Kira Plastinina

2022-07-23

Problem Definition

Specifying the Data Analytic Question

As a Data Science Consultant, I have been employed by Kira Plastinina brand's Sales and Marketing team. They would like to understand their customer's behavior from data that they have collected over the past year. More specifically, they would like to learn the characteristics of customer groups.

Defining the Metric for Success

- Perform clustering stating insights drawn from your analysis and visualizations.
- Upon implementation, provide comparisons between K-Means clustering and Hierarchical clustering highlighting the strengths and limitations of each approach in the context of your analysis.

Understanding the context

There have been opened over 120 Kira Plastinina signature stores in different parts of the world: Russia, Ukraine, Italy, Great Britain, USA, China, Philippines, etc.. The stores present the collections of apparel, shoes, perfumes and accessories. In Russia, there have been opened 34 Kira Plastinina stores in Moscow, plus, the brand's network embraces 100 cities and towns of Russia.

Recording the Experimental Design

- Problem Definition
- Data Sourcing
- Check the Data
- Perform Data Cleaning
- Perform Exploratory Data Analysis (Univariate, Bivariate & Multivariate)
- Implement the Solution
- Challenge the Solution
- Follow up Questions

Data Relevance

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 Valentina'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.

Data Sourcing

```
#Installing packages
#
install.packages('tidyverse', repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)

## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages
```

```
install.packages('ggplot2', repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)
```

```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages
```

```
install.packages('caret', repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)
```

```
## package 'caret' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'caret'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copy
ing C:
## \Users\Lenovo\AppData\Local\R\win-library\4.2\00LOCK\caret\libs\x64\
caret.dll
## to C:\Users\Lenovo\AppData\Local\R\win-library\4.2\caret\libs\x64\ca
ret.dll:
## Permission denied
```

```
## Warning: restored 'caret'
```

```
##
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages
```

```
install.packages('caretEnsemble', repos = "http://cran.us.r-project.org
")
```

```
## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)
```

```
## package 'caretEnsemble' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages
```

```
install.packages('Amelia', repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)
```

```
## package 'Amelia' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages

install.packages('mice', repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)

## package 'mice' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages

install.packages('GGally', repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)

## package 'GGally' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages

install.packages('rpart', repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)

## package 'rpart' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'rpart'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copy
ing C:
## \Users\Lenovo\AppData\Local\R\win-library\4.2\00LOCK\rpart\libs\x64\
rpart.dll
## to C:\Users\Lenovo\AppData\Local\R\win-library\4.2\rpart\libs\x64\rp
art.dll:
## Permission denied

## Warning: restored 'rpart'

##
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages

install.packages("readr", repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)

## package 'readr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages

install.packages("dplyr", repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)

## package 'dplyr' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'dplyr'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copy
ing C:
## \Users\Lenovo\AppData\Local\R\win-library\4.2\00LOCK\dplyr\libs\x64\
dplyr.dll
## to C:\Users\Lenovo\AppData\Local\R\win-library\4.2\dplyr\libs\x64\dp
lyr.dll:
## Permission denied

## Warning: restored 'dplyr'

##
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages

install.packages("dendextend", repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)

## package 'dendextend' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages

install.packages("factoextra", repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)

## package 'factoextra' successfully unpacked and MD5 sums checked
##
```

```
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages

install.packages("cluster", repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)

## package 'cluster' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages

#Install respective libraries
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyve
rse 1.3.2 ──
## ✔ ggplot2 3.3.6     ✔ purrr   0.3.4
## ✔ tibble  3.1.7     ✔ dplyr   1.0.9
## ✔ tidyr   1.2.0     ✔ stringr 1.4.0
## ✔ readr   2.1.2     ✔ forcats 0.5.1
## ── Conflicts ────────────────────────────────────────── tidyverse_co
nflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()

library(ggplot2)
library(caret)

## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift

library(caretEnsemble)

##
## Attaching package: 'caretEnsemble'
##
## The following object is masked from 'package:ggplot2':
##
##     autoplot

library(Amelia)

## Loading required package: Rcpp
## ##
```

```
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.0, built: 2021-05-26)
## ## Copyright (C) 2005-2022 James Honaker, Gary King and Matthew Blac
kwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##

library(mice)

##
## Attaching package: 'mice'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     cbind, rbind

library(GGally)

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

library(rpart)
library(readr)
library(dplyr)
library(dendextend)

##
## ---------------------
## Welcome to dendextend version 1.16.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignett
e.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/
talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tag
s:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(libra
ry(dendextend))
## ---------------------
##
##
```

```
## Attaching package: 'dendextend'
##
## The following object is masked from 'package:rpart':
##
##     prune
##
## The following object is masked from 'package:stats':
##
##     cutree

library(factoextra)

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

library(cluster)

#Read the dataset
shoppers <- read_csv("C://Users//Lenovo//Downloads//DB_prep//online_sho
ppers_intention.csv")

## Rows: 12330 Columns: 18
## ── Column specification ────────────────────────────────────────────
──────────
## Delimiter: ","
## chr  (2): Month, VisitorType
## dbl (14): Administrative, Administrative_Duration, Informational, In
formatio...
## lgl  (2): Weekend, Revenue
##
## ℹ Use `spec()` to retrieve the full column specification for this d
ata.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet
 this message.
```

Check the Data

```
#Find the column names
colnames(shoppers)

##  [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"
```

```
#Checking the head and tail of the data

head(shoppers)
```

```
## # A tibble: 6 × 18
##   Administrative Administrative_D… Informational Informational_D… Pr
oductRelated
##            <dbl>             <dbl>         <dbl>            <dbl>
      <dbl>
## 1              0                 0             0                0
          1
## 2              0                 0             0                0
          2
## 3              0                -1             0               -1
          1
## 4              0                 0             0                0
          2
## 5              0                 0             0                0
         10
## 6              0                 0             0                0
         19
## # … with 13 more variables: ProductRelated_Duration <dbl>, BounceRat
es <dbl>,
## #   ExitRates <dbl>, PageValues <dbl>, SpecialDay <dbl>, Month <chr>,
## #   OperatingSystems <dbl>, Browser <dbl>, Region <dbl>, TrafficType
 <dbl>,
## #   VisitorType <chr>, Weekend <lgl>, Revenue <lgl>
```

```
tail(shoppers)
```

```
## # A tibble: 6 × 18
##   Administrative Administrative_D… Informational Informational_D… Pr
oductRelated
##            <dbl>             <dbl>         <dbl>            <dbl>
      <dbl>
## 1              0                 0             1                0
         16
## 2              3               145             0                0
         53
## 3              0                 0             0                0
          5
## 4              0                 0             0                0
          6
## 5              4                75             0                0
         15
## 6              0                 0             0                0
          3
## # … with 13 more variables: ProductRelated_Duration <dbl>, BounceRat
es <dbl>,
## #   ExitRates <dbl>, PageValues <dbl>, SpecialDay <dbl>, Month <chr>,
```

```
## #   OperatingSystems <dbl>, Browser <dbl>, Region <dbl>, TrafficType
 <dbl>,
## #   VisitorType <chr>, Weekend <lgl>, Revenue <lgl>

#The rows and columns in the data
cat("The dataset has ", dim(shoppers)[1], "rows and ", dim(shoppers)[2],
 " columns")

## The dataset has  12330 rows and  18  columns

# Seeing the structure of the dataset
str(shoppers)

## spec_tbl_df [12,330 × 18] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Administrative         : num [1:12330] 0 0 0 0 0 0 0 1 0 0 ...
##  $ Administrative_Duration: num [1:12330] 0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ Informational          : num [1:12330] 0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration : num [1:12330] 0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ ProductRelated         : num [1:12330] 1 2 1 2 10 19 1 1 2 3 ...
##  $ ProductRelated_Duration: num [1:12330] 0 64 -1 2.67 627.5 ...
##  $ BounceRates            : num [1:12330] 0.2 0 0.2 0.05 0.02 ...
##  $ ExitRates              : num [1:12330] 0.2 0.1 0.2 0.14 0.05 ...
##  $ PageValues             : num [1:12330] 0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialDay             : num [1:12330] 0 0 0 0 0 0 0.4 0 0.8 0.4
 ...
##  $ Month                  : chr [1:12330] "Feb" "Feb" "Feb" "Feb" ...
##  $ OperatingSystems       : num [1:12330] 1 2 4 3 3 2 2 1 2 2 ...
##  $ Browser                : num [1:12330] 1 2 1 2 3 2 4 2 2 4 ...
##  $ Region                 : num [1:12330] 1 1 9 2 1 1 3 1 2 1 ...
##  $ TrafficType            : num [1:12330] 1 2 3 4 4 3 3 5 3 2 ...
##  $ VisitorType            : chr [1:12330] "Returning_Visitor" "Retur
ning_Visitor" "Returning_Visitor" "Returning_Visitor" "Returning_Visitor" ...
##  $ Weekend                : logi [1:12330] FALSE FALSE FALSE FALSE T
RUE FALSE ...
##  $ Revenue                : logi [1:12330] FALSE FALSE FALSE FALSE F
ALSE FALSE ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Administrative = col_double(),
##   ..   Administrative_Duration = col_double(),
##   ..   Informational = col_double(),
##   ..   Informational_Duration = col_double(),
##   ..   ProductRelated = col_double(),
##   ..   ProductRelated_Duration = col_double(),
##   ..   BounceRates = col_double(),
##   ..   ExitRates = col_double(),
##   ..   PageValues = col_double(),
##   ..   SpecialDay = col_double(),
##   ..   Month = col_character(),
##   ..   OperatingSystems = col_double(),
##   ..   Browser = col_double(),
```

```
##   ..   Region = col_double(),
##   ..   TrafficType = col_double(),
##   ..   VisitorType = col_character(),
##   ..   Weekend = col_logical(),
##   ..   Revenue = col_logical()
##   .. )
##  - attr(*, "problems")=<externalptr>

#checking the datatypes on the columns
sapply(shoppers, class)

##          Administrative Administrative_Duration           Informatio
nal
##               "numeric"              "numeric"               "numer
ic"
##  Informational_Duration          ProductRelated ProductRelated_Durat
ion
##               "numeric"              "numeric"               "numer
ic"
##             BounceRates              ExitRates              PageVal
ues
##               "numeric"              "numeric"               "numer
ic"
##              SpecialDay                  Month        OperatingSyst
ems
##               "numeric"            "character"               "numer
ic"
##                 Browser                 Region             TrafficT
ype
##               "numeric"              "numeric"               "numer
ic"
##             VisitorType                Weekend                 Reve
nue
##             "character"              "logical"               "logic
al"

#summary of the dataset
#Basic descriptive statistics and frequencies.

summary(shoppers)

##  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.   :0.00
000
##  1st Qu.:0.000000   1st Qu.:0.01429   1st Qu.:  0.000   1st Qu.:0.00
000
##  Median :0.003119   Median :0.02512   Median :  0.000   Median :0.00
000
##  Mean   :0.022152   Mean   :0.04300   Mean   :  5.889   Mean   :0.06
143
##  3rd Qu.:0.016684   3rd Qu.:0.05000   3rd Qu.:  0.000   3rd Qu.:0.00
000
##  Max.   :0.200000   Max.   :0.20000   Max.   :361.764   Max.   :1.00
000
##  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
##
```

