# WEEK 13 Part2 IP R PROGRAMMING

#1. Defining the question

##a) Specifying the Data Analytic Question. `Classifying customers by studying and understanding their behavior from data collected over the past year while also learning the characteristics of customer groups.` ##b) Defining the metric of success `The project will be a success when we are able to identify the customers who are most likely to complete a transaction by studying their online behavior.` ##c) Understanding the context `The client is a Russian brand that has 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.From the data they'd like to learn the characteristics of customer groups. They plan to eventually formulate their marketing and sales strategies using the findings from this study.` ##d) Recording the experimental design `The process entails:` * Problem Definition * Data Sourcing * Checking the Data * Performing Data Cleaning * Performing Exploratory Data Analysis(EDA) + Univariate Analysis + Bivariate Analysis + Multivariate Analysis * Implementing the Solution * Challenging the Solution * Formulating follow up questions

#2. Data Sourcing `This is the data provided as described above; [Ecommerce Customers Dataset](http://bit.ly/EcommerceCustomersDataset)` ## Data Relevance "' The dataset contains characteristics of individuals who shop from Kira Plastinina. At face value it is appropriate.The appropriateness is better or more accurately judged when measured against the metric of success.

**Data Description:**"' * The dataset consists of **10 numerical** and **8 categorical attributes**. The 'Revenue' attribute can be used as the class label.

- "Administrative", "Administrative Duration", "Informational", "Informational Duration", "Product Related" and "Product Related Duration" represents the number of different types of pages visited by the visitor in that session and total time spent in each of these page categories. The values of these features are derived from the URL information of the pages visited by the user and updated in real-time when a user takes an action, e.g. moving from one page to another.
- The "Bounce Rate", "Exit Rate" and "Page Value" features represent the metrics measured by "Google Analytics" for each page in the e-commerce site.
- The value of the "Bounce Rate" feature for a web page refers to the percentage of visitors who enter the site from that page and then leave ("bounce") without triggering any other requests to the analytics server during that session.
- The value of the "Exit Rate" feature for a specific web page is calculated as for all pageviews to the page, the percentage that was the last in the session.
- The "Page Value" feature represents the average value for a web page that a user visited before completing an e-commerce transaction.
- The "Special Day" feature indicates the closeness of the site visiting time to a specific special day (e.g. Mother's Day, Valentine's Day) in which the sessions are more likely to be finalized with the transaction. The value of this attribute is determined by considering the dynamics of e-commerce such as the duration between the order date and delivery date. For example, for Valentine's day, this value takes a nonzero value between February 2 and February 12, zero before and after this date unless it is close to another special day, and its maximum value of 1 on February 8.

- The dataset also includes the operating system, browser, region, traffic type, visitor type as returning or new visitor, a Boolean value indicating whether the date of the visit is weekend, and month of the year.

```
# increasing memory limit to evade error
memory.limit(size=1800)
```

```
## Warning: 'memory.limit()' is no longer supported
```

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

```
# loading necessary libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```
library(readr)
library(ROCR)
library(PerformanceAnalytics)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'xts'
```

```
## The following objects are masked from 'package:dplyr':
##
##     first, last
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

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

```
library(e1071)
```

```
##
## Attaching package: 'e1071'

## The following objects are masked from 'package:PerformanceAnalytics':
##
##     kurtosis, skewness
```

```
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

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

```
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(ggcorrplot)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(rpart)
library(caTools)
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```
library(class)
library(ISLR)
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-4
```

```
library(Hmisc)
```

```
## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following object is masked from 'package:e1071':
##
##     impute

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

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

```
library(funModeling)
```

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

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(klaR)
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(cluster)
library(factoextra)
```

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

```
library(DataExplorer)
library(ClustOfVar)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
##
## Attaching package: 'GGally'

## The following object is masked from 'package:funModeling':
##
##     range01
```

#3. Checking the data

```
# reading the 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
```

```
#Viewing the first 6 entries
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
```

```
#viewing the last 6 entries
tail(data)
```

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

```r
#viewing the structure of the 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
# viewing the shape ie rows, columns
dim(data)
```

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

```r
# getting the summary statistics
summary(data)
```

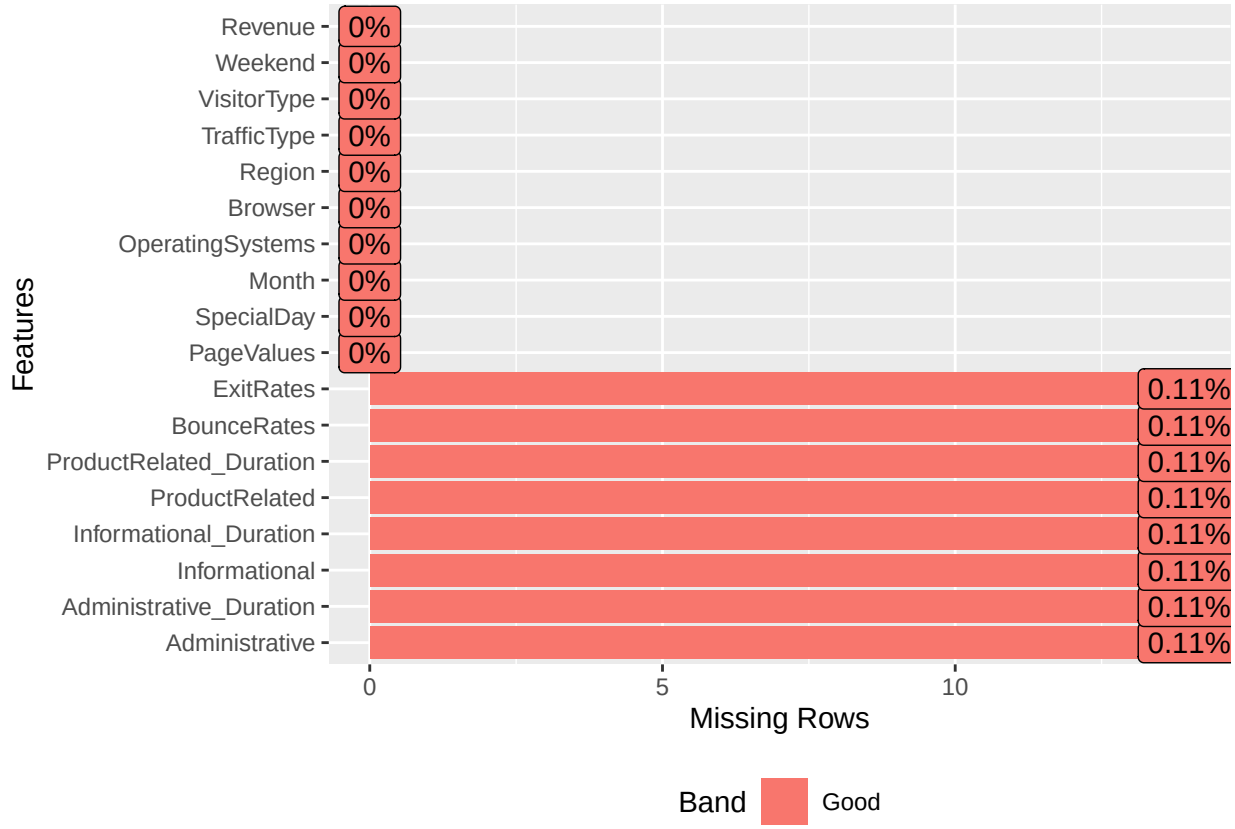
```
##  Administrative   Administrative_Duration Informational
##  Min.   : 0.000   Min.   :  -1.00         Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.:   0.00         1st Qu.: 0.000
##  Median : 1.000   Median :   8.00         Median : 0.000
##  Mean   : 2.318   Mean   :  80.91         Mean   : 0.504
##  3rd Qu.: 4.000   3rd Qu.:  93.50         3rd Qu.: 0.000
##  Max.   :27.000   Max.   :3398.75         Max.   :24.000
##  NA's   :14       NA's   :14              NA's   :14
##  Informational_Duration ProductRelated   ProductRelated_Duration
##  Min.   :  -1.00        Min.   :  0.00   Min.   :   -1.0
##  1st Qu.:   0.00        1st Qu.:  7.00   1st Qu.:  185.0
##  Median :   0.00        Median : 18.00   Median :  599.8
##  Mean   :  34.51        Mean   : 31.76   Mean   : 1196.0
##  3rd Qu.:   0.00        3rd Qu.: 38.00   3rd Qu.: 1466.5
##  Max.   :2549.38        Max.   :705.00   Max.   :63973.5
##  NA's   :14             NA's   :14       NA's   :14
##   BounceRates         ExitRates         PageValues        SpecialDay
##  Min.   :0.000000   Min.   :0.00000   Min.   :  0.000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.01429   1st Qu.:  0.000   1st Qu.:0.00000
##  Median :0.003119   Median :0.02512   Median :  0.000   Median :0.00000
##  Mean   :0.022152   Mean   :0.04300   Mean   :  5.889   Mean   :0.06143
```

```
##  3rd Qu.:0.016684   3rd Qu.:0.05000   3rd Qu.:  0.000   3rd Qu.:0.00000
##  Max.   :0.200000   Max.   :0.20000   Max.   :361.764   Max.   :1.00000
##  NA's   :14         NA's   :14
##     Month           OperatingSystems    Browser           Region
##  Length:12330       Min.   :1.000    Min.   : 1.000   Min.   :1.000
##  Class :character   1st Qu.:2.000    1st Qu.: 2.000   1st Qu.:1.000
##  Mode  :character   Median :2.000    Median : 2.000   Median :3.000
##                     Mean   :2.124    Mean   : 2.357   Mean   :3.147
##                     3rd Qu.:3.000    3rd Qu.: 2.000   3rd Qu.:4.000
##                     Max.   :8.000    Max.   :13.000   Max.   :9.000
##
##   TrafficType    VisitorType         Weekend         Revenue
##  Min.   : 1.00   Length:12330       Mode :logical   Mode :logical
##  1st Qu.: 2.00   Class :character   FALSE:9462      FALSE:10422
##  Median : 2.00   Mode  :character   TRUE :2868      TRUE :1908
##  Mean   : 4.07
##  3rd Qu.: 4.00
##  Max.   :20.00
##
```

#4. Tidying the data/ Data Cleaning

## Completeness

```
# checking the percentage of missing values for all variables by plotting
plot_missing(data)
```




```
#ommiting the missing values
data2 <- na.omit(data)
#rechecking the shape
dim(data2)
```

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

```
# confirming the columns with null values are dropped
colSums(is.na(data2))
```

```
##          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
```

### consistency

```
#checking for duplicates
anyDuplicated(data2)
```

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

```
#removing duplicates
data2 <- unique(data2)
dim(data2)
```

```
## [1] 12199    18
```

```
#converting categorical columns to factors
data2[,11:18] <- sapply(data2[,11:18], as.factor)
head(data2)
```

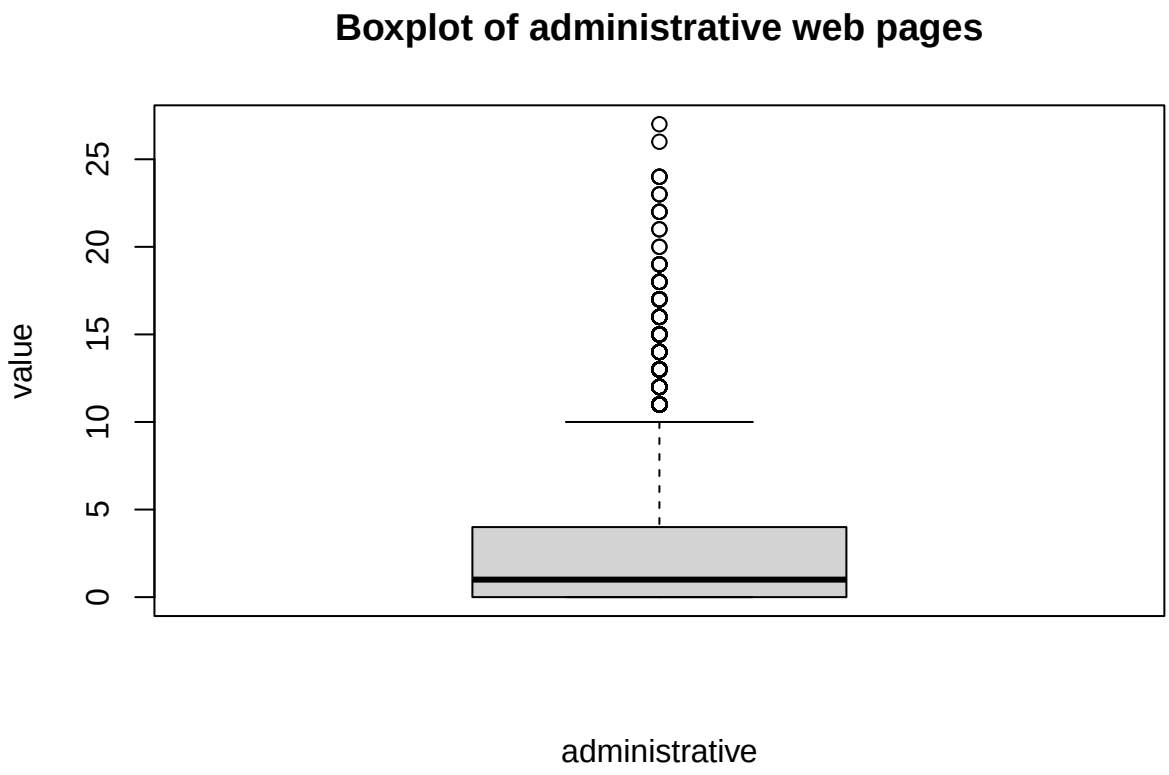
```
##   Administrative Administrative_Duration Informational Informational_Duration
## 1              0                       0             0                      0
## 2              0                       0             0                      0
## 3              0                      -1             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
```

We're doing this because Factor in R is a variable used to categorize and store the data(as a vector of integer values), having a limited number of different values. This is beneficial as we're trying to eventually categorize customers

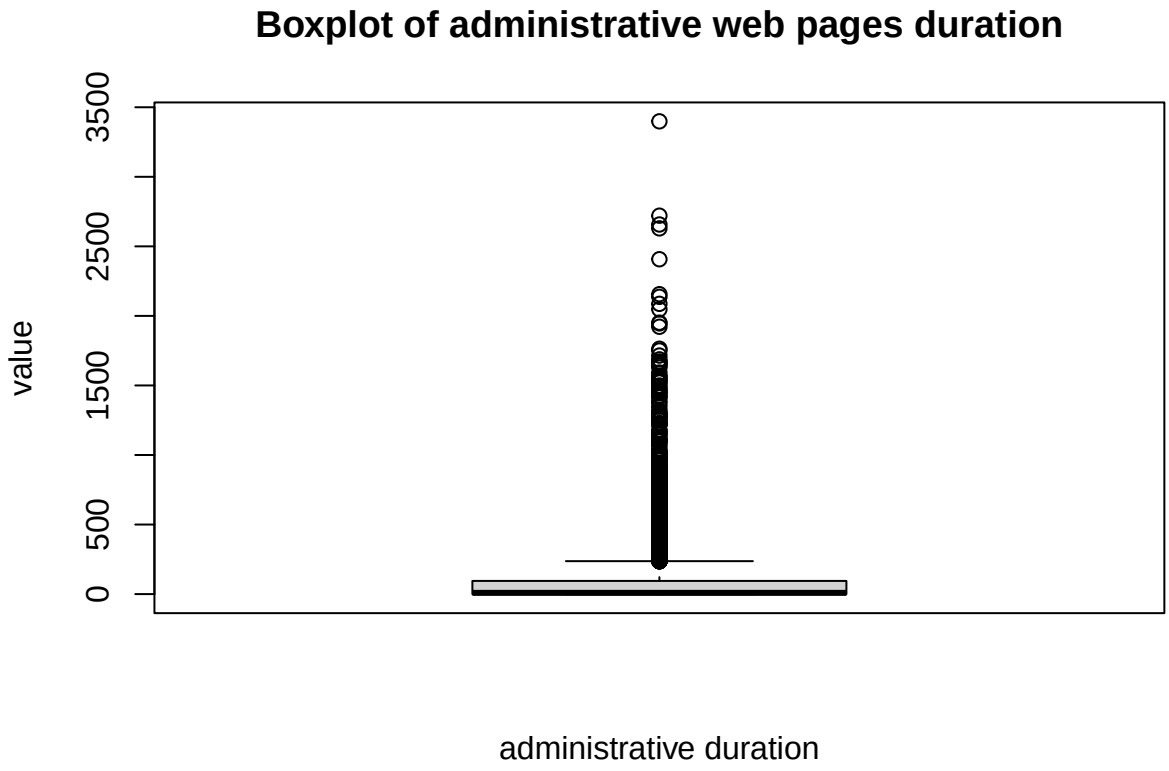
```r
# listing the column names so we can check for outliers
list(colnames(data2))
```

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

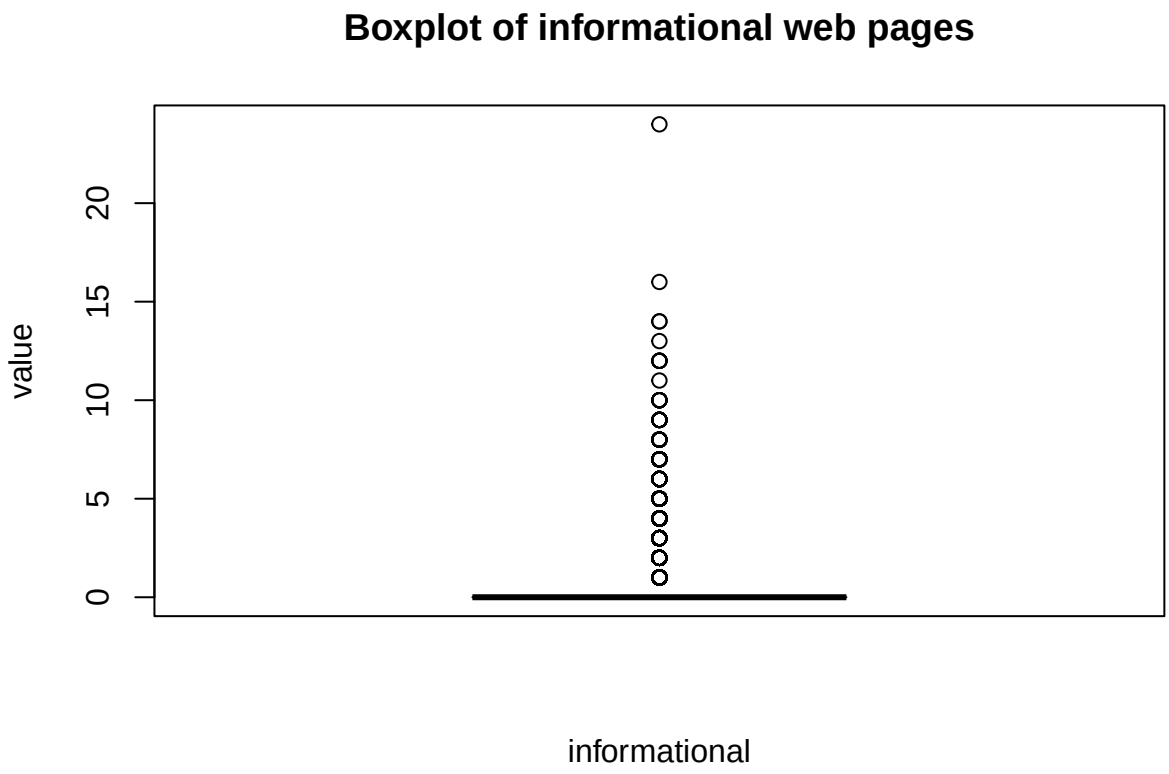
```r
#checking for outliers in each of the variables
boxplot(data2$Administrative, main= 'Boxplot of administrative web pages', xlab='administrative', ylab=
```



