Plastinina

2022-07-25

1. Introduction a)Defining the Question Learn behaviour of customers and learn characteristics of various cstomer groups. b)Defining the Metric of Success If we are able to classify and identify various customer behaviors. 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 • 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
#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" "Return
##  $ Weekend                : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ Revenue                : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

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

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

```r
colnames(data)
```

```
##  [1] "Administrative"          "Administrative_Duration"
##  [3] "Informational"           "Informational_Duration"
##  [5] "ProductRelated"          "ProductRelated_Duration"
##  [7] "BounceRates"             "ExitRates"
##  [9] "PageValues"              "SpecialDay"
## [11] "Month"                   "OperatingSystems"
## [13] "Browser"                 "Region"
## [15] "TrafficType"             "VisitorType"
## [17] "Weekend"                 "Revenue"
```

Data cleaning.

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

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

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

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

```
# 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
```

```
#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
```

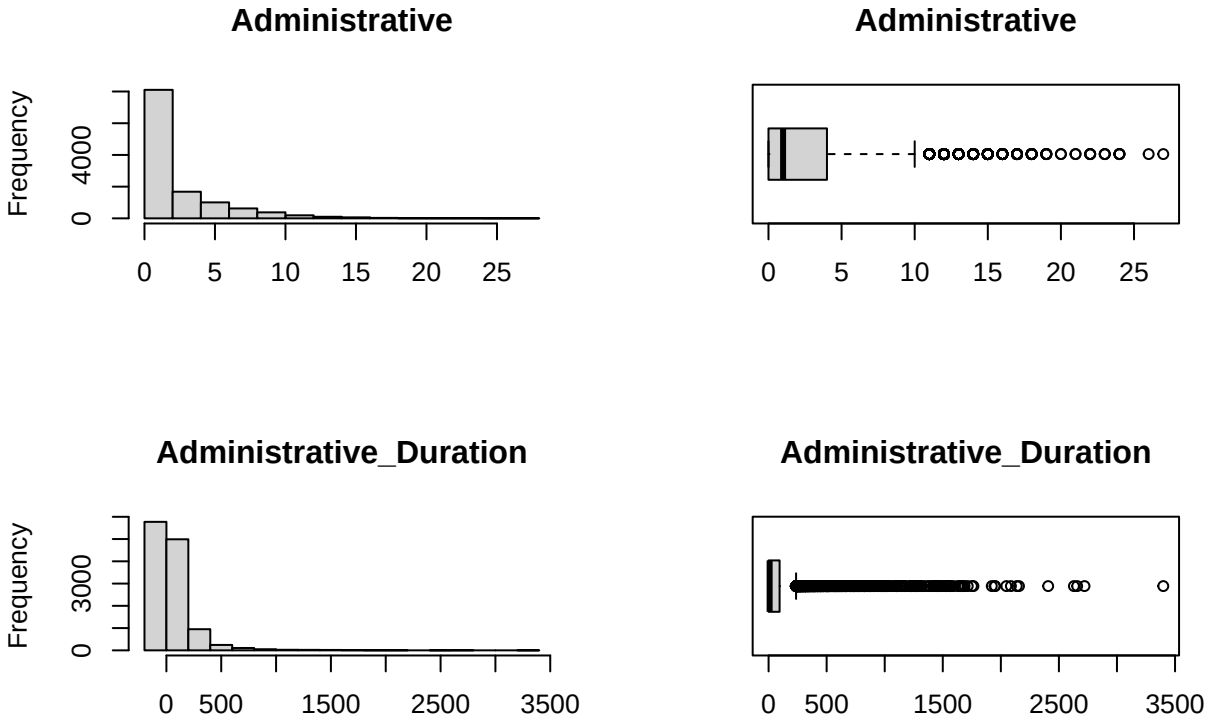
```
# converting variables numerical to categorical.
data$OperatingSystems <- as.factor(data$OperatingSystems)
data$Browser <- as.factor(data$Browser)
data$Region <- as.factor(data$Region)
data$TrafficType <- as.factor(data$TrafficType)
data$Weekend <- as.factor(data$Weekend)
data$Revenue <- as.factor(data$Revenue)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

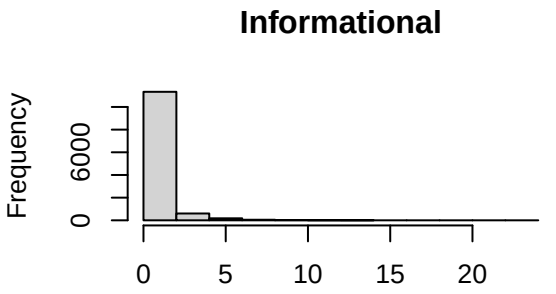
```
## 'data.frame':    12199 obs. of  18 variables:
##  $ Administrative         : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Administrative_Duration: num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ Informational          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration : num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ 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 ...
```

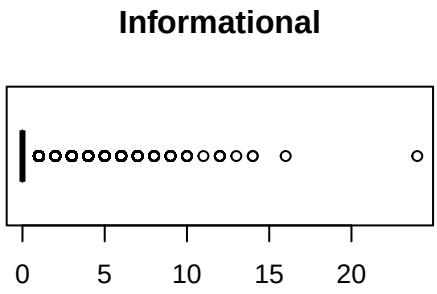
```
##  $ ExitRates              : num  0.2 0.1 0.2 0.14 0.05 ...
##  $ PageValues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialDay             : num  0 0 0 0 0 0 0.4 0 0.8 0.4 ...
##  $ Month                  : chr  "Feb" "Feb" "Feb" "Feb" ...
##  $ OperatingSystems       : Factor w/ 8 levels "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" "Return
##  $ 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" ...
```