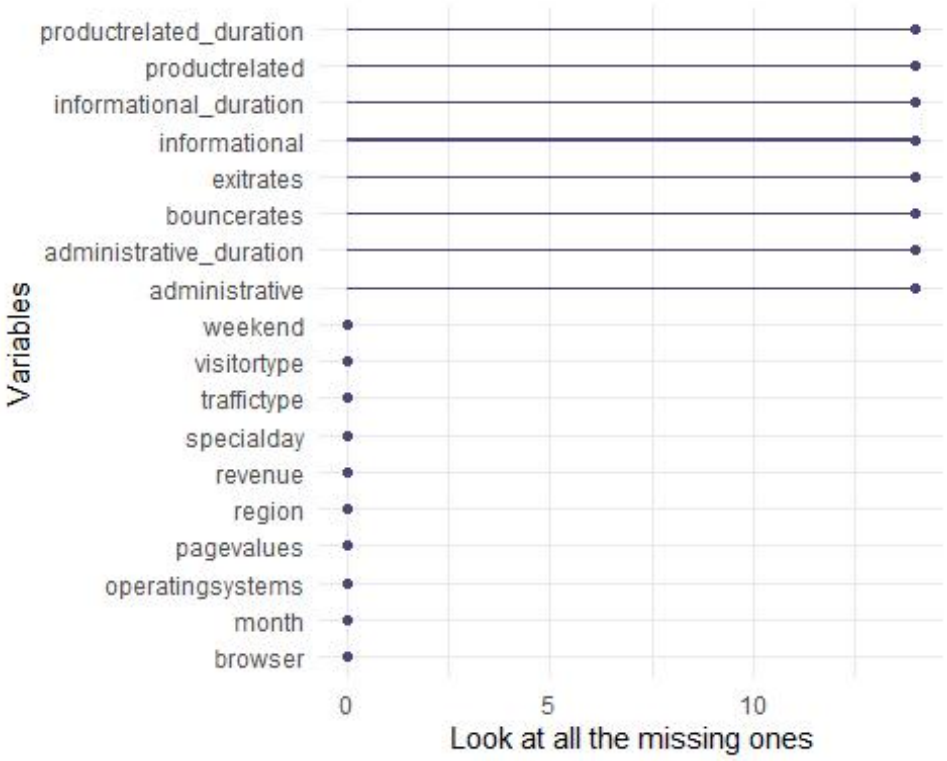
```
# list the levels for the class
levels(shoppers$revenue)
```

```
## Warning: Unknown or uninitialised column: `revenue`.

## NULL
```

Perform Data Cleaning

```
names(shoppers)<- tolower(names(shoppers)) # make the column names to l
owercase

# Checking the number of missing per column/variable
colSums(is.na(shoppers))

##          administrative administrative_duration           informatio
nal
##                      14                      14
 14
##  informational_duration         productrelated productrelated_durat
ion
##                      14                      14
 14
##             bouncerates               exitrates              pageval
ues
##                      14                      14
  0
##              specialday                   month        operatingsyst
ems
##                       0                       0
  0
##                 browser                  region             traffict
ype
##                       0                       0
  0
##             visitortype                 weekend                  reve
nue
##                       0                       0
  0

#Using visualizations to see missing data
install.packages("naniar", repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/Lenovo/AppData/Local/R/win-library
/4.2'
## (as 'lib' is unspecified)

## package 'naniar' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Lenovo\AppData\Local\Temp\RtmpsTjWrl\downloaded_packages

library(naniar)

gg_miss_var(shoppers) + labs(y = "Look at all the missing ones")
```

```
## Warning: It is deprecated to specify `guide = FALSE` to remove a gui
de. Please
## use `guide = "none"` instead.
```



Dealing with missing values

```
#Omit missing values
shoppers <- na.omit(shoppers)

# Checking the number of missing per column/variable again
colSums(is.na(shoppers))
```

```
##          administrative administrative_duration           informatio
nal
##                       0                       0
  0
##  informational_duration         productrelated productrelated_durat
ion
##                       0                       0
  0
##             bouncerates               exitrates              pageval
ues
##                       0                       0
  0
##              specialday                   month        operatingsyst
ems
##                       0                       0
  0
```

```
##                 browser                  region             traffict
ype
##                       0                       0
  0
##             visitortype                 weekend                 reve
nue
##                       0                       0
  0
```

```
#Checking for duplicates in data

duplicated_rows <- shoppers[duplicated(shoppers),]
duplicated_rows
```

```
## # A tibble: 117 × 18
##    administrative administrative_… informational informational_d… pr
oductrelated
##             <dbl>            <dbl>         <dbl>            <dbl>
       <dbl>
##  1              0                0             0                0
           1
##  2              0                0             0                0
           1
##  3              0                0             0                0
           1
##  4              0                0             0                0
           1
##  5              0                0             0                0
           1
##  6              0                0             0                0
           1
##  7              0                0             0                0
           1
##  8              0                0             0                0
           1
##  9              0                0             0                0
           2
## 10              0                0             0                0
           1
## # … with 107 more rows, and 13 more variables: productrelated_durati
on <dbl>,
## #   bouncerates <dbl>, exitrates <dbl>, pagevalues <dbl>, specialday
 <dbl>,
## #   month <chr>, operatingsystems <dbl>, browser <dbl>, region <dbl>,
## #   traffictype <dbl>, visitortype <chr>, weekend <lgl>, revenue <lg
l>
```

```
#Dealing with duplicates
shoppers<- shoppers[!duplicated(shoppers), ]
```

```
duplicated_rows <- shoppers[duplicated(shoppers),]
duplicated_rows
```

```
## # A tibble: 0 × 18
## # … with 18 variables: administrative <dbl>, administrative_duration
 <dbl>,
## #   informational <dbl>, informational_duration <dbl>, productrelate
d <dbl>,
## #   productrelated_duration <dbl>, bouncerates <dbl>, exitrates <db
l>,
## #   pagevalues <dbl>, specialday <dbl>, month <chr>, operatingsystem
s <dbl>,
## #   browser <dbl>, region <dbl>, traffictype <dbl>, visitortype <ch
r>,
## #   weekend <lgl>, revenue <lgl>
```

```
dim(shoppers)
```

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

Checking for outliers

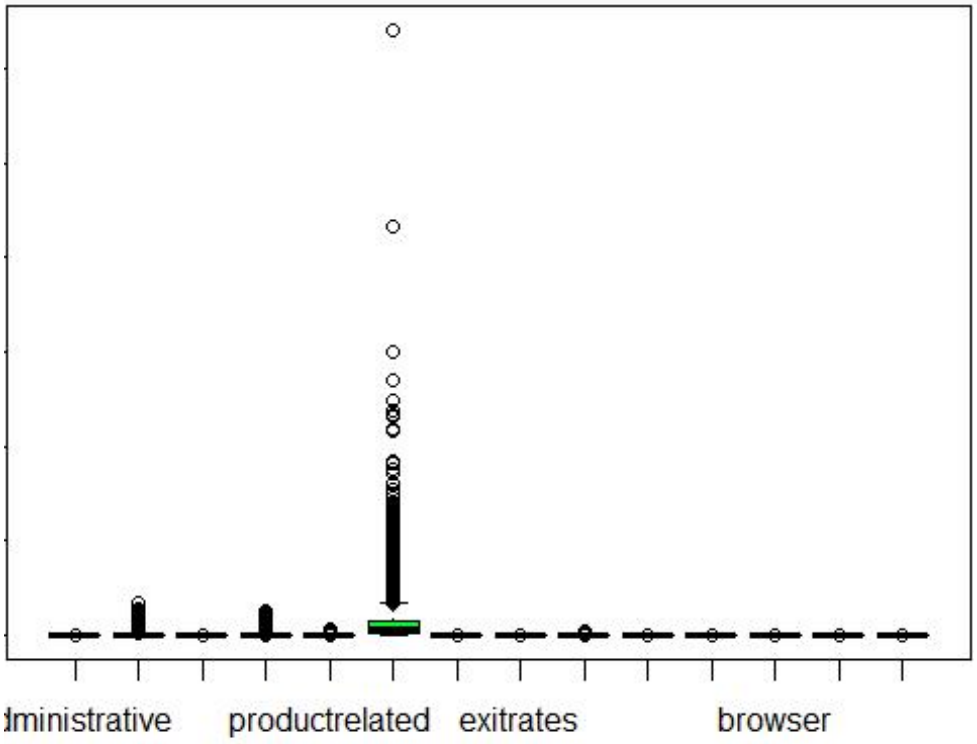
```
# Creating a Subset
df <-subset(shoppers, select = -c(month, visitortype, weekend, revenue))

print("Modified Data Frame")
```

```
## [1] "Modified Data Frame"
```

```
head(df)
```

```
## # A tibble: 6 × 14
##   administrative administrative_d… informational informational_d… pr
oductrelated
##            <dbl>             <dbl>         <dbl>            <dbl>
       <dbl>
## 1              0                 0             0                0
           1
## 2              0                 0             0                0
           2
## 3              0                -1             0               -1
           1
## 4              0                 0             0                0
           2
## 5              0                 0             0                0
          10
## 6              0                 0             0                0
          19
## # … with 9 more variables: productrelated_duration <dbl>, bouncerate
s <dbl>,
## #   exitrates <dbl>, pagevalues <dbl>, specialday <dbl>,
```

```
## #   operatingsystems <dbl>, browser <dbl>, region <dbl>, traffictype
 <dbl>

#Visualize the outliers

# Margins area
par(oma=c(3,0,0,0)) # all sides have 3 lines of space
par(mar=c(0,0,0,0) + 0.1)

fig <- boxplot(df, col = rainbow(ncol(df)))
```




```
# Listing the outliers
OutVals = boxplot(df, plot=FALSE, col = ncol(df))$out
OutVals

##     [1]  1.200000e+01  1.600000e+01  1.300000e+01  1.100000e+01  1.3
00000e+01
##     [6]  1.800000e+01  1.400000e+01  1.200000e+01  1.200000e+01  1.6
00000e+01
##    [11]  1.300000e+01  1.700000e+01  1.900000e+01  1.500000e+01  1.1
00000e+01
##    [16]  1.100000e+01  1.200000e+01  1.300000e+01  1.100000e+01  1.4
00000e+01
##    [21]  1.100000e+01  1.500000e+01  1.100000e+01  1.100000e+01  1.1
00000e+01
##    [26]  1.200000e+01  1.100000e+01  2.400000e+01  1.100000e+01  1.1
00000e+01
```

```
e+01  1.300000e+01
## [22286]  1.300000e+01  1.000000e+01  8.000000e+00  2.000000e+01  1.0
00000e+01
## [22291]  2.000000e+01  1.000000e+01  1.000000e+01  1.000000e+01  1.0
00000e+01
## [22296]  8.000000e+00  8.000000e+00  1.300000e+01  1.300000e+01  8.0
00000e+00
## [22301]  1.300000e+01  1.000000e+01  8.000000e+00  2.000000e+01  8.0
00000e+00
## [22306]  1.300000e+01  8.000000e+00  1.000000e+01  1.300000e+01  8.0
00000e+00
## [22311]  1.000000e+01  2.000000e+01  1.300000e+01  8.000000e+00  1.0
00000e+01
## [22316]  1.000000e+01  1.000000e+01  1.300000e+01  8.000000e+00  1.1
00000e+01
## [22321]  1.300000e+01  1.000000e+01  2.000000e+01  1.100000e+01  1.0
00000e+01
## [22326]  1.000000e+01  8.000000e+00  1.300000e+01  1.100000e+01
```

```
#Removing outliers from each column

shoppers1 = shoppers
shoppers1<- shoppers[-which(shoppers$administrative %in% OutVals ),]
shoppers1<- shoppers[-which(shoppers$administrative_duration %in% OutVa
ls ),]
shoppers1<- shoppers[-which(shoppers$informational%in% OutVals ),]
shoppers1<- shoppers[-which(shoppers$informational_duration %in% OutVal
s ),]
shoppers1<- shoppers[-which(shoppers$productrelated %in% OutVals ),]
shoppers1<- shoppers[-which(shoppers$productrelated_duration %in% OutVa
ls),]

#check the difference after removing outliers
print(dim(shoppers))

## [1] 12199    18

print(dim(shoppers1))

## [1] 8956   18
```

