1. Introduction

a)Defining the Question Grouping similar customer behavior that lead to an outcome of either buying the brand or not. **b)Defining the Metric of Success** To create accurate groups of different customer characteristics that show whether a customer is likely to purchase the brand or not.

c)Understanding the Context

Kira Plastinina's sales and marketing team would like to understand their customer's behavior from data that they have collected over the past year.

d)Experimental Design

- Problem Definition
- Data Preparation and Cleaning
  - Loading data
  - Checking for missing values
  - Checking for duplicates
- Perform Exploratory Data Analysis
  - Univariate analysis
  - Bivariate analysis
  - Multivariate analysis
- Modelling
  - KNN
  - Naive Bayes
  - Decision Trees
- Unsupervised Learning
  - K-Means clustering
  - Hierachial Clustering
- Conclusion

2.Data Preparation and Cleaning

```
#loading data
data<-read.csv("http://bit.ly/EcommerceCustomersDataset")
head(data)
```

```
##   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
# checking number of rows and columns
dim(data)
```

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

```r
#previewing our dataset
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" "Returni
##  $ Weekend                : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ Revenue                : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

```r
# checking for duplicated records
anyDuplicated(data)
```

```
## [1] 159
```

We have 159 duplicated records

```r
# removing duplicates
data <- unique(data)
dim(data)
```

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

```r
# checking for missing values
colSums(is.na(data))
```

```
##          Administrative Administrative_Duration           Informational 
##                      12                      12                      12 
##  Informational_Duration          ProductRelated ProductRelated_Duration 
##                      12                      12                      12 
##             BounceRates               ExitRates              PageValues 
##                      12                      12                       0 
##              SpecialDay                   Month        OperatingSystems 
##                       0                       0                       0 
##                 Browser                  Region             TrafficType 
##                       0                       0                       0 
##             VisitorType                 Weekend                 Revenue 
##                       0                       0                       0
```

```r
#dropping our missing values
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
```

We dropped the missing values since we have a large number of records

```r
# converting some of the variables from numerical to categorical
data$OperatingSystems <- as.factor(data$OperatingSystems)
data$Browser <- as.factor(data$Browser)
data$Region <- as.factor(data$Region)
data$TrafficType <- as.factor(data$TrafficType)
data$Weekend <- as.factor(data$Weekend)
data$Revenue <- as.factor(data$Revenue)
str(data)
```

```
## 'data.frame':    12199 obs. of  18 variables:
##  $ Administrative         : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Administrative_Duration: num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ Informational          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration : num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ ProductRelated         : int  1 2 1 2 10 19 1 1 2 3 ...
##  $ ProductRelated_Duration: num  0 64 -1 2.67 627.5 ...
##  $ BounceRates            : num  0.2 0 0.2 0.05 0.02 ...
##  $ ExitRates              : num  0.2 0.1 0.2 0.14 0.05 ...
##  $ PageValues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialDay             : num  0 0 0 0 0 0 0.4 0 0.8 0.4 ...
##  $ Month                  : chr  "Feb" "Feb" "Feb" "Feb" ...
##  $ OperatingSystems       : Factor w/ 8 levels "1","2","3","4",..: 1 2 4 3 3 2 2 1 2 2 ...
##  $ Browser                : Factor w/ 13 levels "1","2","3","4",..: 1 2 1 2 3 2 4 2 2 4 ...
##  $ Region                 : Factor w/ 9 levels "1","2","3","4",..: 1 1 9 2 1 1 3 1 2 1 ...
##  $ TrafficType            : Factor w/ 20 levels "1","2","3","4",..: 1 2 3 4 4 3 3 5 3 2 ...
##  $ VisitorType            : chr  "Returning_Visitor" "Returning_Visitor" "Returning_Visitor" "Returni
##  $ Weekend                : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 2 1 1 2 1 1 ...
##  $ Revenue                : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "na.action")= 'omit' Named int [1:12] 1050 1116 1117 1118 1119 1443 1444 1445 1446 1996 .
##   ..- attr(*, "names")= chr [1:12] "1066" "1133" "1134" "1135" ...
```

3.Exploratory Data Analysis

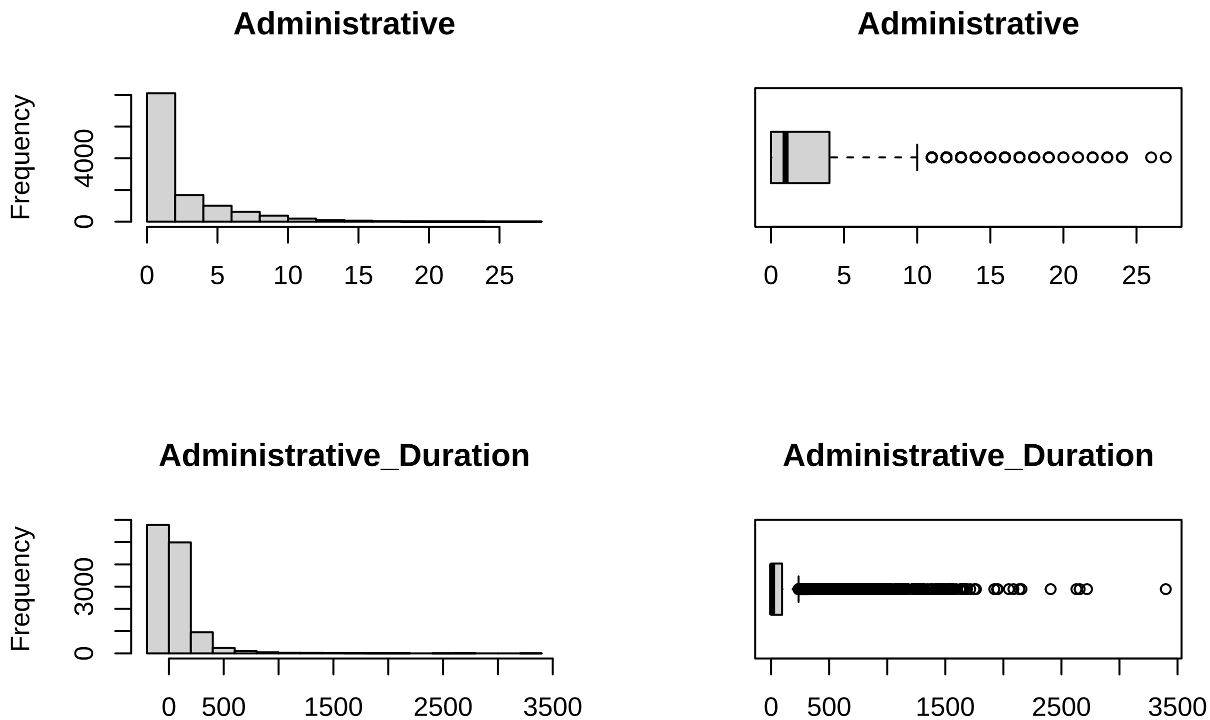
```r
# summary of the data
summary(data)
```

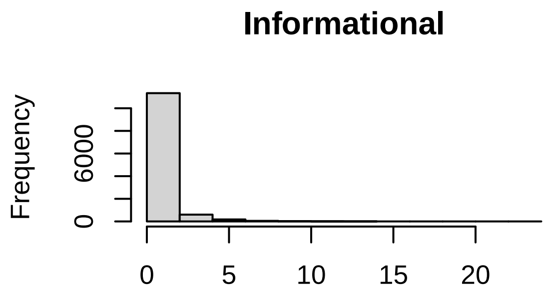
```
##  Administrative  Administrative_Duration Informational
##  Min.   : 0.00   Min.   :  -1.00         Min.   : 0.0000
##  1st Qu.: 0.00   1st Qu.:   0.00         1st Qu.: 0.0000
##  Median : 1.00   Median :   9.00         Median : 0.0000
##  Mean   : 2.34   Mean   :  81.68         Mean   : 0.5088
##  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
```

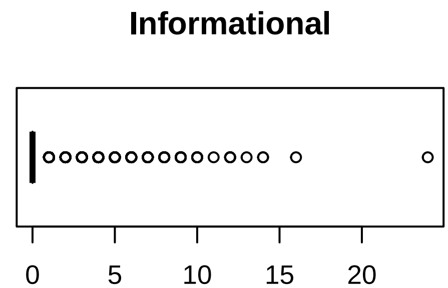
```
##  Max.   :2549.38        Max.   :705.00   Max.   :63973.5
##
##   BounceRates        ExitRates         PageValues        SpecialDay
##  Min.   :0.00000   Min.   :0.00000   Min.   :  0.000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.01422   1st Qu.:  0.000   1st Qu.:0.00000
##  Median :0.00293   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       2      :6536     2      :7878   1      :4711
##  Class :character   1      :2548     1      :2426   3      :2382
##  Mode  :character   3      :2530     4      : 730   4      :1168
##                     4      : 478     5      : 466   2      :1127
##                     8      :  75     6      : 174   6      : 800
##                     6      :  19     10     : 163   7      : 758
##                     (Other):  13     (Other): 362   (Other):1253
##   TrafficType   VisitorType         Weekend       Revenue
##  2      :3907   Length:12199       FALSE:9343   FALSE:10291
##  1      :2383   Class :character   TRUE :2856   TRUE : 1908
##  3      :2017   Mode  :character
##  4      :1066
##  13     : 728
##  10     : 450
##  (Other):1648
```

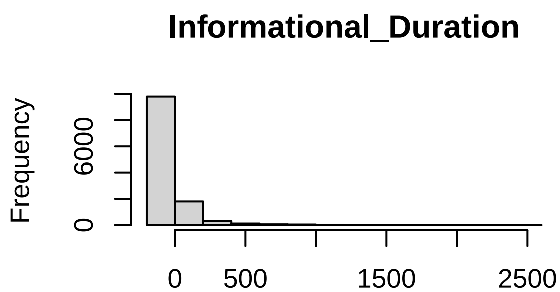
3.1 Univariate Analysis

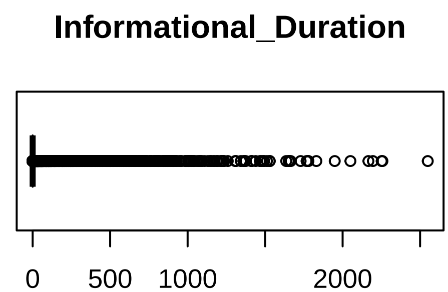
```r
# previewing the numerical variables histograms and barplots
par(mfrow=c(2,2))
for(i in 1:10) {
hist(data[, i], main=names(data)[i], xlab = NULL)
boxplot(data[,i], main=names(data)[i], horizontal = TRUE)
}
```

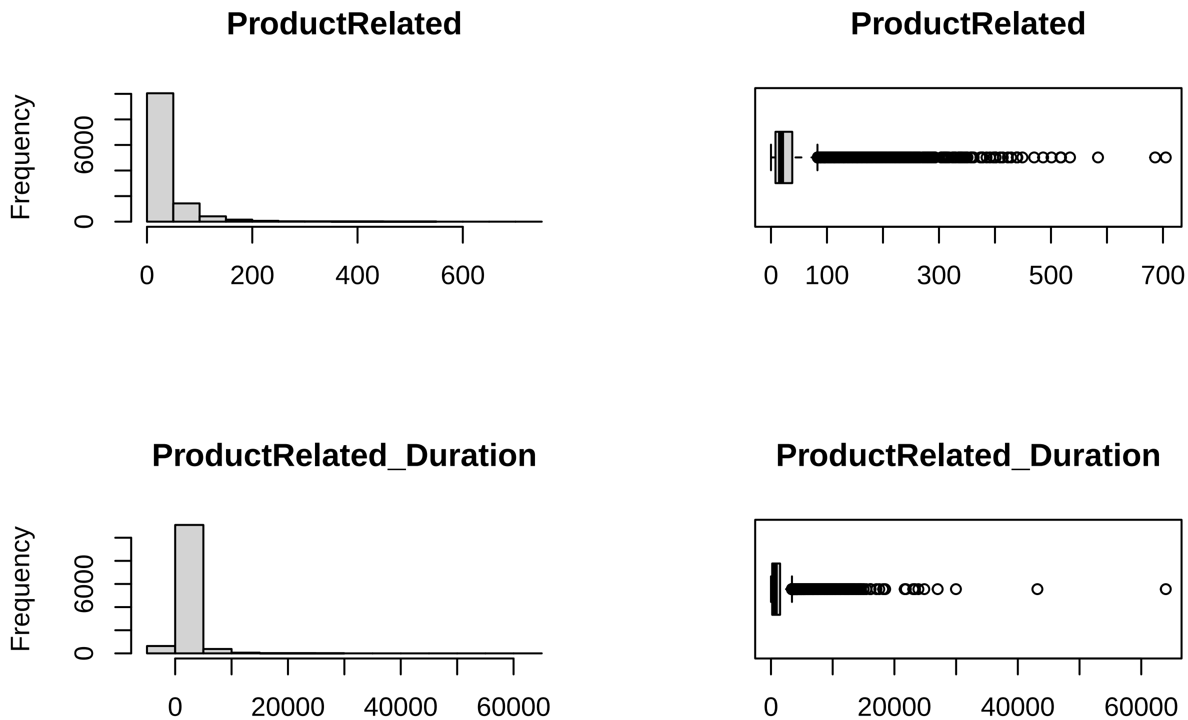

Administrative
Frequency
0
4000
0 5 10 15 20 25
Administrative
0 5 10 15 20 25
Administrative_Duration
Frequency
0
3000
0 500 1500 2500 3500
Administrative_Duration
0 500 1500 2500 3500

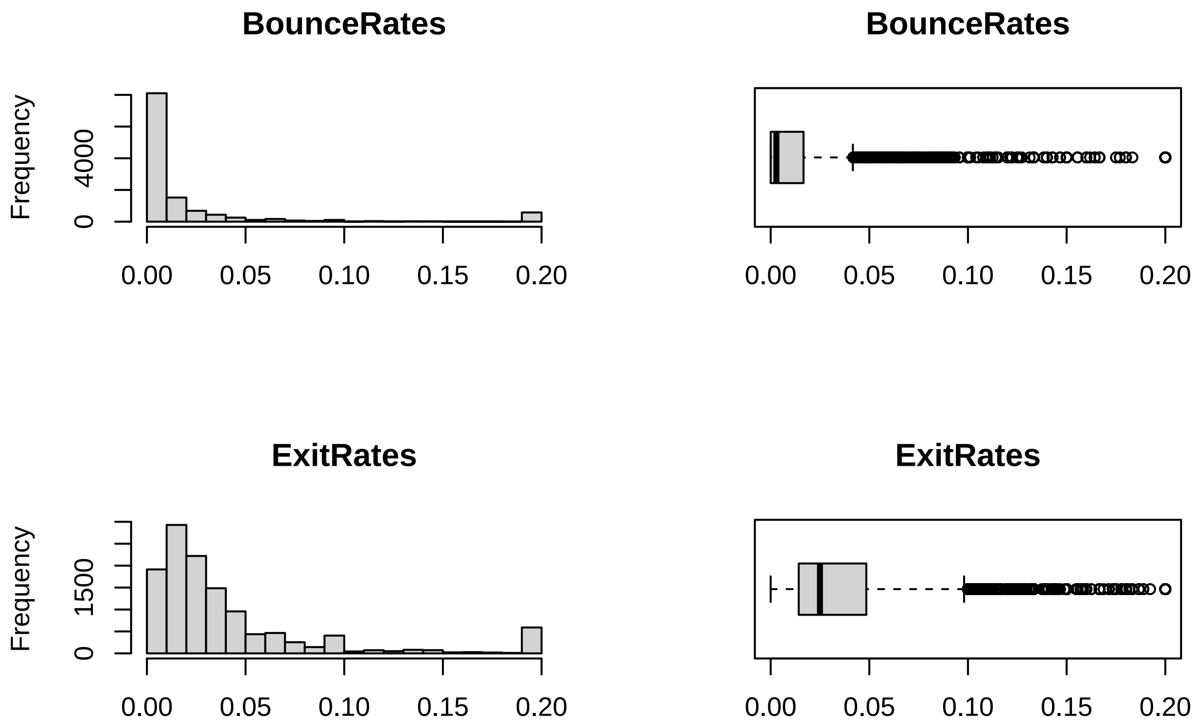

Informational
Frequency
0
6000
0
5
10
15
20

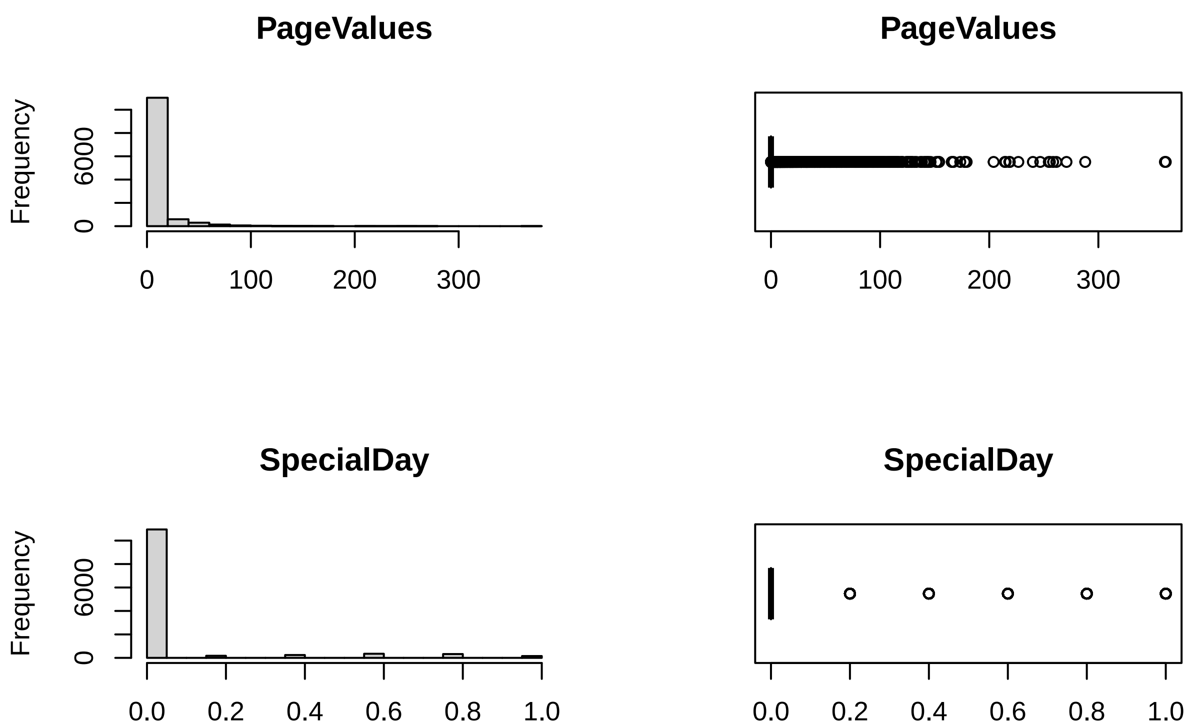

Informational
0
5
10
15
20


Informational_Duration
Frequency
0
6000
0
500
1500
2500


Informational_Duration
0
500
1000
2000


ProductRelated
Frequency
0
6000
0
200
400
600
ProductRelated
0
100
300
500
700
ProductRelated_Duration
Frequency
0
6000
0
20000
40000
60000
ProductRelated_Duration
0
20000
40000
60000


BounceRates
Frequency
0
4000
0.00
0.05
0.10
0.15
0.20
BounceRates
0.00
0.05
0.10
0.15
0.20
ExitRates
Frequency
0
1500
0.00
0.05
0.10
0.15
0.20
ExitRates
0.00
0.05
0.10
0.15
0.20

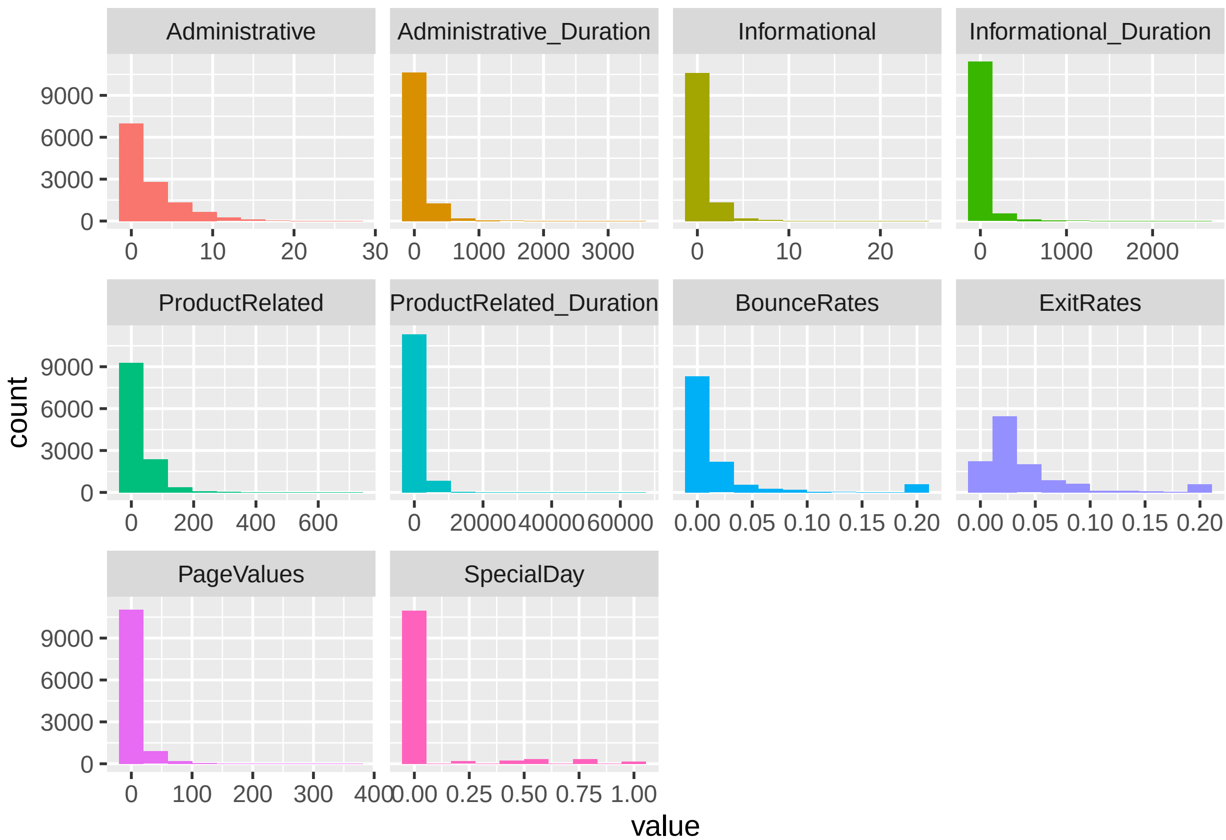



