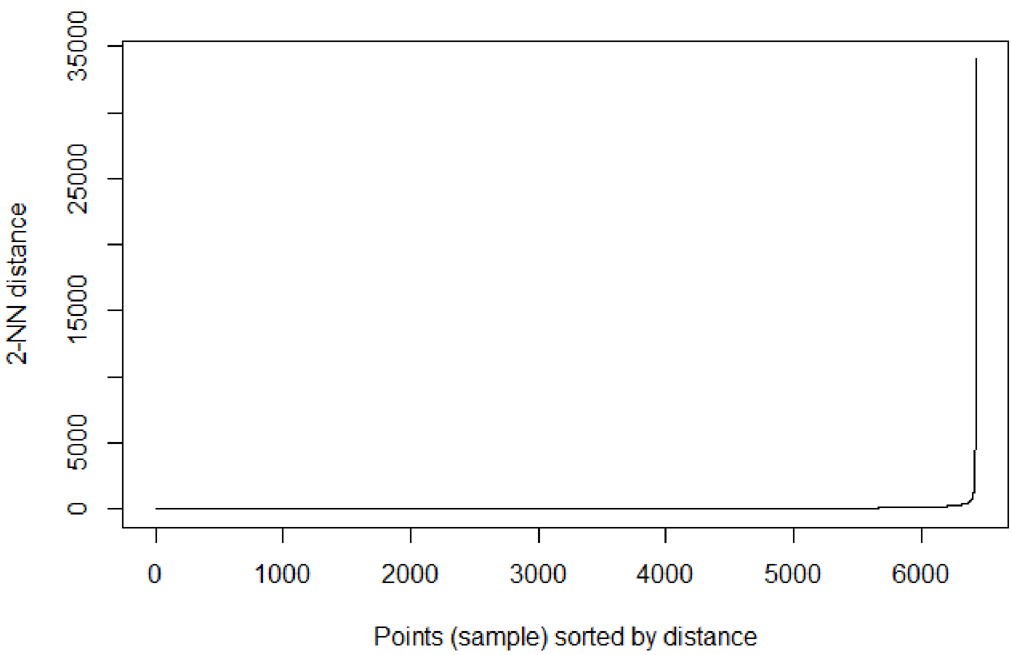
```
# The resulting dendogram shows too many clusters.
```

Strengths and Limitations

```
# Strengths
# It is easier and faster to use for a smaller dataset

# Limitations
# Since our dataset is large, hierarchical clustering is not the right fit since it creates to
# Its computationally expensive for very large datasets.
```

7. Challenge the Solution

```
# We challenge the solution using DBSCAN algorithm to see if it performs better clustering

# Loading necessary libraries
pacman :: p_load(dbscan)

# obtaining optimal nearest neighbours

kNNdistplot(final[,1:10],k=2)
```




```
# shows optimal distance at approx 2000 for k value which we already know as 2 based on revenu
```

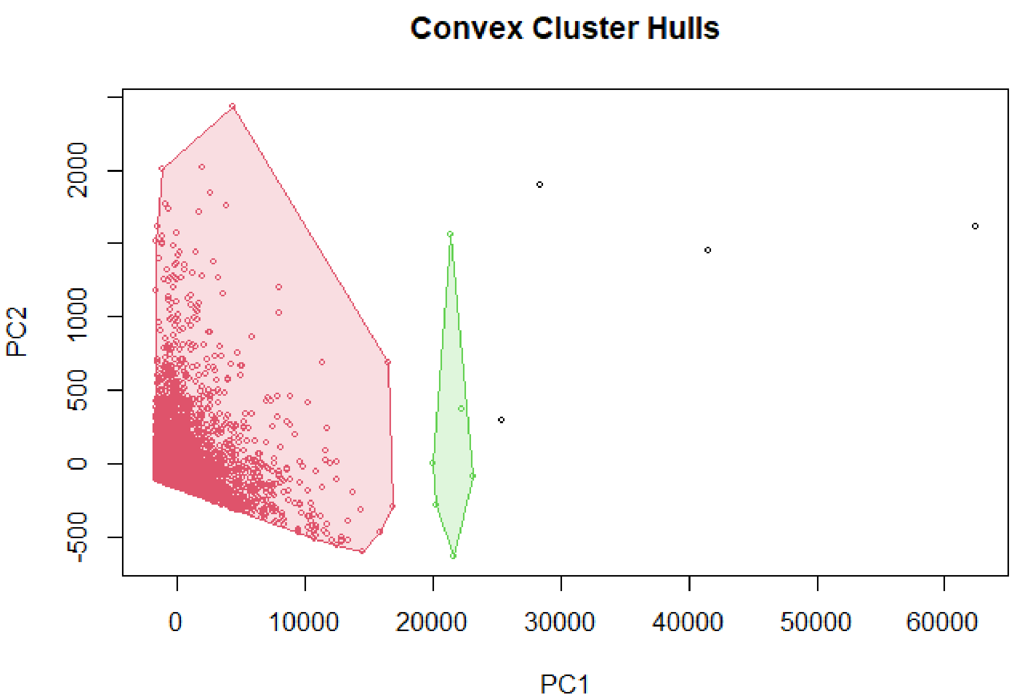
```
# We want minimum 2 Cluster points with in a distance of eps(2000)
#

result_db <- dbscan(final[,1:10],eps=2000,MinPts = 2, borderPoints = TRUE)
```

```
## Warning in dbscan(final[, 1:10], eps = 2000, MinPts = 2, borderPoints = TRUE):
## converting argument MinPts (fpc) to minPts (dbscan)!
```

```
result_db
```

```
## DBSCAN clustering for 6426 objects.
## Parameters: eps = 2000, minPts = 2
## The clustering contains 2 cluster(s) and 4 noise points.
##
##    0    1    2
##    4 6416    6
##
## Available fields: cluster, eps, minPts
```

```
# We also plot our clusters using hullplot()

hullplot(final[,1:10],result_db$cluster)
```



8. Follow Up Questions/Summary

```
# This dataset was not the right dataset to answer or provide a solution to the problem. This 

# We therefore require a new dataset that has a roughly equal measure of the revenue outcome.
```