Perform Exploratory Data Analysis

Univariate Analysis

```
# Creating a Subset
df1 <-subset(shoppers1, select = -c(month, visitortype, weekend, revenu
e))
print("Modified Data Frame")

## [1] "Modified Data Frame"
```

```
head(df1)
```

```
## # A tibble: 6 × 14
##   administrative administrative_d… informational informational_d… pr
oductrelated
##            <dbl>             <dbl>         <dbl>            <dbl>
       <dbl>
## 1              0                 0             0                0
           1
## 2              0                 0             0                0
           2
## 3              0                 0             0                0
          10
## 4              0                 0             0                0
          19
## 5              0                 0             0                0
          16
## 6              2                53             0                0
          23
## # … with 9 more variables: productrelated_duration <dbl>, bouncerate
s <dbl>,
## #   exitrates <dbl>, pagevalues <dbl>, specialday <dbl>,
## #   operatingsystems <dbl>, browser <dbl>, region <dbl>, traffictype
 <dbl>
```

```
#Descriptive statistics for each column

desc_stats <- data.frame(
  Min = apply(df1, 2, min),    # minimum
  Max = apply(df1, 2, max),     # Maximum
  Med = apply(df1, 2, median), # median
  Mean = apply(df1, 2, mean),  # mean
  variance= apply(df1, 2, var),  #Variance
  SD = apply(df1, 2, sd)   # Standard deviation
  )

desc_stats <- round(desc_stats, 1)
desc_stats
```

```
##                          Min    Max   Med  Mean variance    SD
## administrative             0   24.0   1.0   2.4     10.1   3.2
## administrative_duration    0 2156.2  18.0  82.4  27089.0 164.6
## informational              0   16.0   0.0   0.5      1.4   1.2
## informational_duration     0 2252.0   0.0  30.4  15301.2 123.7
## productrelated             0  223.0  21.0  27.6    589.3  24.3
## productrelated_duration    0 3401.3 735.9 961.0 663437.4 814.5
## bouncerates                0    0.2   0.0   0.0      0.0   0.0
## exitrates                  0    0.2   0.0   0.0      0.0   0.0
## pagevalues                 0  361.8   0.0   6.9    407.4  20.2
## specialday                 0    1.0   0.0   0.1      0.0   0.2
## operatingsystems           1    8.0   2.0   2.1      0.9   0.9
```

```
## browser                  1   13.0   2.0   2.3      2.9   1.7
## region                   1    9.0   3.0   3.2      5.8   2.4
## traffictype              1   20.0   2.0   4.1     16.1   4.0
```

Mode

```
#Get the mode

getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}


attach(shoppers1)
 print("The mode of the administrative column is " )

## [1] "The mode of the administrative column is "

getmode(administrative);

## [1] 0

 print("The mode of the administrative_duration column is " )

## [1] "The mode of the administrative_duration column is "

 getmode(administrative_duration);

## [1] 0

print("The mode of the informational column is " )

## [1] "The mode of the informational column is "

getmode(informational);

## [1] 0

  print("The mode of the informational_duration column is " )

## [1] "The mode of the informational_duration column is "

 getmode(informational_duration);

## [1] 0

print("The mode of the productrelated column is " )

## [1] "The mode of the productrelated column is "

 getmode(productrelated);

## [1] 1
```

```
print("The mode of the productrelated_duration column is " )
## [1] "The mode of the productrelated_duration column is "
 getmode(productrelated_duration);
## [1] 0
print("The mode of the bouncerates column is " )
## [1] "The mode of the bouncerates column is "
 getmode(bouncerates);
## [1] 0
 print("The mode of the exitrates column is " )
## [1] "The mode of the exitrates column is "
  getmode(exitrates);
## [1] 0.2
 print("The mode of the pagevalues column is " )
## [1] "The mode of the pagevalues column is "
  getmode(pagevalues)
## [1] 0
   print("The mode of the specialday column is " )
## [1] "The mode of the specialday column is "
  getmode(specialday)
## [1] 0
 print("The mode of the operatingsystems column is " )
## [1] "The mode of the operatingsystems column is "
  getmode(operatingsystems)
## [1] 2
 print("The mode of the browser column is " )
## [1] "The mode of the browser column is "
  getmode(browser)
## [1] 2
```

```
 print("The mode of the region column is " )
```

```
## [1] "The mode of the region column is "
```

```
  getmode(region)
```

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

```
 print("The mode of the traffictype column is " )
```

```
## [1] "The mode of the traffictype column is "
```

```
  getmode(traffictype)
```

```
## [1] 2
```

```
#check the quantiles  values of every column
 Quantiles = apply(df1, 2, quantile)     # Quantile
Quantiles
```

```
##      administrative administrative_duration informational
## 0%                0                   0.000             0
## 25%               0                   0.000             0
## 50%               1                  18.000             0
## 75%               4                 100.150             0
## 100%             24                2156.167            16
##      informational_duration productrelated productrelated_duration b
ouncerates
## 0%                    0.000              0                  0.0000 0.
000000000
## 25%                   0.000             11                318.6958 0.
000000000
## 50%                   0.000             21                735.9083 0.
003846154
## 75%                   0.000             37               1414.7888 0.
016666667
## 100%               2252.033            223               3401.3000 0.
200000000
##       exitrates pagevalues specialday operatingsystems browser regio
n
## 0%   0.00000000     0.0000          0                1       1
1
## 25%  0.01250000     0.0000          0                2       2
1
## 50%  0.02307692     0.0000          0                2       2
3
## 75%  0.04060606     0.0000          0                3       2
4
## 100% 0.20000000   361.7637          1                8      13
9
##      traffictype
## 0%             1
```

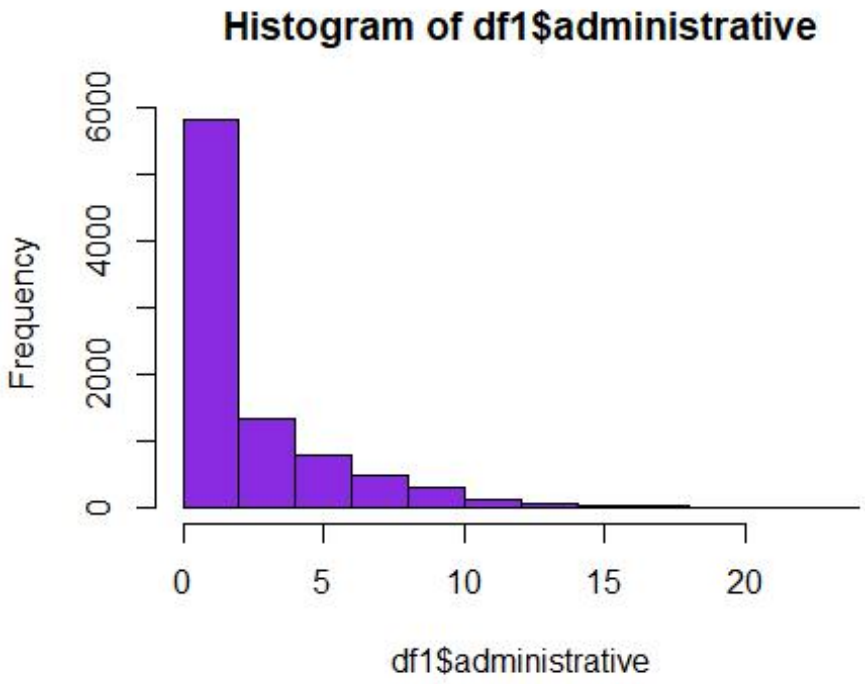
```
## 25%            2
## 50%            2
## 75%            4
## 100%          20
```

```
#check the range of  values of every column
Range = apply(df1, 2, range)    # Range
Range
```

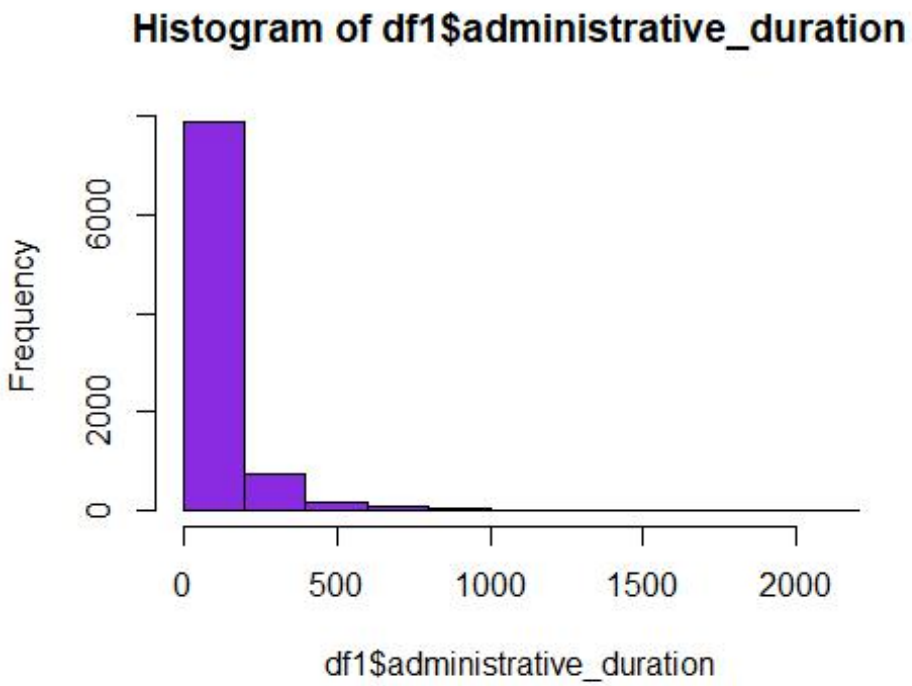
```
##      administrative administrative_duration informational
## [1,]              0                   0.000             0
## [2,]             24                2156.167            16
##      informational_duration productrelated productrelated_duration b
ouncerates
## [1,]                  0.000              0                      0.0
      0.0
## [2,]               2252.033            223                   3401.3
      0.2
##      exitrates pagevalues specialday operatingsystems browser region
## [1,]       0.0     0.0000          0                1       1      1
## [2,]       0.2   361.7637          1                8      13      9
##      traffictype
## [1,]           1
## [2,]          20
```

Administatrive

```
#See the administrative distribution
hist(df1$administrative, col='blueviolet')
```

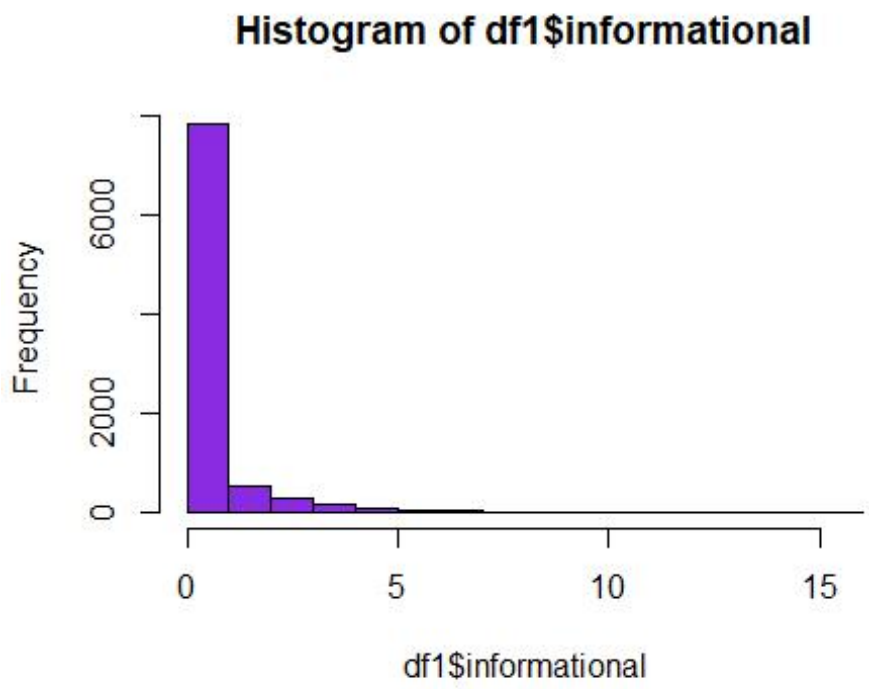



```
#See the administrative_duration distribution
hist(df1$administrative_duration, col='blueviolet')
```