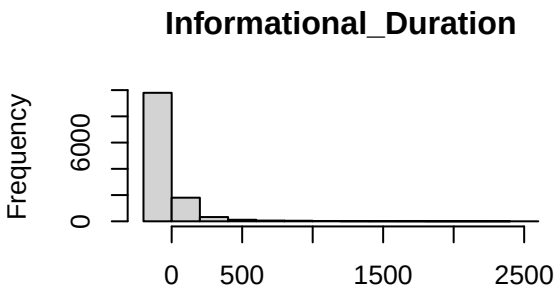
3.Exploratory Data Analysis 3.1 Univariate Analysis

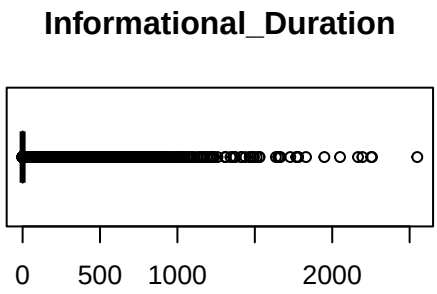
```
# previewing the numerical variables histograms and barplots
par(mfrow=c(2,2))
for(i in 1:10) {
hist(data[, i], main=names(data)[i], xlab = NULL)
boxplot(data[,i], main=names(data)[i], horizontal = TRUE)
}
```

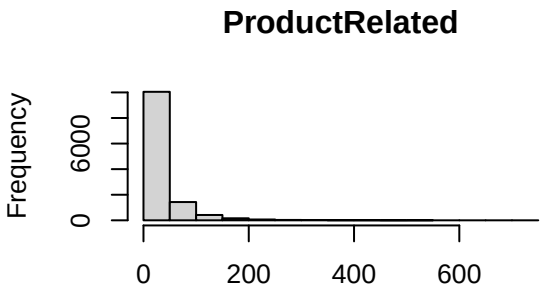


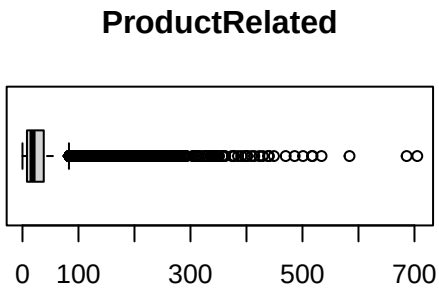

Informational
Frequency
0
6000
0
5
10
15
20

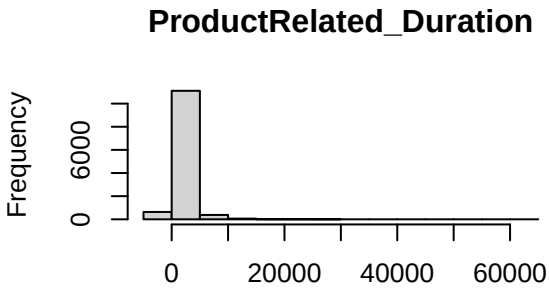

Informational
0
5
10
15
20

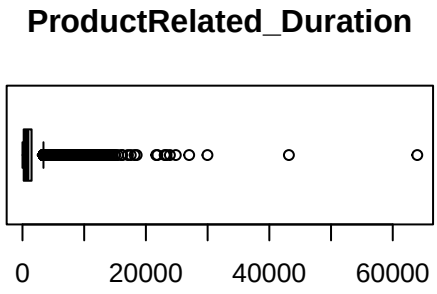

Informational_Duration
Frequency
0
6000
0
500
1500
2500

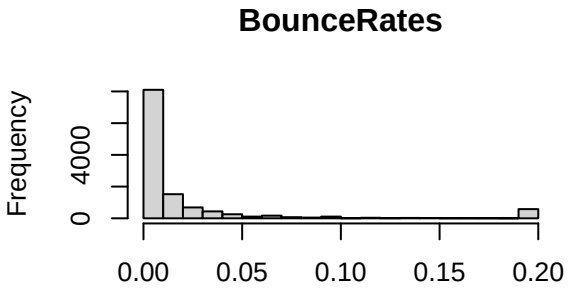

Informational_Duration
0
500
1000
2000

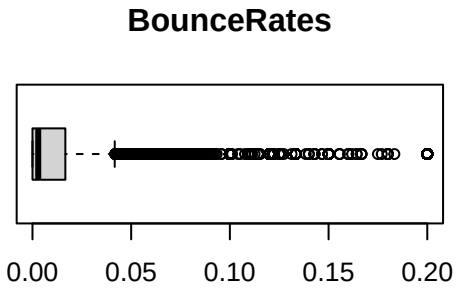

ProductRelated
Frequency
0
6000
0
200
400
600

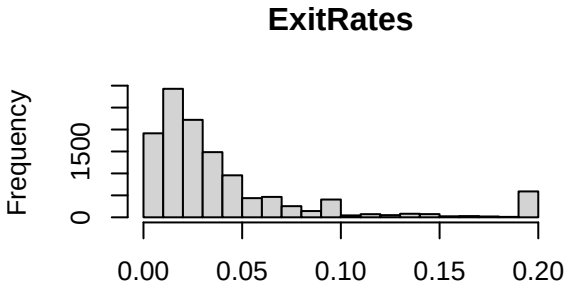

ProductRelated
0
100
300
500
700

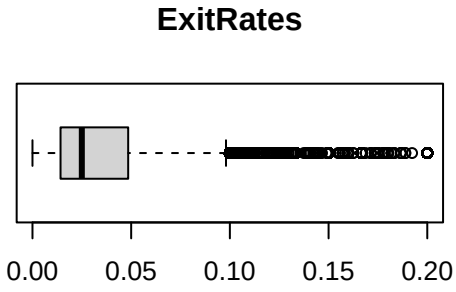

ProductRelated_Duration
Frequency
0
6000
0
20000
40000
60000

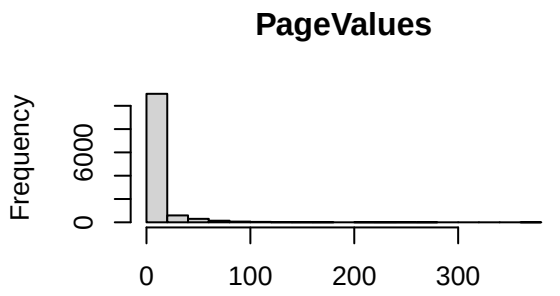

ProductRelated_Duration
0
20000
40000
60000

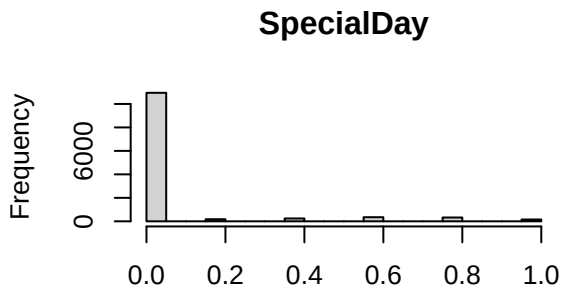

BounceRates
Frequency
0
4000
0.00
0.05
0.10
0.15
0.20

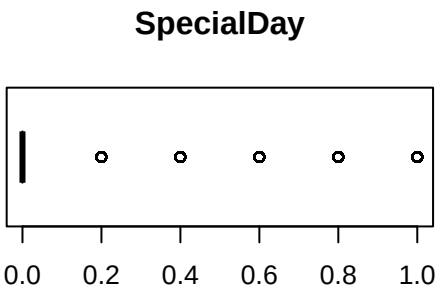

BounceRates
0.00
0.05
0.10
0.15
0.20


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





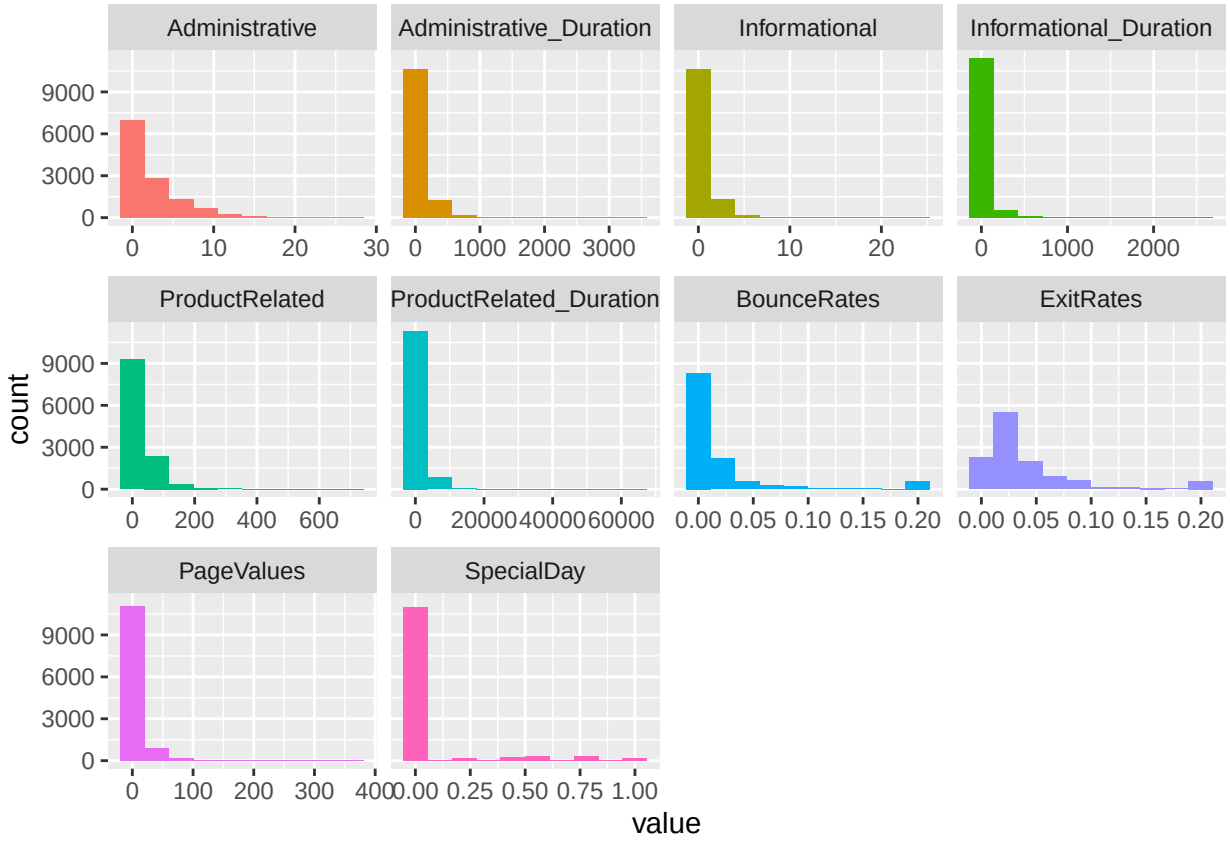