```r
# loading package that helps in creating frequency tables
library(funModeling)
```

```
## Loading required package: Hmisc

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

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

## funModeling v.1.9.4 :)
## Examples and tutorials at livebook.datascienceheroes.com
##  / Now in Spanish: librovivodecienciadedatos.ai
```

```r
#continuous variables histograms
data_num <- data[1:10]
plot_num(data_num)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



the values are positively skewed

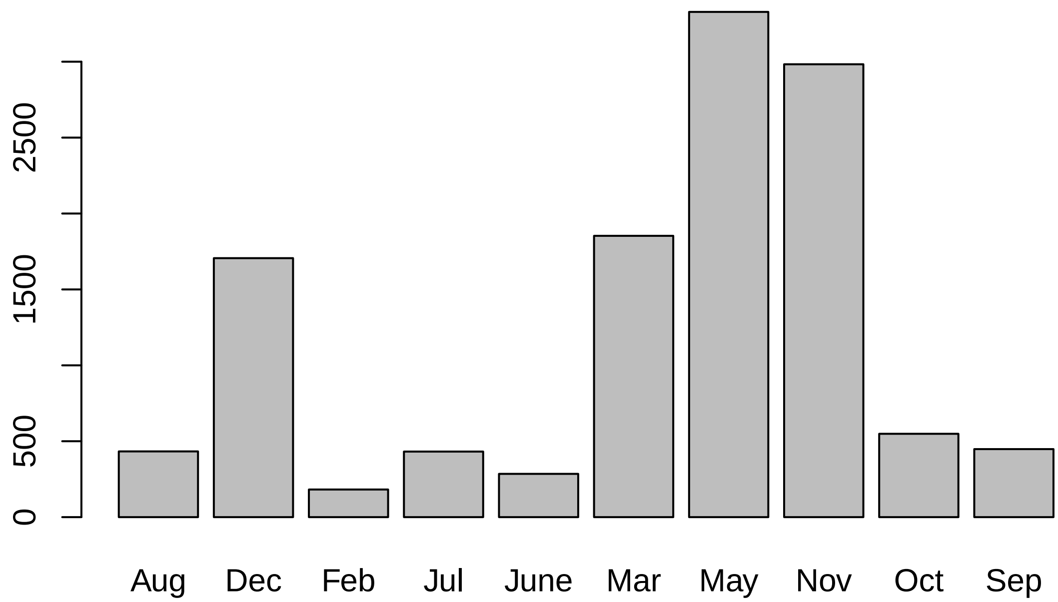
```r
# creating tables for our categorical values
month_table <- table(data$Month)
os_table <- table(data$OperatingSystems)
browser_table <- table(data$Browser)
region_table <- table(data$Region)
traffic_table <- table(data$TrafficType)
visitor_table <- table(data$VisitorType)
weekend_table <- table(data$Weekend)
revenue_table <- table(data$Revenue)
```

```r
# adjusting plot size
set_plot_dimensions <- function(width_choice, height_choice) {
options(repr.plot.width = width_choice, repr.plot.height = height_choice)
}
```

```r
# barplot of Month
set_plot_dimensions(5, 4)
month_table
```

```
##
##  Aug  Dec  Feb  Jul June  Mar  May  Nov  Oct  Sep
##  433 1706  182  432  285 1853 3328 2983  549  448
```

```r
barplot(month_table)
```

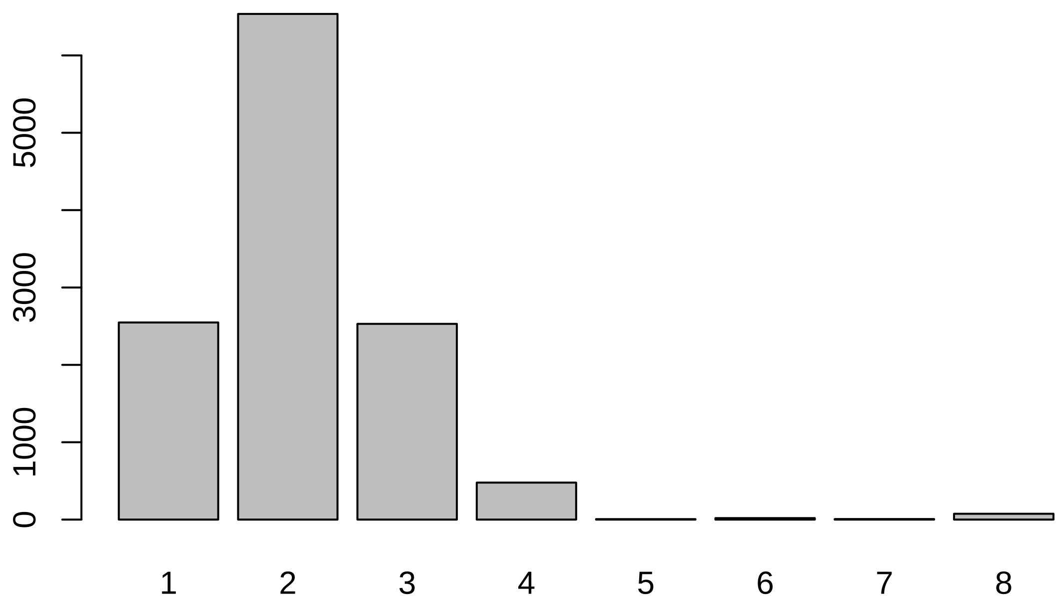


May is the most frequently occuring month with February being the least

```r
# barplot of Operating System
set_plot_dimensions(5, 4)
os_table
```

```
##
##    1    2    3    4    5    6    7    8
## 2548 6536 2530  478    6   19    7   75
```

```r
barplot(os_table)
```

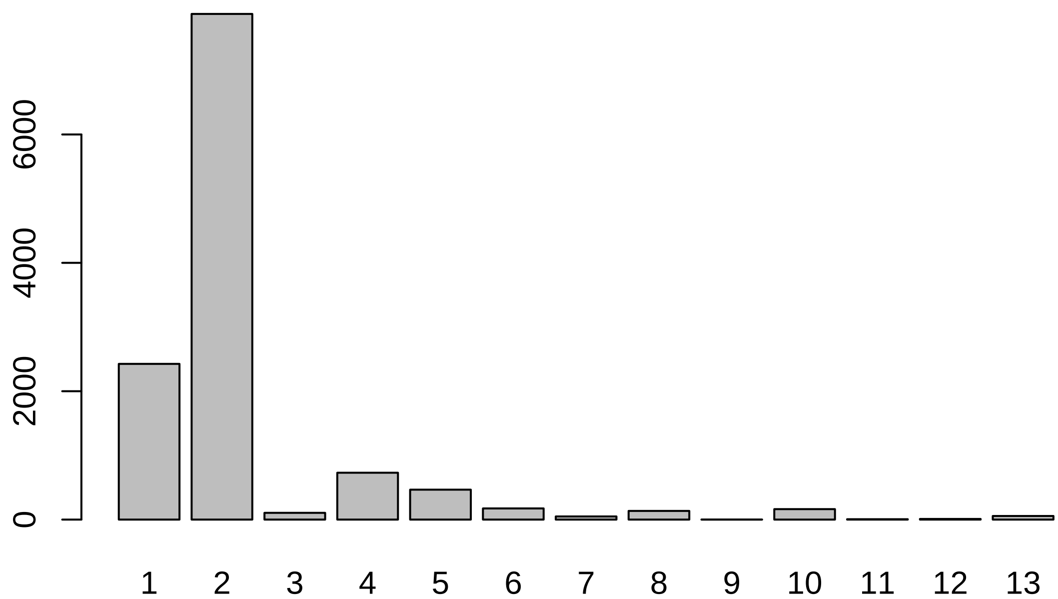


The most used OS is Operating System 2 with the least used is OS 5

```
# barplot of Browser
set_plot_dimensions(5, 4)
browser_table
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   12   13
## 2426 7878  105  730  466  174   49  135    1  163    6   10   56
```

```
barplot(browser_table)
```