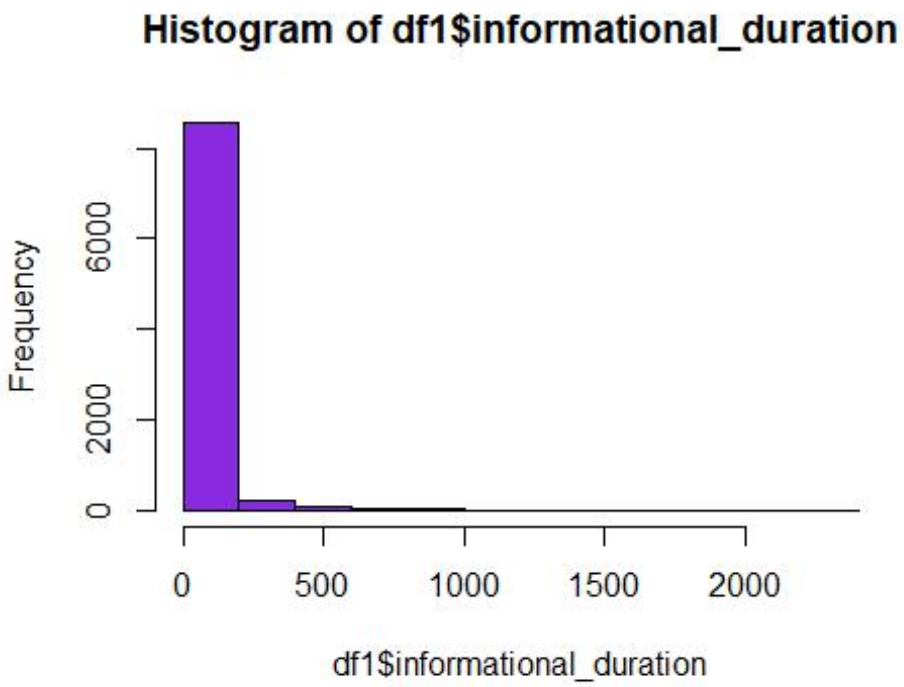


Informational

```
hist(df1$informational, col='blueviolet')  #See the informational distr
ibution
```

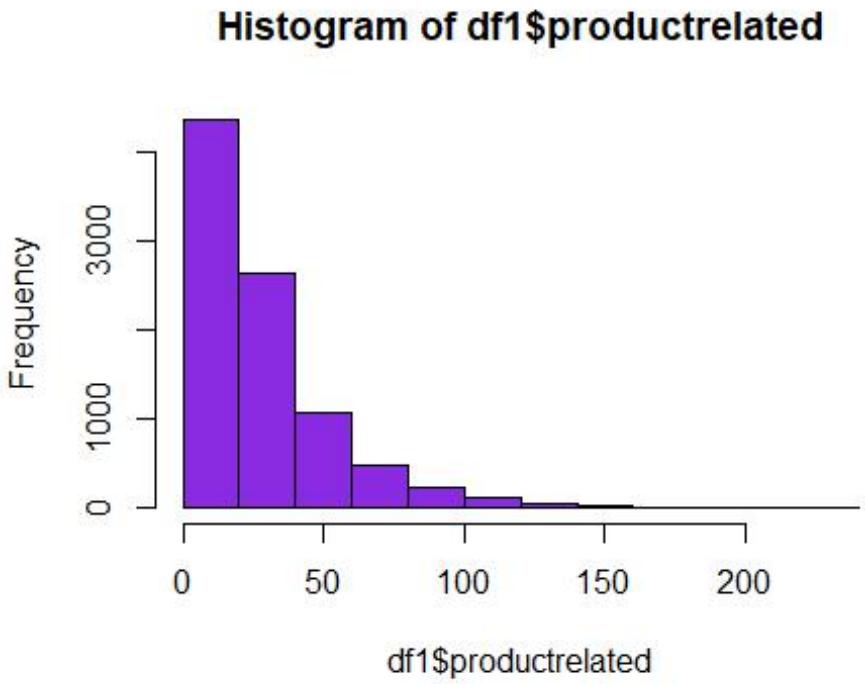



```
#See the informational_duration distribution
hist(df1$informational_duration, col='blueviolet')
```

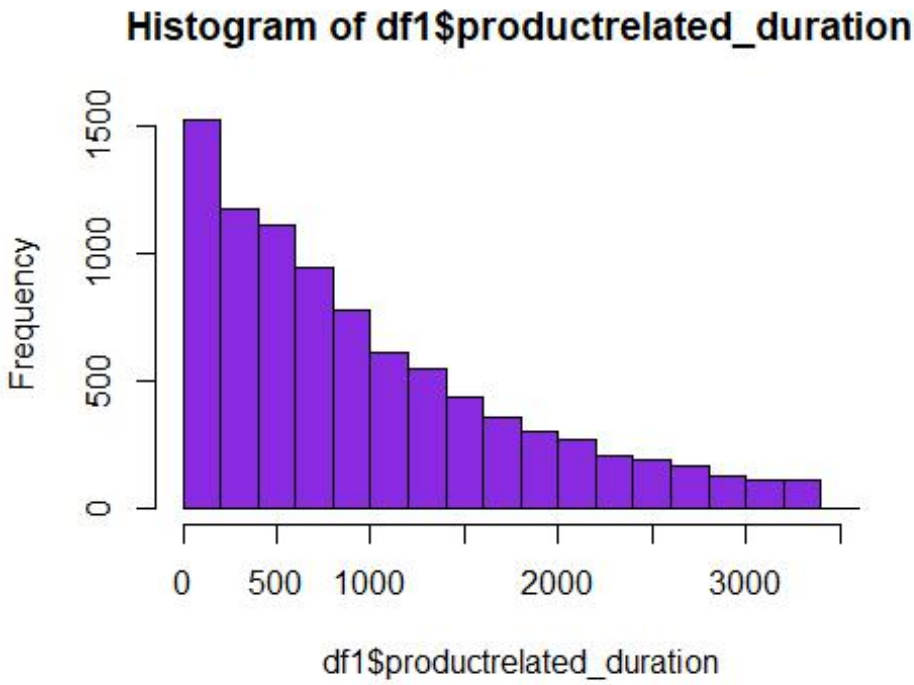


Product related

```
hist(df1$productrelated, col='blueviolet') #See the productrelated dist
ribution
```

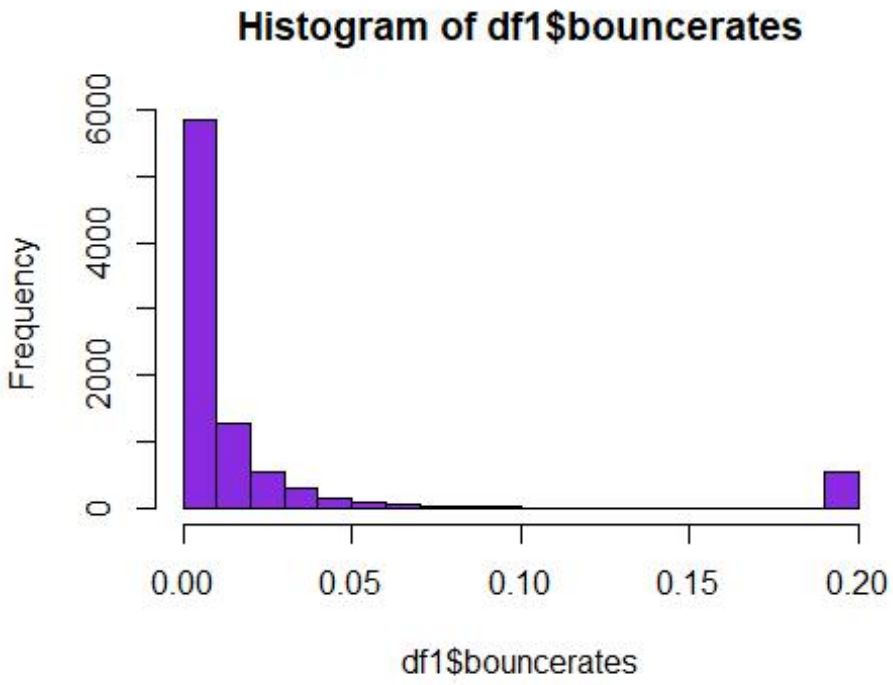



```
hist(df1$productrelated_duration, col='blueviolet')  #See the productre
lated_duration distribution
```



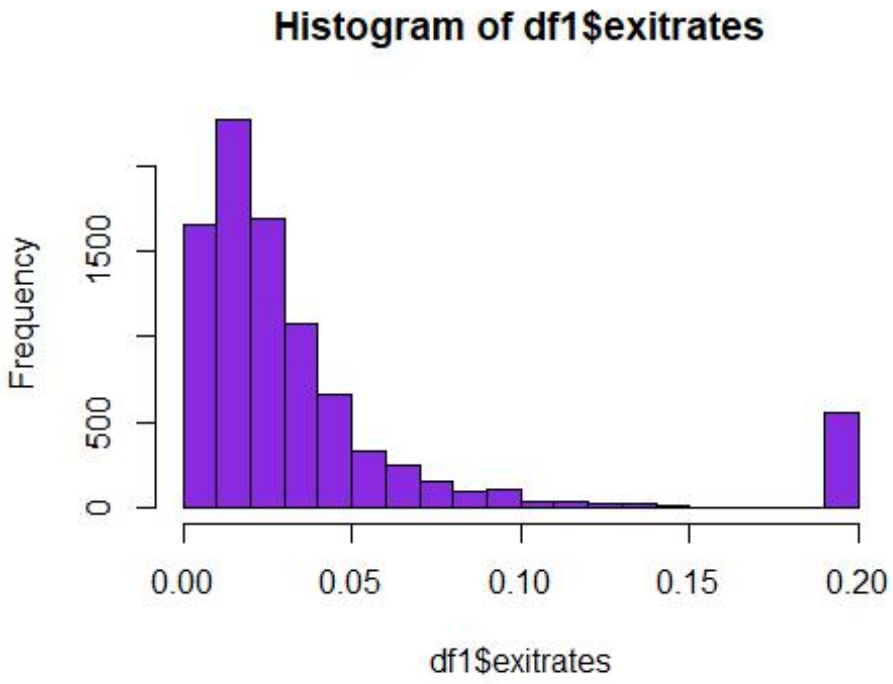
Bouncerates

```
hist(df1$bouncerates, col='blueviolet')
```