```
library(funModeling)
```

```
## 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
# Checking for variable skeweness with histograms on the numerical variables.
data_num <- data[1:10]
plot_num(data_num)
```

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




```r
# It shows that our variables have positive skewness to them
```

```r
# Categorical variables
# 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)
```

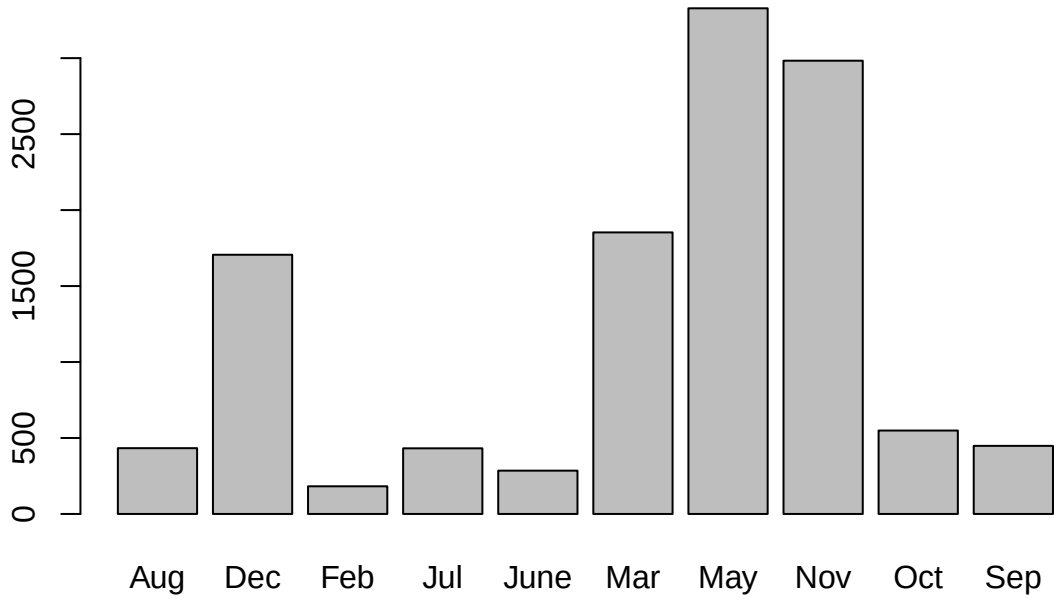
```r
# adjusting plot size
set_plot_dimensions <- function(width_choice, height_choice) {
options(repr.plot.width = width_choice, repr.plot.height = height_choice)
}
```

Plotting few of the tables.

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

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

```
barplot(month_table)
```

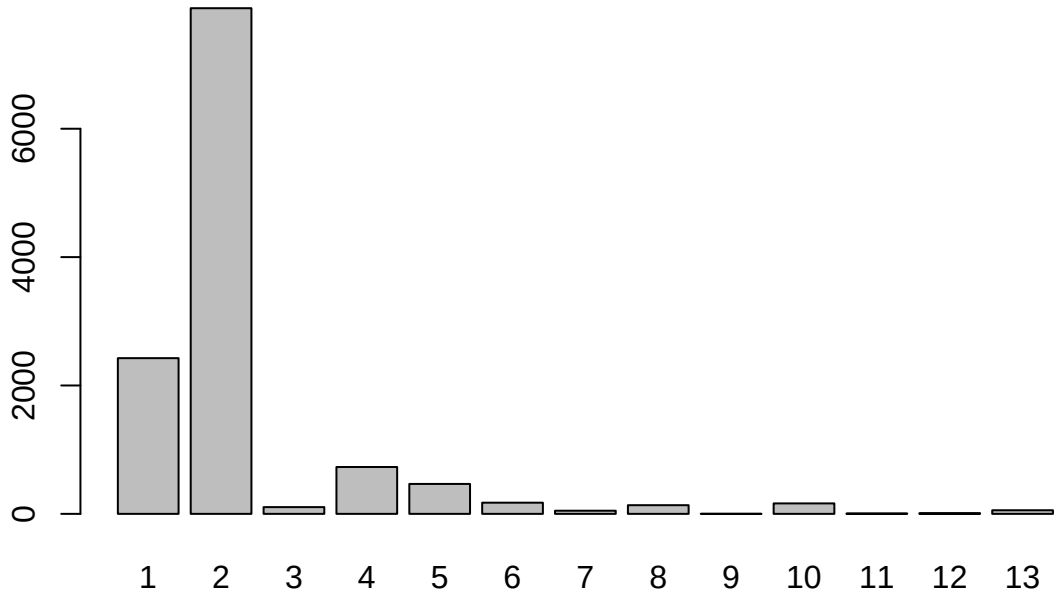



```
# May records the highest frequency February having the least.
```

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




```
#  it shows browser two is the most used browser
```

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

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

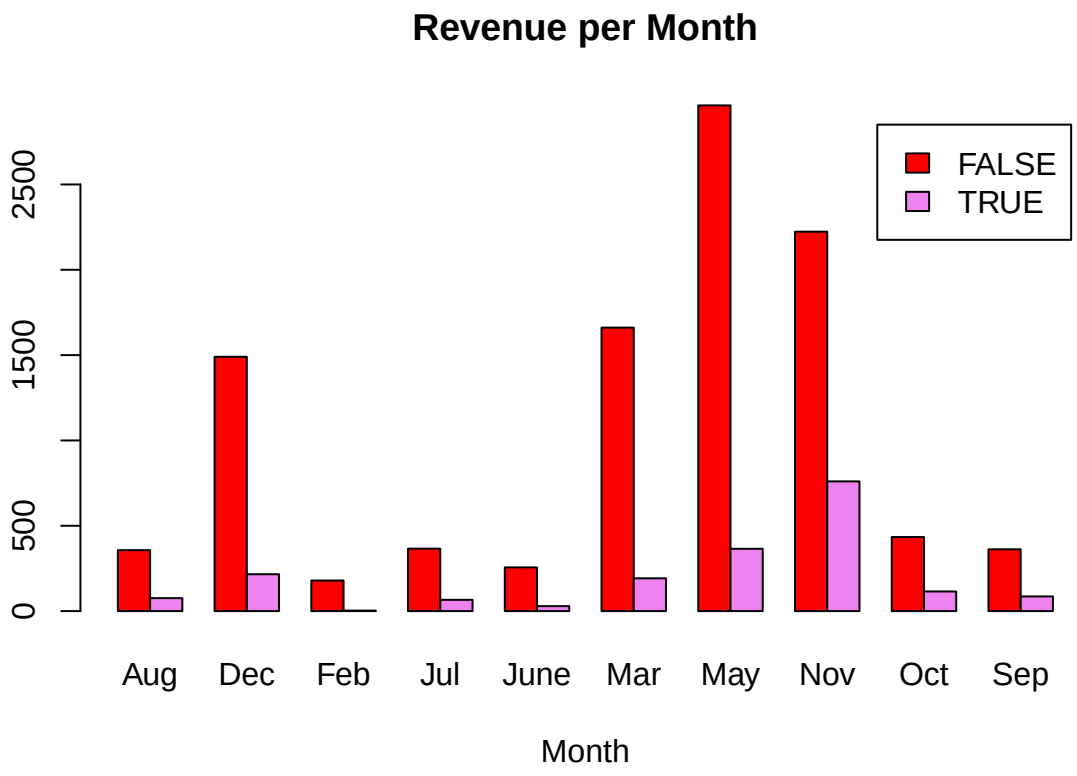
```
barplot(visitor_table)
```