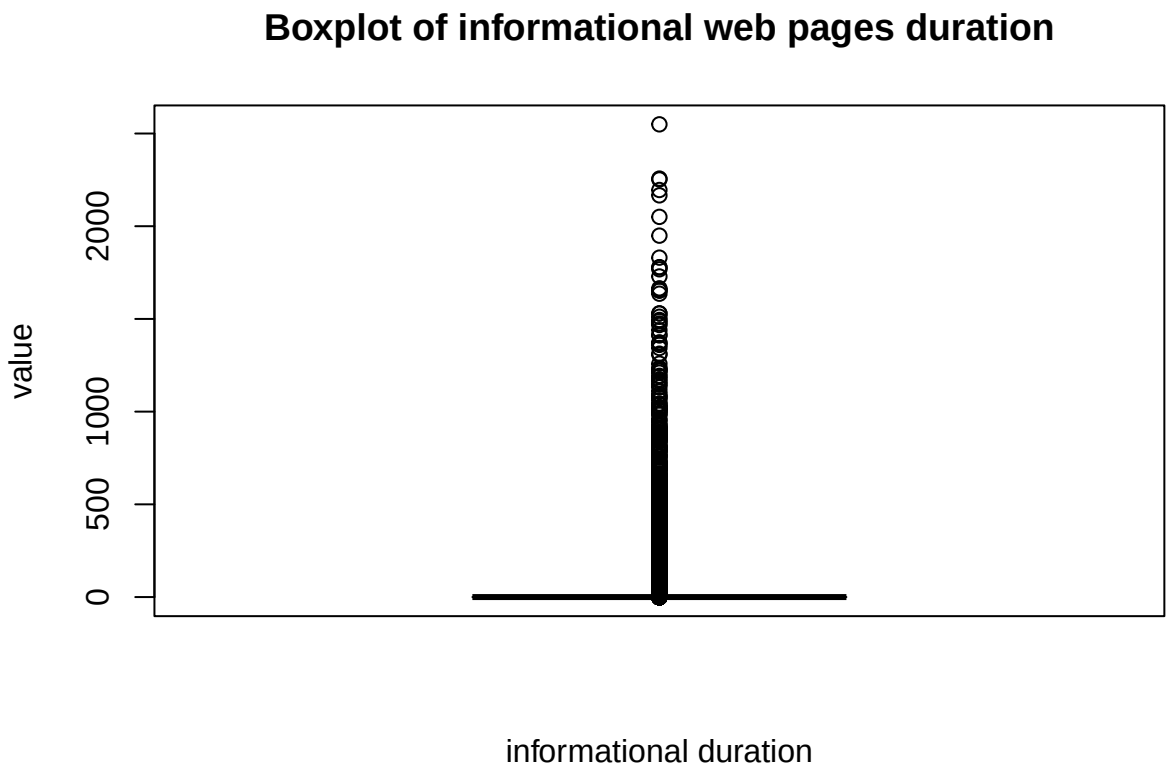

```r
boxplot(data2$Administrative_Duration, main= 'Boxplot of administrative web pages duration', xlab='admi
```



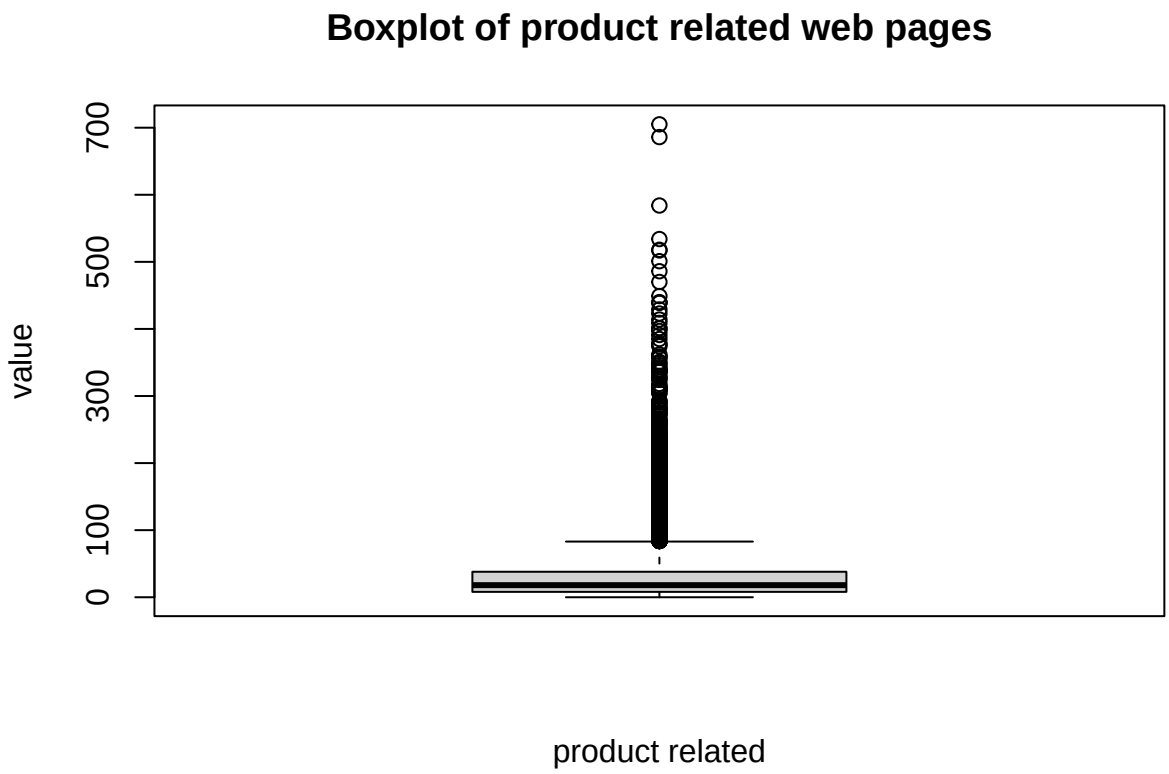

```
boxplot(data2$Informational, main= 'Boxplot of informational web pages', xlab='informational', ylab='val
```



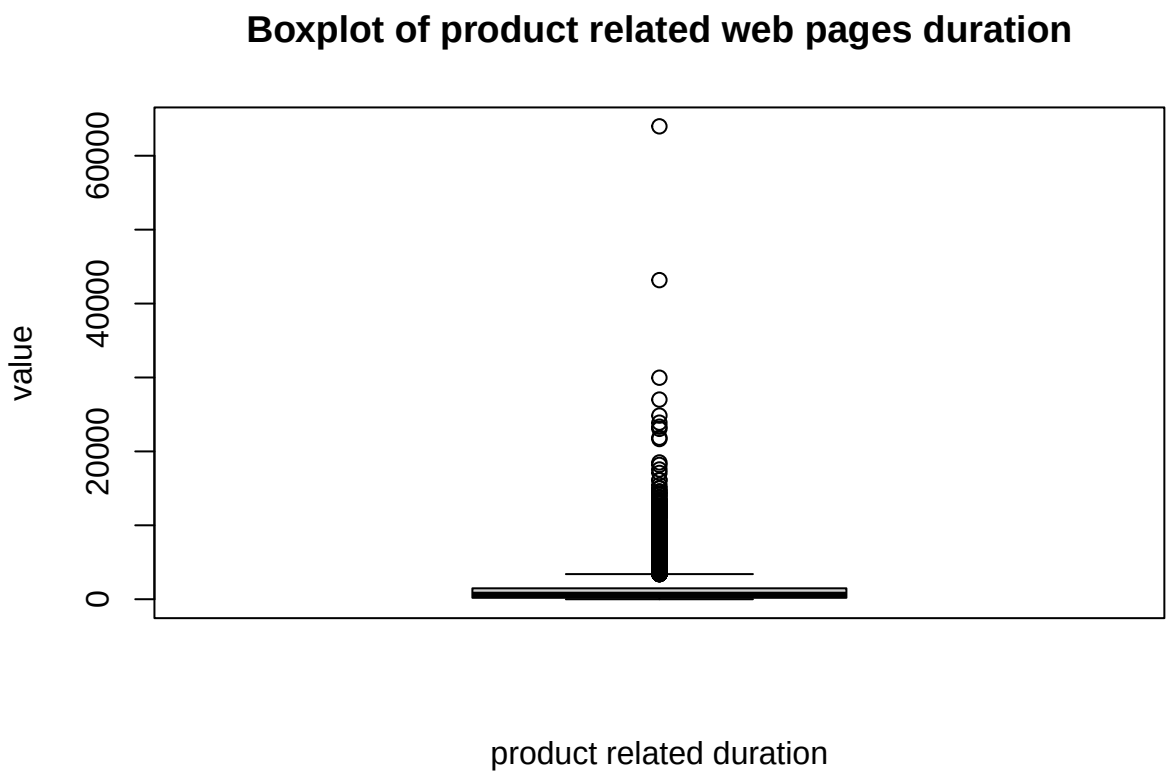

```
boxplot(data2$Informational_Duration, main= 'Boxplot of informational web pages duration', xlab='informa
```



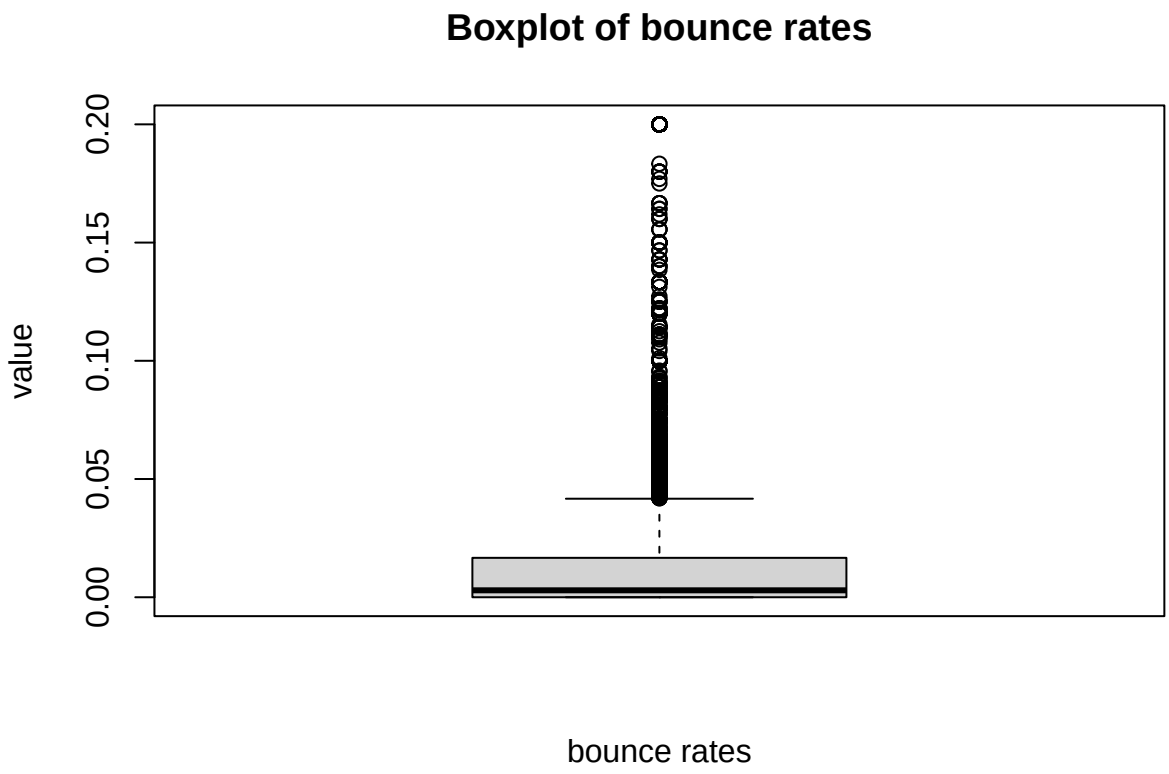

```
boxplot(data2$ProductRelated, main= 'Boxplot of product related web pages', xlab='product related', yla
```



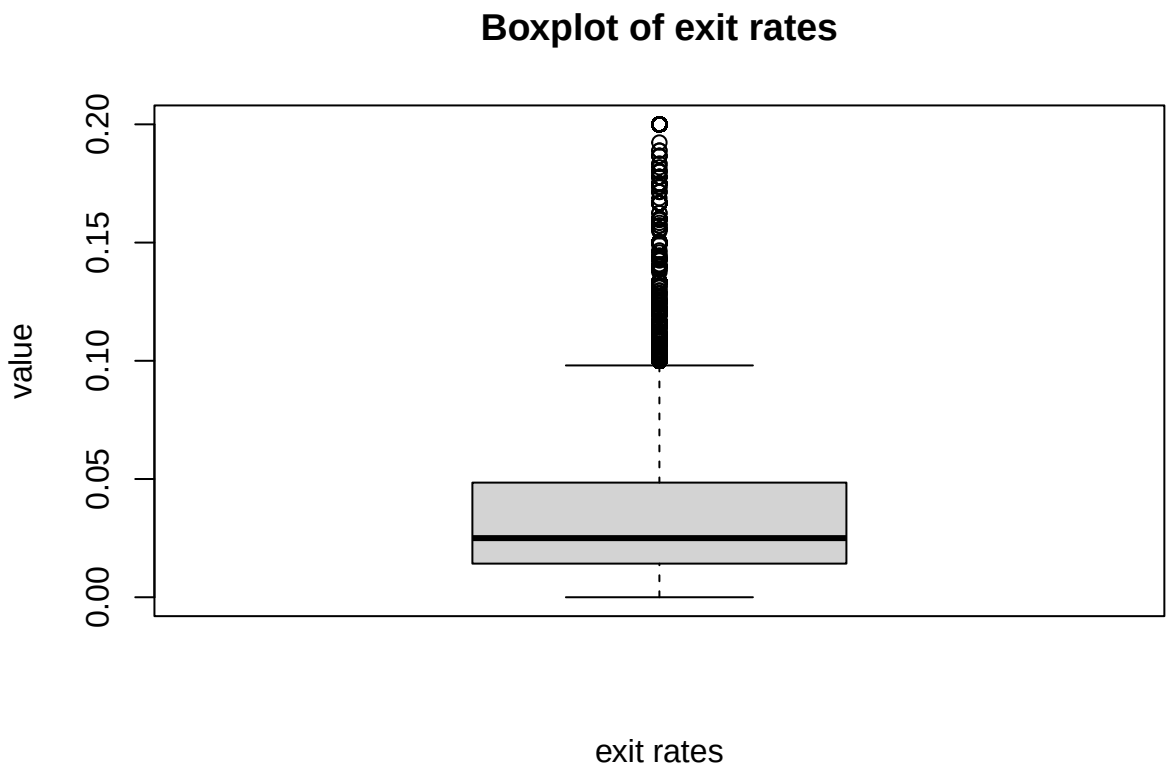

```
boxplot(data2$ProductRelated_Duration, main= 'Boxplot of product related web pages duration', xlab='proc
```



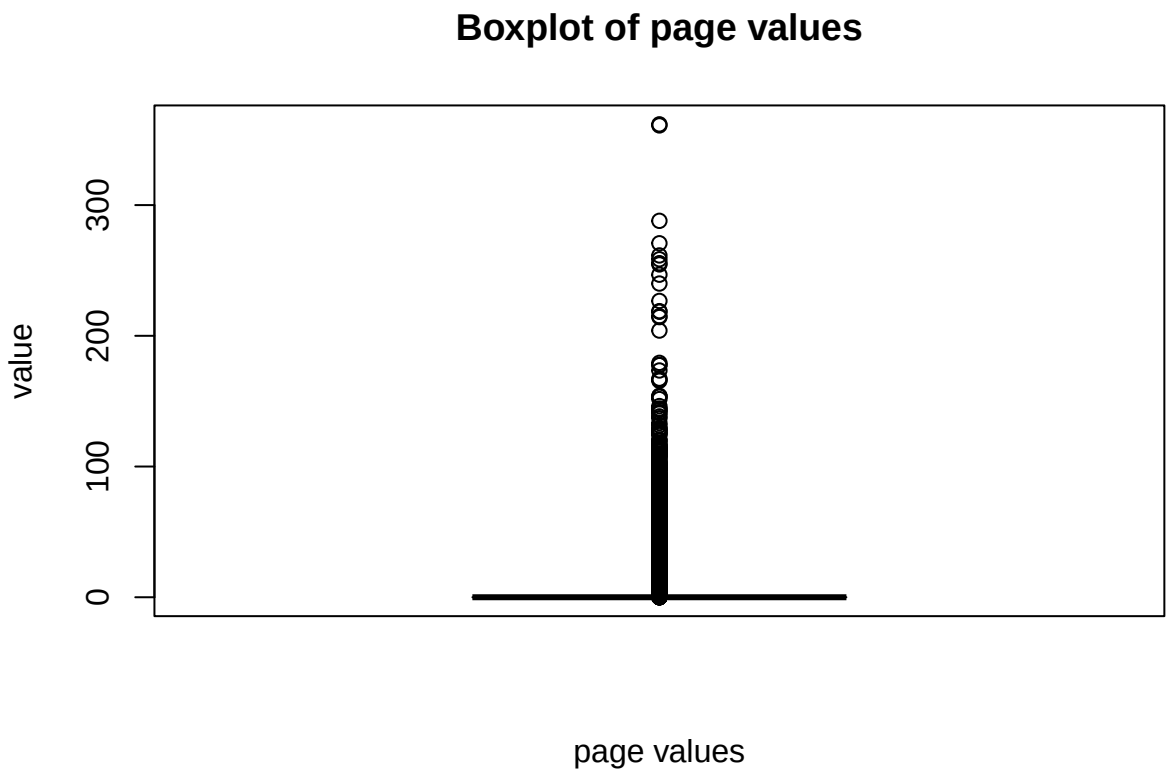

```
boxplot(data2$BounceRates, main= 'Boxplot of bounce rates', xlab='bounce rates', ylab='value')
```



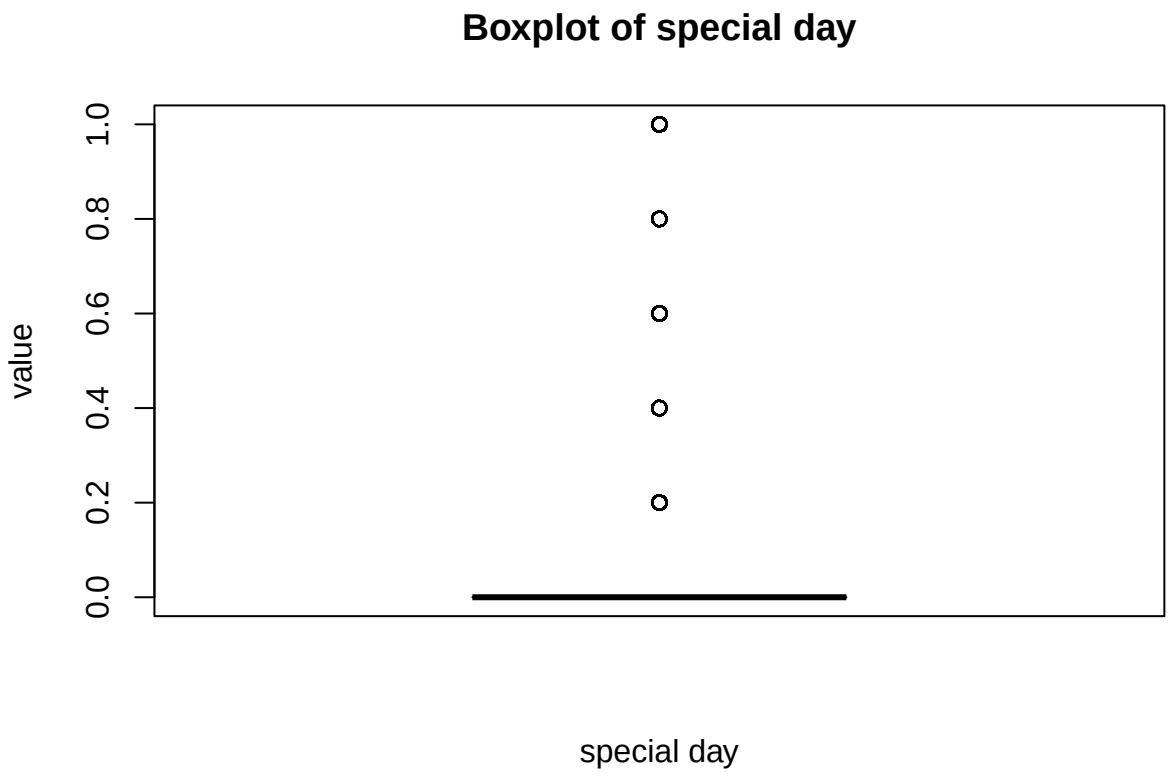

```
boxplot(data2$ExitRates, main= 'Boxplot of exit rates', xlab='exit rates', ylab='value')
```




```
boxplot(data2$PageValues, main= 'Boxplot of page values', xlab='page values', ylab='value')
```




```
boxplot(data2$SpecialDay, main= 'Boxplot of special day', xlab='special day', ylab='value')
```




There are outliers in every column but we will not be removing them since they might contain insights useful to the project.