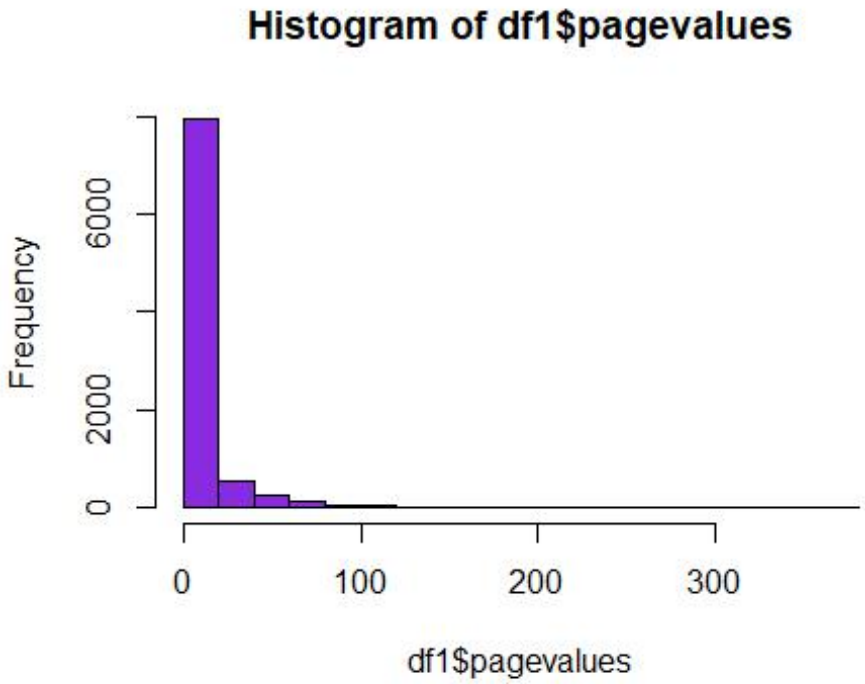
Exitrates

```
hist(df1$exitrates, col='blueviolet')
```



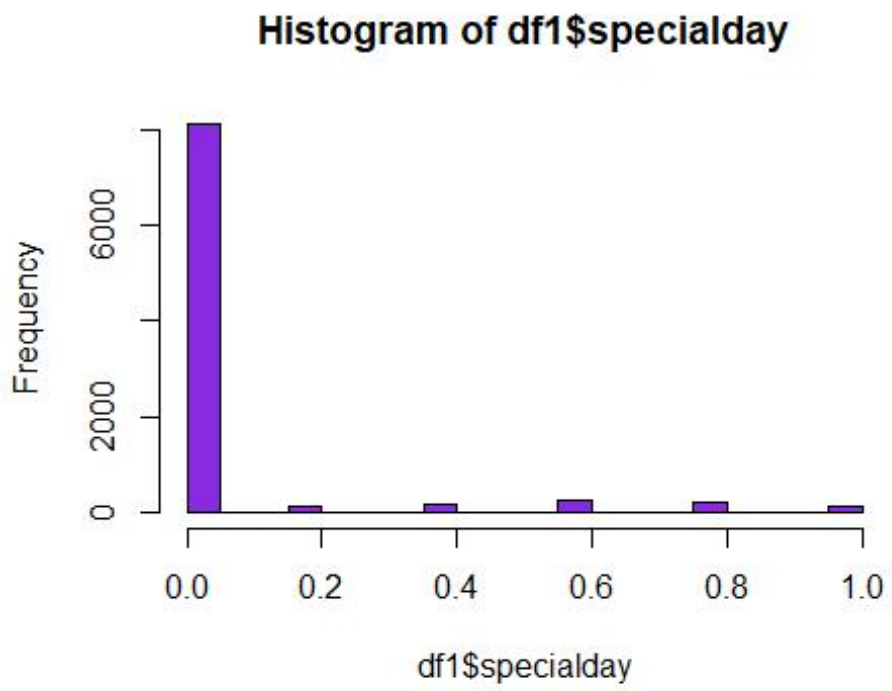
page-values

```
hist(df1$pagevalues, col='blueviolet')
```



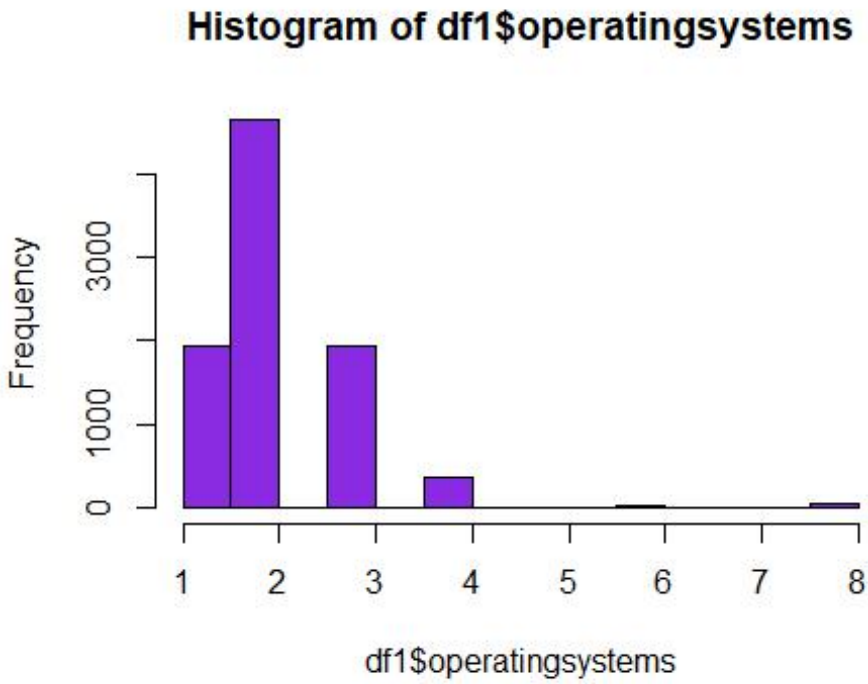
Special day

```
#See the daily_time_spent_on_site distribution
hist(df1$specialday, col='blueviolet')
```

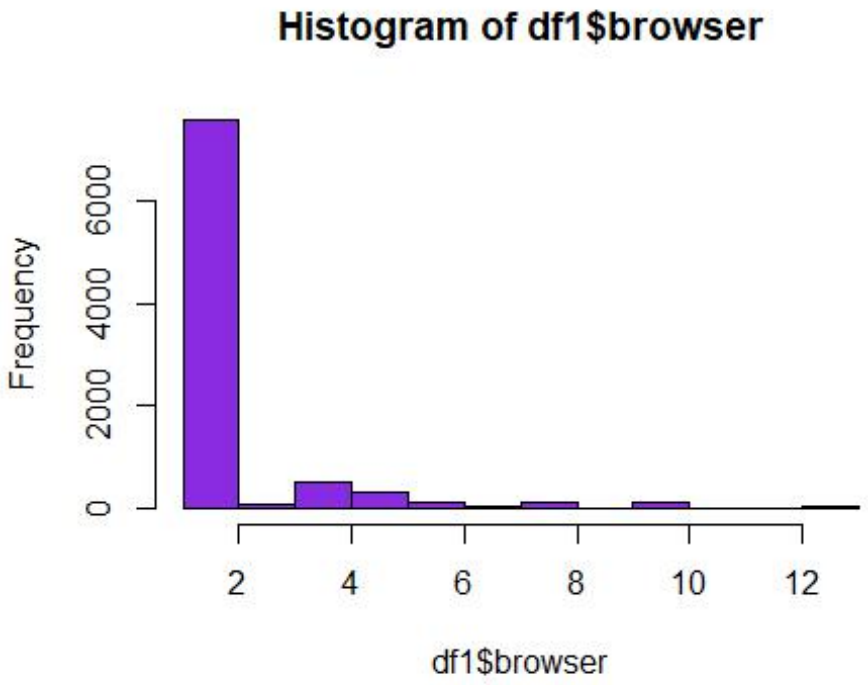


Computers

```
#See the operatingsystems distribution
hist(df1$operatingsystems, col='blueviolet')
```

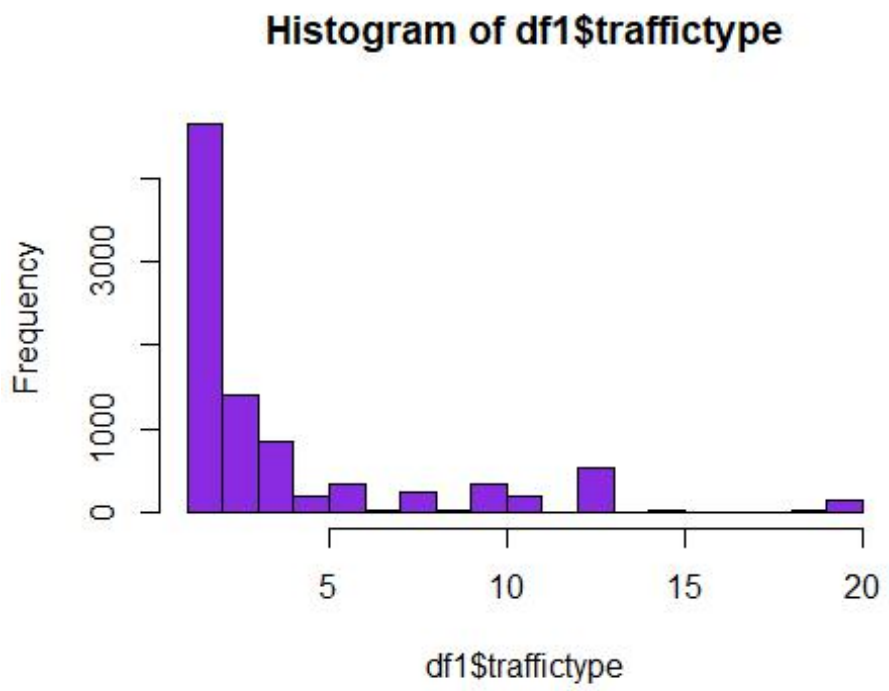



```r
#See the browser distribution
hist(df1$browser, col='blueviolet')
```

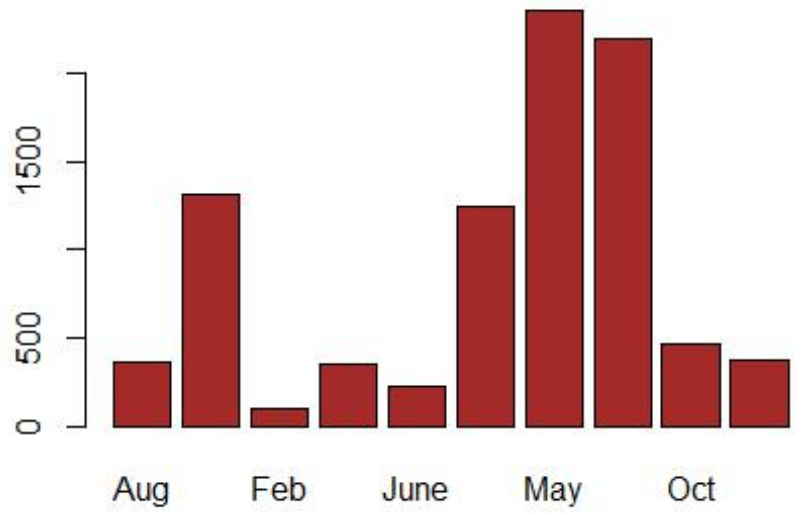


Traffic type

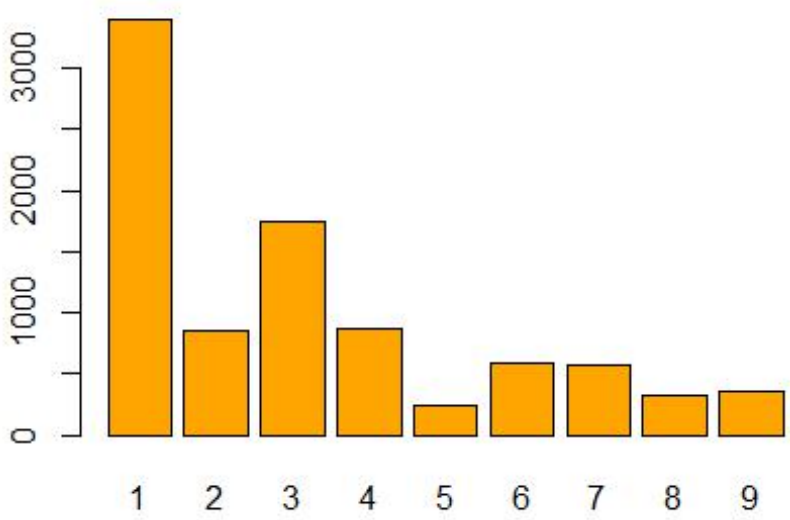
```
#See the traffictype distribution
hist(df1$traffictype, col='blueviolet')
```