```
# Shows that most visitors returned
```

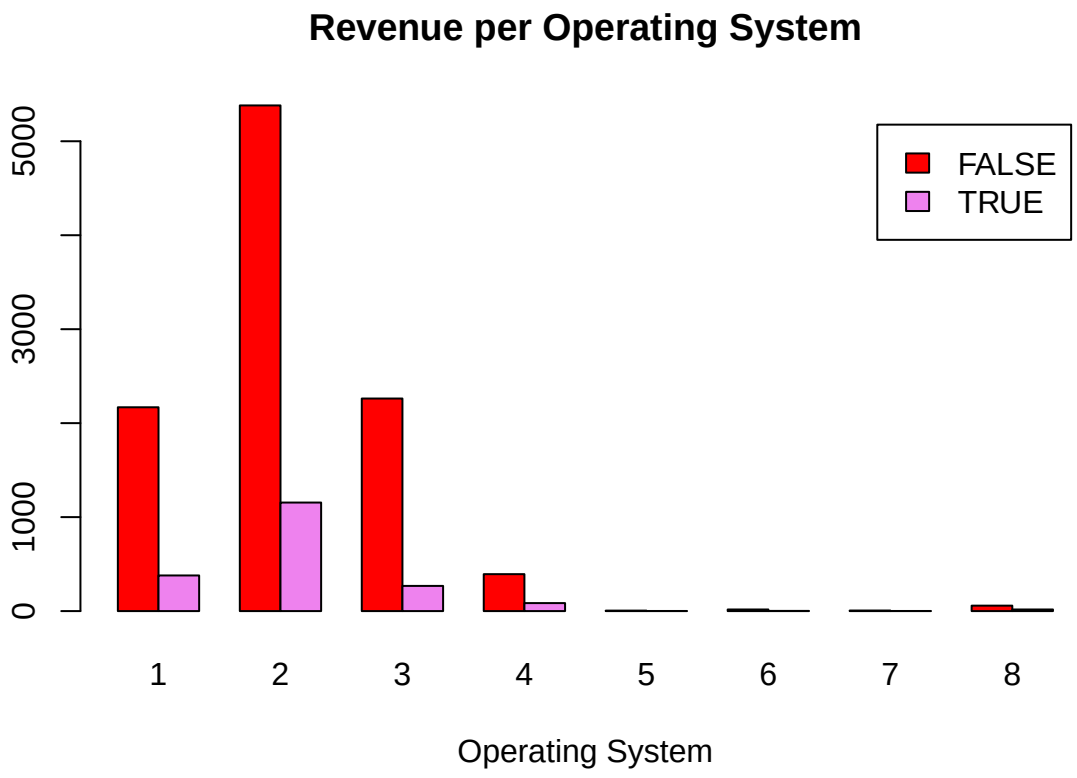
3.2 Bivariate analysis We will show relationship between variables.