#5. EDA ##Univariate Analysis ### a) Measures of central tendancy

```
#finding the mean of each of the columns
mean <- colMeans(data2[sapply(data2, is.numeric)])
mean
```

```
##          Administrative Administrative_Duration           Informational
##            2.340028e+00            8.168214e+01            5.088122e-01
##  Informational_Duration          ProductRelated ProductRelated_Duration
##            3.483734e+01            3.205845e+01            1.207508e+03
##             BounceRates               ExitRates              PageValues
##            2.044674e-02            4.149678e-02            5.952500e+00
##              SpecialDay
##            6.197229e-02
```

```
#loading the tidyverse and robustbase(for the colMedians function) libraries
library(robustbase)
```

```
##
## Attaching package: 'robustbase'

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

```
library(tidyverse)
#Finding the median
median <- data2%>%
  select_if(is.numeric) %>%
  as.matrix()%>%
  colMedians()
print(median)
```

```
##          Administrative Administrative_Duration           Informational
##            1.000000e+00            9.000000e+00            0.000000e+00
##  Informational_Duration          ProductRelated ProductRelated_Duration
##            0.000000e+00            1.800000e+01            6.095417e+02
##             BounceRates               ExitRates              PageValues
##            2.930403e-03            2.500000e-02            0.000000e+00
##              SpecialDay
##            0.000000e+00
```

```
# finding the mode of each column
# defining the mode function
mode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
# listing columns and calling it cn
#cn <- list("Administrative","Administrative_Duration","Informational","Informational_Duration","Produc

# creating the mode loop
#for (i in columns) {
#  print(mode(data2$i))
#}

mode(data2$Administrative)
```

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

```
mode(data2$Administrative_Duration)
```

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

```
mode(data2$Informational)
```

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

```
mode(data2$Informational_Duration)
```

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

```
mode(data2$ProductRelated)
```

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

```
mode(data2$ProductRelated_Duration)
```

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

```
mode(data2$BounceRates)
```

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

```
mode(data2$ExitRates)
```

```
## [1] 0.2
```

```
mode(data2$PageValues)
```

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

```
mode(data2$SpecialDay)
```

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

## b) Measures of dispersion

```
#finding the minimum value for each column
min(data2$Administrative)
```

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

```
min(data2$Administrative_Duration)
```

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

```
min(data2$Informational)
```

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

```
min(data2$Informational_Duration)
```

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

```
min(data2$ProductRelated)
```

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

```r
min(data2$ProductRelated_Duration)
```

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

```r
min(data2$BounceRates)
```

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

```r
min(data2$ExitRates)
```

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

```r
min(data2$PageValues)
```

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

```r
min(data2$SpecialDay)
```

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

```r
#finding the maximum value in each column
max(data2$Administrative)
```

```
## [1] 27
```

```r
max(data2$Administrative_Duration)
```

```
## [1] 3398.75
```

```r
max(data2$Informational)
```

```
## [1] 24
```

```r
max(data2$Informational_Duration)
```

```
## [1] 2549.375
```

```r
max(data2$ProductRelated)
```

```
## [1] 705
```

```r
max(data2$ProductRelated_Duration)
```

```
## [1] 63973.52
```

```r
max(data2$BounceRates)
```

```
## [1] 0.2
```

```r
max(data2$ExitRates)
```

```
## [1] 0.2
```

```r
max(data2$PageValues)
```

```
## [1] 361.7637
```

```r
max(data2$SpecialDay)
```

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

```r
#finding the ranges of each column

cat("Administrative:",range(data2$Administrative))
```

```
## Administrative: 0 27
```

```r
cat("|||Informational:", range(data2$Informational))
```

```
## |||Informational: 0 24
```

```r
cat("|||Informational_Duration:", range(data2$Informational_Duration))
```

```
## |||Informational_Duration: -1 2549.375
```

```r
cat("|||ProductRelated:", range(data2$ProductRelated))
```

```
## |||ProductRelated: 0 705
```

```r
cat("|||ProductRelated_Duration:", range(data2$ProductRelated_Duration))
```

```
## |||ProductRelated_Duration: -1 63973.52
```

```r
cat("|||BounceRates:", range(data2$BounceRates))
```

```
## |||BounceRates: 0 0.2
```

```r
cat("|||ExitRates:", range(data2$ExitRates))
```

```
## |||ExitRates: 0 0.2
```

```
cat("|||PageValues:", range(data2$PageValues))
```

```
## |||PageValues: 0 361.7637
```

```
cat("|||SpecialDay:", range(data2$SpecialDay))
```

```
## |||SpecialDay: 0 1
```

the ranges coinside with the minimum and maximum values of each column

```
#finding the quantiles
print("|||Administrative:")
```

```
## [1] "|||Administrative:"
```

```
quantile(data2$Administrative)
```

```
##   0%  25%  50%  75% 100%
##    0    0    1    4   27
```

```
print("|||Administrative_Duration:")
```

```
## [1] "|||Administrative_Duration:"
```

```
quantile(data2$Administrative_Duration)
```

```
##      0%     25%     50%     75%    100%
##   -1.00    0.00    9.00   94.75 3398.75
```

```
print("|||Informational:")
```

```
## [1] "|||Informational:"
```

```
quantile(data2$Informational)
```

```
##   0%  25%  50%  75% 100%
##    0    0    0    0   24
```

```
print("|||Informational_Duration:")
```

```
## [1] "|||Informational_Duration:"
```

```
quantile(data2$Informational_Duration)
```

```
##       0%      25%      50%      75%     100%
##   -1.000    0.000    0.000    0.000 2549.375
```

```
print("|||ProductRelated:")
```

```
## [1] "|||ProductRelated:"
```

```
quantile(data2$ProductRelated)
```

```
##   0%  25%  50%  75% 100%
##    0    8   18   38  705
```

```
print("|||ProductRelated_Duration:")
```

```
## [1] "|||ProductRelated_Duration:"
```

```
quantile(data2$ProductRelated_Duration)
```

```
##         0%        25%        50%        75%       100%
##    -1.0000   193.5833   609.5417  1477.5648 63973.5222
```

```
print("|||BounceRates:")
```

```
## [1] "|||BounceRates:"
```

```
quantile(data2$BounceRates)
```

```
##          0%         25%         50%         75%        100%
## 0.000000000 0.000000000 0.002930403 0.016666667 0.200000000
```

```
print("|||ExitRates:")
```

```
## [1] "|||ExitRates:"
```

```
quantile(data2$ExitRates)
```

```
##         0%        25%        50%        75%       100%
## 0.00000000 0.01422258 0.02500000 0.04848485 0.20000000
```

```
print("|||PageValues:")
```

```
## [1] "|||PageValues:"
```

```
quantile(data2$PageValues)
```

```
##       0%      25%      50%      75%     100%
##   0.0000   0.0000   0.0000   0.0000 361.7637
```

```r
print("|||SpecialDay:")
```

```
## [1] "|||SpecialDay:"
```

```r
quantile(data2$SpecialDay)
```

```
##   0%  25%  50%  75% 100%
##    0    0    0    0    1
```

```r
#finding the standard deviation
sd(data2$Administrative)
```

```
## [1] 3.330851
```

```r
sd(data2$Administrative_Duration)
```

```
## [1] 177.5282
```

```r
sd(data2$Informational)
```

```
## [1] 1.275817
```

```r
sd(data2$Informational_Duration)
```

```
## [1] 141.4585
```

```r
sd(data2$ProductRelated)
```

```
## [1] 44.60091
```

```r
sd(data2$ProductRelated_Duration)
```

```
## [1] 1919.927
```

```r
sd(data2$BounceRates)
```

```
## [1] 0.0454025
```

```r
sd(data2$ExitRates)
```

```
## [1] 0.04624716
```

```r
sd(data2$PageValues)
```

```
## [1] 18.65779
```

```r
sd(data2$SpecialDay)
```

```
## [1] 0.1997106
```

```r
#finding variance
var(data2$Administrative)
```

```
## [1] 11.09457
```

```r
var(data2$Administrative_Duration)
```

```
## [1] 31516.25
```

```r
var(data2$Informational)
```

```
## [1] 1.62771
```

```r
var(data2$Informational_Duration)
```

```
## [1] 20010.51
```

```r
var(data2$ProductRelated)
```

```
## [1] 1989.241
```

```r
var(data2$ProductRelated_Duration)
```

```
## [1] 3686121
```

```r
var(data2$BounceRates)
```

```
## [1] 0.002061387
```

```r
var(data2$ExitRates)
```

```
## [1] 0.0021388
```

```r
var(data2$PageValues)
```

```
## [1] 348.1132
```

```r
var(data2$SpecialDay)
```

```
## [1] 0.03988432
```

```
#finding kurtosis
library(e1071)
skewness(data2$Administrative)
```

```
## [1] 1.946009
```

```
skewness(data2$Administrative_Duration)
```

```
## [1] 5.589523
```

```
skewness(data2$Informational)
```

```
## [1] 4.012958
```

```
skewness(data2$Informational_Duration)
```

```
## [1] 7.536508
```

```
skewness(data2$ProductRelated)
```

```
## [1] 4.331601
```

```
skewness(data2$ProductRelated_Duration)
```

```
## [1] 7.250512
```

```
skewness(data2$BounceRates)
```

```
## [1] 3.152486
```

```
skewness(data2$ExitRates)
```

```
## [1] 2.232851
```

```
skewness(data2$PageValues)
```

```
## [1] 6.347882
```

```
skewness(data2$SpecialDay)
```

```
## [1] 3.284077
```

```
#finding skewness
kurtosis(data2$Administrative)
```

```
## [1] 4.634854
```

```
kurtosis(data2$Administrative_Duration)
```

```
## [1] 50.08518
```

```
kurtosis(data2$Informational)
```

```
## [1] 26.63768
```

```
kurtosis(data2$Informational_Duration)
```

```
## [1] 75.45122
```

```
kurtosis(data2$ProductRelated)
```

```
## [1] 31.04345
```

```
kurtosis(data2$ProductRelated_Duration)
```

```
## [1] 136.5679
```

```
kurtosis(data2$BounceRates)
```

```
## [1] 9.253055
```

```
kurtosis(data2$ExitRates)
```

```
## [1] 4.623003
```

```
kurtosis(data2$PageValues)
```

```
## [1] 64.92917
```

```
kurtosis(data2$SpecialDay)
```

```
## [1] 9.783958
```

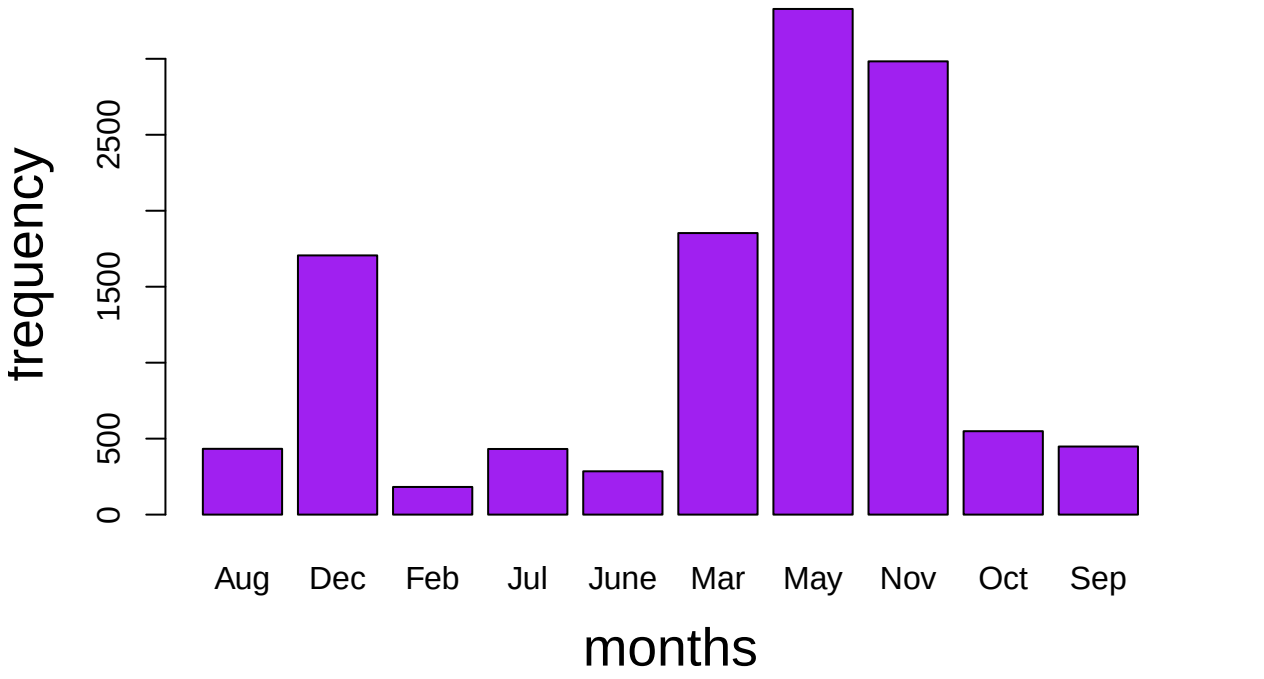
## Univariate Analysis Graphicals

### a) Categorical columns

```
#frequency table of month
month.freq <- table(data2$Month)
sort(month.freq, decreasing = TRUE)[1:5]
```

```
##
##  May  Nov  Mar  Dec  Oct
## 3328 2983 1853 1706  549
```

```
#Bar chart to show frequency distribution of months
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(month.freq), main="Frequency of time when data was captured.",
        xlab="months",
        ylab="frequency",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("purple"))
```

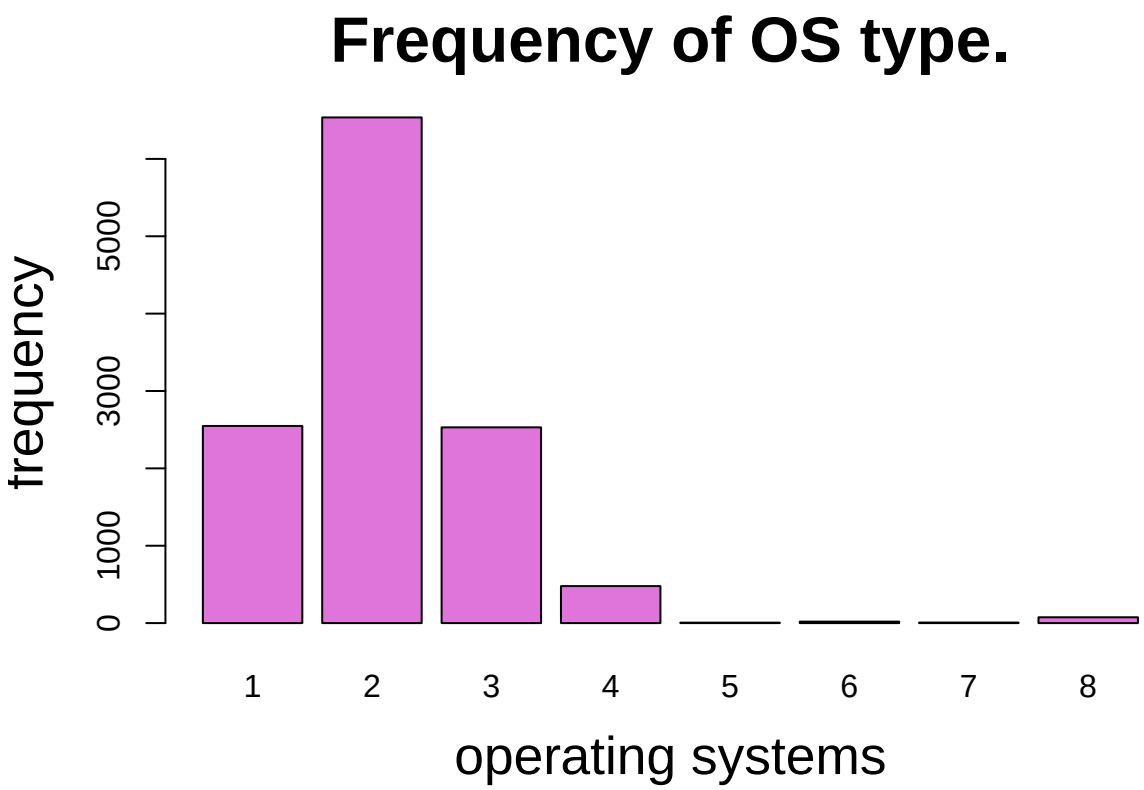



**Observation:** May, November and March had the most web page visits while February and June had the least.

```
#frequency table of Operating systems
os.freq <- table(data2$OperatingSystems)
sort(os.freq, decreasing = TRUE)[1:5]
```

```
##
##    2    1    3    4    8
## 6536 2548 2530  478   75
```

```
#Bar chart to show frequency distribution of operating systems
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(os.freq), main="Frequency of OS type.",
        xlab="operating systems",
        ylab="frequency",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#DF75DA"))
```

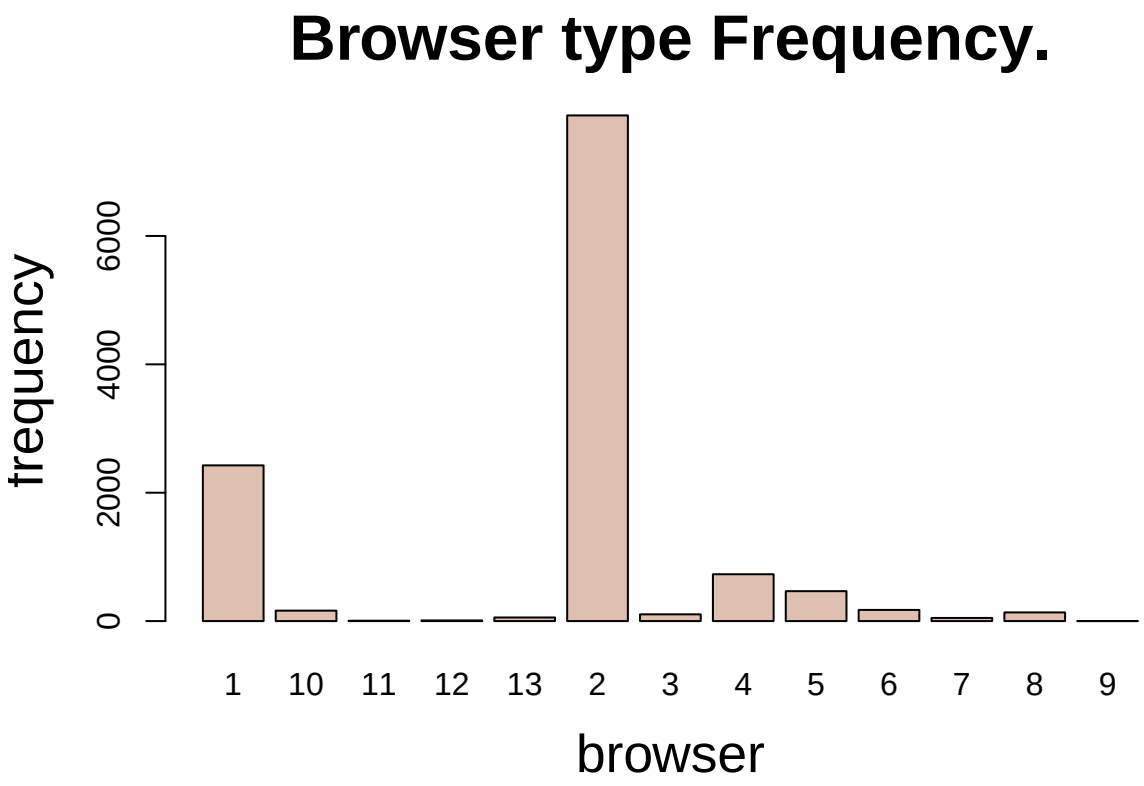



**Observation:** Type 2 operating systenm is the most used followed by 1 and 3.