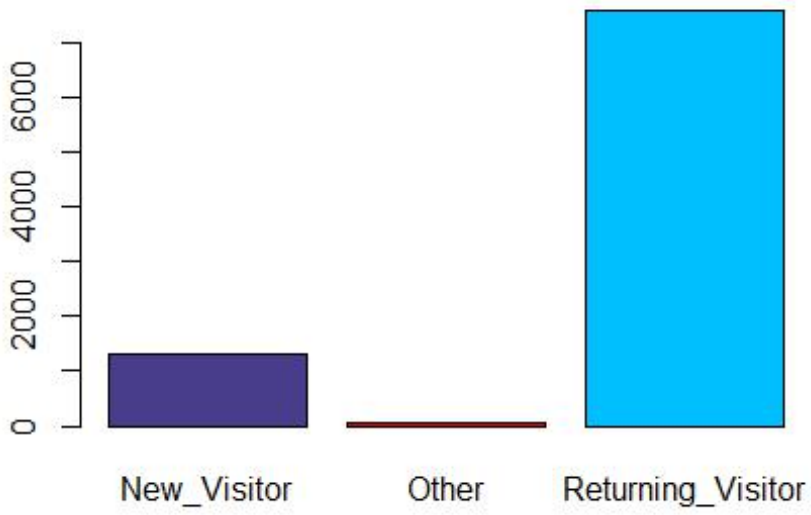

```
# Getting specific column - month
month1 <-shoppers1$month

# Applying the table() function will compute the frequency distribution
 of the month variable
# ---
#
month_frequency1 <- table(month1)


# Then applying the barplot function to produce its bar graph
# ---
#
barplot(month_frequency1, col=c("brown"))
```



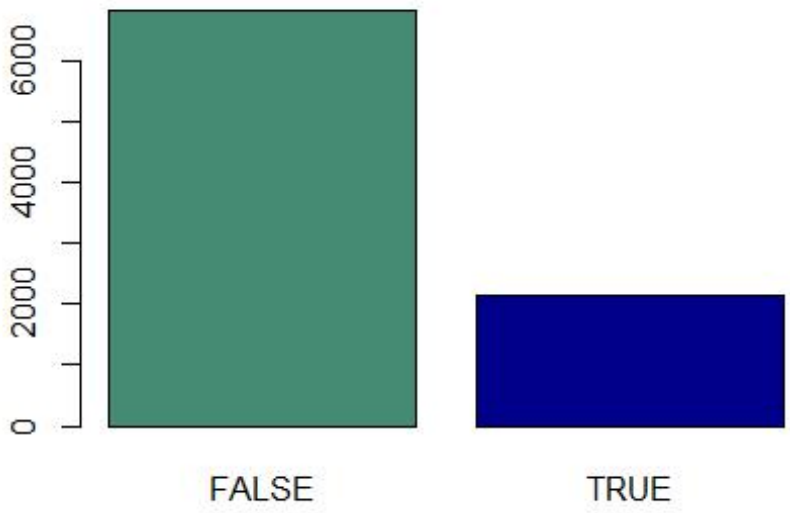

```
# Getting specific column - region
regions <-shoppers1$region

# Applying the table() function will compute the frequency distribution
 of the region variable
# ---
#
regions_frequency1 <- table(regions)


# Then applying the barplot function to produce its bar graph
# ---
#
barplot(regions_frequency1, col=c("orange"))
```



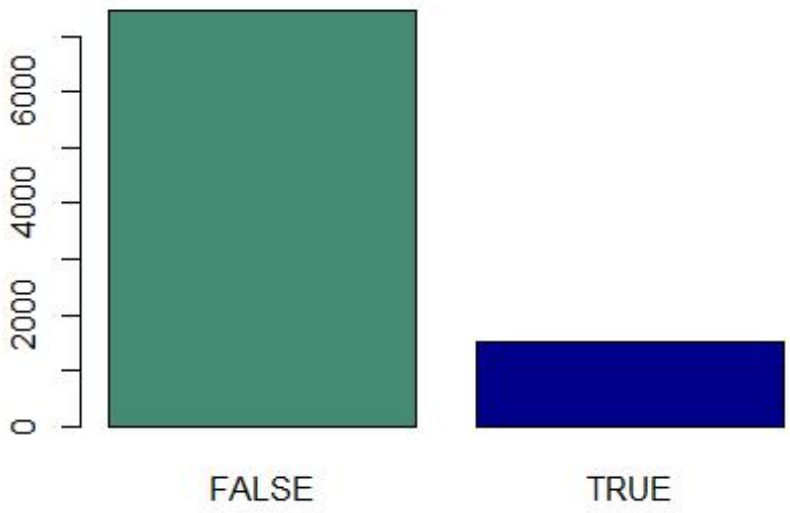

```
# Getting specific column - visitortype
visitors <-shoppers1$visitortype

# Applying the table() function will compute the frequency distribution
 of the visitortype variable
# ---
#
visitors_frequency1 <- table(visitors)


# Then applying the barplot function to produce its bar graph
# ---
#
barplot(visitors_frequency1, col=c("darkslateblue", "red", "deepskyblue
"))
```




```
# Getting specific column - weekend
weekends <- shoppers1$weekend

# Applying the table() function will compute the frequency distribution
 of the weekend variable
# ---
#
weekend_frequency1 <- table(weekends)


# Then applying the barplot function to produce its bar graph
# ---
#
barplot(weekend_frequency1, col=c("aquamarine4", "blue4"))
```



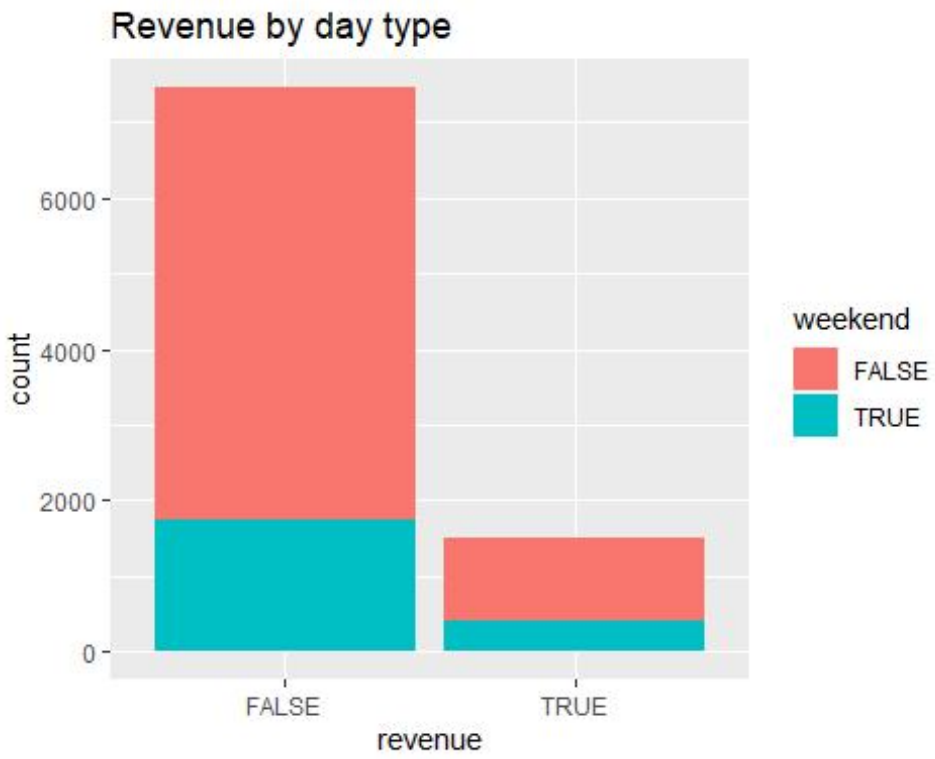

```
# Getting specific column - revenue
revenues <- shoppers1$revenue

# Applying the table() function will compute the frequency distribution
 of the revenue variable
# ---
#
revenues_frequency1 <- table(revenues)


# Then applying the barplot function to produce its bar graph
# ---
#
barplot(revenues_frequency1, col=c("aquamarine4", "blue4"))
```



Bivariate Analysis

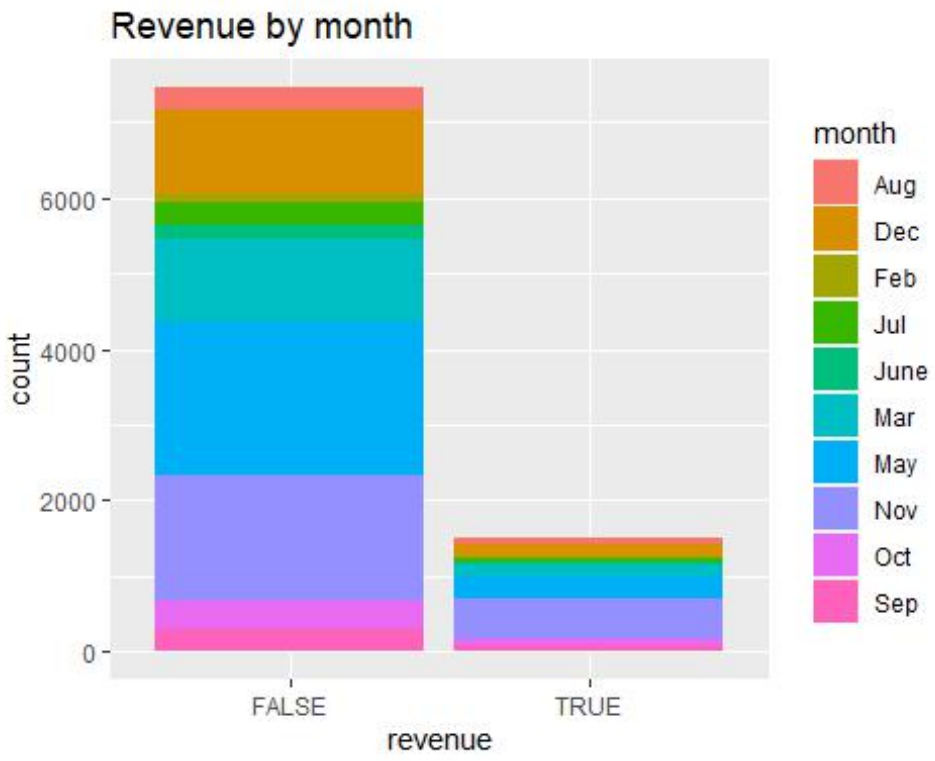
Revenue by day type

```
library(magrittr)

##
## Attaching package: 'magrittr'

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

## The following object is masked from 'package:tidyr':
##
##     extract

shoppers1 %>%
   ggplot(aes(revenue)) +
   geom_bar(aes(fill= weekend))+
   labs(title = "Revenue by day type")
```