```
library(ggplot2)
```

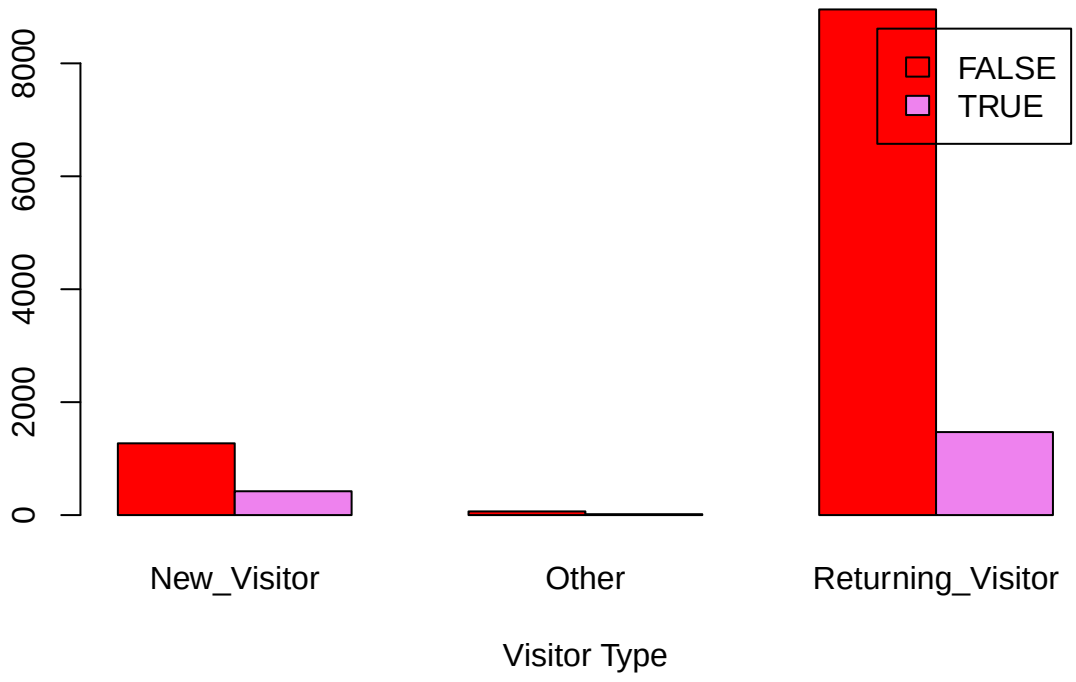
```
# plotting the distribution of Revenue per Month
# November had the highest returns.
set_plot_dimensions(4, 4)
rev_month <- table(data$Revenue, data$Month)
barplot(rev_month, main = "Revenue per Month", col = c("red", "violet"), beside = TRUE,
legend = rownames(rev_month), xlab = "Month")
```



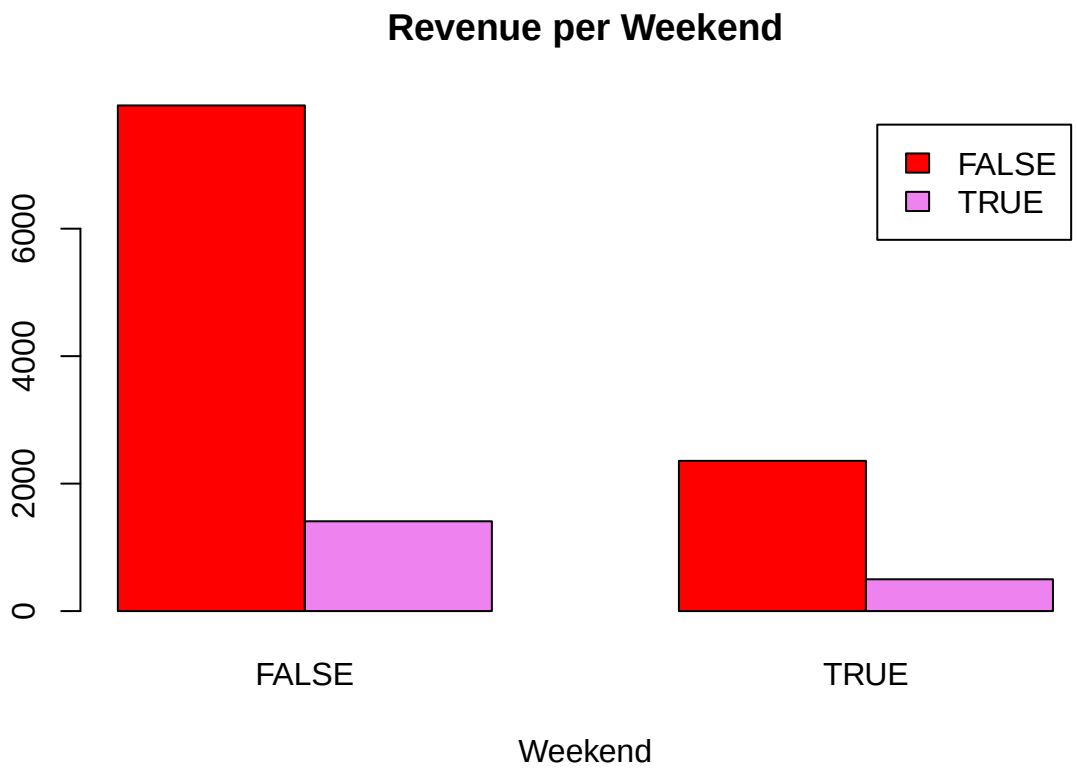

```
# plotting the distribution of Revenue per Operating System
# 2nd operating system brought the highest revenue
set_plot_dimensions(6, 6)
rev_os <- table(data$Revenue, data$OperatingSystems)
barplot(rev_os, main = "Revenue per Operating System", col = c("red", "violet"), beside = TRUE,
legend = rownames(rev_os), xlab = "Operating System")
```



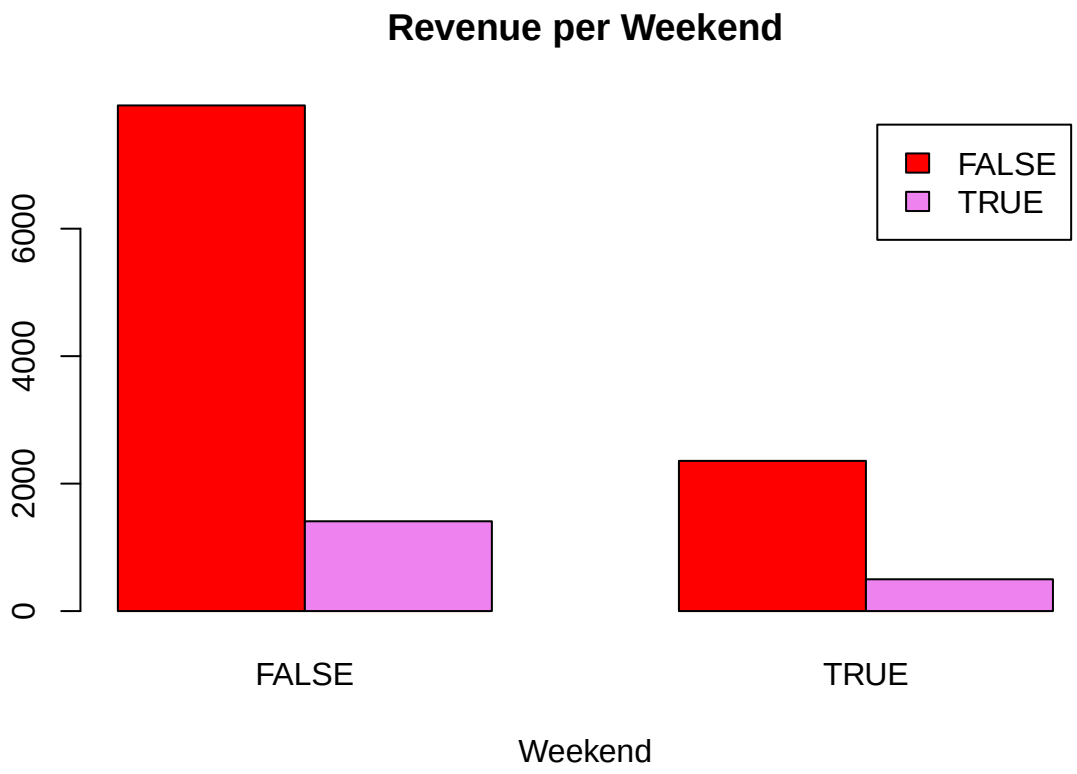