```
#frequency table of browser
browser.freq <- table(data2$Browser)
sort(browser.freq, decreasing = TRUE)[1:5]
```

```
##
##    2    1    4    5    6
## 7878 2426  730  466  174
```

```
#Bar chart to show frequency distribution of browsers
options(repr.plot.width = 10, repr.plot.height = 15)
barplot(c(browser.freq), main="Browser type Frequency.",
        xlab="browser",
        ylab="frequency",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#DDC0B2"))
```

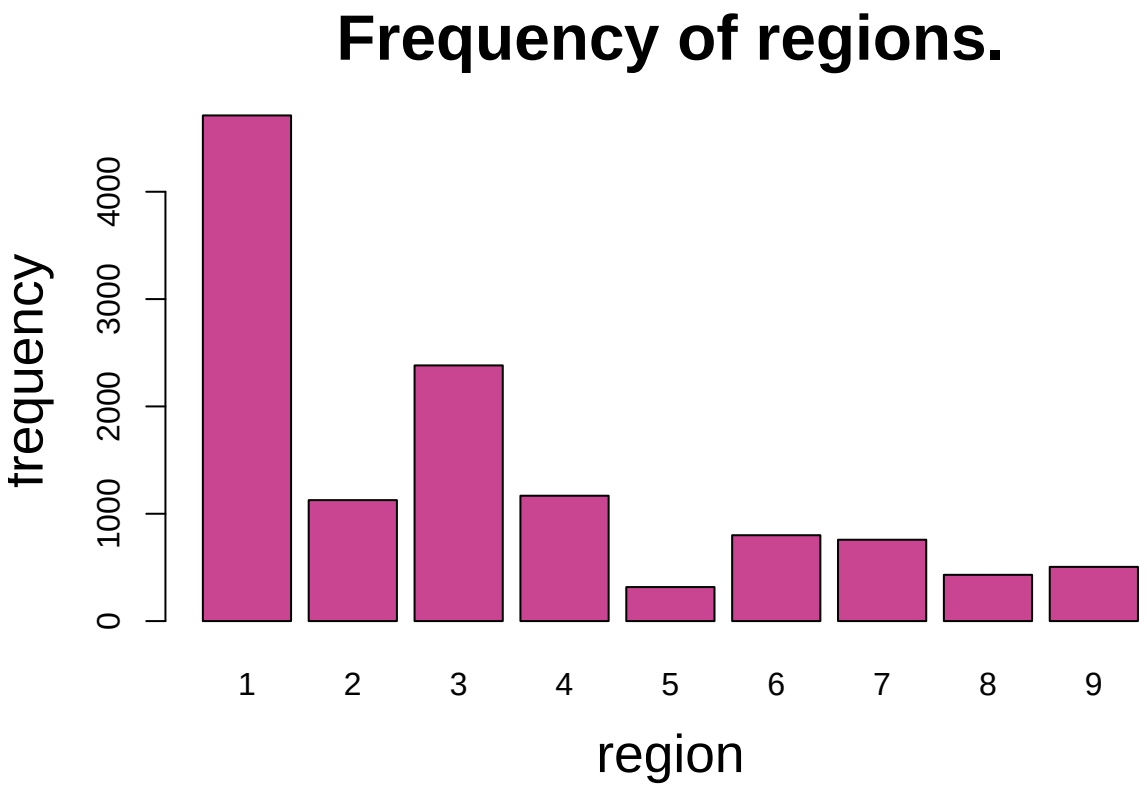



**Observation:** Browser type 2 is by far the most used followed by type 1.

```
#frequency table of region
region.freq <- table(data2$Region)
sort(region.freq, decreasing = TRUE)[1:5]
```

```
##
##    1    3    4    2    6
## 4711 2382 1168 1127  800
```

```
#Bar chart to show frequency distribution of regions
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(region.freq), main="Frequency of regions.",
        xlab="region",
        ylab="frequency",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#C94591"))
```

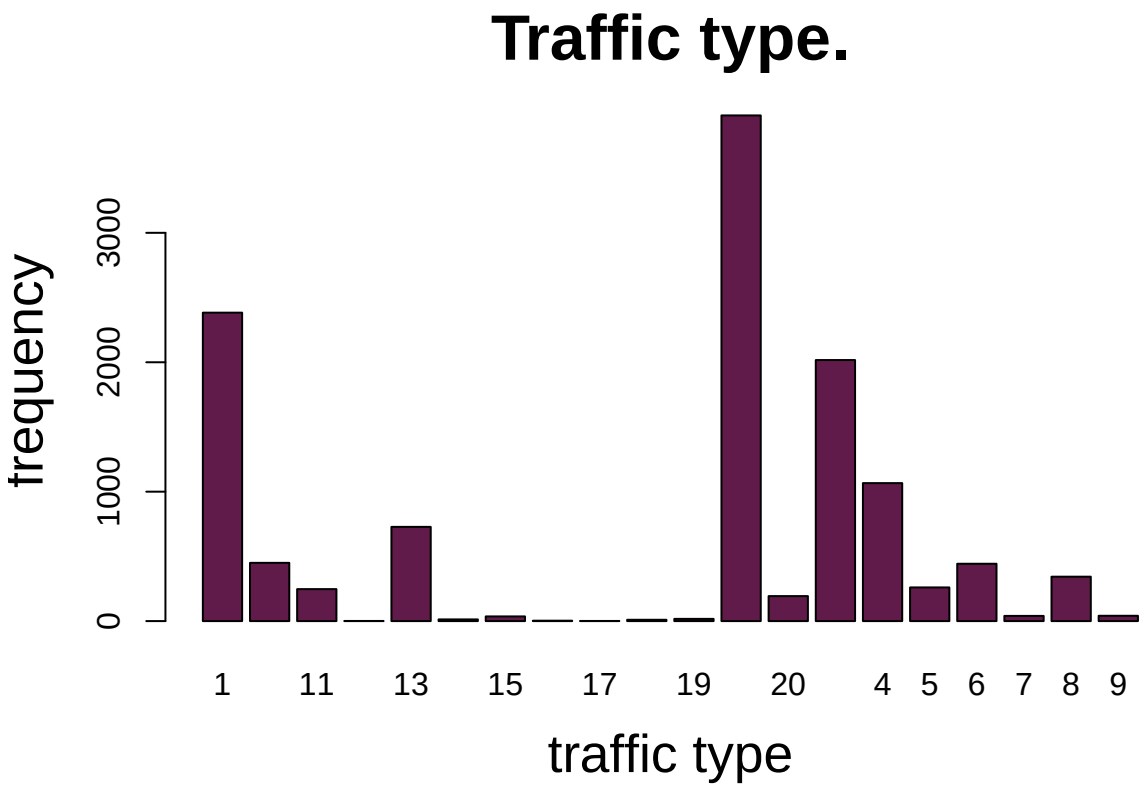



**Observation** The highest number of individuals in the dataset came from region 1 followed by region 3. The least people come from region 5

```
#frequency table of traffic type
traffic.freq <- table(data2$TrafficType)
sort(traffic.freq, decreasing = TRUE)[1:5]
```

```
##
##    2    1    3    4   13
## 3907 2383 2017 1066  728
```

```
#Bar chart to show frequency distribution of traffic type
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(traffic.freq), main="Traffic type.",
        xlab="traffic type",
        ylab="frequency",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#601B4A"))
```

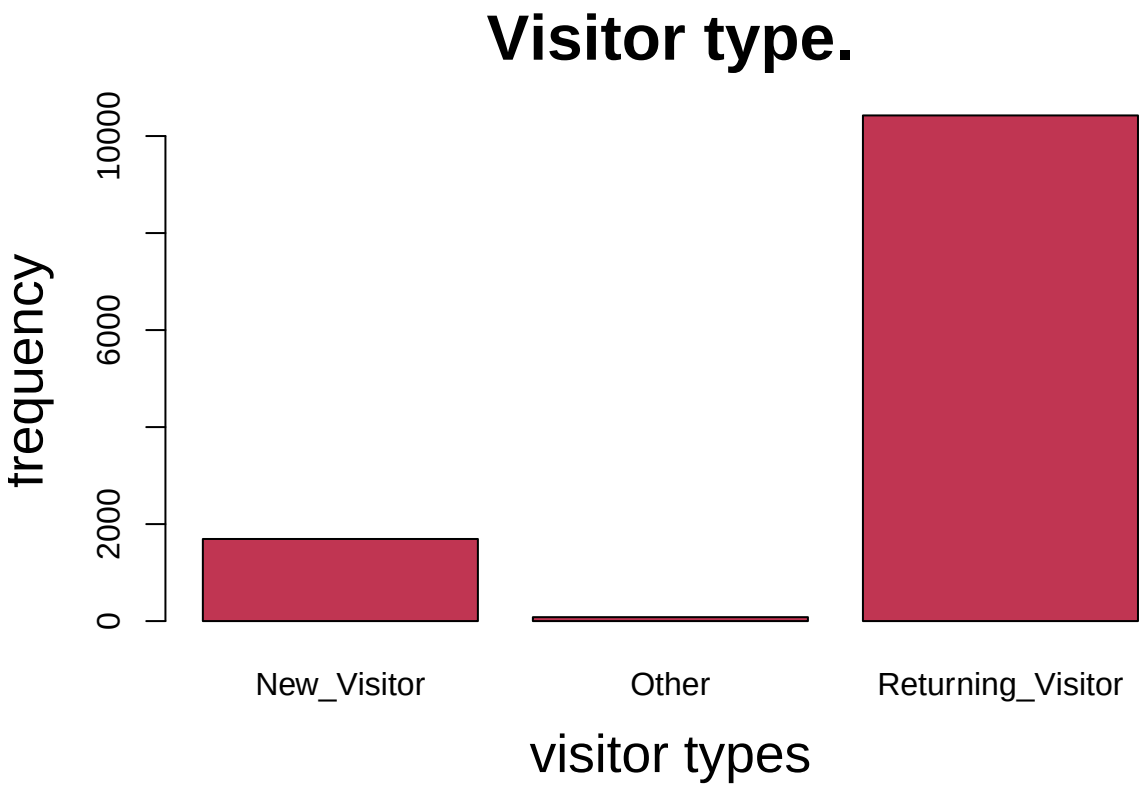



**Observation** Traffic type 2 is the most frequent followed by 1 and 3.

```
#frequency table of visitor type
visitor.freq <- table(data2$VisitorType)
sort(visitor.freq, decreasing = TRUE)[1:5]
```

```
##
## Returning_Visitor       New_Visitor             Other              <NA>
##             10425              1693                81
##              <NA>
##
```

```
#Bar chart to show frequency distribution of visitor type
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(visitor.freq), main="Visitor type.",
        xlab="visitor types",
        ylab="frequency",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#C03552"))
```

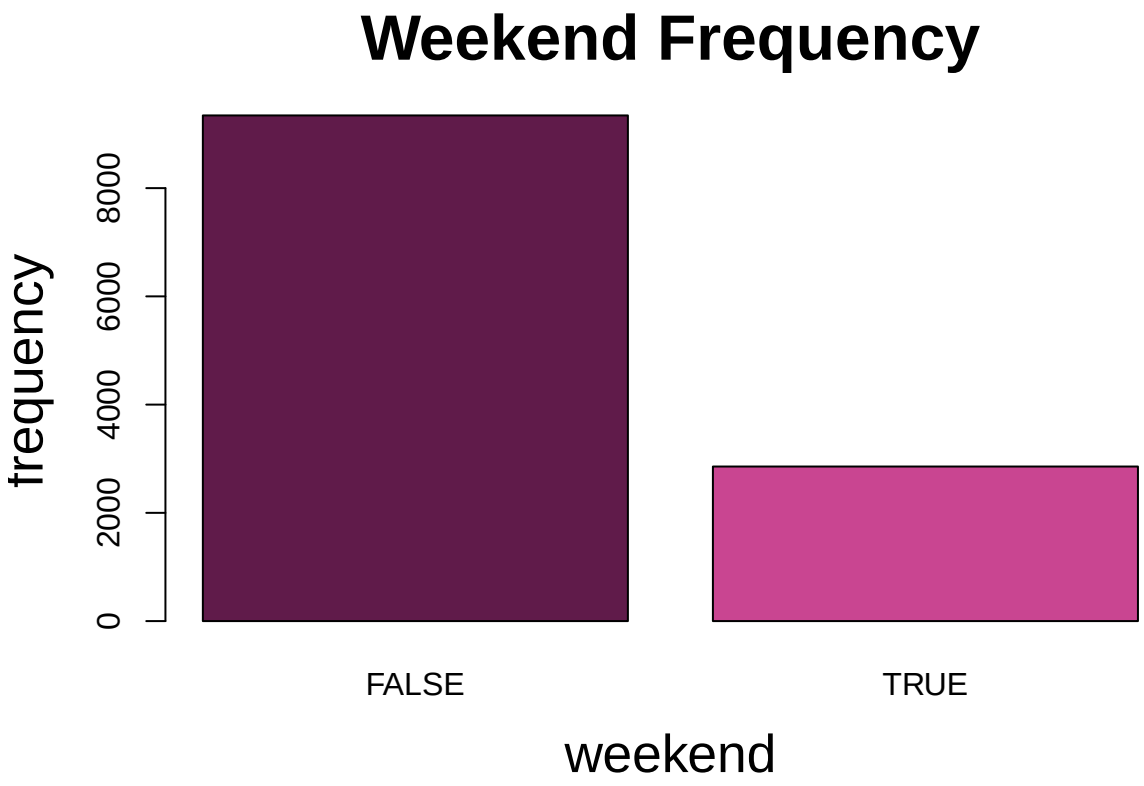



**Observation:** Returning visitors formed the largest portion of the data, followed by New visitors then other.

```
#frequency table of weekend
weekend.freq <- table(data2$Weekend)
sort(weekend.freq, decreasing = TRUE)[1:5]
```

```
##
## FALSE  TRUE  <NA>  <NA>  <NA>
##  9343  2856
```

```
#Bar chart to show frequency distribution of weekend
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(weekend.freq), main="Weekend Frequency",
        xlab="weekend",
        ylab="frequency",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#601B4A", "#C94591"))
```

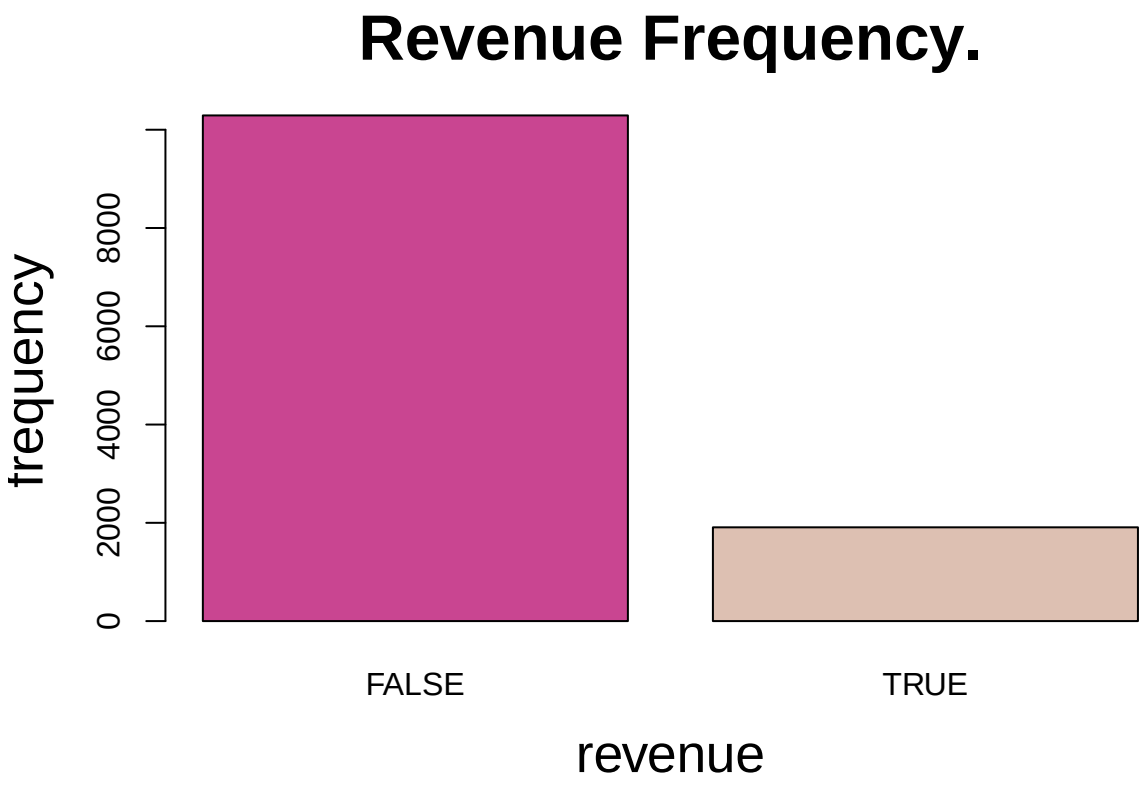



**Observations** More web page visits were made during the weekday.

```
#frequency table of revenue
rev.freq <- table(data2$Revenue)
sort(rev.freq, decreasing = TRUE)[1:5]
```

```
##
## FALSE  TRUE  <NA>  <NA>  <NA>
## 10291  1908
```

```
#Bar chart to show frequency distribution of revenue
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(rev.freq), main="Revenue Frequency.",
        xlab="revenue",
        ylab="frequency",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#C94591", "#DDC0B2"))
```

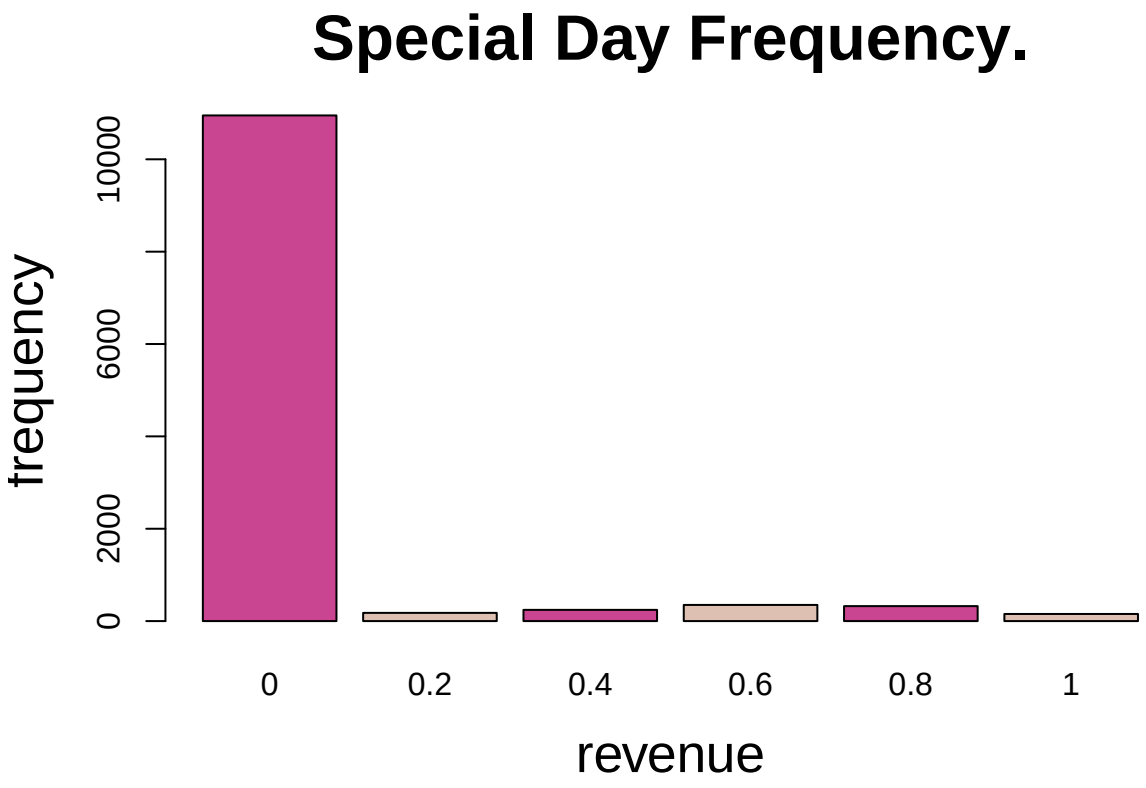



**Observation:** The company didn't generate revenue from most of the people who's data was captured.