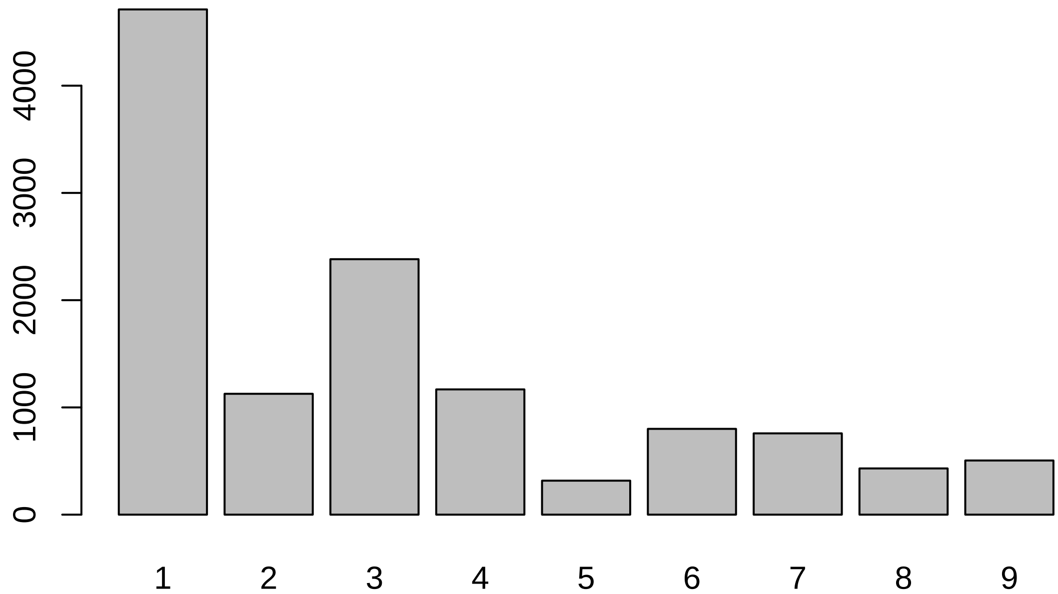


The most used browser is browser 2 and browser 9 was the least used browser

```
# barplot of Region
set_plot_dimensions(5, 4)
region_table
```

```
##
##    1    2    3    4    5    6    7    8    9
## 4711 1127 2382 1168  317  800  758  431  505
```

```
barplot(region_table)
```

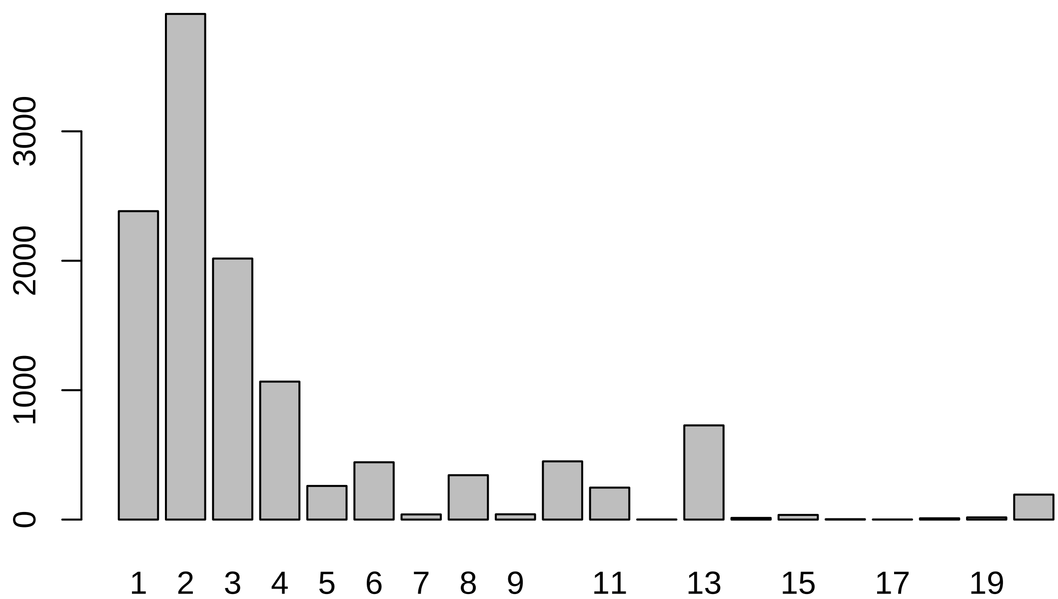


The most occuring region is region 1 and region 5 being the least occuring

```
# barplot of TrafficType
set_plot_dimensions(5, 4)
traffic_table
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 2383 3907 2017 1066  260  443   40  343   41  450  247    1  728   13   36    3
##   17   18   19   20
##    1   10   17  193
```

```
barplot(traffic_table)
```

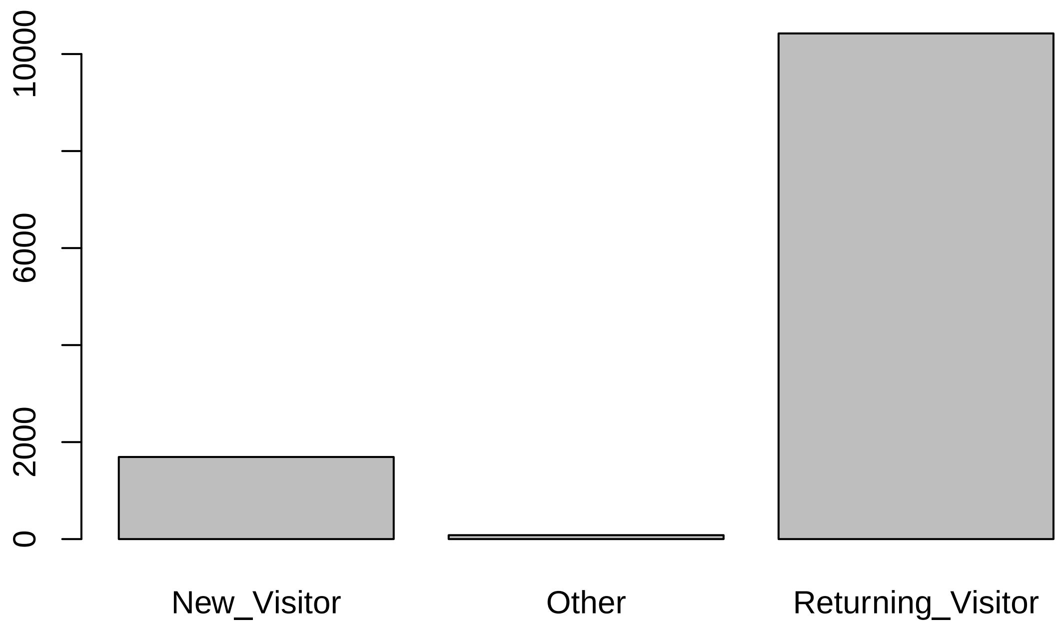


Traffic type 2 is frequently occuring with traffic type 12 and 17 being the least occuring

```
# barplot of VisitorType
set_plot_dimensions(5, 4)
visitor_table
```

```
##
##       New_Visitor             Other Returning_Visitor
##              1693                81             10425
```

```
barplot(visitor_table)
```

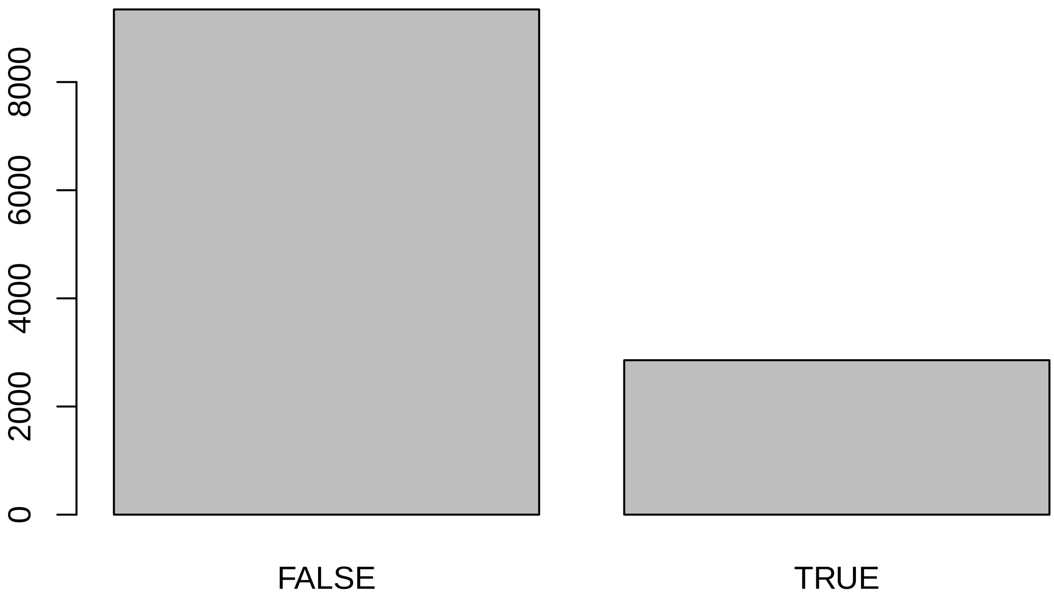


Most of the visitors did return

```
# barplot of Weekend
set_plot_dimensions(5, 4)
weekend_table
```

```
##
## FALSE  TRUE
##  9343  2856
```

```
barplot(weekend_table)
```

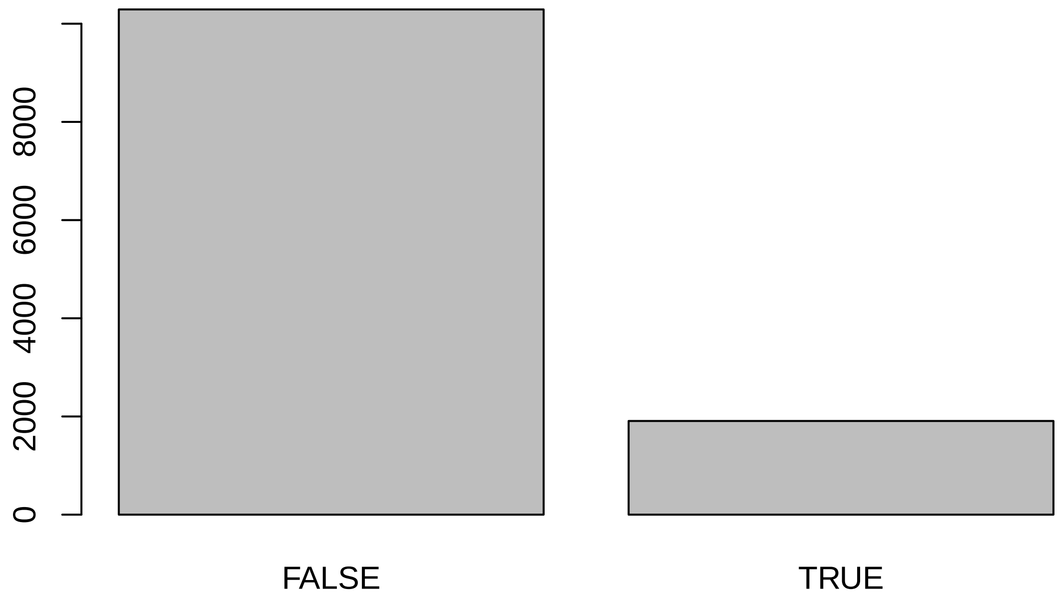


Weekdays appear to be more than the weekends

```
# barplot of Revenue
set_plot_dimensions(5, 4)
revenue_table
```

```
##
## FALSE  TRUE
## 10291  1908
```

```
barplot(revenue_table)
```