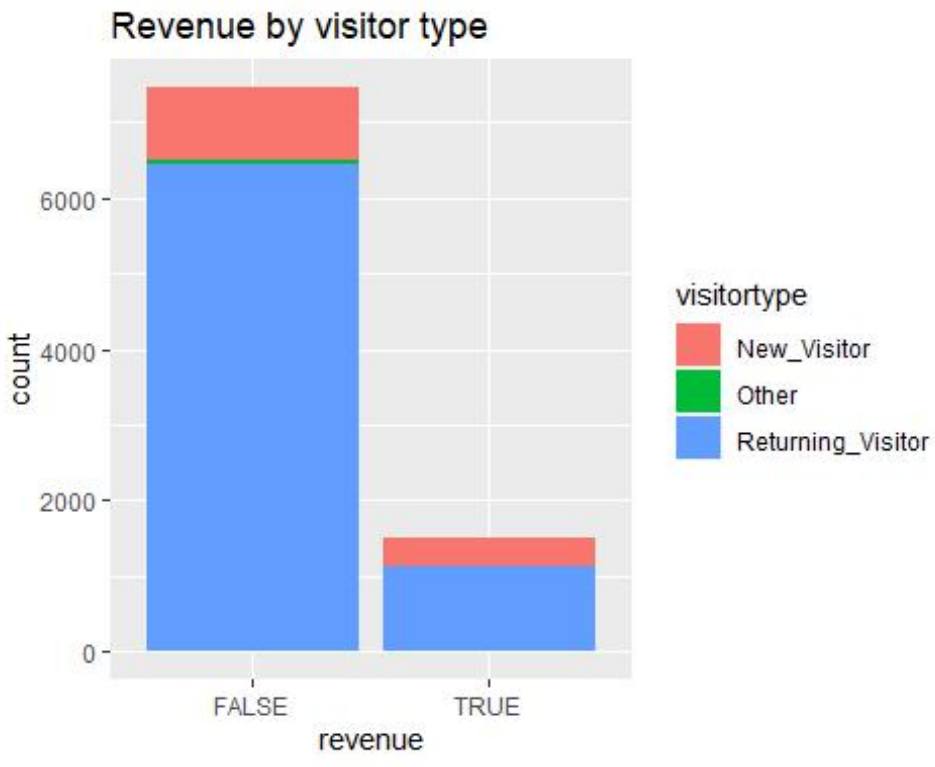
Revenue by month

```
library(magrittr)
shoppers1 %>%
   ggplot(aes(revenue)) +
   geom_bar(aes(fill= month))+
   labs(title = "Revenue by month")
```



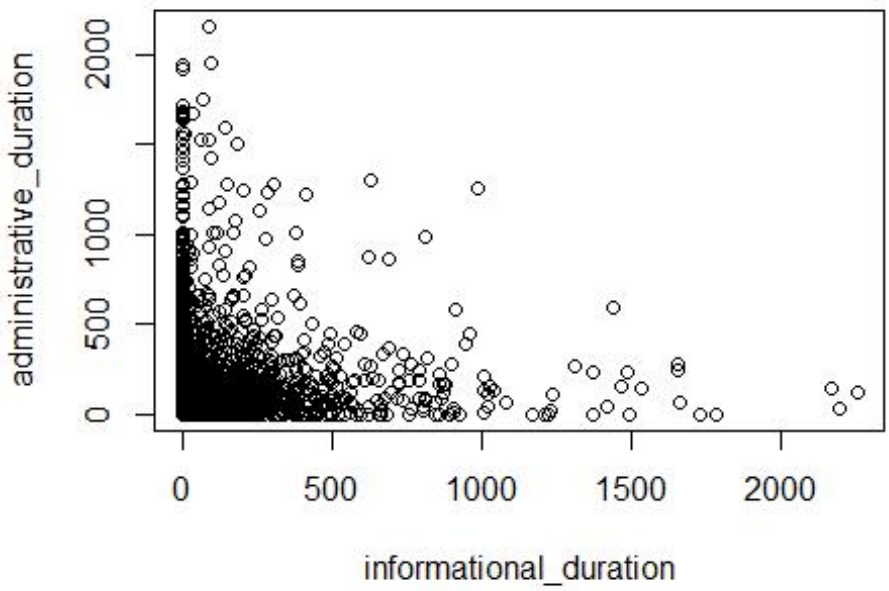
Revenue by visitor type

```
library(magrittr)
shoppers1 %>%
   ggplot(aes(revenue)) +
   geom_bar(aes(fill= visitortype))+
   labs(title = "Revenue by visitor type")
```



Time spent on Informational_ duration versus administrative_duration

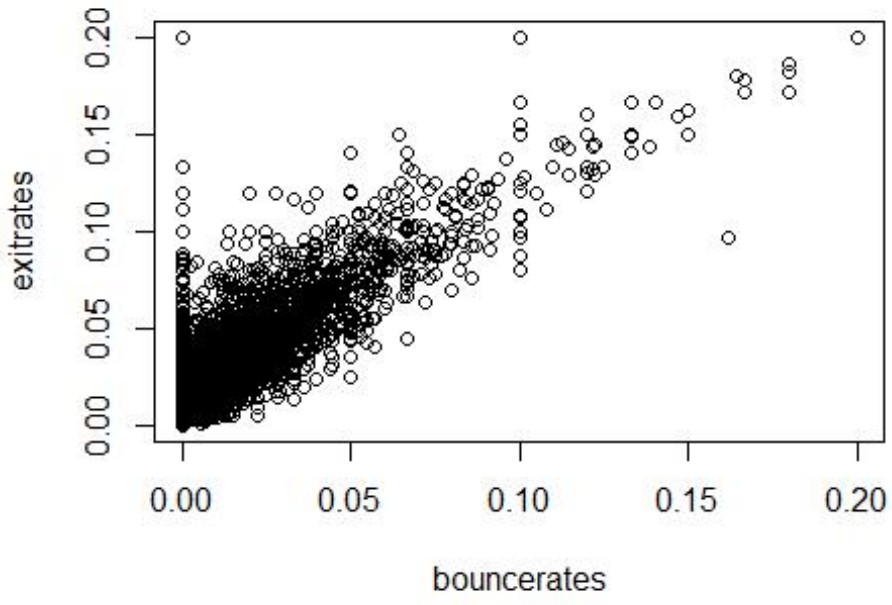
```
plot(informational_duration,administrative_duration,  xlab="information
al_duration", ylab="administrative_duration") # Drawwing a scatterplot
```




```
cor(informational_duration,administrative_duration) #Correlation Coeffi
cient

## [1] 0.1332411

cov(informational_duration,administrative_duration)

## [1] 2712.677
```

Bouncerates versus exit rates

```
plot(bouncerates,exitrates,  xlab="bouncerates", ylab="exitrates") # Dr
awwing a scatterplot
```




```
cor(bouncerates,exitrates) #Correlation Coefficient
```

```
## [1] 0.9479605
```

```
cov(bouncerates,exitrates)
```

```
## [1] 0.002185605
```

Multivariate Analysis

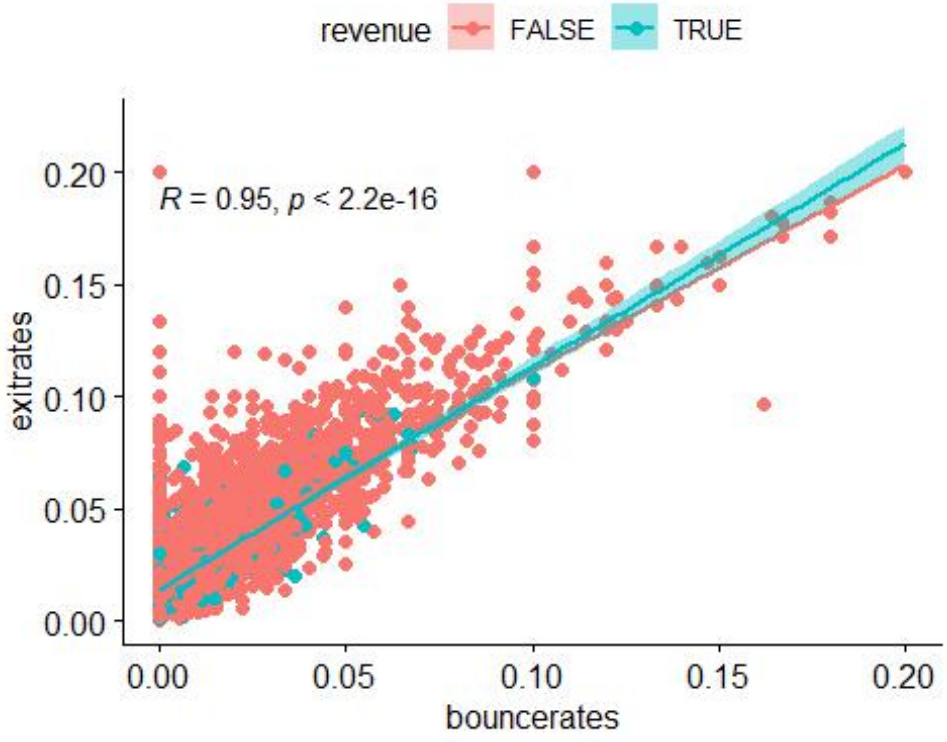
```
library("ggpubr")
```

```
##
## Attaching package: 'ggpubr'
```

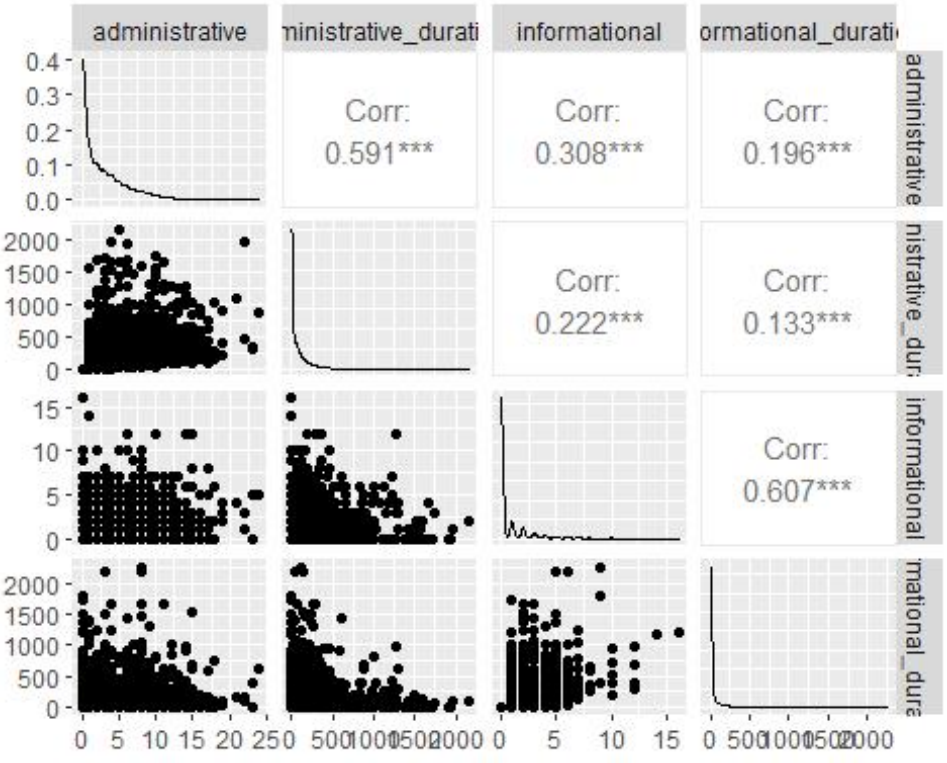
```
## The following object is masked from 'package:dendextend':
##
##     rotate
```

```
ggscatter(shoppers1, x = "bouncerates", y = "exitrates",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "bouncerates", ylab = "exitrates",  color = "revenue")
```