```
#frequency table of revenue
rev.freq <- table(data2$SpecialDay)
sort(rev.freq, decreasing = TRUE)[1:5]
```

```
##
##     0   0.6   0.8   0.4   0.2
## 10950   350   324   243   178
```

```
#Bar chart to show frequency distribution of revenue
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(rev.freq), main="Special Day Frequency.",
        xlab="revenue",
        ylab="frequency",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#C94591", "#DDC0B2"))
```

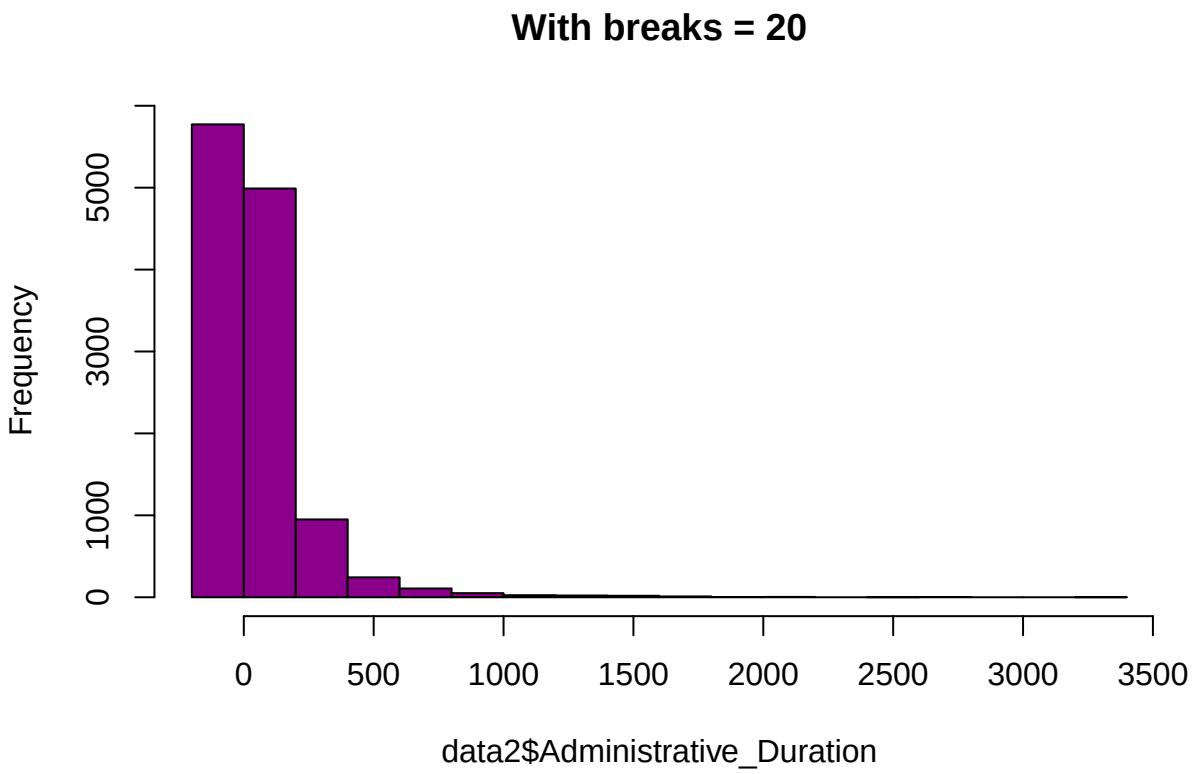



**Numerical variables**

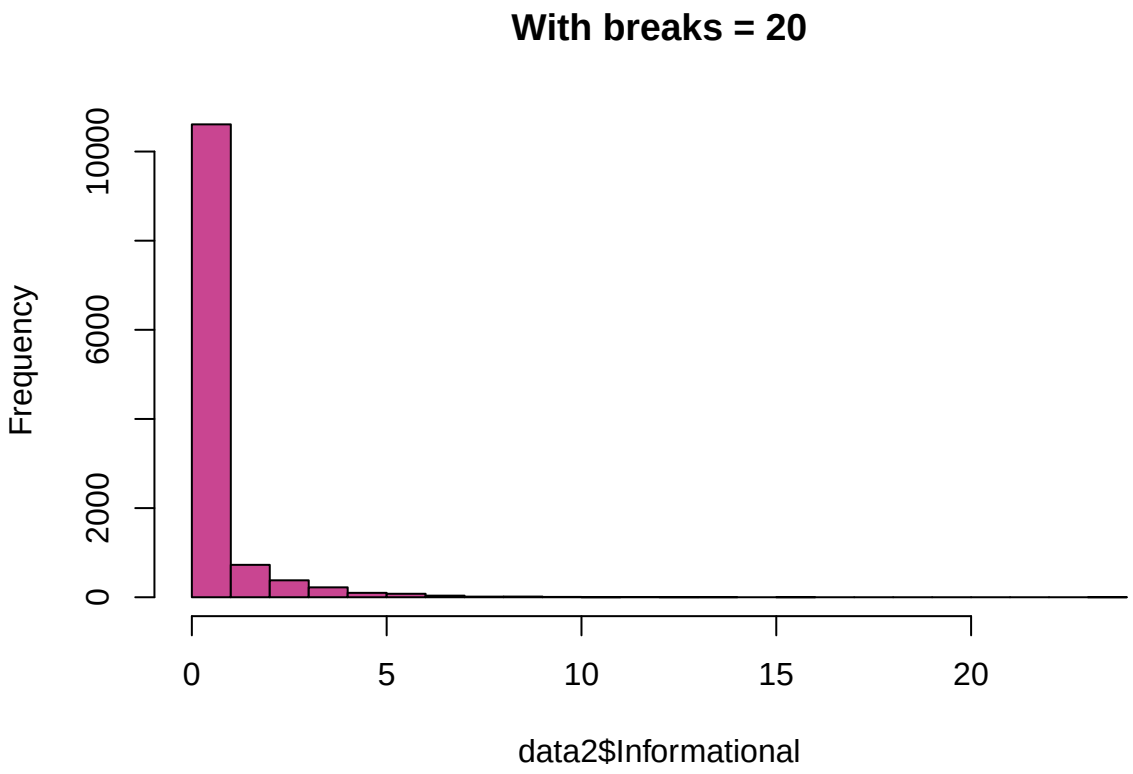
```
# getting column names so we can generate histograms in order to
colnames(data2)
```

```
##  [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"
```

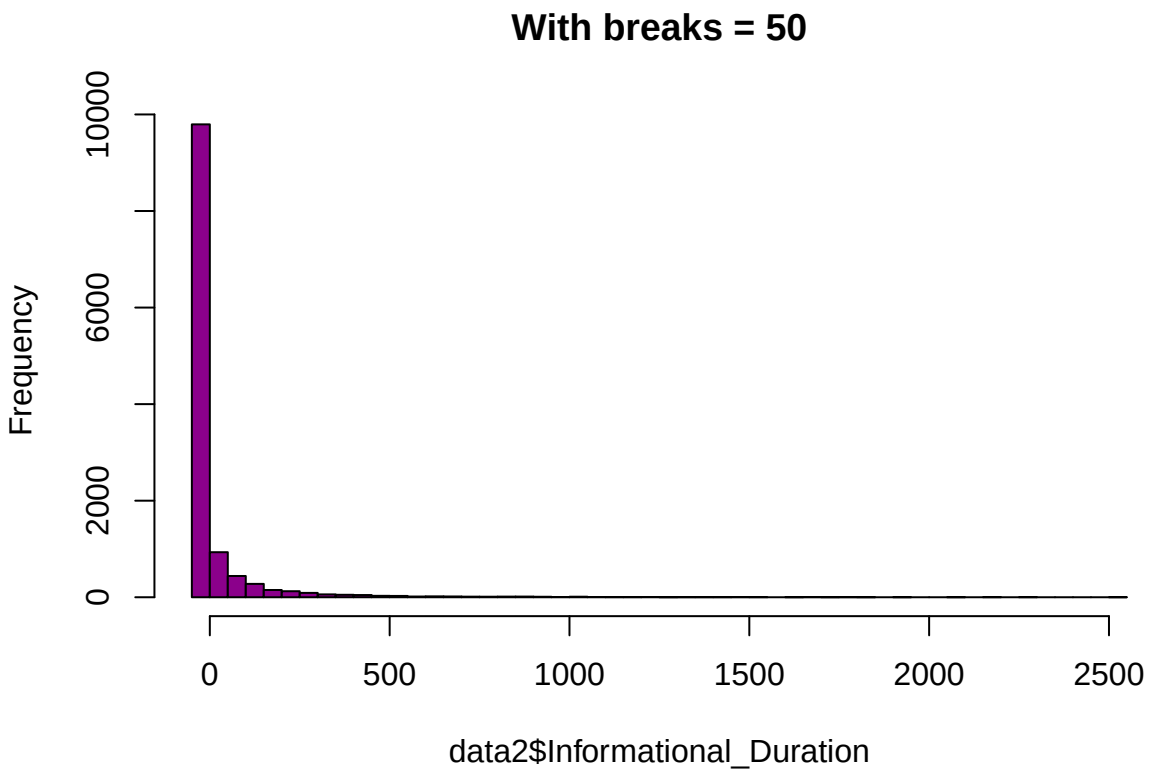
```
#creating a histogram of administrative duration variable
options(repr.plot.width = 10, repr.plot.height = 20)
hist(data2$Administrative_Duration,breaks=20, main="With breaks = 20",col="darkmagenta")
```



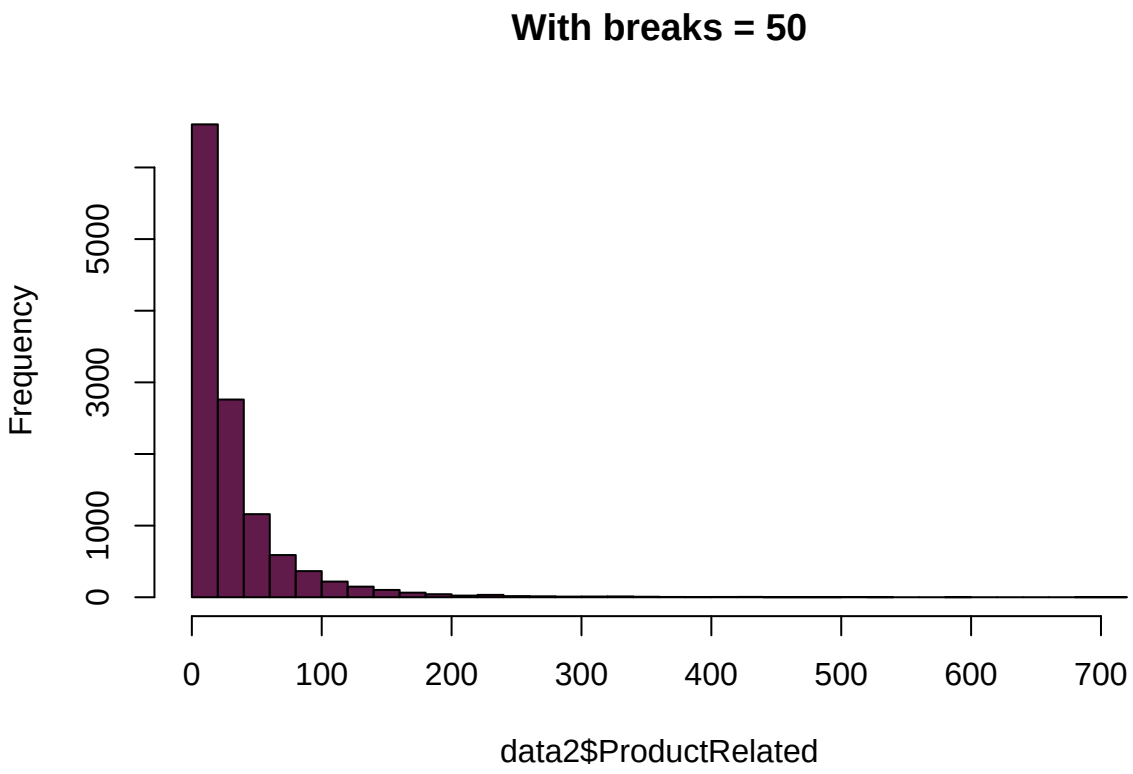

```
#creating a histogram of informational variable
hist(data2$Informational, breaks=20, main="With breaks = 20",col="#C94591")
```



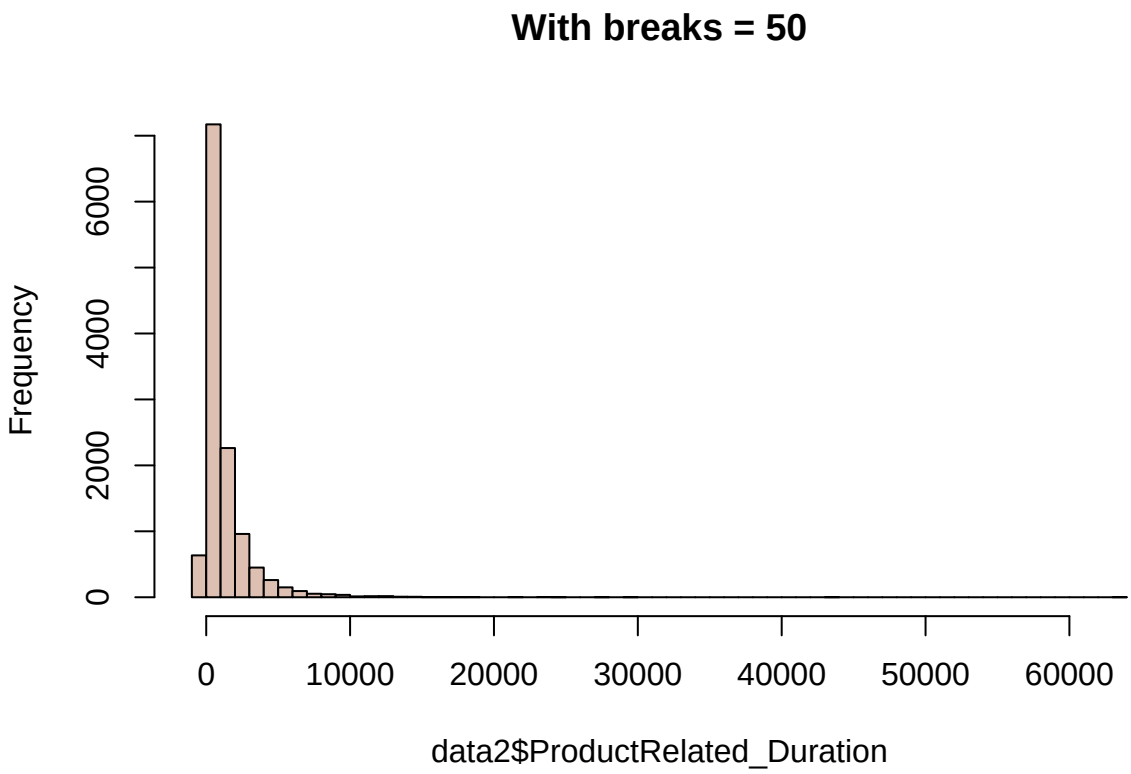

```
#creating a histogram of informational duration variable
hist(data2$Informational_Duration,breaks=50, main="With breaks = 50",col="darkmagenta")
```



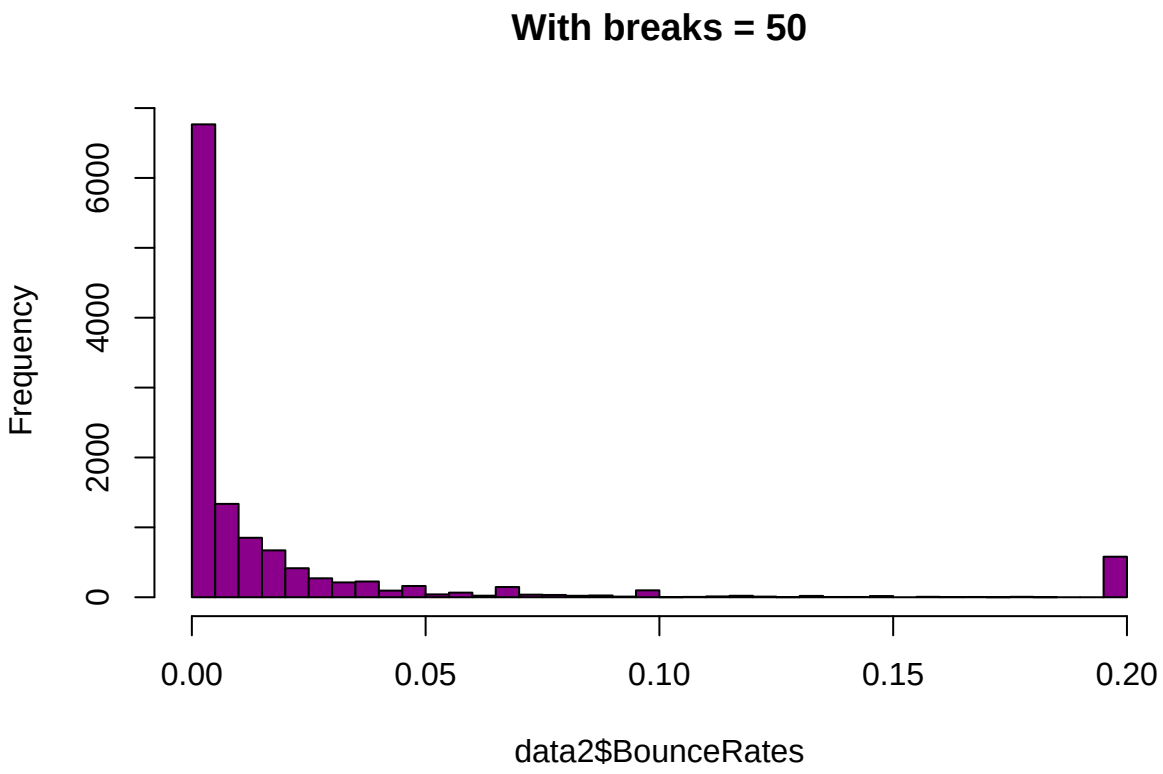

```
#creating a histogram of product related variable
hist(data2$ProductRelated,breaks=50, main="With breaks = 50",col="#601B4A")
```



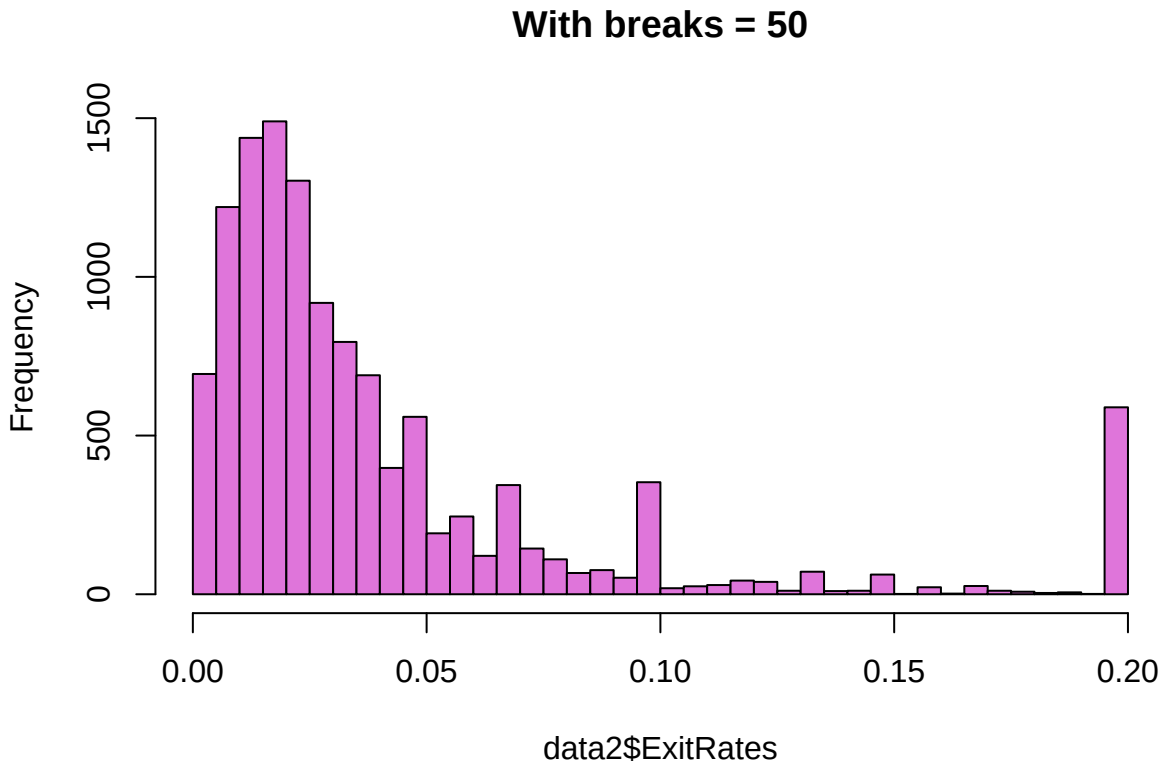