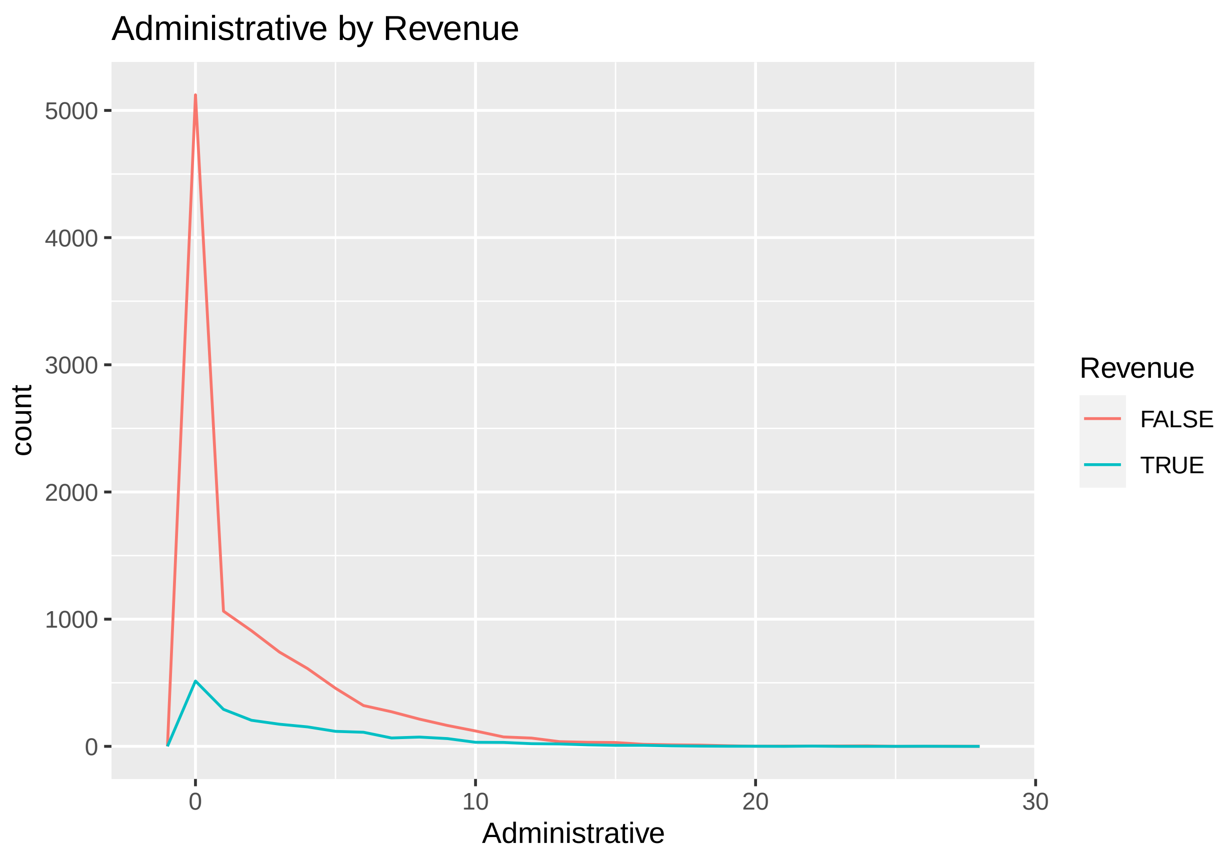


The false revenues are more than the true ones

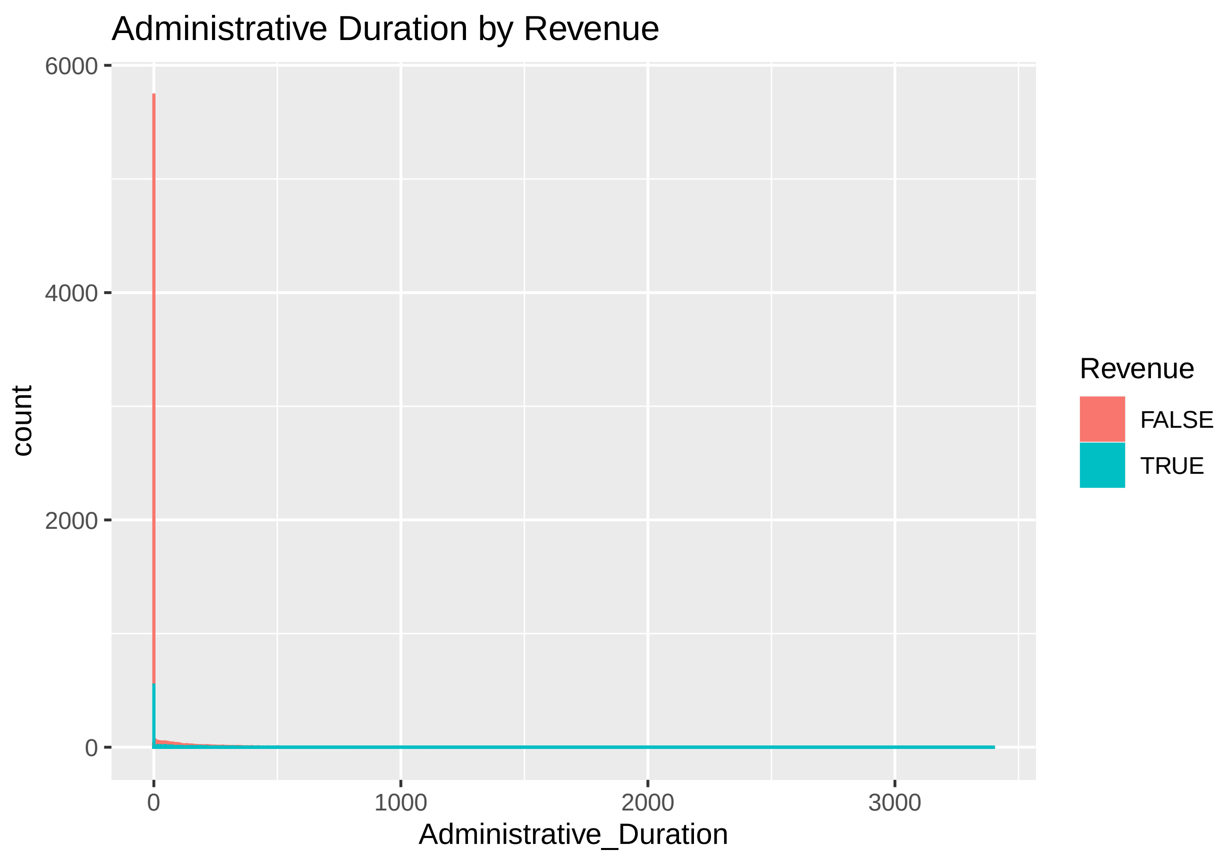
3.2 Bivariate Analysis

```
# importing library
library(ggplot2)
```

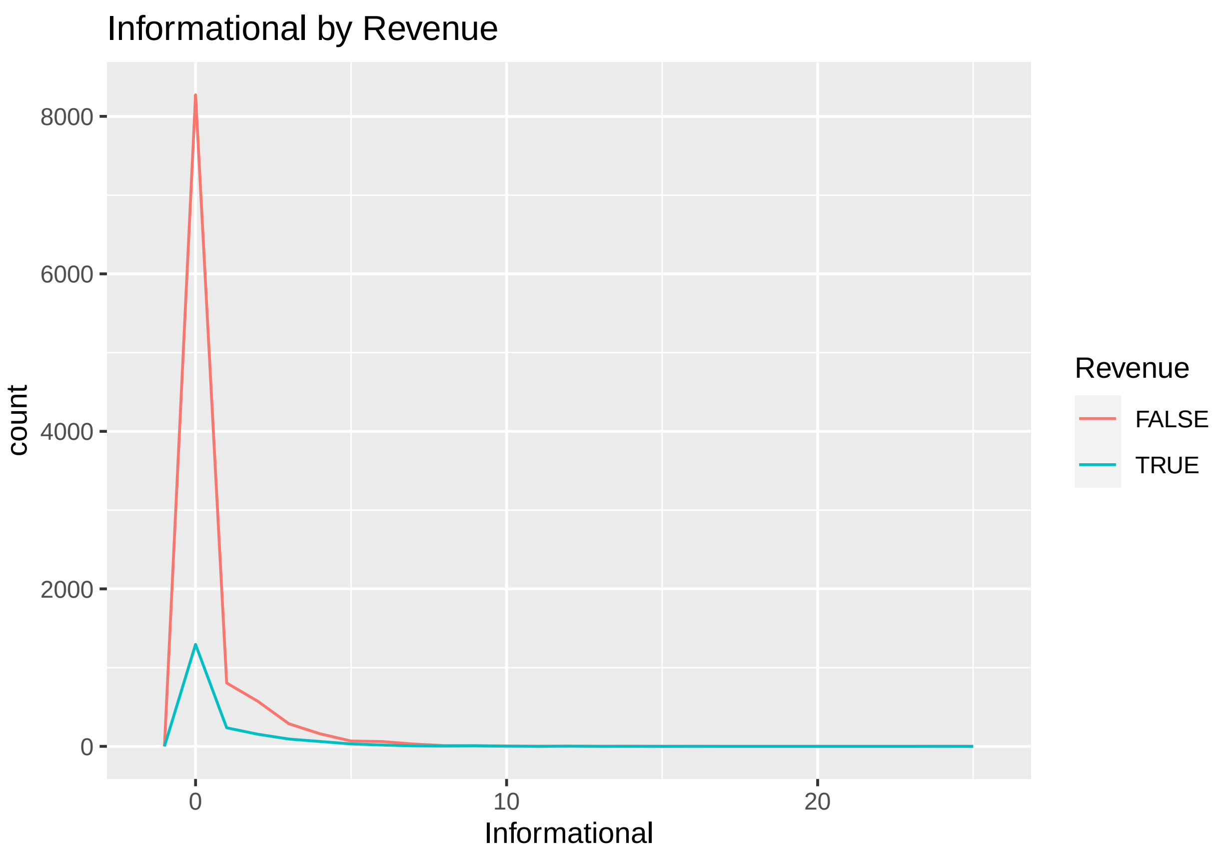
```
# plotting Administrative by Revenue
set_plot_dimensions(5, 4)
ggplot(data, aes(x = Administrative, fill = Revenue, color = Revenue)) +
geom_freqpoly(binwidth = 1) +
labs(title = "Administrative by Revenue")
```



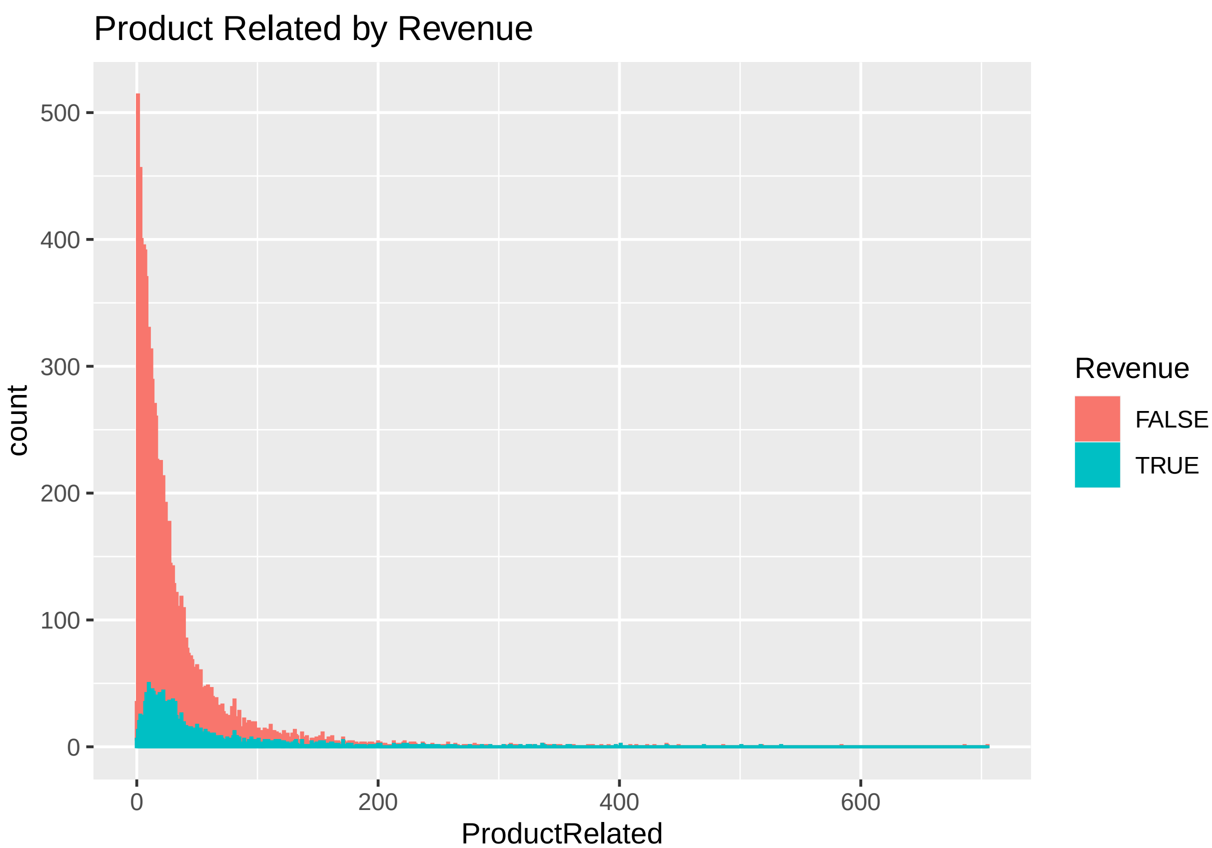

```r
# plotting Administrative Duration by Revenue
set_plot_dimensions(5, 4)
ggplot(data, aes(x = Administrative_Duration, fill = Revenue, color = Revenue)) +
geom_histogram(binwidth = 1) +
labs(title = "Administrative Duration by Revenue")
```



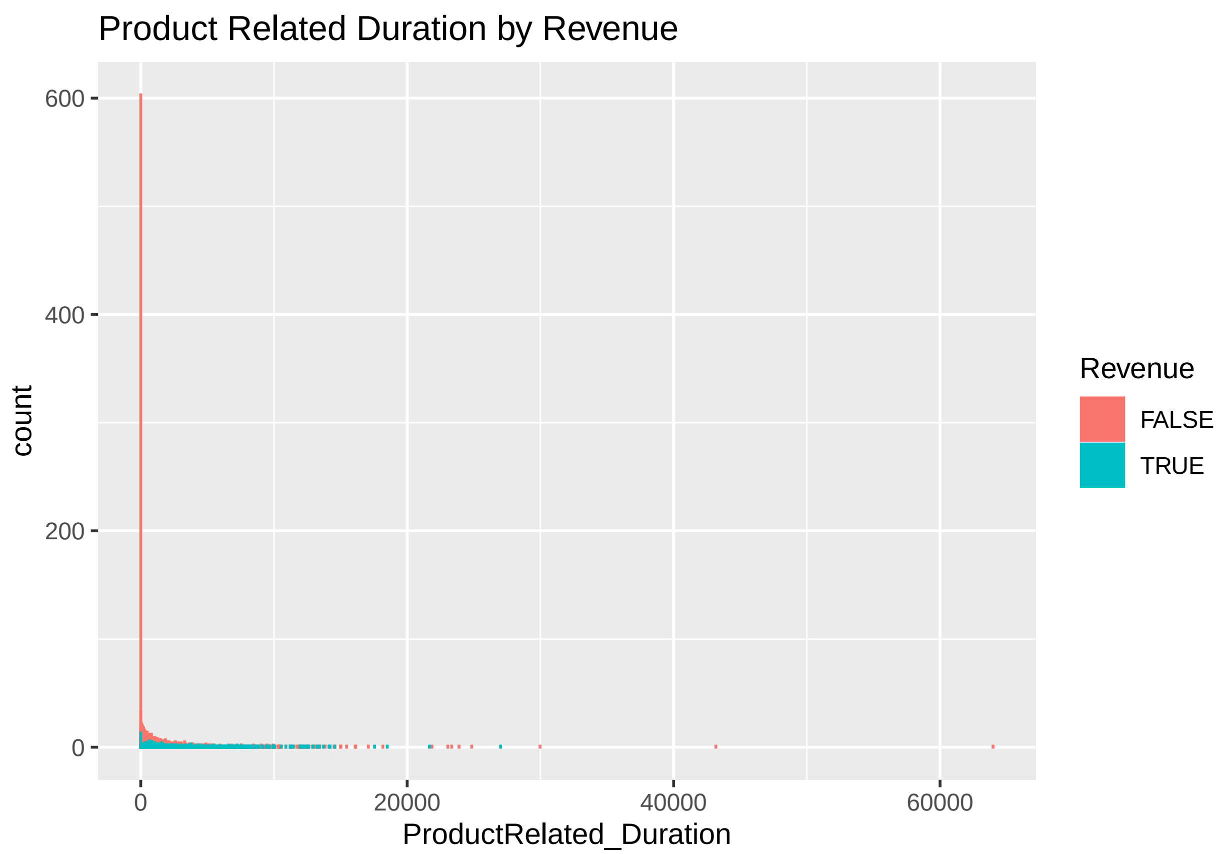

```r
# plotting Informational by Revenue
set_plot_dimensions(5, 4)
ggplot(data, aes(x = Informational, fill = Revenue, color = Revenue)) +
geom_freqpoly(binwidth = 1) +
labs(title = "Informational by Revenue")
```



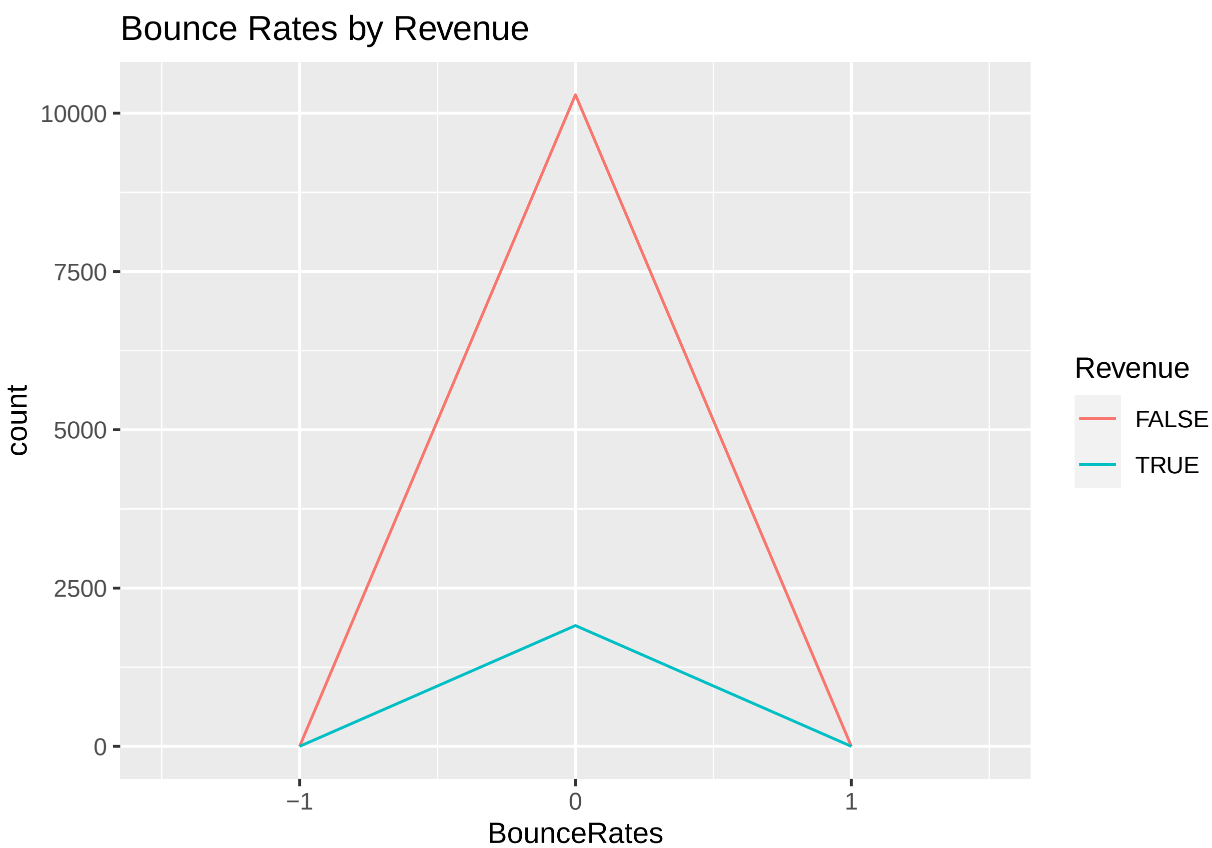

```
# plotting product related by Revenue
set_plot_dimensions(5, 4)
ggplot(data, aes(x = ProductRelated, fill = Revenue, color = Revenue)) +
geom_histogram(binwidth = 1) +
labs(title = "Product Related by Revenue")
```