```
# plotting the distribution of Revenue per Visitor Type
# Returning traders brought the highest revenue
set_plot_dimensions(6, 6)
rev_visitor <- table(data$Revenue, data$VisitorType)
barplot(rev_visitor, main = "Revenue per Visitor Type", col = c("red", "violet"), beside = TRUE,
legend = rownames(rev_visitor), xlab = "Visitor Type")
```



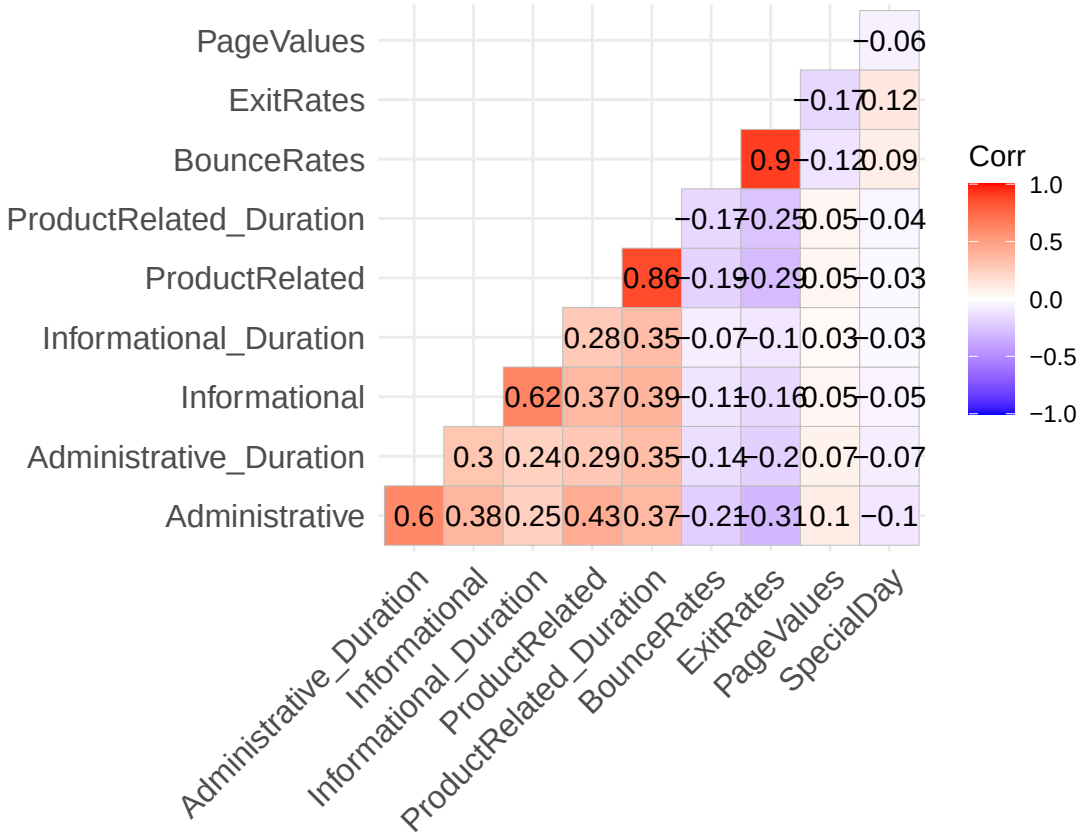

```
# plotting the distribution of Revenue per Weekend
set_plot_dimensions(6, 6)
rev_weekend <- table(data$Revenue, data$Weekend)
barplot(rev_weekend, main = "Revenue per Weekend", col = c("red", "violet"), beside = TRUE,
legend = rownames(rev_weekend), xlab = "Weekend")
```




```
# plotting the distribution of Revenue per Week
# Week day recorded the highest revenue
set_plot_dimensions(6, 6)
rev_weekend <- table(data$Revenue, data$Weekend)
barplot(rev_weekend, main = "Revenue per Weekend", col = c("red", "violet"), beside = TRUE,
legend = rownames(rev_weekend), xlab = "Weekend")
```




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




```
# We will drop the highly correlated columns
# They bring multicollinearity to our datset.
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
```

```
# 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
```

```
# visualizing the correlation of the new dataset
set_plot_dimensions(4, 4)
new_corr_data <- cor(data_num)
ggcorrplot(round(new_corr_data, 2) ,lab = T,type = 'lower')
```

4. Modelling 4.1 Feature Engineering

```
library(lattice)
library(caret)
```

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

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

```
# applying our function to our attributes.
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 SpecialDay
## 1              0             0    0.001418440  1.00000000          0          0
## 2              0             0    0.002836879  0.00000000          0          0
## 3              0             0    0.001418440  1.00000000          0          0
## 4              0             0    0.002836879  0.25000000          0          0
## 5              0             0    0.014184397  0.10000000          0          0
## 6              0             0    0.026950355  0.07894737          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
```

```
# 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,]
```

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

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

```
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
```

KNN

```r
warning = FALSE
```

```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
```

```
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
```

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

```
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
```

```
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
```

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

Modelling using knn

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

```
model <- knn(train = train, test = test,cl = train_rev, k = 20)
```

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

```
##         model
## test_rev FALSE TRUE
##    FALSE  3074   13
##    TRUE    546   26
```

```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"
```

UNSUPERVISED

```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 SpecialDay
## 1              0             0    0.001418440  1.00000000          0          0
## 2              0             0    0.002836879  0.00000000          0          0
## 3              0             0    0.001418440  1.00000000          0          0
## 4              0             0    0.002836879  0.25000000          0          0
## 5              0             0    0.014184397  0.10000000          0          0
## 6              0             0    0.026950355  0.07894737          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
```

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

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