```
#creating a histogram of product related duration variable
hist(data2$ProductRelated_Duration,breaks=50, main="With breaks = 50",col="#DDC0B2")
```



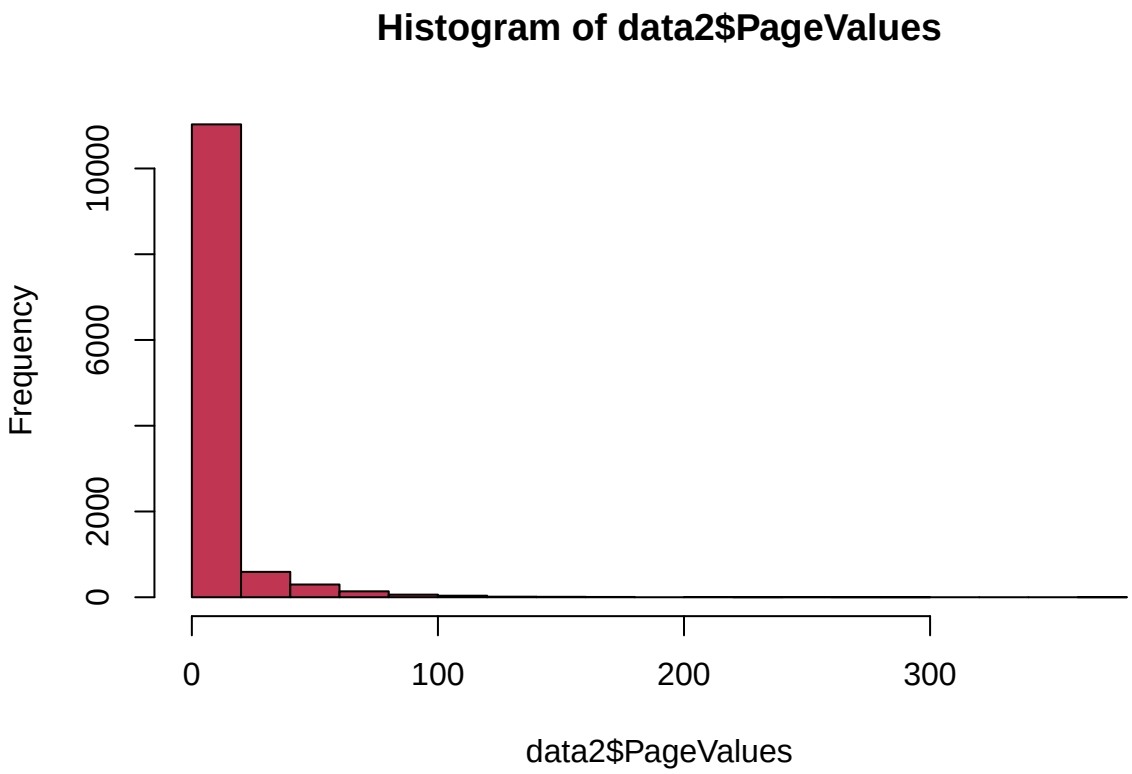

```
#creating a histogram of bounce rates variable
hist(data2$BounceRates,breaks=50, main="With breaks = 50",col="darkmagenta")
```



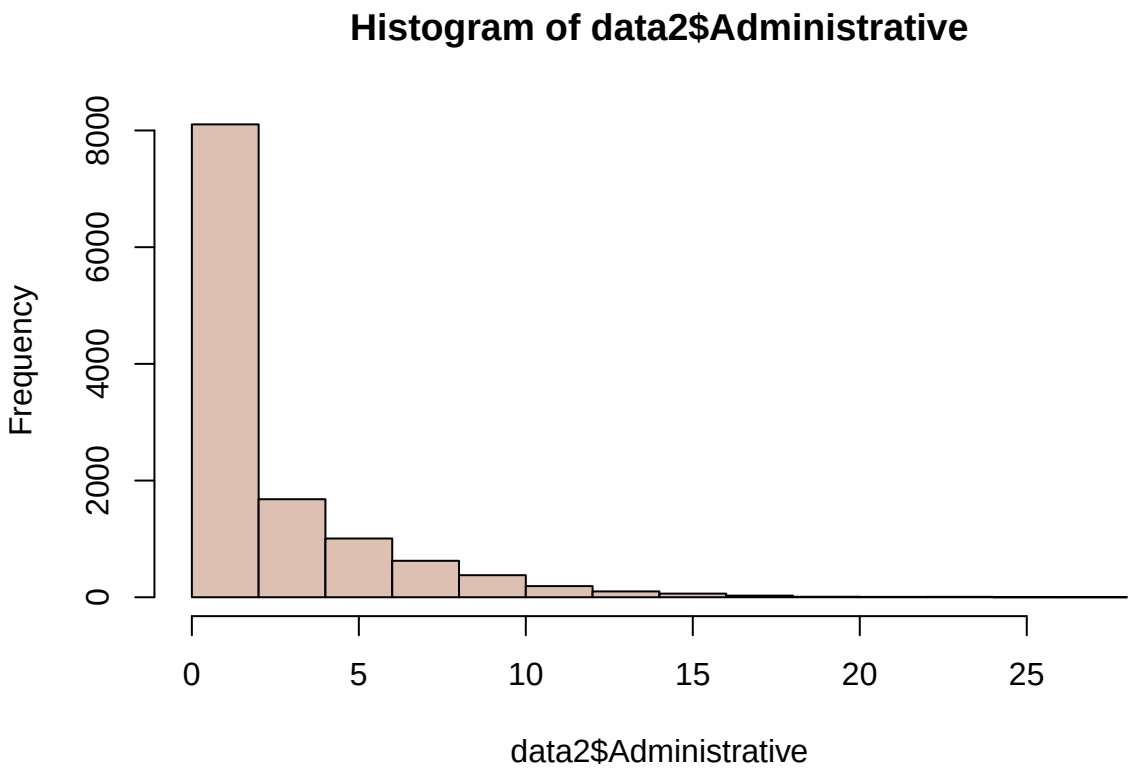

```
#creating a histogram of exit rates variable
hist(data2$ExitRates,breaks=50, main="With breaks = 50",col="#DF75DA")
```




```
#creating a histogram of page values variable
hist(data2$PageValues,col="#C03552")
```




```
#creating a histogram of special day variable
hist(data2$Administrative, col="#DDC0B2")
```



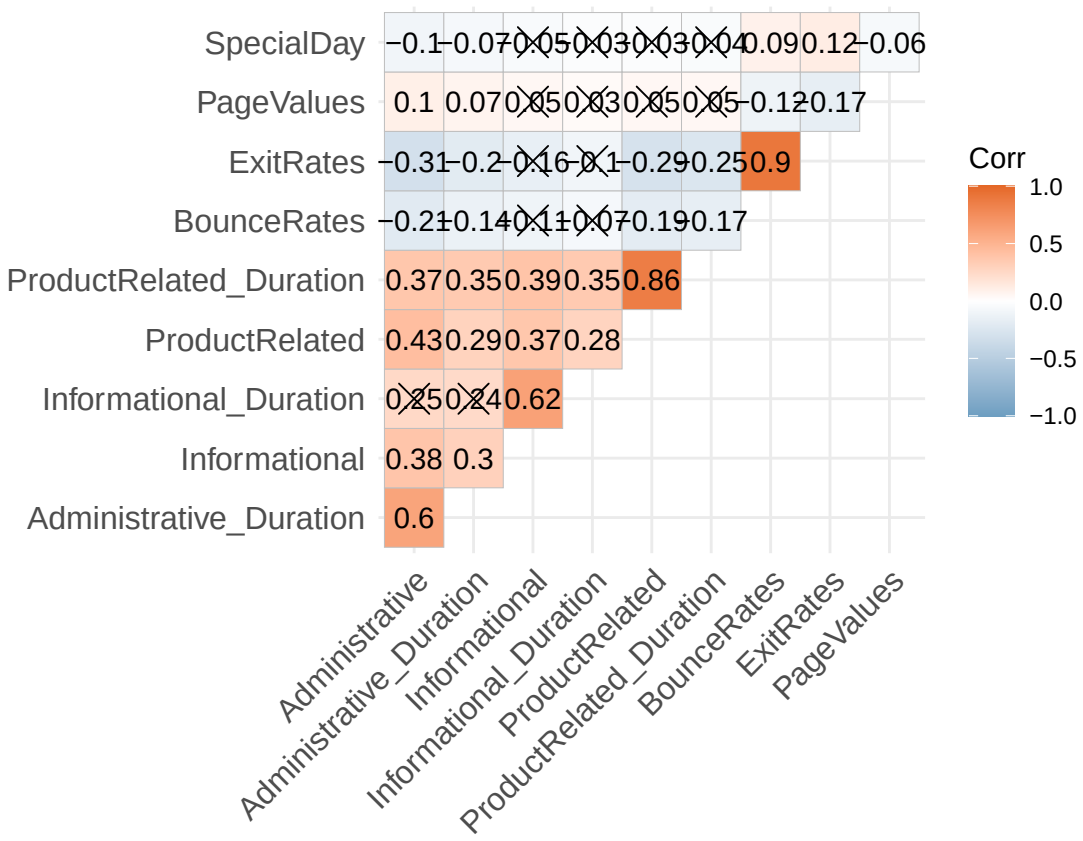

#6. Bivariate & Multivariate Analysis

```
#finding the covariance
cov <- cov(data2[,unlist(lapply(data2, is.numeric))])
cov
```

```
##                          Administrative Administrative_Duration Informational
## Administrative             11.09456996              355.034186   1.594806280
## Administrative_Duration   355.03418646            31516.250360  68.273361883
## Informational               1.59480628               68.273362   1.627709681
## Informational_Duration    120.04936778             5956.517671 111.656022657
## ProductRelated             63.61170357             2270.731540  21.202182071
## ProductRelated_Duration  2372.71642208           120492.067559 945.703033133
## BounceRates                -0.03231259               -1.106938  -0.006343127
## ExitRates                  -0.04794942               -1.658656  -0.009414909
## PageValues                  6.02328225              219.168388   1.128072018
## SpecialDay                 -0.06457297               -2.649741  -0.012580917
##                          Informational_Duration ProductRelated
## Administrative                    120.0493678     63.6117036
## Administrative_Duration          5956.5176708   2270.7315396
## Informational                     111.6560227     21.2021821
## Informational_Duration          20010.5068642   1760.6514935
## ProductRelated                   1760.6514935   1989.2412959
## ProductRelated_Duration         94127.8699847  73668.6330189
## BounceRates                        -0.4506041     -0.3918681
## ExitRates                          -0.6733911     -0.5902590
```

```
## PageValues                          79.3484334     45.0324519
## SpecialDay                          -0.8840522     -0.2309712
##                          ProductRelated_Duration   BounceRates     ExitRates
## Administrative                        2372.71642 -3.231259e-02  -0.047949418
## Administrative_Duration             120492.06756 -1.106938e+00  -1.658655837
## Informational                         945.70303 -6.343127e-03  -0.009414909
## Informational_Duration              94127.86998 -4.506041e-01  -0.673391128
## ProductRelated                      73668.63302 -3.918681e-01  -0.590258984
## ProductRelated_Duration           3686121.49674 -1.520023e+01 -21.783499809
## BounceRates                           -15.20023  2.061387e-03   0.001896814
## ExitRates                             -21.78350  1.896814e-03   0.002138800
## PageValues                           1821.19283 -9.825801e-02  -0.149769655
## SpecialDay                            -14.65110  7.964769e-04   0.001078620
##                             PageValues    SpecialDay
## Administrative              6.02328225 -6.457297e-02
## Administrative_Duration   219.16838756 -2.649741e+00
## Informational               1.12807202 -1.258092e-02
## Informational_Duration     79.34843344 -8.840522e-01
## ProductRelated             45.03245187 -2.309712e-01
## ProductRelated_Duration 1821.19282970 -1.465110e+01
## BounceRates               -0.09825801  7.964769e-04
## ExitRates                 -0.14976966  1.078620e-03
## PageValues               348.11318376 -2.404591e-01
## SpecialDay                -0.24045911  3.988432e-02
```

```r
#Finding correlation
corr <- cor(data2[, unlist(lapply(data2, is.numeric))])

p.mat <- cor_pmat(corr, method = "spearman")
ggcorrplot(corr, method = "square", type = "upper",
           colors = c("#6D9EC1", "white", "#E46726"),
           lab = TRUE, p.mat=p.mat, sig.level = .05)
```




```
#selecting the true values from the revenue column
revenue <- data2[data2$Revenue == 'TRUE',]
head(revenue)
```

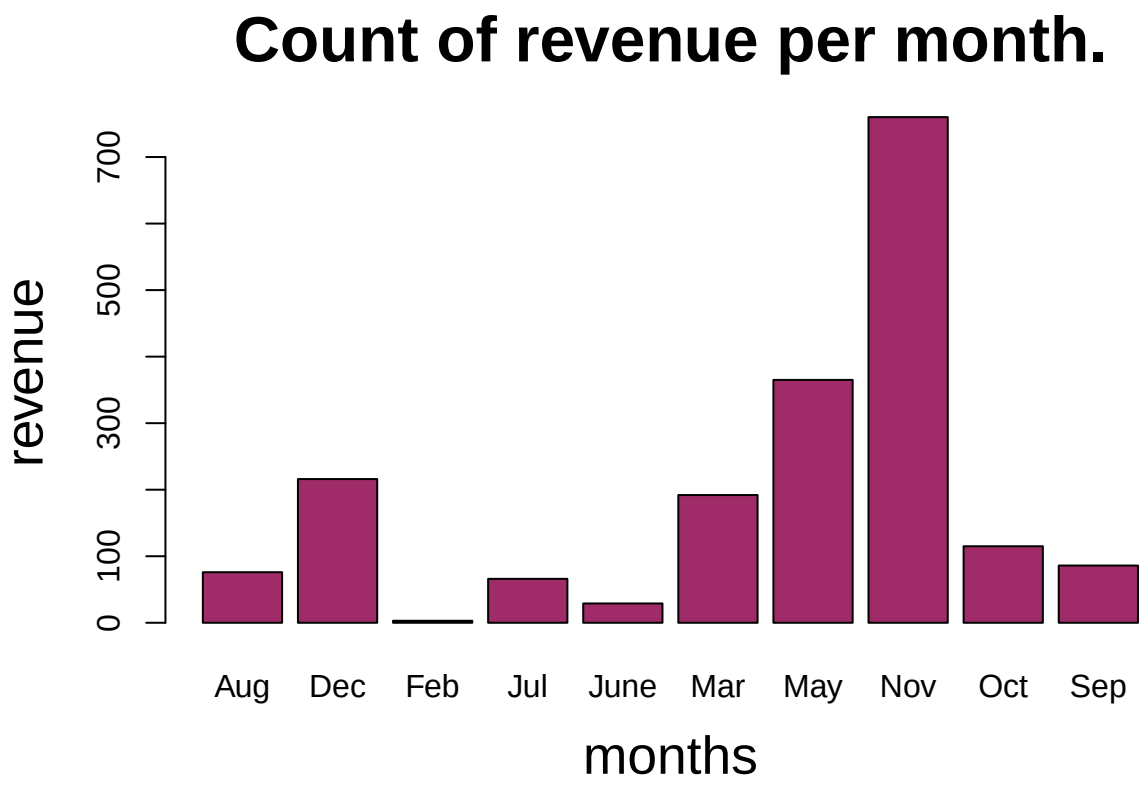
```
##     Administrative Administrative_Duration Informational Informational_Duration
## 66               3                87.83333             0                    0.0
## 77              10              1005.66667             0                    0.0
## 102              4                61.00000             0                    0.0
## 189              9               111.50000             1                   48.5
## 197              2                56.00000             1                  144.0
## 199              0                 0.00000             0                    0.0
##     ProductRelated ProductRelated_Duration BounceRates   ExitRates PageValues
## 66              27               798.3333 0.000000000 0.012643678  22.916036
## 77              36              2111.3417 0.004347826 0.014492754  11.439412
## 102             19               607.0000 0.000000000 0.026984127  17.535959
## 189             49              1868.8197 0.000000000 0.020708874   1.706015
## 197             67              2563.7833 0.000000000 0.005797101  19.342650
## 199             17               840.2333 0.000000000 0.001666667 109.176000
##     SpecialDay Month OperatingSystems Browser Region TrafficType
## 66         0.8   Feb                2       2      3           1
## 77         0.0   Feb                2       6      1           2
## 102        1.0   Feb                1       1      7           4
## 189        0.0   Mar                2       2      7           2
## 197        0.0   Mar                2       2      4           2
## 199        0.0   Mar                2       2      9           2
##           VisitorType Weekend Revenue
```

```
## 66  Returning_Visitor   FALSE    TRUE
## 77  Returning_Visitor   FALSE    TRUE
## 102 Returning_Visitor    TRUE    TRUE
## 189 Returning_Visitor   FALSE    TRUE
## 197       New_Visitor   FALSE    TRUE
## 199       New_Visitor   FALSE    TRUE
```

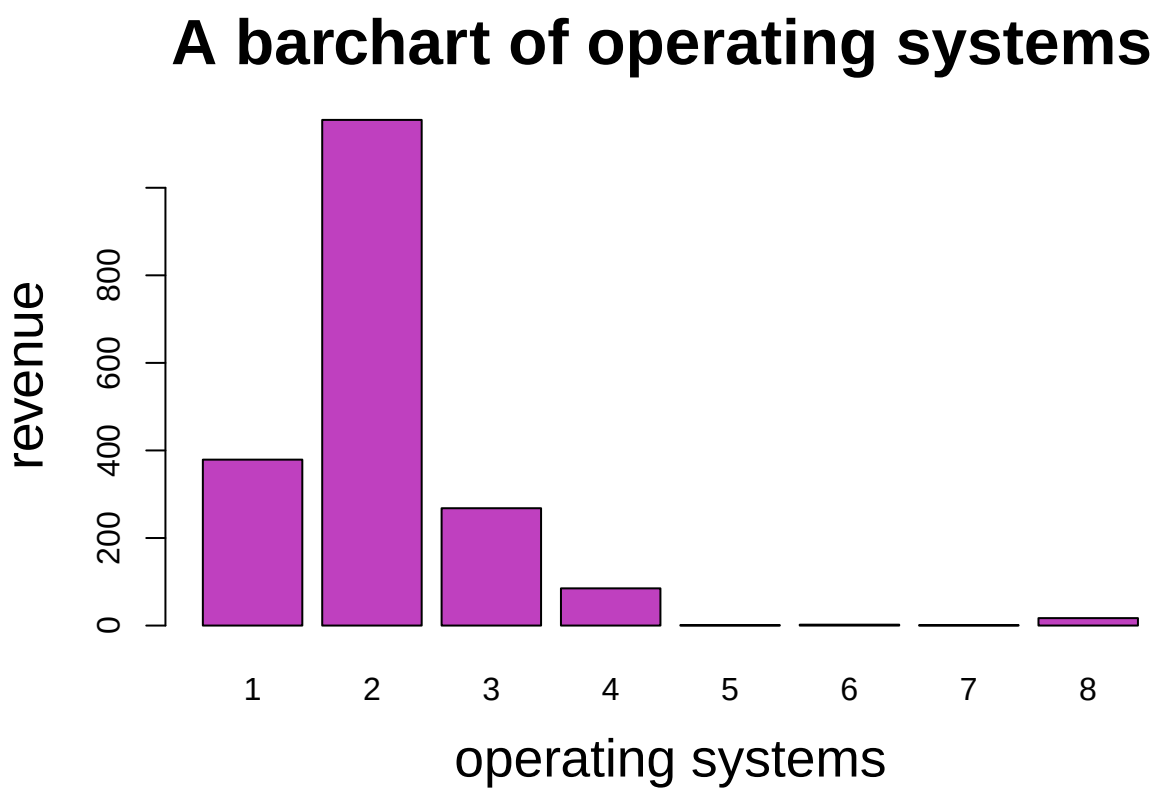
```
# finding out the dataframe with the revenue's dimentions
dim(revenue)
```