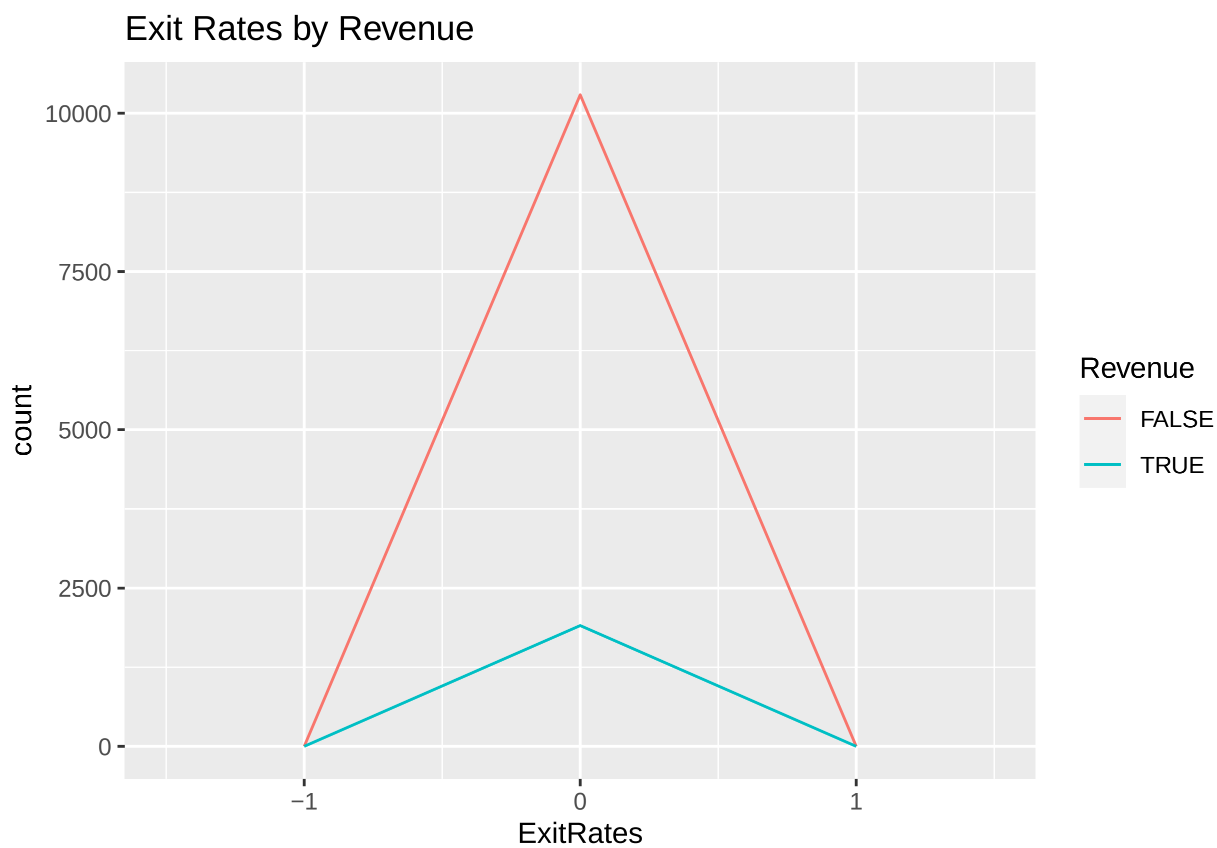

```r
# plotting product related duration by Revenue
set_plot_dimensions(5, 4)
ggplot(data, aes(x = ProductRelated_Duration, fill = Revenue, color = Revenue)) +
geom_histogram(binwidth = 1) +
labs(title = "Product Related Duration by Revenue")
```



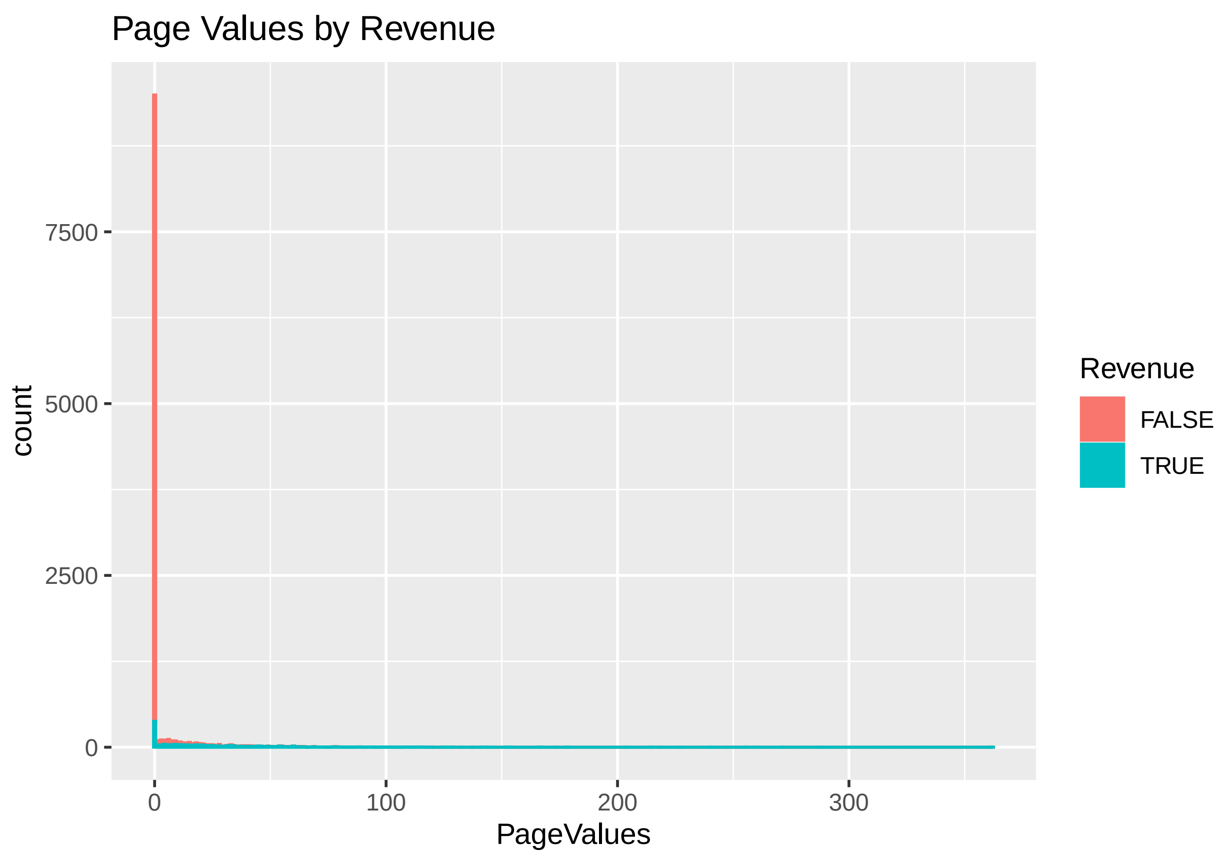

```
# plotting bounce rates by Revenue
set_plot_dimensions(5, 4)
ggplot(data, aes(x = BounceRates, fill = Revenue, color = Revenue)) +
geom_freqpoly(binwidth = 1) +
labs(title = "Bounce Rates by Revenue")
```



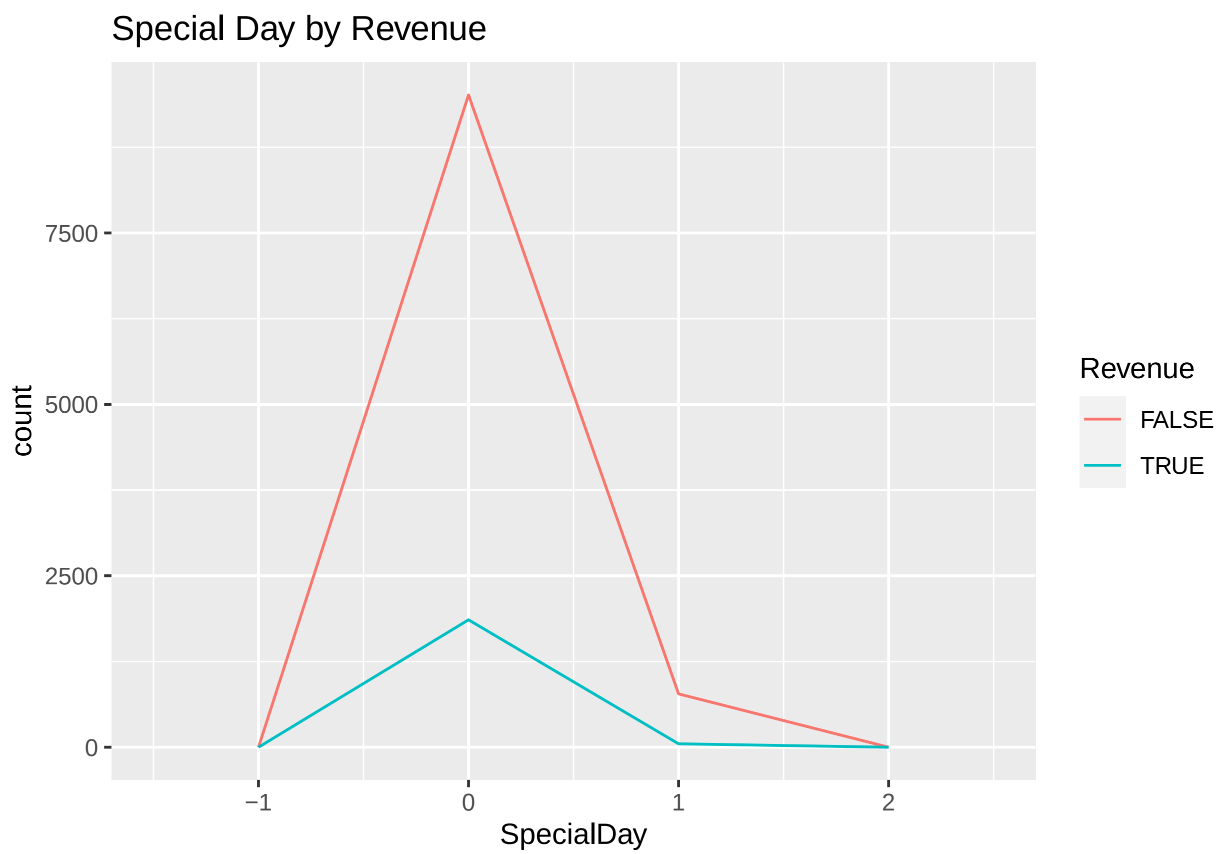

```
# plotting exit rates by Revenue
set_plot_dimensions(5, 4)
ggplot(data, aes(x = ExitRates, fill = Revenue, color = Revenue)) +
geom_freqpoly(binwidth = 1) +
labs(title = "Exit Rates by Revenue")
```



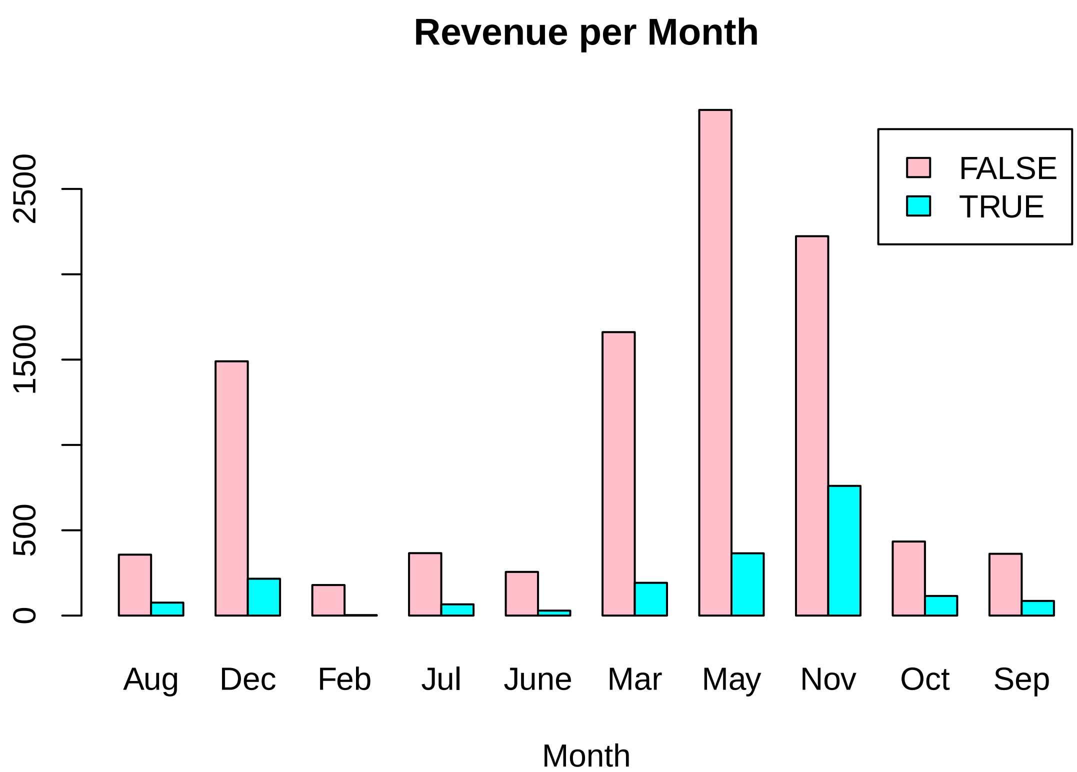

```r
# plotting page Values by Revenue
set_plot_dimensions(5, 4)
ggplot(data, aes(x = PageValues, fill = Revenue, color = Revenue)) +
geom_histogram(binwidth = 1) +
labs(title = "Page Values by Revenue")
```




```r
# plotting special day by Revenue
set_plot_dimensions(5, 4)
ggplot(data, aes(x = SpecialDay, fill = Revenue, color = Revenue)) +
geom_freqpoly(binwidth = 1) +
labs(title = "Special Day by Revenue")
```