```
str(data_new)
```

```
## 'data.frame':    12199 obs. of  13 variables:
##  $ Administrative  : num  0 0 0 0 0 ...
##  $ Informational   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ProductRelated  : num  0.00142 0.00284 0.00142 0.00284 0.01418 ...
##  $ BounceRates     : num  1 0 1 0.25 0.1 ...
##  $ 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           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ OperatingSystems: num  1 2 4 3 3 2 2 1 2 2 ...
##  $ Browser         : num  1 2 1 2 3 2 4 2 2 4 ...
##  $ Region          : num  1 1 9 2 1 1 3 1 2 1 ...
##  $ TrafficType     : num  1 2 3 4 4 3 3 5 3 2 ...
##  $ VisitorType     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Weekend         : num  NA NA NA NA NA NA NA NA NA NA ...
```

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

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

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

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

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

Scalling our data

```
# 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
## 1     -0.7025315    -0.3988128     -0.6963635  3.954699721 -0.3190356
## 2     -0.7025315    -0.3988128     -0.6739424 -0.450343788 -0.3190356
## 3     -0.7025315    -0.3988128     -0.6963635  3.954699721 -0.3190356
## 4     -0.7025315    -0.3988128     -0.6739424  0.650917089 -0.3190356
## 5     -0.7025315    -0.3988128     -0.4945739 -0.009839437 -0.3190356
## 6     -0.7025315    -0.3988128     -0.2927843 -0.102577188 -0.3190356
##   SpecialDay Month OperatingSystems    Browser     Region TrafficType
## 1 -0.3103105   NaN       -1.2396607 -0.7939682 -0.8962939 -0.76562243
## 2 -0.3103105   NaN       -0.1371074 -0.2093703 -0.8962939 -0.51660683
## 3 -0.3103105   NaN        2.0679992 -0.7939682  2.4336556 -0.26759123
## 4 -0.3103105   NaN        0.9654459 -0.2093703 -0.4800502 -0.01857564
## 5 -0.3103105   NaN        0.9654459  0.3752276 -0.8962939 -0.01857564
## 6 -0.3103105   NaN       -0.1371074 -0.2093703 -0.8962939 -0.26759123
##   VisitorType Weekend
## 1         NaN     NaN
## 2         NaN     NaN
## 3         NaN     NaN
## 4         NaN     NaN
## 5         NaN     NaN
## 6         NaN     NaN
```

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

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]  993 2031 9175
```

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

```
##   Administrative Informational ProductRelated BounceRates  PageValues
## 1     -0.3554452   -0.19358522    -0.10680010   0.2352751 -0.21647087
## 2     -0.4355676   -0.28612300    -0.38495245   1.1723814 -0.17256443
## 3      0.1348877    0.08428838     0.09677285  -0.2849847  0.06162768
##   SpecialDay Month OperatingSystems     Browser      Region TrafficType
## 1   3.156942     0       0.03055175  0.02258496 -0.05500378  0.09050975
## 2  -0.228952     0       0.45949824  0.21691699  0.10588810  1.39447447
```

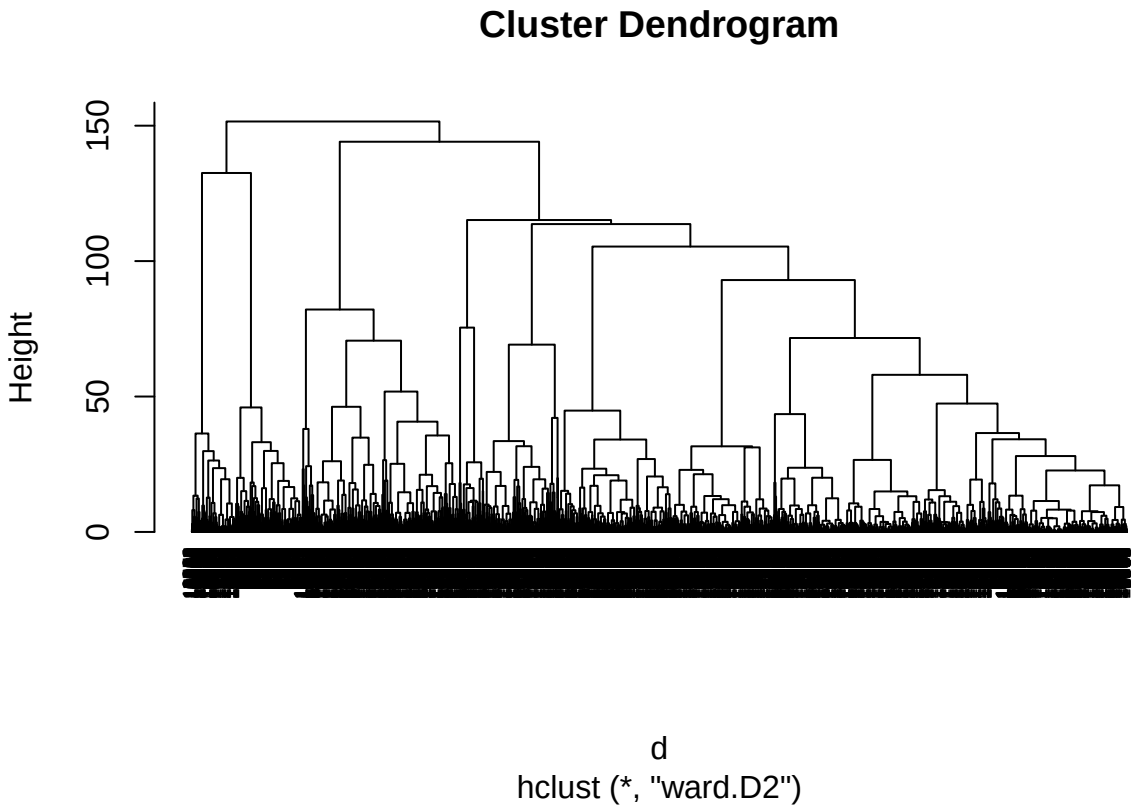
```
## 3  -0.290991     0      -0.10502221 -0.05046161 -0.01748665 -0.31847998
##   VisitorType Weekend
## 1           0       0
## 2           0       0
## 3           0       0
```

Hierachial Clustering

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

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

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



In conclusion, Most revenue was generated during the week days than the weekends Operating system and browser used the most was the number two. Most revenue was generated by the returning visitors.