```
## [1] 1908   18
```

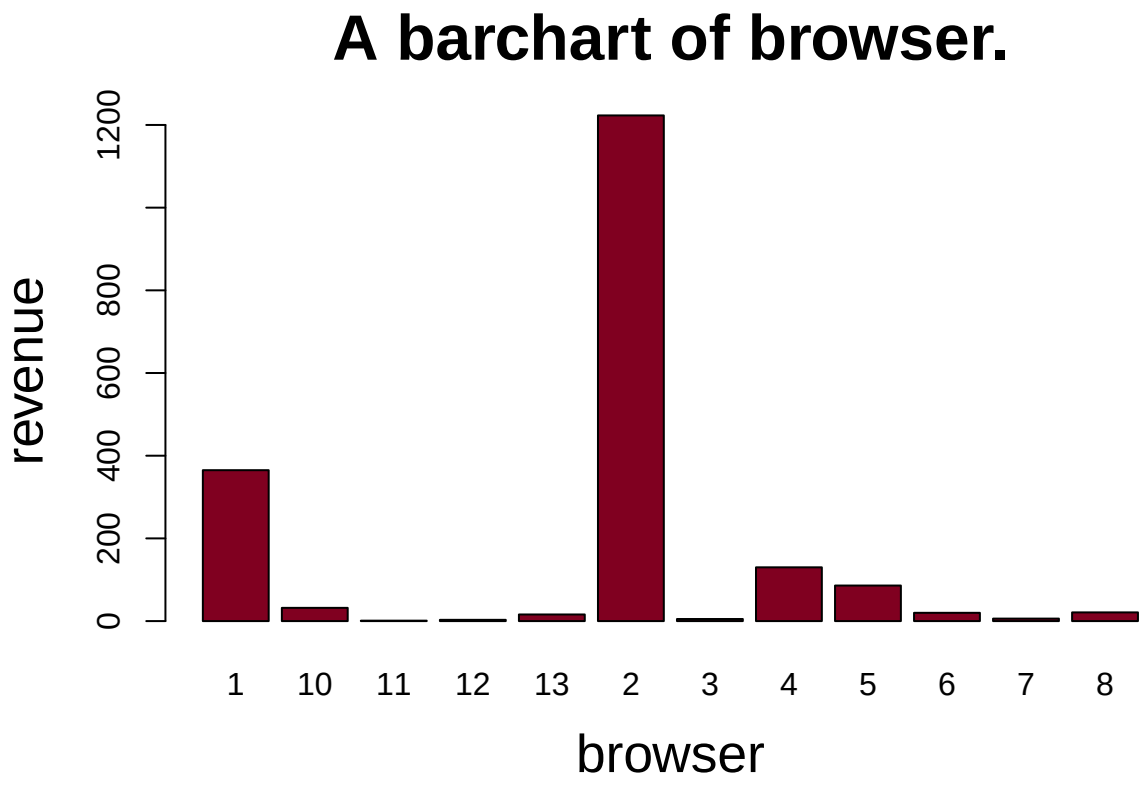
```
#comparison between month and revenue brought in
#frequency table of month
month1.freq <- table(revenue$Month)
#Bar chart to show frequency distribution of months
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(month1.freq), main="Count of revenue per month.",
        xlab="months",
        ylab="revenue",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#9F2B68"))
```



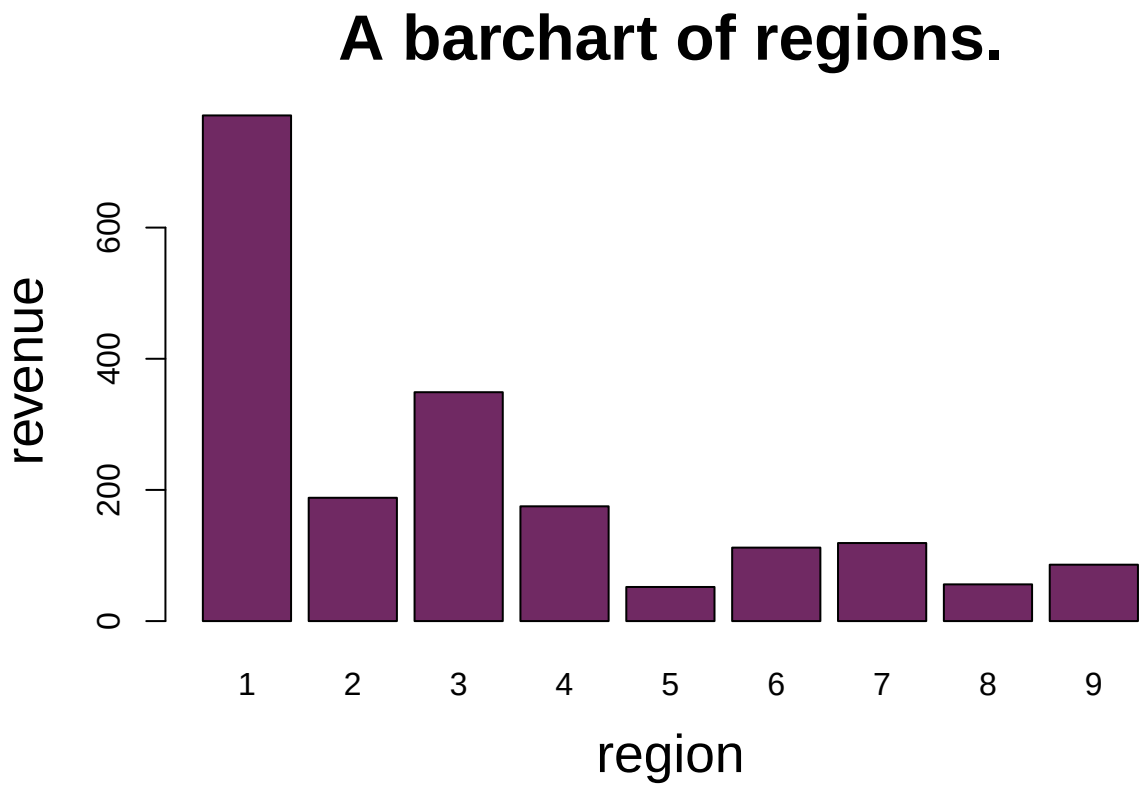
```r
#comparison between operating systems and revenue brought in
#frequency table of Operating systems
os1.freq <- table(revenue$OperatingSystems)
#Bar chart to show frequency distribution of operating systems
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(os1.freq), main="A barchart of operating systems.",
        xlab="operating systems",
        ylab="revenue",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#BF40BF"))
```



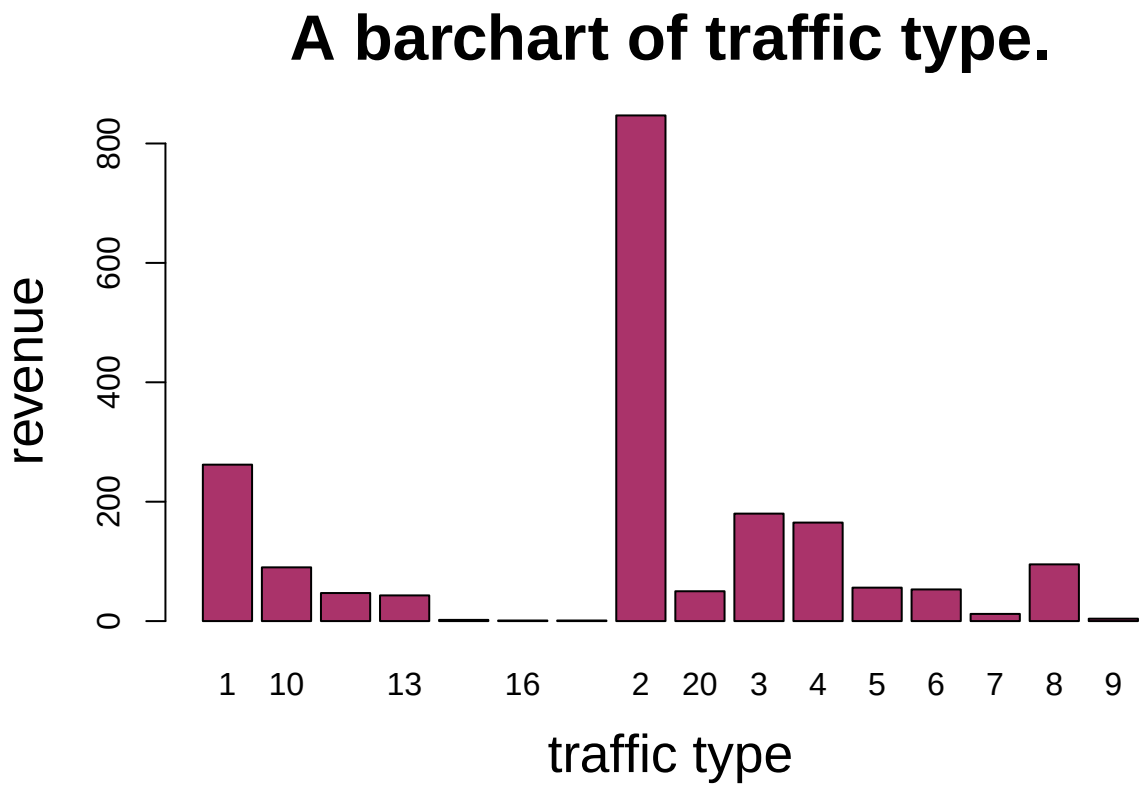

```r
#comparison between browser and revenue brought in
#frequency table of browser
browser1.freq <- table(revenue$Browser)
#Bar chart to show frequency distribution of browsers
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(browser1.freq), main="A barchart of browser.",
        xlab="browser",
        ylab="revenue",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#800020"))
```




```r
#comparison between region and revenue brought in
#frequency table of region
region1.freq <- table(revenue$Region)
#Bar chart to show frequency distribution of regions
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(region1.freq), main="A barchart of regions.",
        xlab="region",
        ylab="revenue",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#702963"))
```



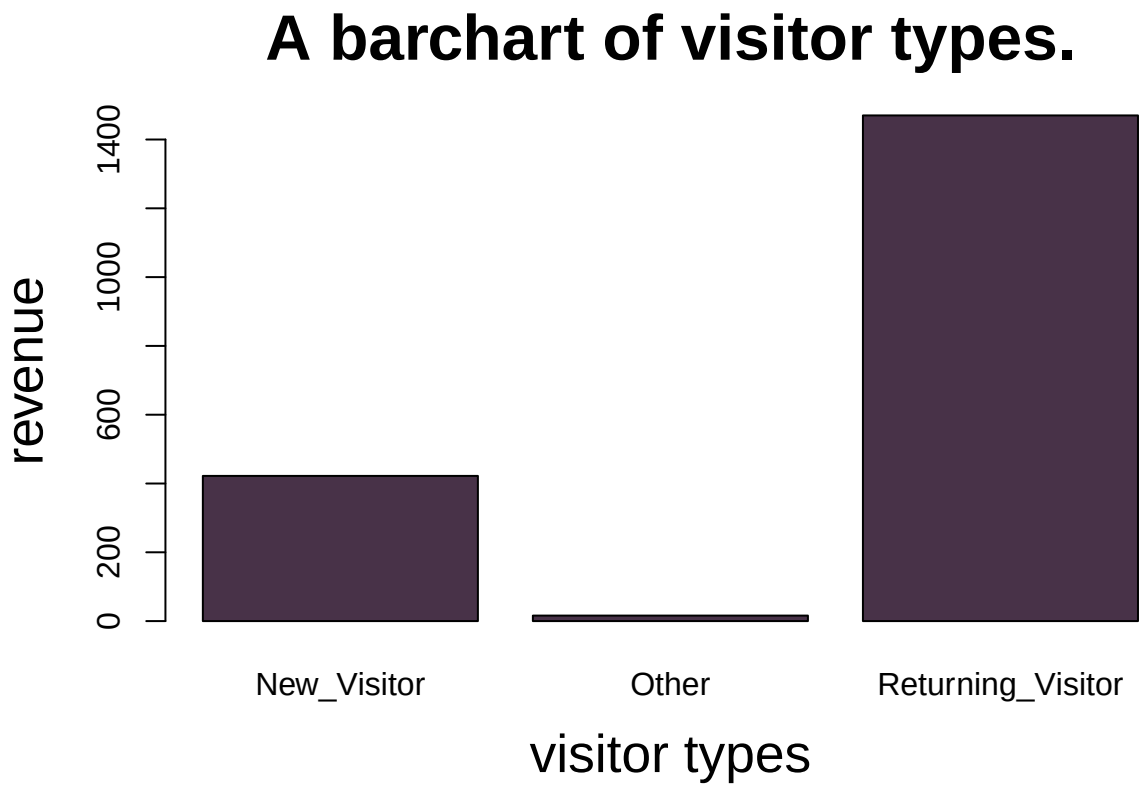

```
#comparison between traffic type and revenue brought in
#frequency table of traffic type
traffic1.freq <- table(revenue$TrafficType)
#Bar chart to show frequency distribution of traffic type
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(traffic1.freq), main="A barchart of traffic type.",
        xlab="traffic type",
        ylab="revenue",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#AA336A"))
```



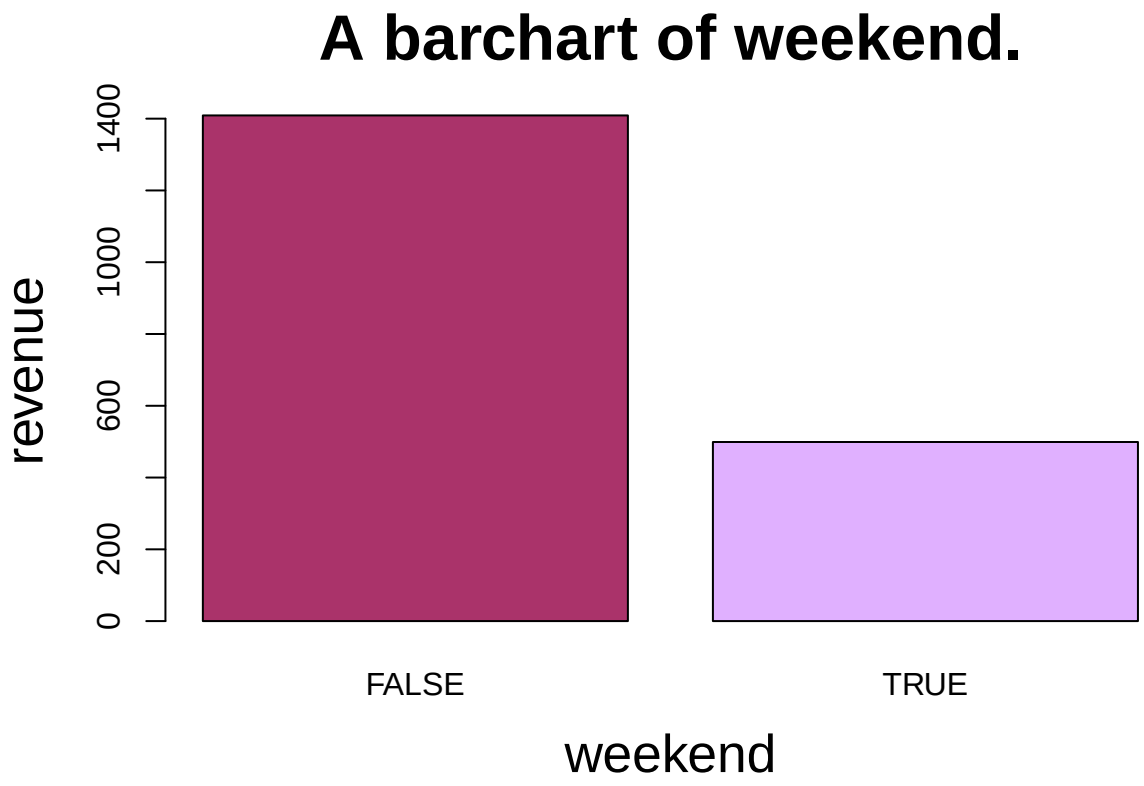

```
#plotting to see which type of visitors brought in revenue

revenue <- na.omit(revenue)
#creating a frequency table of visitor type
visitor1.freq <- table(revenue$VisitorType)
sort(visitor1.freq, decreasing = TRUE)[1:5]
```

```
##
## Returning_Visitor       New_Visitor             Other              <NA>
##              1470               422                16
##              <NA>
##
```

```
#Bar chart to show frequency distribution of visitor type
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(visitor1.freq), main="A barchart of visitor types.",
        xlab="visitor types",
        ylab="revenue",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#483248"))
```

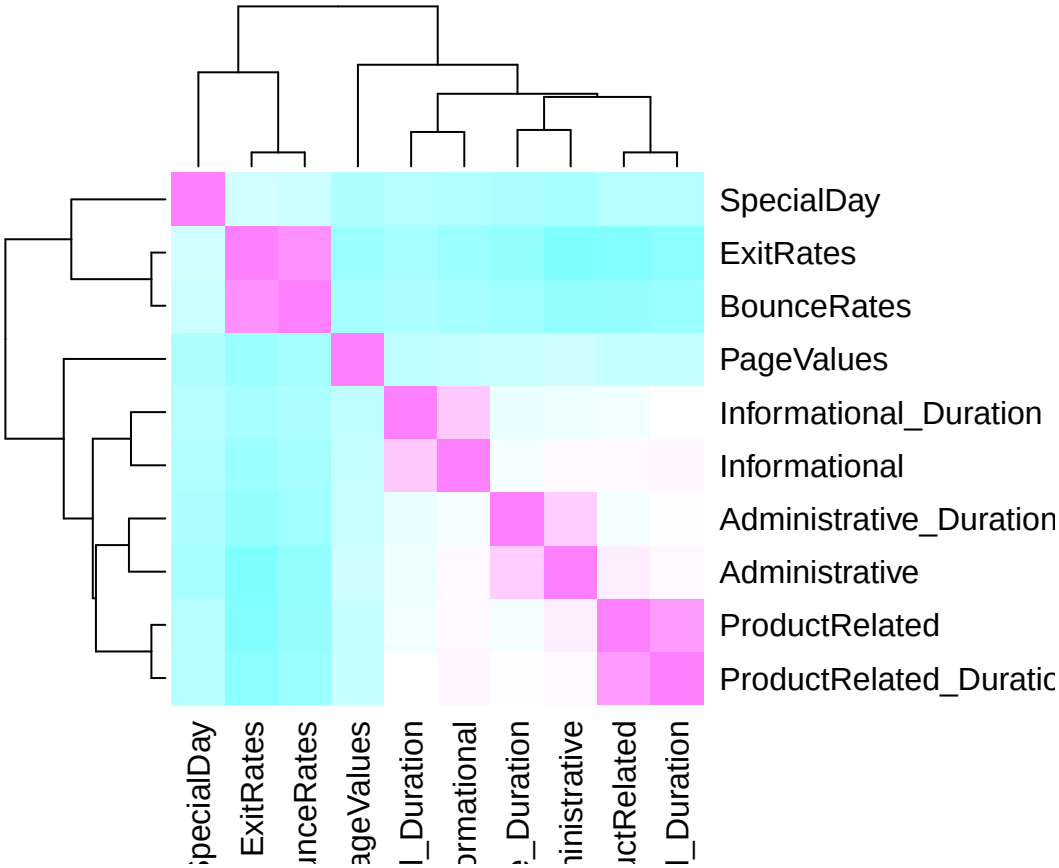



```
#comparison between weekend and revenue brought in
#frequency table of weekend
weekend1.freq <- table(revenue$Weekend)
#Bar chart to show frequency distribution of weekend
options(repr.plot.width = 10, repr.plot.height = 10)
barplot(c(weekend1.freq), main="A barchart of weekend.",
        xlab="weekend",
        ylab="revenue",
        cex.main=2, cex.lab=1.7,cex.sub=1.2,
        col=c("#AA336A", "#E0B0FF"))
```




Plotting a heat map using the correlation matrix

```
heatmap(corr, symm=TRUE, col = cm.colors(256))
```




**Observations from Bivariate Analysis** "' The number of product related websites visited is highly correlated to the product related duration spent. A great portion of revenue was gotten from region 1May had most web page visits and as expected most revenue was made in that month. Operating system type 2 and browser type 2 users brought in the most revenue.A larger amount of revenue was made during the weekdays. Exit rates and bounce rates are highly correlated.The traffic type that brought in most revenue was type 2 and returning visitors brought in the most followed by the new then other visitors.

# 7. Implementing the Solution

##a) Feature engineering

```
#removing the revenue column cause we dont need it when performing clustering
data3 <- data2[,1:17]
head(data3)
```

```
##   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
## 1 Returning_Visitor   FALSE
## 2 Returning_Visitor   FALSE
## 3 Returning_Visitor   FALSE
## 4 Returning_Visitor   FALSE
## 5 Returning_Visitor    TRUE
## 6 Returning_Visitor   FALSE
```