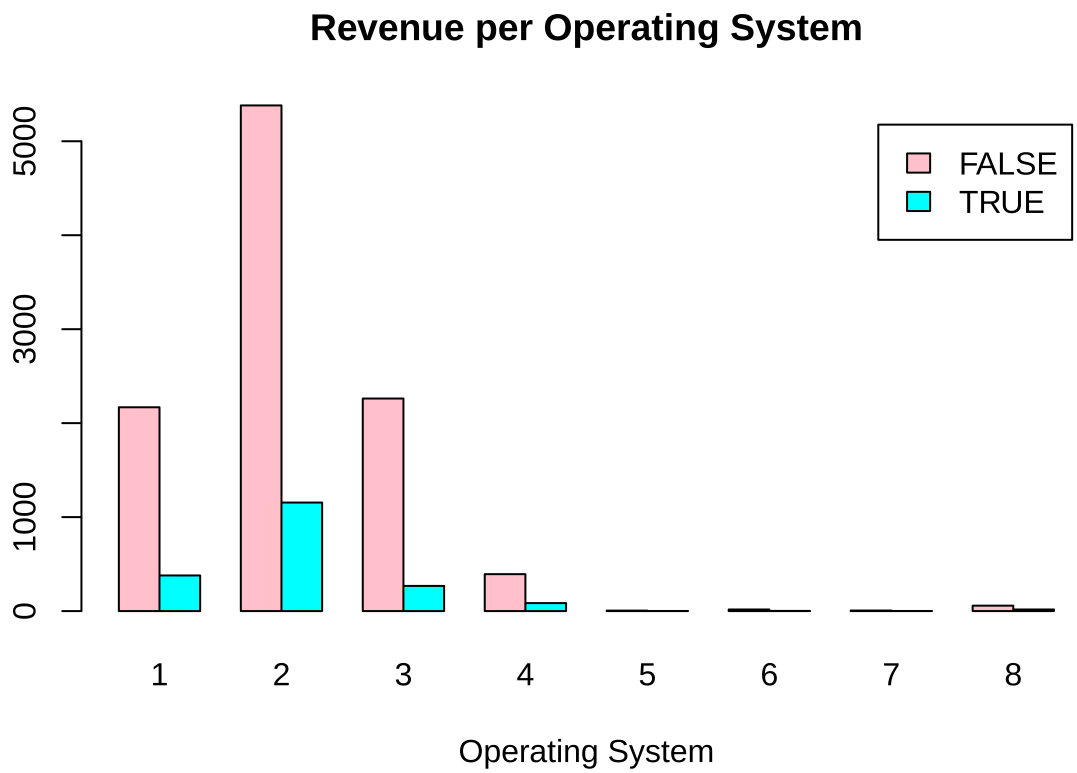



```r
# plotting the distribution of Revenue per Month
set_plot_dimensions(6, 6)
rev_month <- table(data$Revenue, data$Month)
barplot(rev_month, main = "Revenue per Month", col = c("pink", "cyan"), beside = TRUE,
legend = rownames(rev_month), xlab = "Month")
```



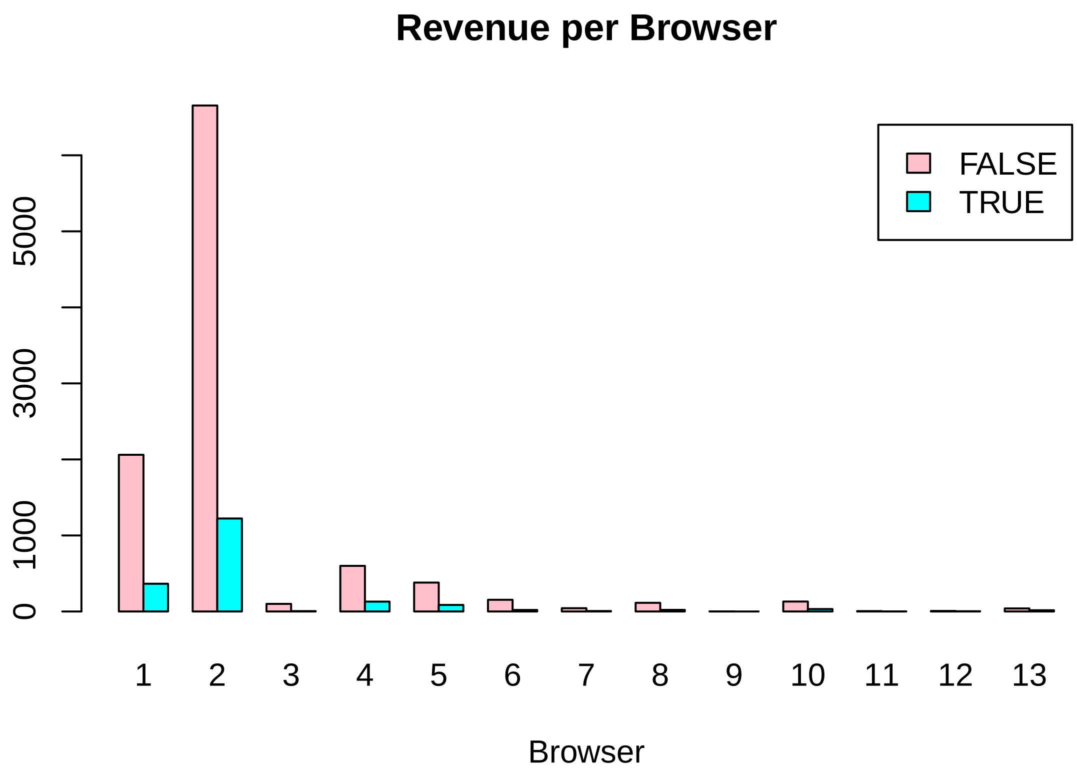
November returned the highest amounts of revenue with February returning the least amount

```
# plotting the distribution of Revenue per Operating System
set_plot_dimensions(6, 6)
rev_os <- table(data$Revenue, data$OperatingSystems)
barplot(rev_os, main = "Revenue per Operating System", col = c("pink", "cyan"), beside = TRUE,
legend = rownames(rev_os), xlab = "Operating System")
```



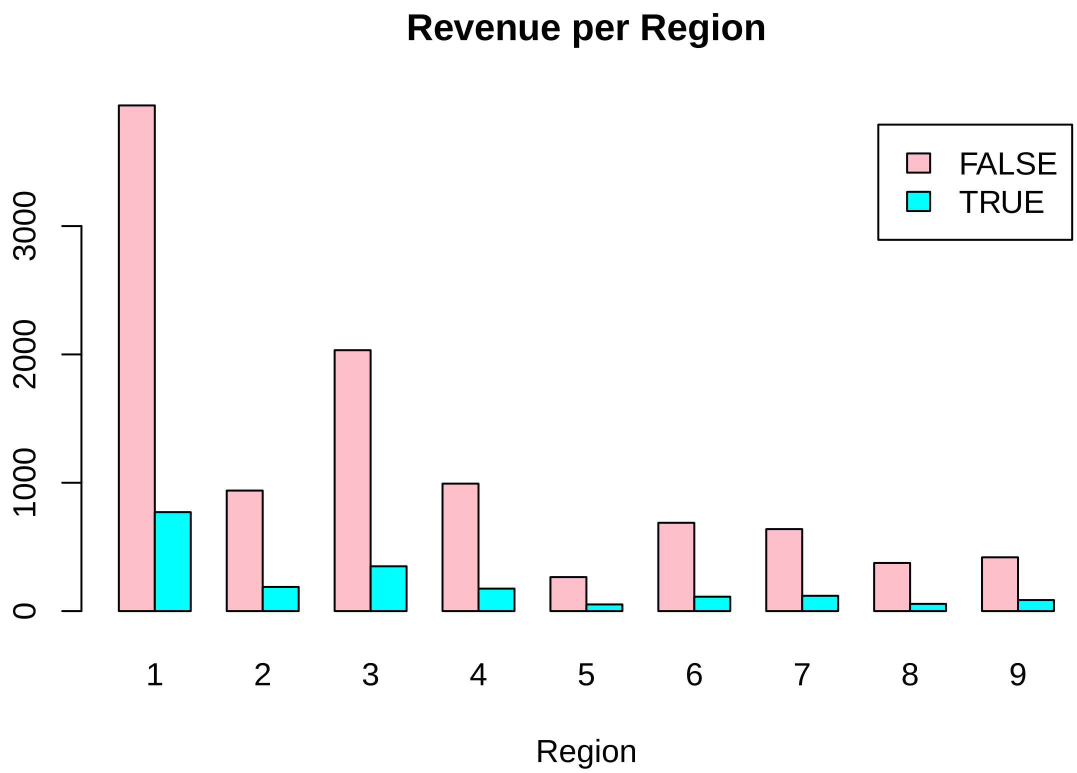
OS 2 returned the highest amount of revenue

```r
# plotting the distribution of Revenue per Browser
set_plot_dimensions(6, 6)
rev_browser <- table(data$Revenue, data$Browser)
barplot(rev_browser, main = "Revenue per Browser", col = c("pink", "cyan"), beside = TRUE,
legend = rownames(rev_browser), xlab = "Browser")
```



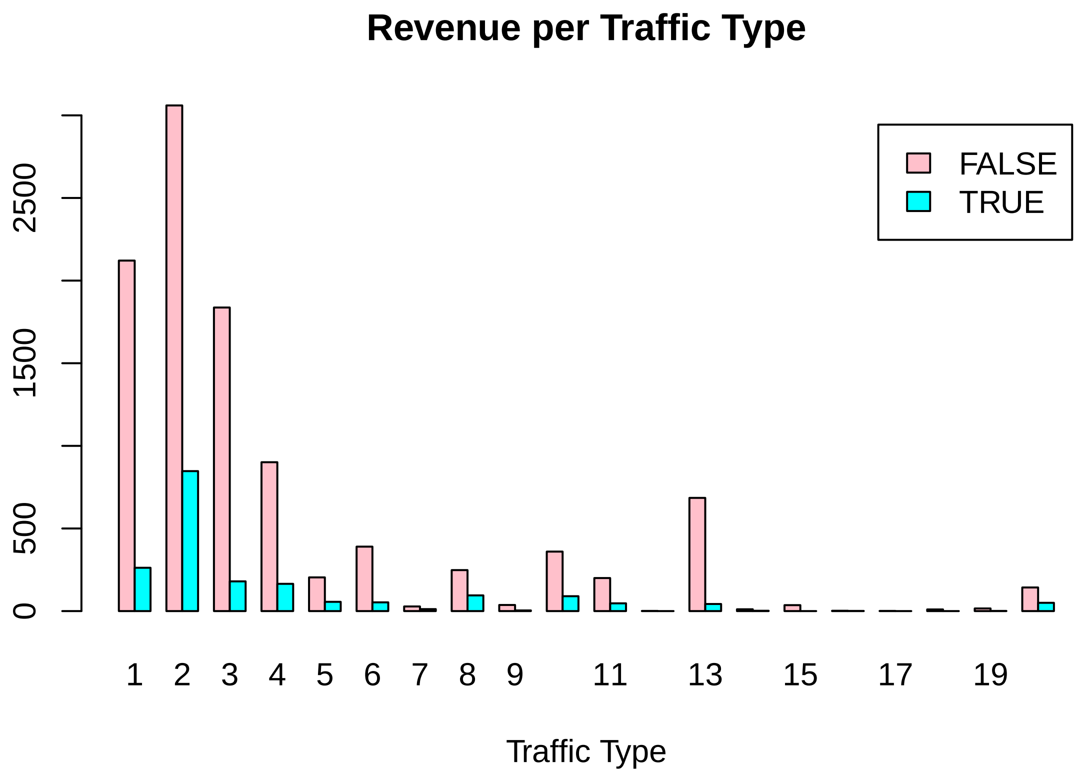
Browser 2 returns the highest revenue

```r
# plotting the distribution of Revenue per Region
set_plot_dimensions(6, 6)
rev_region <- table(data$Revenue, data$Region)
barplot(rev_region, main = "Revenue per Region", col = c("pink", "cyan"), beside = TRUE,
legend = rownames(rev_region), xlab = "Region")
```



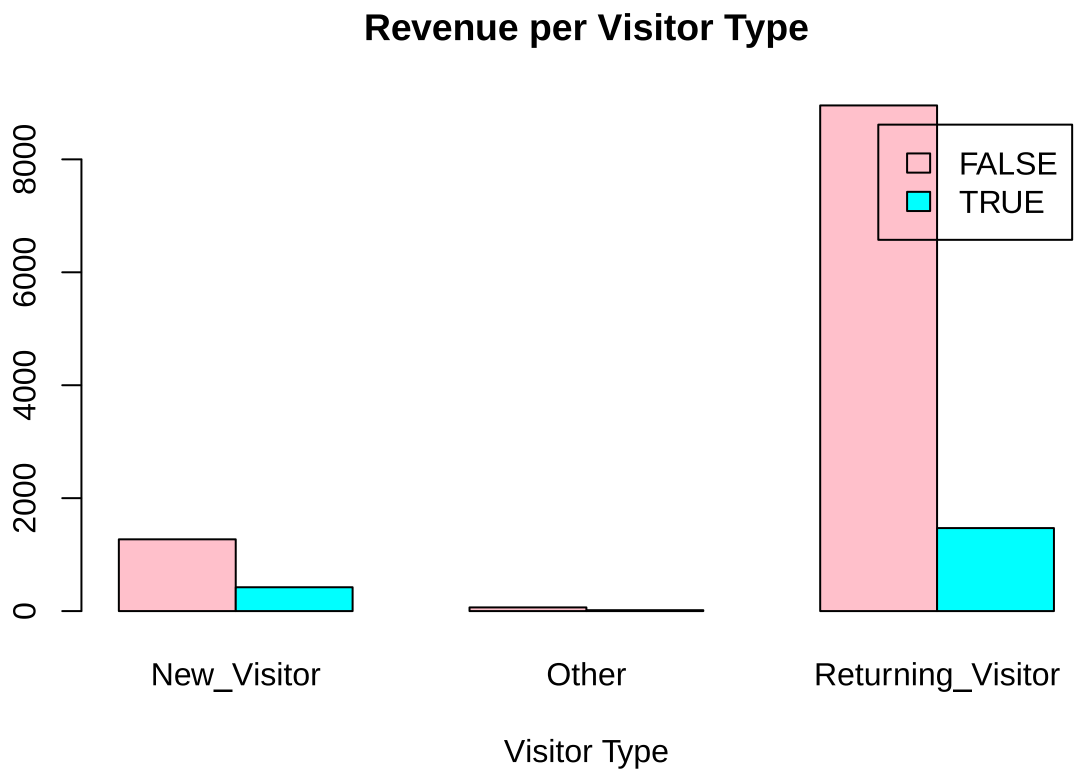
Region 1 returns the highest revenue amount and region 5 returning the least amount

```
# plotting the distribution of Revenue per Traffic Type
set_plot_dimensions(6, 6)
rev_traffic <- table(data$Revenue, data$TrafficType)
barplot(rev_traffic, main = "Revenue per Traffic Type", col = c("pink", "cyan"), beside = TRUE,
legend = rownames(rev_traffic), xlab = "Traffic Type")
```