```
revenue3 <- data3$Revenue
# adjustng the DTs
data3[,12:15] <- sapply(data3[,12:15], as.character)
data3[,12:15] <- sapply(data3[,12:15], as.numeric)
head(data3)
```

```
##   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
## 1 Returning_Visitor   FALSE
## 2 Returning_Visitor   FALSE
## 3 Returning_Visitor   FALSE
## 4 Returning_Visitor   FALSE
## 5 Returning_Visitor    TRUE
## 6 Returning_Visitor   FALSE
```

```r
#encoding factor columns using the one hot encoder
library(caret)
dmy = dummyVars(" ~ .", data = data3)
data3.encod = data.frame(predict(dmy, newdata = data3))
```

```r
# checking its dimentions
dim(data3.encod)
```

```
## [1] 12199    29
```

```r
# checking the new df's structure
str(data3.encod)
```

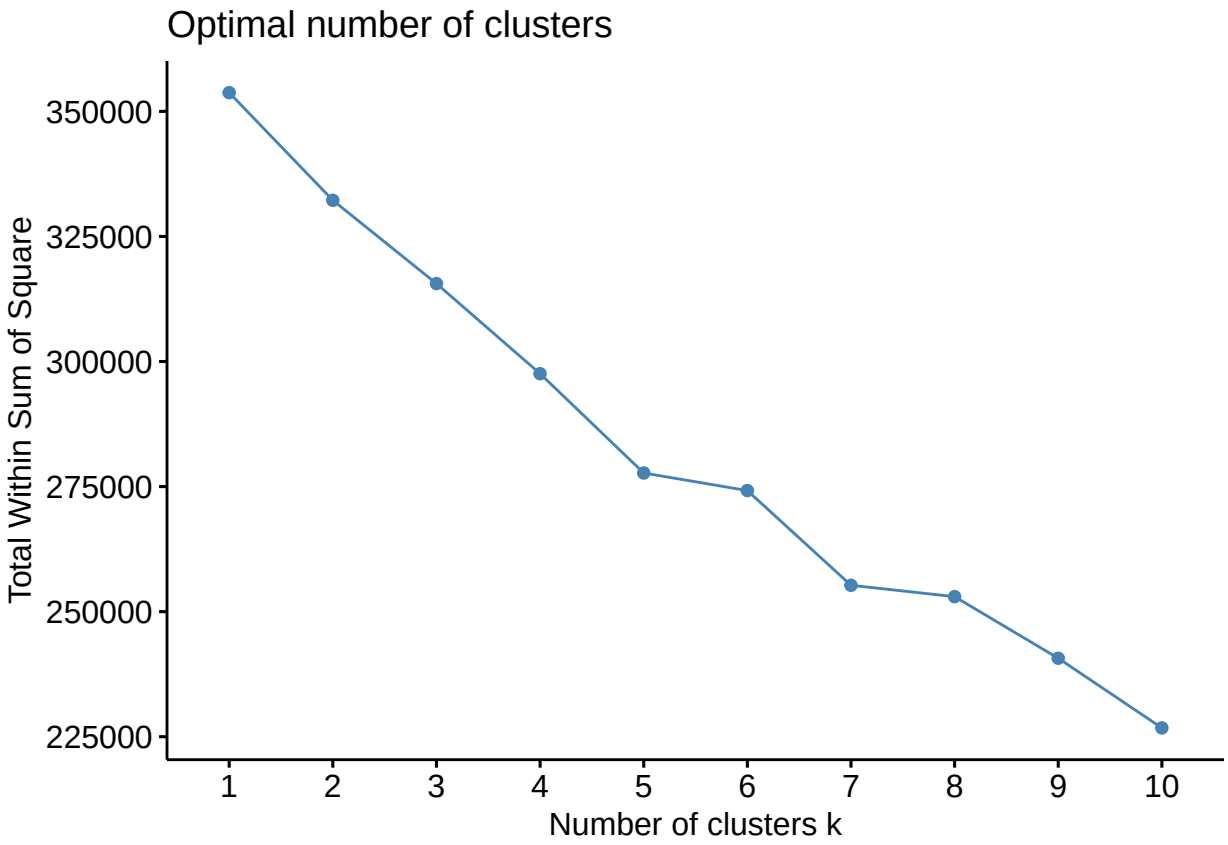
```
## 'data.frame':    12199 obs. of  29 variables:
##  $ Administrative              : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ Administrative_Duration     : num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ Informational               : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration      : num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ ProductRelated              : num  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 ...
##  $ MonthAug                    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MonthDec                    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MonthFeb                    : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ MonthJul                    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MonthJune                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MonthMar                    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MonthMay                    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MonthNov                    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MonthOct                    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MonthSep                    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ 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 ...
##  $ VisitorTypeNew_Visitor      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VisitorTypeOther            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VisitorTypeReturning_Visitor: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ WeekendFALSE                : num  1 1 1 1 0 1 1 0 1 1 ...
##  $ WeekendTRUE                 : num  0 0 0 0 1 0 0 1 0 0 ...
```

```r
#scaling the data
sc <- scale(data3.encod)
head(sc)
```

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

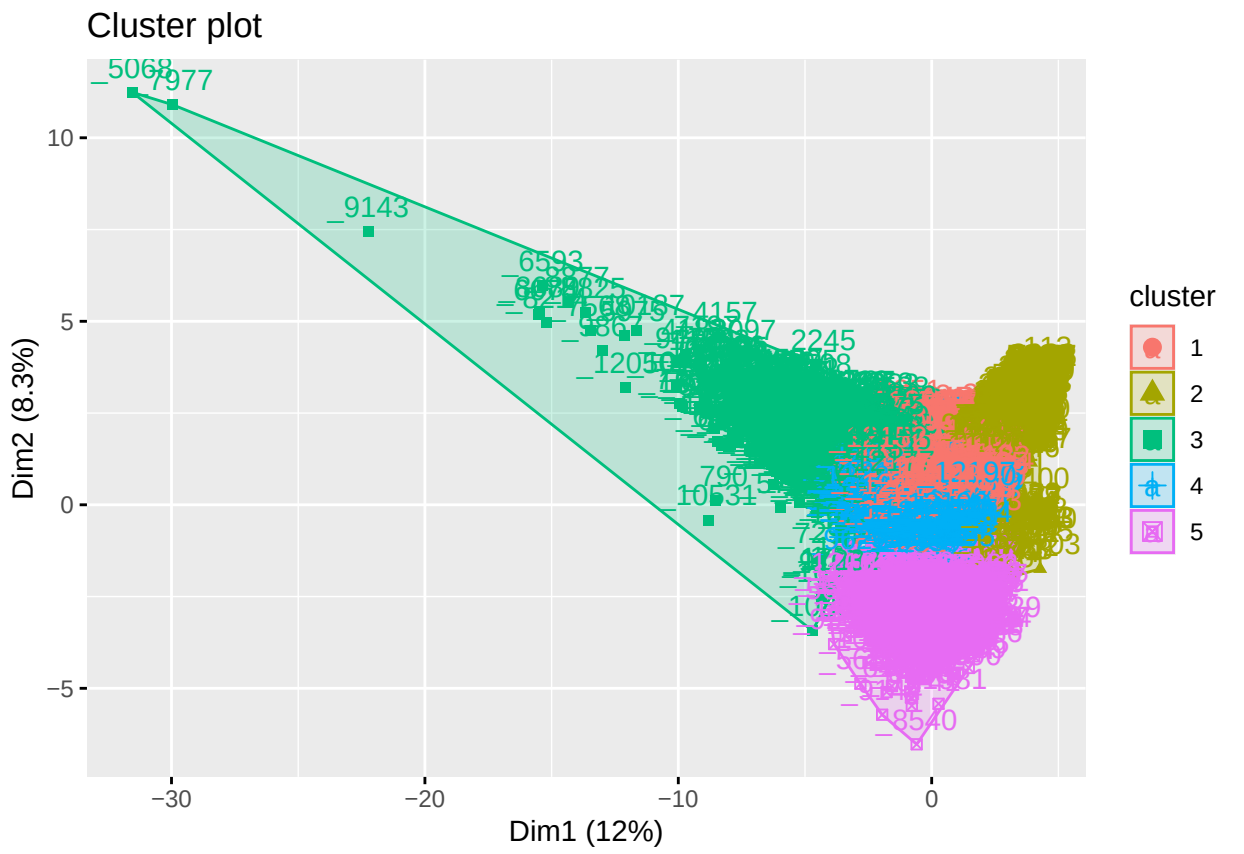
```
## 4     -0.7025315              -0.4601081    -0.3988128             -0.2462725
## 5     -0.7025315              -0.4601081    -0.3988128             -0.2462725
## 6     -0.7025315              -0.4601081    -0.3988128             -0.2462725
##   ProductRelated ProductRelated_Duration  BounceRates  ExitRates PageValues
## 1     -0.6963635              -0.6289343  3.954699721  3.4273070 -0.3190356
## 2     -0.6739424              -0.5955997 -0.450343788  1.2650121 -0.3190356
## 3     -0.6963635              -0.6294551  3.954699721  3.4273070 -0.3190356
## 4     -0.6739424              -0.6275453  0.650917089  2.1299300 -0.3190356
## 5     -0.4945739              -0.3020990 -0.009839437  0.1838646 -0.3190356
## 6     -0.2927843              -0.5486101 -0.102577188 -0.3661929 -0.3190356
##   SpecialDay   MonthAug   MonthDec MonthFeb   MonthJul  MonthJune   MonthMar
## 1 -0.3103105 -0.1918279 -0.4032013 8.125396 -0.1915981 -0.1546592 -0.4231883
## 2 -0.3103105 -0.1918279 -0.4032013 8.125396 -0.1915981 -0.1546592 -0.4231883
## 3 -0.3103105 -0.1918279 -0.4032013 8.125396 -0.1915981 -0.1546592 -0.4231883
## 4 -0.3103105 -0.1918279 -0.4032013 8.125396 -0.1915981 -0.1546592 -0.4231883
## 5 -0.3103105 -0.1918279 -0.4032013 8.125396 -0.1915981 -0.1546592 -0.4231883
## 6 -0.3103105 -0.1918279 -0.4032013 8.125396 -0.1915981 -0.1546592 -0.4231883
##     MonthMay   MonthNov   MonthOct   MonthSep OperatingSystems    Browser
## 1 -0.6124739 -0.5689022 -0.2170728 -0.1952467       -1.2396607 -0.7939682
## 2 -0.6124739 -0.5689022 -0.2170728 -0.1952467       -0.1371074 -0.2093703
## 3 -0.6124739 -0.5689022 -0.2170728 -0.1952467        2.0679992 -0.7939682
## 4 -0.6124739 -0.5689022 -0.2170728 -0.1952467        0.9654459 -0.2093703
## 5 -0.6124739 -0.5689022 -0.2170728 -0.1952467        0.9654459  0.3752276
## 6 -0.6124739 -0.5689022 -0.2170728 -0.1952467       -0.1371074 -0.2093703
##       Region TrafficType VisitorTypeNew_Visitor VisitorTypeOther
## 1 -0.8962939 -0.76562243             -0.4014135      -0.08175404
## 2 -0.8962939 -0.51660683             -0.4014135      -0.08175404
## 3  2.4336556 -0.26759123             -0.4014135      -0.08175404
## 4 -0.4800502 -0.01857564             -0.4014135      -0.08175404
## 5 -0.8962939 -0.01857564             -0.4014135      -0.08175404
## 6 -0.8962939 -0.26759123             -0.4014135      -0.08175404
##   VisitorTypeReturning_Visitor WeekendFALSE WeekendTRUE
## 1                    0.4124972    0.5528638  -0.5528638
## 2                    0.4124972    0.5528638  -0.5528638
## 3                    0.4124972    0.5528638  -0.5528638
## 4                    0.4124972    0.5528638  -0.5528638
## 5                    0.4124972   -1.8086156   1.8086156
## 6                    0.4124972    0.5528638  -0.5528638
```

```
#distance: computing the Euclidean distance between observations.
dst<-dist(sc)
# calculating how many clusters we'll need using or within sum squares
library(factoextra)
fviz_nbclust(sc, kmeans, method = "wss")+labs(subtite = "Elbow method")
```




```
# kmeans
km.out<- kmeans(sc, centers =  5, nstart=100)
```

```
#visualizualizing the clustering algorithm results
km.clusters<- km.out$cluster
rownames(sc)<-paste(data3$revenue, 1:dim(data3)[1], sep="_")
fviz_cluster(list(data=sc, cluster=km.clusters))
```



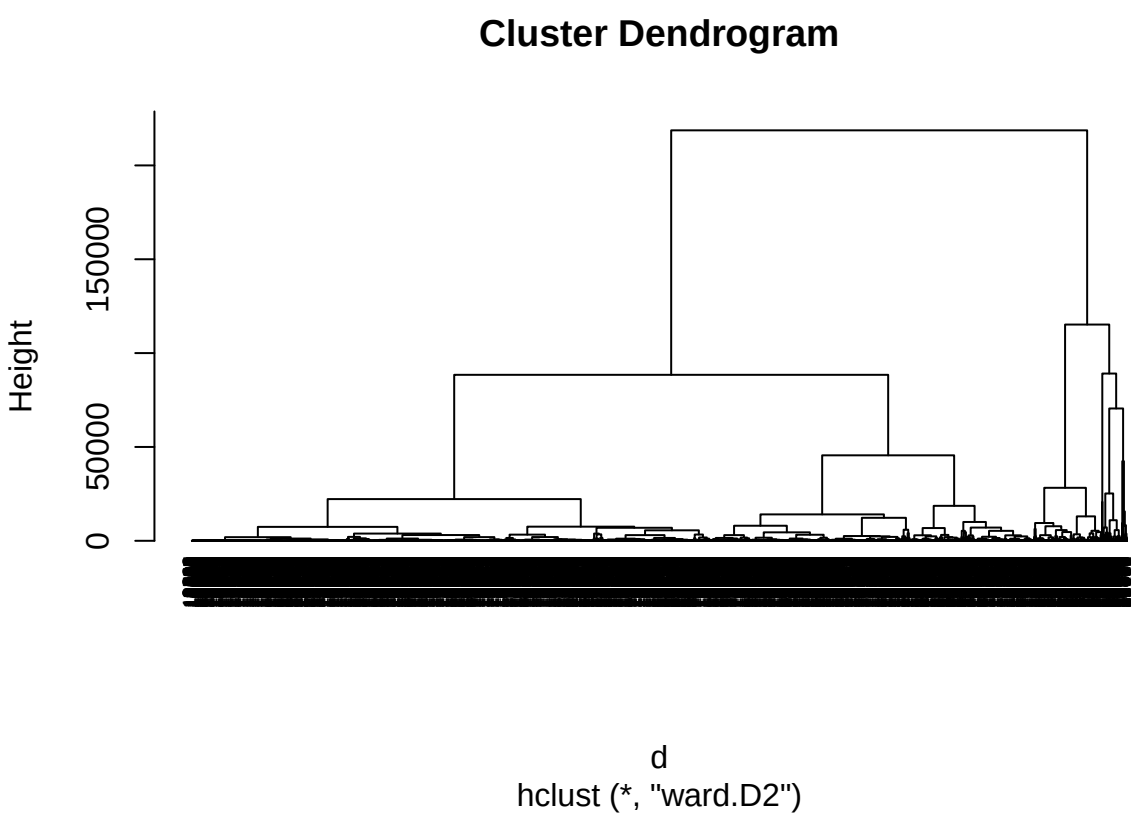

### Hierachical clustering

since the clustering visual on Kmeans clustering are crammed and hard to draw conclusions from, we'll use hierrarchical clustering

```
# First we use the dist() function to compute the Euclidean distance between observations,
# d will be the first argument in the hclust() function dissimilarity matrix
d <- dist(data3.encod, method = "euclidean")
# Using Ward's method to perform hierarchical clustering
hier <- hclust(d, method = "ward.D2" )
hier
```

```
##
## Call:
## hclust(d = d, method = "ward.D2")
##
## Cluster method   : ward.D2
## Distance         : euclidean
## Number of objects: 12199
```

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

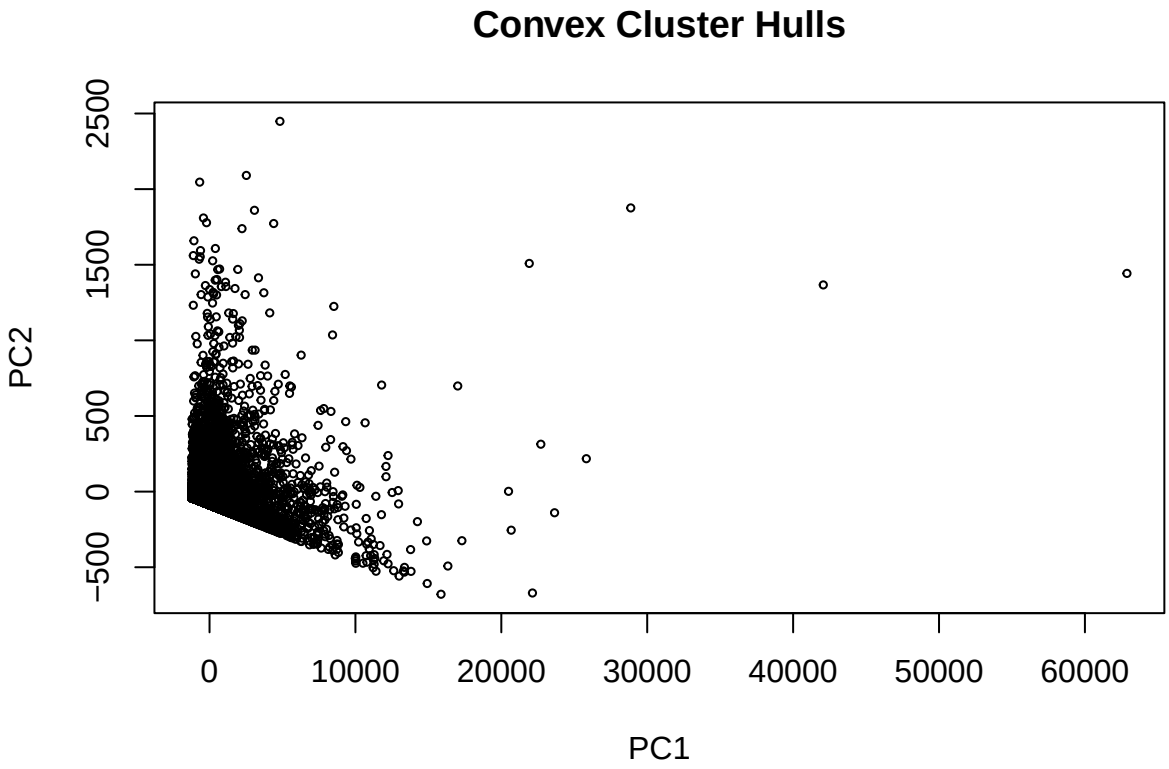



#8. Challenging the solution `We'll use dbscan to challange the solution and see if we can get better results`

```
library('dbscan')
# Applying our DBSCAN algorithm. We'll apply a minimum of 4 points with in a distance of eps(0.4)
db<-dbscan(data3.encod,eps=0.4, minPts = 4)
db
```

```
## DBSCAN clustering for 12199 objects.
## Parameters: eps = 0.4, minPts = 4
## The clustering contains 1 cluster(s) and 12195 noise points.
##
##     0     1
## 12195     4
##
## Available fields: cluster, eps, minPts
```

```
# plotting our clusters as shown
hullplot(data3.encod,db$cluster)
```




**Conclusion**

- Kmeans performed better than hierachical clustering in this study as it can handle larger datasets as compared to hierachical clustering and DBSCAN which performed poorest.

**Recommendations**

*The use of Kmeans clustering in identifying revenue generating customers since it has proven its ability to handle large datasets.

- The allocation of more resources towards marketing the brand on the weekends,focusing on region 1, to returning and new visitors.

##9. Follow up questions

###a) Did we have the right data? `Yes we did. Our data set had a good number of variables that helped us study the customers and preditc who was most likely to generate revenue. The data has certainly proven to be appropriate.` ###b) Do we need other data to answer our question? `No, although more data wouldn't hurt. Especially for exploratory analysis then we can gauge whether or not It would be fit to model with.` ###c) Did we have the right question? `We were able to answer the research question therefor the question was indeed correct.`