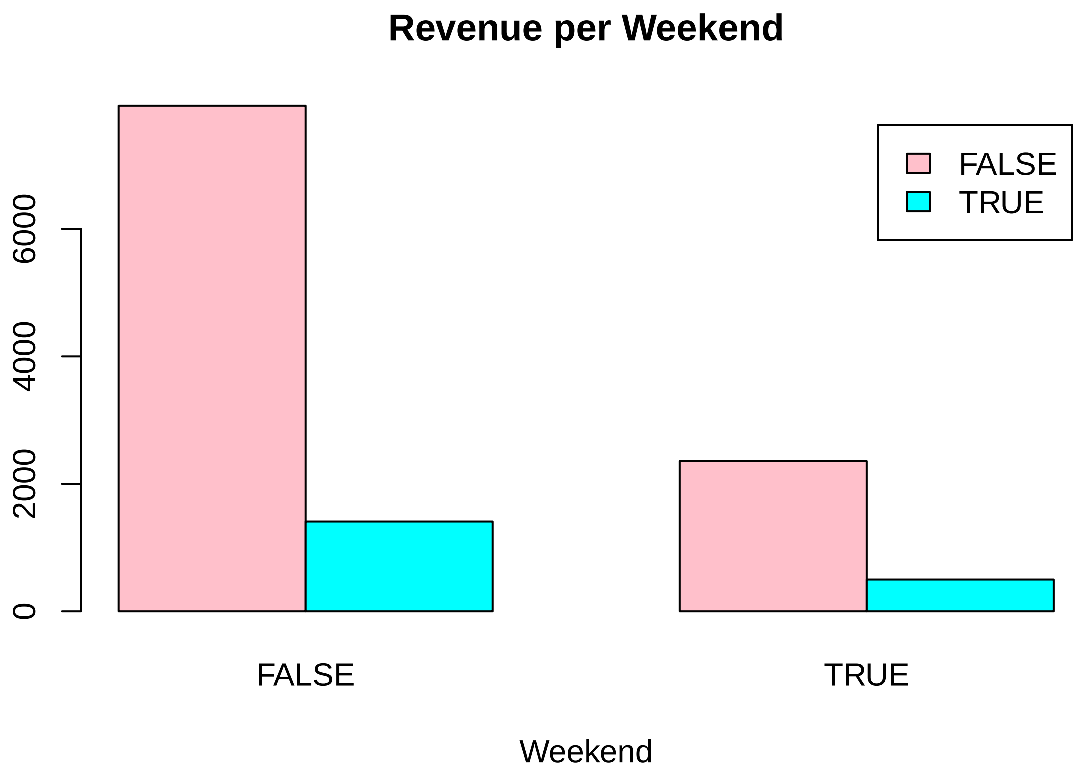
Traffic Type 2 returns the highest revenue amount

```
# plotting the distribution of Revenue per Visitor Type
set_plot_dimensions(6, 6)
rev_visitor <- table(data$Revenue, data$VisitorType)
barplot(rev_visitor, main = "Revenue per Visitor Type", col = c("pink", "cyan"), beside = TRUE,
legend = rownames(rev_visitor), xlab = "Visitor Type")
```



returning visitors brought the biggest revenue

```
# plotting the distribution of Revenue per Weekend
set_plot_dimensions(6, 6)
rev_weekend <- table(data$Revenue, data$Weekend)
barplot(rev_weekend, main = "Revenue per Weekend", col = c("pink", "cyan"), beside = TRUE,
legend = rownames(rev_weekend), xlab = "Weekend")
```



Weekdays return biggest revenue

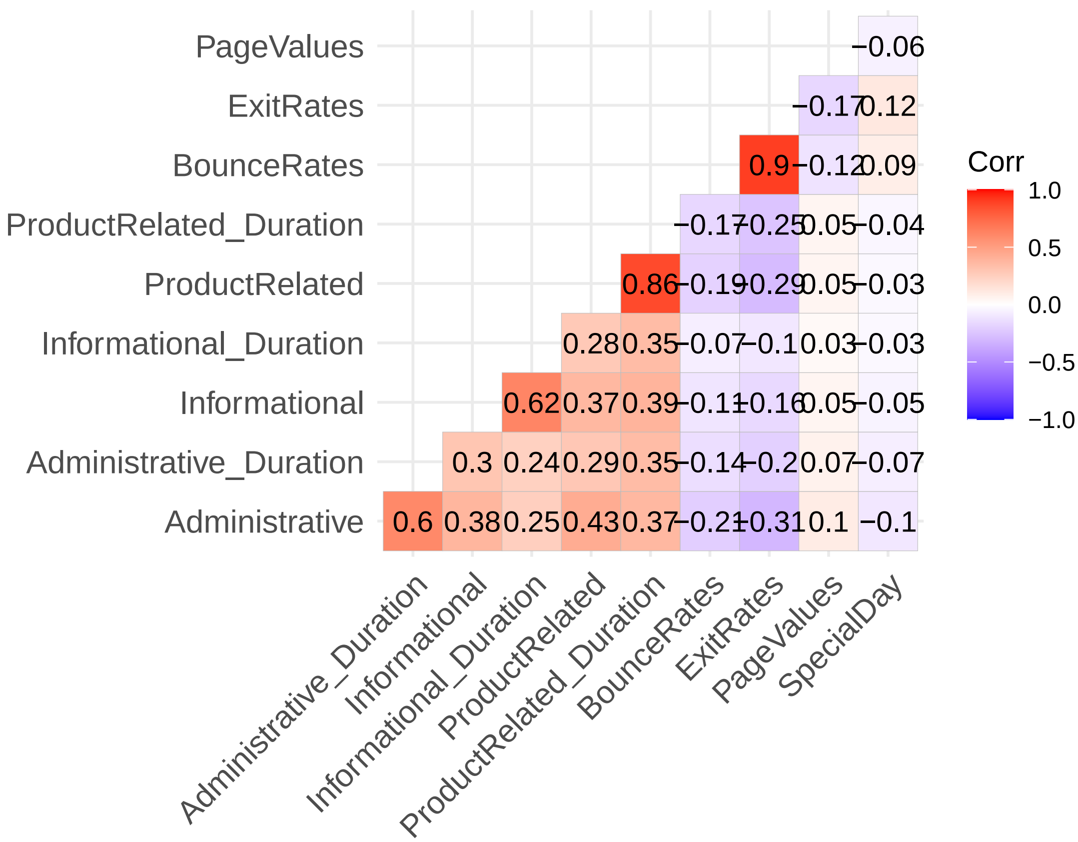
Checking for correlation

```r
# geting the numerical columns
data_num <- data[,1:10]
head(data_num)
```

```
##   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
## 1          0
```

```
## 2          0
## 3          0
## 4          0
## 5          0
## 6          0
```

```r
# creating a heatmap
library(ggcorrplot)
set_plot_dimensions(6, 6)
corr_data <- cor(data_num)
ggcorrplot(round(corr_data, 2) ,lab = T,type = 'lower')
```



BounceRates is very highly correlated to ExitRates Administrative is correlated with Administrative_Duration Informational is highly correlated to Informational_Duration ProductRelated is highly correlated with ProductRated_Duration To reduce redundancy and dimensionality, we will drop one variable of each of the highly correlated pairs.

```r
# dropping the highly correlated columns
to_drop <- c("Administrative_Duration", "Informational_Duration", "ProductRelated_Duration", "ExitRates
data <- data[, !names(data) %in% to_drop]
head(data)
```

```
##   Administrative Informational ProductRelated BounceRates PageValues SpecialDay
## 1              0             0              1  0.20000000          0          0
## 2              0             0              2  0.00000000          0          0
## 3              0             0              1  0.20000000          0          0
```

```
## 4             0             0              2  0.05000000          0          0
## 5             0             0             10  0.02000000          0          0
## 6             0             0             19  0.01578947          0          0
##   Month OperatingSystems Browser Region TrafficType       VisitorType Weekend
## 1   Feb                1       1      1           1 Returning_Visitor   FALSE
## 2   Feb                2       2      1           2 Returning_Visitor   FALSE
## 3   Feb                4       1      9           3 Returning_Visitor   FALSE
## 4   Feb                3       2      2           4 Returning_Visitor   FALSE
## 5   Feb                3       3      1           4 Returning_Visitor    TRUE
## 6   Feb                2       2      1           3 Returning_Visitor   FALSE
##   Revenue
## 1   FALSE
## 2   FALSE
## 3   FALSE
## 4   FALSE
## 5   FALSE
## 6   FALSE
```

```r
# getting the numerical columns from the new dataframe
data_num <- data[,1:6]
head(data_num)
```

```
##   Administrative Informational ProductRelated BounceRates PageValues SpecialDay
## 1             0             0              1  0.20000000          0          0
## 2             0             0              2  0.00000000          0          0
## 3             0             0              1  0.20000000          0          0
## 4             0             0              2  0.05000000          0          0
## 5             0             0             10  0.02000000          0          0
## 6             0             0             19  0.01578947          0          0
```

```r
# visualizing the correlation of the new dataset
set_plot_dimensions(6, 6)
new_corr_data <- cor(data_num)
ggcorrplot(round(new_corr_data, 2) ,lab = T,type = 'lower')
```

The dropping of the variables has reduced multicollinearity in our set

4. Modelling

4.1 Feature Engineering

```r
# importing libraries
library(lattice)
library(caret)
```

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

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

```r
# shuffling our data to randomize the records
shuffle_index <- sample(1:nrow(data))
data <- data[shuffle_index, ]
dim(data)
```

```
## [1] 12199    14
```

```r
# Normalizing our dataset
normalize <- function(x){
return ((x-min(x)) / (max(x)-min(x)))
}
data$Administrative <- normalize(data$Administrative)
data$Informational <- normalize(data$Informational)
data$ProductRelated <- normalize(data$ProductRelated)
data$BounceRates <- normalize(data$BounceRates)
data$PageValues <- normalize(data$PageValues)
data$SpecialDay <- normalize(data$SpecialDay)
head(data)
```

```
##       Administrative Informational ProductRelated BounceRates PageValues
## 4690      0.03703704    0.08333333    0.005673759  0.00000000 0.00000000
## 5009      0.25925926    0.00000000    0.038297872  0.00000000 0.00000000
## 10428     0.03703704    0.00000000    0.089361702  0.01587301 0.02333665
## 8603      0.33333333    0.08333333    0.214184397  0.06410257 0.00000000
## 5350      0.00000000    0.00000000    0.008510638  0.05555555 0.00000000
## 331       0.00000000    0.00000000    0.001418440  1.00000000 0.00000000
##       SpecialDay Month OperatingSystems Browser Region TrafficType
## 4690         0.0   May                2       2      6           6
## 5009         0.2   May                1       1      1           3
## 10428        0.0   Nov                2       2      1           1
## 8603         0.0   Nov                1       2      1           1
## 5350         0.2   May                2       6      6           4
## 331          0.0   Mar                3       2      3           1
##             VisitorType Weekend Revenue
## 4690  Returning_Visitor   FALSE   FALSE
## 5009  Returning_Visitor   FALSE   FALSE
## 10428 Returning_Visitor   FALSE    TRUE
## 8603  Returning_Visitor    TRUE   FALSE
## 5350  Returning_Visitor   FALSE   FALSE
## 331   Returning_Visitor   FALSE   FALSE
```

```r
# splitting our data into 70:30 training and test sets
intrain <- createDataPartition(y = data$Revenue, p = 0.7, list = FALSE)
training <- data[intrain,]
testing <- data[-intrain,]
```

```r
# checking the dimensions of our training and testing sets
dim(training)
```

```
## [1] 8540   14
```

```r
dim(testing)
```

```
## [1] 3659   14
```

```r
# checking the dimensions of our split
prop.table(table(data$Revenue)) * 100
```

```
##
##    FALSE     TRUE
## 84.35937 15.64063
```

```r
prop.table(table(training$Revenue)) * 100
```

```
##
##    FALSE     TRUE
## 84.35597 15.64403
```

```r
prop.table(table(testing$Revenue)) * 100
```

```
##
##    FALSE     TRUE
## 84.36731 15.63269
```

4.2 KNN

```r
# splitting into train and test sets without the target variable
train <- training[, -14]
test <- testing[, -14]
```

```r
#storing our train and test sets
train_rev <- training[, 14]
test_rev <- testing[, 14]
```

```r
# checking all predictor variables are numerical
train$Month <- as.numeric(train$Month)
```

```
## Warning: NAs introduced by coercion
```

```r
train$OperatingSystems <- as.numeric(train$OperatingSystems)
train$Browser <- as.numeric(train$Browser)
train$Region <- as.numeric(train$Region)
train$TrafficType <- as.numeric(train$TrafficType)
train$VisitorType <- as.numeric(train$VisitorType)
```

```
## Warning: NAs introduced by coercion
```

```r
train$Weekend <- as.numeric(train$Weekend)
test$Month <- as.numeric(test$Month)
```

```
## Warning: NAs introduced by coercion
```

```r
test$OperatingSystems <- as.numeric(test$OperatingSystems)
test$Browser <- as.numeric(test$Browser)
test$Region <- as.numeric(test$Region)
test$TrafficType <- as.numeric(test$TrafficType)
test$VisitorType <- as.numeric(test$VisitorType)
```

```
## Warning: NAs introduced by coercion
```

```r
test$Weekend <- as.numeric(test$Weekend)
```

```r
# checking the dimensions
dim(train)
```

```
## [1] 8540   13
```

```r
dim(test)
```

```
## [1] 3659   13
```

```r
length(train_rev)
```

```
## [1] 8540
```

```r
length(test_rev)
```

```
## [1] 3659
```

```r
colSums(is.na(train))
```

```
##   Administrative    Informational   ProductRelated      BounceRates
##                0                0                0                0
##       PageValues       SpecialDay            Month OperatingSystems
##                0                0             8540                0
##          Browser           Region      TrafficType      VisitorType
##                0                0                0             8540
##          Weekend
##                0
```

```r
colSums(is.na(test))
```

```
##   Administrative    Informational   ProductRelated      BounceRates
##                0                0                0                0
##       PageValues       SpecialDay            Month OperatingSystems
##                0                0             3659                0
##          Browser           Region      TrafficType      VisitorType
##                0                0                0             3659
##          Weekend
##                0
```

```r
train[is.na(train)] <- 0
test[is.na(test)] <-0
```

```r
# importing library
library(class)
require(class)
```

```r
# modelling using KNN
model <- knn(train = train, test = test,cl = train_rev, k = 20)
```

```r
table(factor(model))
```

```
##
## FALSE  TRUE
##  3624    35
```

```r
knn_table <- table(test_rev, model)
knn_table
```

```
##          model
## test_rev FALSE TRUE
##    FALSE  3076   11
##    TRUE    548   24
```

```r
# checking the accuracy
knn_acc <- sum(diag(knn_table)/(sum(rowSums(knn_table)))) * 100
print(paste("KNN accuracy score:", knn_acc))
```

```
## [1] "KNN accuracy score: 84.7226018037715"
```

The model has an 85% accuracy level

4.3 Naive Bayes

```r
# Creating objects x which holds the predictor variables and y which holds the response variables
x = training[,-14]
y = training$Revenue
```

```r
# building our model
model = train(x,y,'nb',trControl=trainControl(method='cv',number=10))
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 139
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 29
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 88
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 139
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 304
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 354

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 456

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 475

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 478

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 602

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 832

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 853

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 643

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 132

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 208

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 291

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 370

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 408

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 459

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 472

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 643

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 757
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 846

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 508

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 79

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 365

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 508

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 573

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 596

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 777

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 548

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 659

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 727

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 126

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 326

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 382

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 496

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 618

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 753
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 436

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 467

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 62

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 89

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 315

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 436

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 467

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 715

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 837

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 24

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 74

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 157

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 172

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 253

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 269

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 352

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 389
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 432

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 555

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 645

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 660

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 821

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 141

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 638

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 226

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 298

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 353

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 396

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 511

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 532

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 583

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 32

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 140

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 276
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 499

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 534

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 760
```

```
# making  the predictions
Predict <- predict(model,newdata = testing )
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1026
```

```
# checking the accuracy using the confusion matrix
confusionMatrix(Predict, testing$Revenue )
```

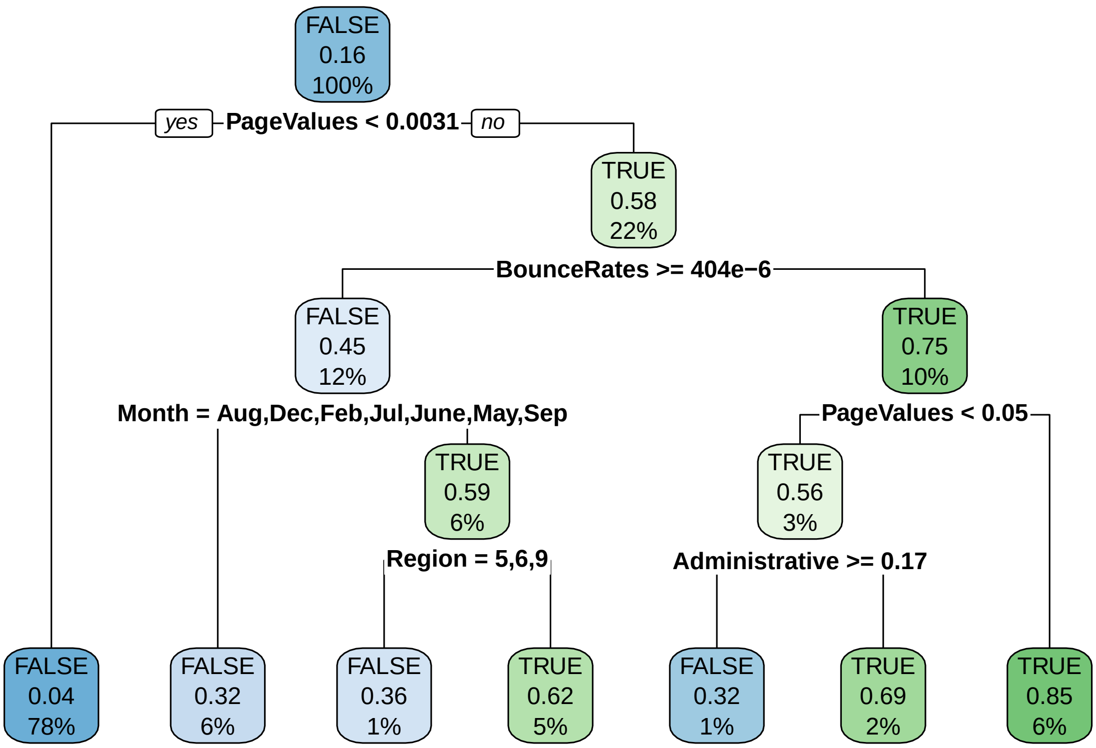
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  2818  209
##      TRUE    269  363
##
##                Accuracy : 0.8694
##                  95% CI : (0.858, 0.8801)
##     No Information Rate : 0.8437
##     P-Value [Acc > NIR] : 6.715e-06
##
##                   Kappa : 0.525
##
##  Mcnemar's Test P-Value : 0.006963
##
##             Sensitivity : 0.9129
##             Specificity : 0.6346
##          Pos Pred Value : 0.9310
##          Neg Pred Value : 0.5744
##              Prevalence : 0.8437
##          Detection Rate : 0.7702
##    Detection Prevalence : 0.8273
##       Balanced Accuracy : 0.7737
##
##        'Positive' Class : FALSE
##
```

The model has 88% accuracy

4.4 Decision Trees

```r
# importing libraries
library(rpart)
library(rpart.plot)
```

```r
# fitting and training the model with the decision tree classifier
fit <- rpart(Revenue ~ ., data = training, method = 'class')
rpart.plot(fit, extra = 106)
```




```r
# making predictions
predict_unseen <- predict(fit, testing, type = 'class')
```

```r
#comparing predicted values to the actual values
table_mat <- table(testing$Revenue, predict_unseen)
table_mat
```

```
##        predict_unseen
##         FALSE TRUE
##   FALSE  2937  150
##   TRUE    245  327
```

```r
# calculating the accuracy
accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
print(paste('Accuracy:', accuracy_Test))
```

```
## [1] "Accuracy: 0.892047007379065"
```

Our model has 89.8% accuracy

5. Unsupervised Learning

```r
# creating set with no revenue column since it has labels
data_new <- data[, -14]
data_new.class <- data[, "Revenue"]
head(data_new)
```

```
##       Administrative Informational ProductRelated BounceRates PageValues
## 4690      0.03703704    0.08333333    0.005673759  0.00000000 0.00000000
## 5009      0.25925926    0.00000000    0.038297872  0.00000000 0.00000000
## 10428     0.03703704    0.00000000    0.089361702  0.01587301 0.02333665
## 8603      0.33333333    0.08333333    0.214184397  0.06410257 0.00000000
## 5350      0.00000000    0.00000000    0.008510638  0.05555555 0.00000000
## 331       0.00000000    0.00000000    0.001418440  1.00000000 0.00000000
##       SpecialDay Month OperatingSystems Browser Region TrafficType
## 4690         0.0   May                2       2      6           6
## 5009         0.2   May                1       1      1           3
## 10428        0.0   Nov                2       2      1           1
## 8603         0.0   Nov                1       2      1           1
## 5350         0.2   May                2       6      6           4
## 331          0.0   Mar                3       2      3           1
##             VisitorType Weekend
## 4690  Returning_Visitor   FALSE
## 5009  Returning_Visitor   FALSE
## 10428 Returning_Visitor   FALSE
## 8603  Returning_Visitor    TRUE
## 5350  Returning_Visitor   FALSE
## 331   Returning_Visitor   FALSE
```

```r
# previewing our target class
head(data_new.class)
```

```
## [1] FALSE FALSE TRUE  FALSE FALSE FALSE
## Levels: FALSE TRUE
```

```r
# converting the factors into numerics
data_new$Month <- as.numeric(as.character(data_new$Month))
```

```
## Warning: NAs introduced by coercion
```

```r
data_new$OperatingSystems <- as.numeric(as.character(data_new$OperatingSystems))
data_new$Browser <- as.numeric(as.character(data_new$Browser))
data_new$Region <- as.numeric(as.character(data_new$Region))
data_new$TrafficType <- as.numeric(as.character(data_new$TrafficType))
data_new$VisitorType <- as.numeric(as.character(data_new$VisitorType))
```

```
## Warning: NAs introduced by coercion
```

```r
data_new$Weekend <- as.numeric(as.character(data_new$Weekend))
```

```
## Warning: NAs introduced by coercion
```

```r
str(data_new)
```

```
## 'data.frame':    12199 obs. of  13 variables:
##  $ Administrative  : num  0.037 0.259 0.037 0.333 0 ...
##  $ Informational   : num  0.0833 0 0 0.0833 0 ...
##  $ ProductRelated  : num  0.00567 0.0383 0.08936 0.21418 0.00851 ...
##  $ BounceRates     : num  0 0 0.0159 0.0641 0.0556 ...
##  $ PageValues      : num  0 0 0.0233 0 0 ...
##  $ SpecialDay      : num  0 0.2 0 0 0.2 0 0 0 0 0 ...
##  $ Month           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ OperatingSystems: num  2 1 2 1 2 3 2 2 1 1 ...
##  $ Browser         : num  2 1 2 2 6 2 2 2 1 1 ...
##  $ Region          : num  6 1 1 1 6 3 4 1 1 2 ...
##  $ TrafficType     : num  6 3 1 1 4 1 13 2 1 1 ...
##  $ VisitorType     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Weekend         : num  NA NA NA NA NA NA NA NA NA NA ...
```

```r
# checking for missing values
anyNA(data_new)
```

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

```r
#dealing with the missing values
data_new[is.na(data_new)] <- 0
```

```r
# checking for missing values
anyNA(data_new)
```

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

Scalling our data

```r
# import library
library(dplyr)
```

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

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

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

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

```r
# scalling our data
rescale_data <- scale(data_new)
```

```r
# previewing our rescaled set
head(rescale_data)
```

```
##       Administrative Informational ProductRelated BounceRates PageValues
## 4690      -0.4023079     1.1688099     -0.6291003  -0.4503438 -0.3190356
## 5009       1.3990334    -0.3988128     -0.1134158  -0.4503438 -0.3190356
## 10428     -0.4023079    -0.3988128      0.6937426  -0.3804225  0.1334484
## 8603       1.9994805     1.1688099      2.6667965  -0.1679692 -0.3190356
## 5350      -0.7025315    -0.3988128     -0.5842582  -0.2056192 -0.3190356
## 331       -0.7025315    -0.3988128     -0.6963635   3.9546997 -0.3190356
##       SpecialDay Month OperatingSystems    Browser      Region TrafficType
## 4690  -0.3103105   NaN       -0.1371074 -0.2093703  1.18492456  0.47945556
## 5009   0.6911386   NaN       -1.2396607 -0.7939682 -0.89629390 -0.26759123
## 10428 -0.3103105   NaN       -0.1371074 -0.2093703 -0.89629390 -0.76562243
## 8603  -0.3103105   NaN       -1.2396607 -0.2093703 -0.89629390 -0.76562243
## 5350   0.6911386   NaN       -0.1371074  2.1290212  1.18492456 -0.01857564
## 331   -0.3103105   NaN        0.9654459 -0.2093703 -0.06380652 -0.76562243
##       VisitorType Weekend
## 4690          NaN     NaN
## 5009          NaN     NaN
## 10428         NaN     NaN
## 8603          NaN     NaN
## 5350          NaN     NaN
## 331           NaN     NaN
```

```r
# replacing the NaN values with 0
rescale_data[is.na(rescale_data)] <- 0
```

```r
#previewing the set
head(rescale_data)
```

```
##       Administrative Informational ProductRelated BounceRates PageValues
## 4690      -0.4023079     1.1688099     -0.6291003  -0.4503438 -0.3190356
## 5009       1.3990334    -0.3988128     -0.1134158  -0.4503438 -0.3190356
## 10428     -0.4023079    -0.3988128      0.6937426  -0.3804225  0.1334484
## 8603       1.9994805     1.1688099      2.6667965  -0.1679692 -0.3190356
## 5350      -0.7025315    -0.3988128     -0.5842582  -0.2056192 -0.3190356
## 331       -0.7025315    -0.3988128     -0.6963635   3.9546997 -0.3190356
##       SpecialDay Month OperatingSystems    Browser      Region TrafficType
## 4690  -0.3103105     0       -0.1371074 -0.2093703  1.18492456  0.47945556
## 5009   0.6911386     0       -1.2396607 -0.7939682 -0.89629390 -0.26759123
## 10428 -0.3103105     0       -0.1371074 -0.2093703 -0.89629390 -0.76562243
## 8603  -0.3103105     0       -1.2396607 -0.2093703 -0.89629390 -0.76562243
## 5350   0.6911386     0       -0.1371074  2.1290212  1.18492456 -0.01857564
## 331   -0.3103105     0        0.9654459 -0.2093703 -0.06380652 -0.76562243
```

```
##       VisitorType Weekend
## 4690            0       0
## 5009            0       0
## 10428           0       0
## 8603            0       0
## 5350            0       0
## 331             0       0
```

5.1 K-Means Clustering

```r
# applying k-means with k = 3
k_result <- kmeans(rescale_data, 3)
```

```r
# previewing the number of records in each cluster
k_result$size
```

```
## [1] 7835 2670 1694
```

```r
#previewing the clusters
k_result$centers
```

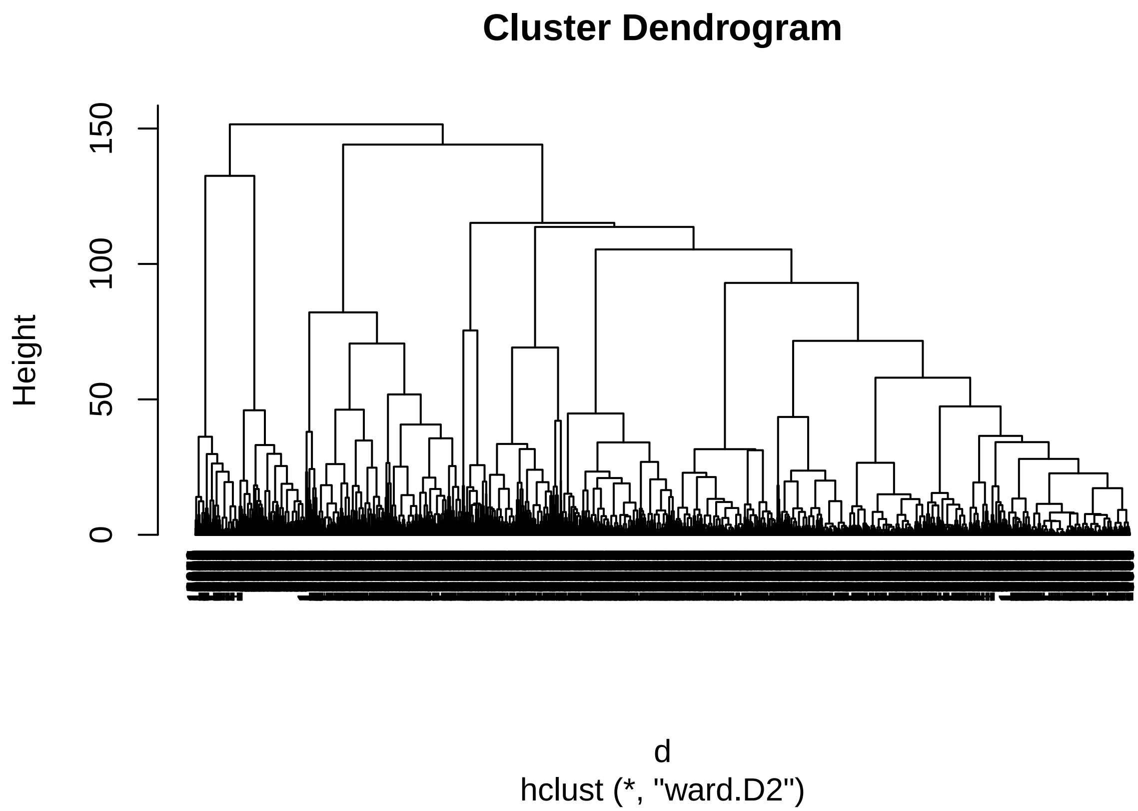
```
##   Administrative Informational ProductRelated  BounceRates  PageValues
## 1     -0.2771222    -0.2668603     -0.2315535  0.066946384 -0.07837697
## 2     -0.1774214    -0.2209140     -0.2084576  0.002645722  0.04513979
## 3      1.5613739     1.5824624      1.3995297 -0.313806963  0.29135793
##    SpecialDay Month OperatingSystems     Browser     Region TrafficType
## 1  0.05141268     0      -0.03156626 -0.07237964 -0.4872754 -0.00519522
## 2 -0.04100694     0       0.10570208  0.27888937  1.5598557  0.09194252
## 3 -0.17315807     0      -0.02060385 -0.10480526 -0.2048478 -0.12088665
##   VisitorType Weekend
## 1           0       0
## 2           0       0
## 3           0       0
```

5.2 Hierachial Clustering

```r
# Compute the euclidean distance
d <- dist(rescale_data, method = "euclidean")
```

```r
# compute hierarchical clustering using the Ward method
hier <- hclust(d, method = "ward.D2" )
```

```r
# plotting the dendogram
plot(hier, cex = 0.6, hang = -1)
```



6. Conclusion

- The month of November returns the highest number of revenues.
- Most of the revenue was from region 1.
- Returning visitors generated more revenue.
- Traffic 2 has the highest number of revenues.
- More revenue was generated during the weekdays than the weekends.