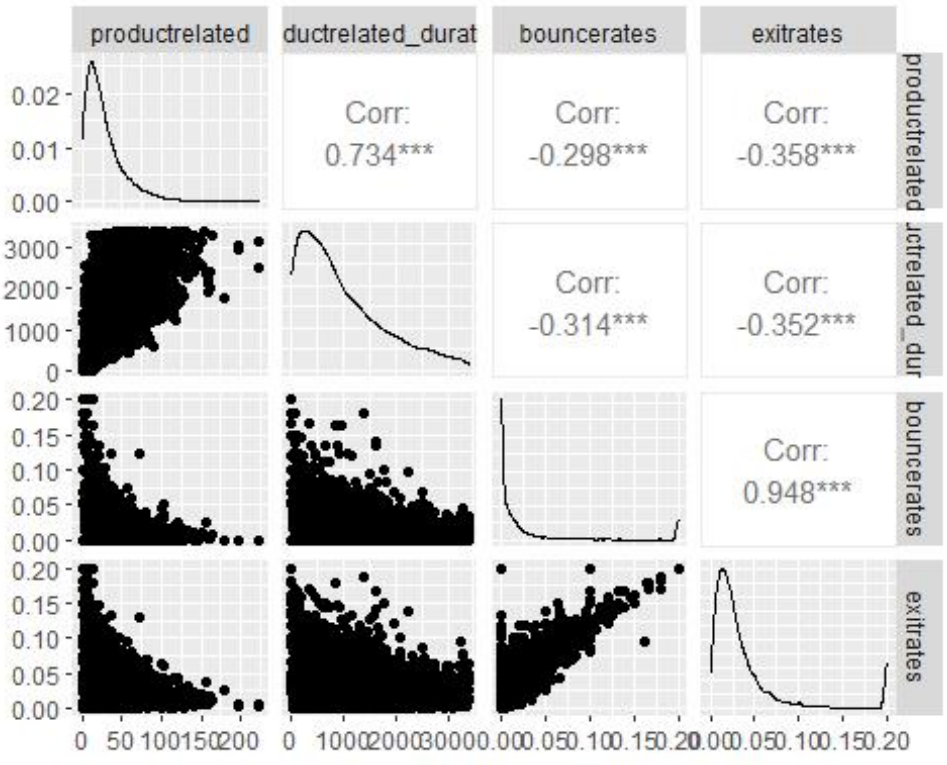
```
## `geom_smooth()` using formula 'y ~ x'
```



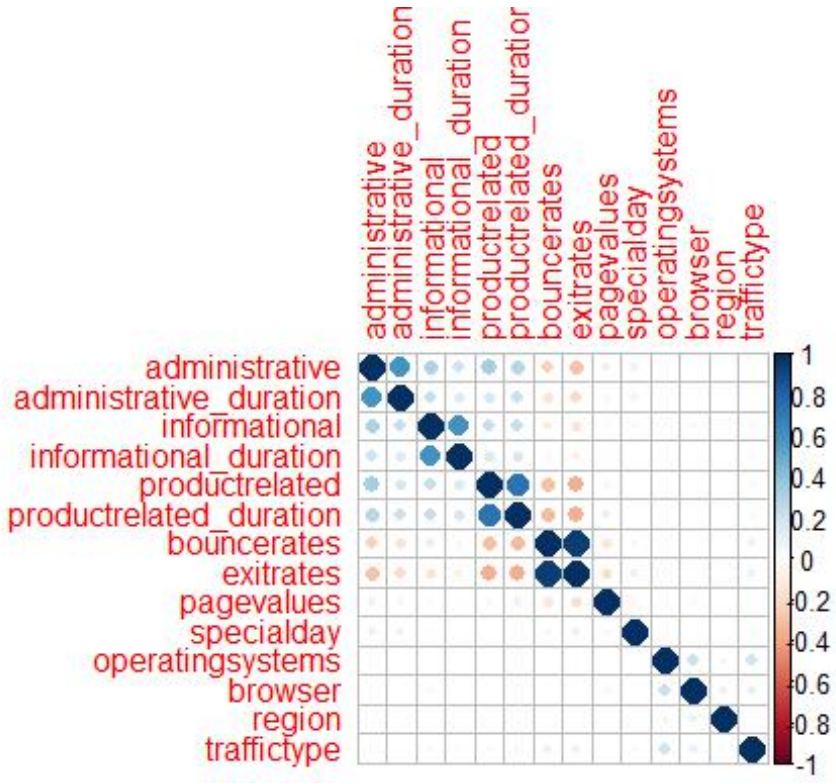

```
ggpairs(shoppers1[1:4])
```




```
ggpairs(shoppers1[5:8])
```




```
M<-cor(df1) #find the correlation
library(corrplot)

## corrplot 0.92 loaded

# Margins area
par(oma=c(0,0,0,0)) # all sides have 3 lines of space
par(mar=c(0,0,0,0) + 0.1)
corrplot(M, method="circle") #Compute and visualize the correlation co-
eeficients
```



Implement the Solution

```
# we would not require Class Label(output) during execution of our algo
rithm.
# We will, therefore, remove Class Attribute specialday and store it in
 another variable.


shopper1.new<- shoppers1[1:15]
shopper1.class<- shoppers1[, "revenue"]
head(shopper1.new)
```

```
## # A tibble: 6 × 15
##   administrative administrative_d… informational informational_d… pr
oductrelated
##            <dbl>             <dbl>         <dbl>            <dbl>
       <dbl>
## 1              0                 0             0                0
           1
## 2              0                 0             0                0
           2
## 3              0                 0             0                0
          10
## 4              0                 0             0                0
          19
## 5              0                 0             0                0
          16
```

```
## 6              2                53             0                 0
          23
## # … with 10 more variables: productrelated_duration <dbl>, bouncerat
es <dbl>,
## #   exitrates <dbl>, pagevalues <dbl>, specialday <dbl>, month <chr>,
## #   operatingsystems <dbl>, browser <dbl>, region <dbl>, traffictype
 <dbl>
```

```
head(shopper1.class)
```

```
## # A tibble: 6 × 1
##   revenue
##   <lgl>
## 1 FALSE
## 2 FALSE
## 3 FALSE
## 4 FALSE
## 5 FALSE
## 6 FALSE
```

One hot encoding

One-hot encoding is the process of converting a categorical variable with multiple categories into multiple variables, each with a value of 1 or 0.

```
dummy <- dummyVars(" ~ .", data=shopper1.new)
newdata <- data.frame(predict(dummy, newdata = shopper1.new))

glimpse(newdata) # look at the data after its encoded
```

```
## Rows: 8,956
## Columns: 24
## $ administrative          <dbl> 0, 0, 0, 0, 0, 2, 0, 0, 0, 4, 1, 0,
0, 0, 0, 0…
## $ administrative_duration <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 53.0, 0.0,
0.0, 0.0, …
## $ informational           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0…
## $ informational_duration  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0…
## $ productrelated          <dbl> 1, 2, 10, 19, 16, 23, 13, 20, 8, 32,
 45, 8, 10…
## $ productrelated_duration <dbl> 0.000000, 2.666667, 627.500000, 154.
216667, 40…
## $ bouncerates             <dbl> 0.200000000, 0.050000000, 0.02000000
0, 0.01578…
## $ exitrates               <dbl> 0.200000000, 0.140000000, 0.05000000
0, 0.02456…
## $ pagevalues              <dbl> 0.00000, 0.00000, 0.00000, 0.00000,
0.00000, 0…
## $ specialday              <dbl> 0.0, 0.0, 0.0, 0.0, 0.4, 0.0, 0.0, 0.
```

```
0, 1.0, 0…
## $ monthAug                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0…
## $ monthDec                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0…
## $ monthFeb                <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1…
## $ monthJul                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0…
## $ monthJune               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0…
## $ monthMar                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0…
## $ monthMay                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0…
## $ monthNov                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0…
## $ monthOct                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0…
## $ monthSep                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0…
## $ operatingsystems        <dbl> 1, 3, 3, 2, 1, 1, 1, 2, 2, 2, 3, 2,
1, 2, 2, 3…
## $ browser                 <dbl> 1, 2, 3, 2, 1, 1, 1, 4, 2, 2, 2, 2,
1, 2, 2, 2…
## $ region                  <dbl> 1, 2, 1, 1, 4, 9, 1, 4, 5, 1, 1, 1,
3, 1, 7, 1…
## $ traffictype             <dbl> 1, 4, 4, 3, 3, 3, 4, 4, 1, 3, 1, 1,
1, 1, 1, 4…
```

Scaling the data

```
df <- data.frame(scale(newdata))
head(df)

##   administrative administrative_duration informational informational
_duration
## 1     -0.7496555             -0.5004229    -0.4059591             -
0.2453589
## 2     -0.7496555             -0.5004229    -0.4059591             -
0.2453589
## 3     -0.7496555             -0.5004229    -0.4059591             -
0.2453589
## 4     -0.7496555             -0.5004229    -0.4059591             -
0.2453589
## 5     -0.7496555             -0.5004229    -0.4059591             -
0.2453589
## 6     -0.1191818             -0.1784053    -0.4059591             -
0.2453589
##   productrelated productrelated_duration bouncerates  exitrates page
```

```
values
## 1     -1.0972467              -1.1798029  3.67211244  3.4029522 -0.3
403108
## 2     -1.0560531              -1.1765290  0.58611386  2.1380254 -0.3
403108
## 3     -0.7265042              -0.4094074 -0.03108585  0.2406352 -0.3
403108
## 4     -0.3557617              -0.9904677 -0.11771037 -0.2956641 -0.3
403108
## 5     -0.4793426              -0.6791993 -0.05680251 -0.2688492 -0.3
403108
## 6     -0.1909873               0.8683873 -0.27110797 -0.4695656 -0.3
403108
##   specialday   monthAug   monthDec monthFeb   monthJul  monthJune
monthMar
## 1 -0.2940763 -0.2043382 -0.4135287 9.316342 -0.2016551 -0.1612525 -0.
4006703
## 2 -0.2940763 -0.2043382 -0.4135287 9.316342 -0.2016551 -0.1612525 -0.
4006703
## 3 -0.2940763 -0.2043382 -0.4135287 9.316342 -0.2016551 -0.1612525 -0.
4006703
## 4 -0.2940763 -0.2043382 -0.4135287 9.316342 -0.2016551 -0.1612525 -0.
4006703
## 5  1.7887609 -0.2043382 -0.4135287 9.316342 -0.2016551 -0.1612525 -0.
4006703
## 6 -0.2940763 -0.2043382 -0.4135287 9.316342 -0.2016551 -0.1612525 -0.
4006703
##     monthMay   monthNov  monthOct  monthSep operatingsystems    brow
ser
## 1 -0.5967479 -0.5692385 -0.232407 -0.206992       -1.2218704 -0.7891
816
## 2 -0.5967479 -0.5692385 -0.232407 -0.206992        0.9446454 -0.2003
363
## 3 -0.5967479 -0.5692385 -0.232407 -0.206992        0.9446454  0.3885
091
## 4 -0.5967479 -0.5692385 -0.232407 -0.206992       -0.1386125 -0.2003
363
## 5 -0.5967479 -0.5692385 -0.232407 -0.206992       -1.2218704 -0.7891
816
## 6 -0.5967479 -0.5692385 -0.232407 -0.206992       -1.2218704 -0.7891
816
##       region traffictype
## 1 -0.9055925 -0.76778840
## 2 -0.4891790 -0.01971258
## 3 -0.9055925 -0.01971258
## 4 -0.9055925 -0.26907119
## 5  0.3436481 -0.26907119
## 6  2.4257158 -0.26907119
```

Normalize the data

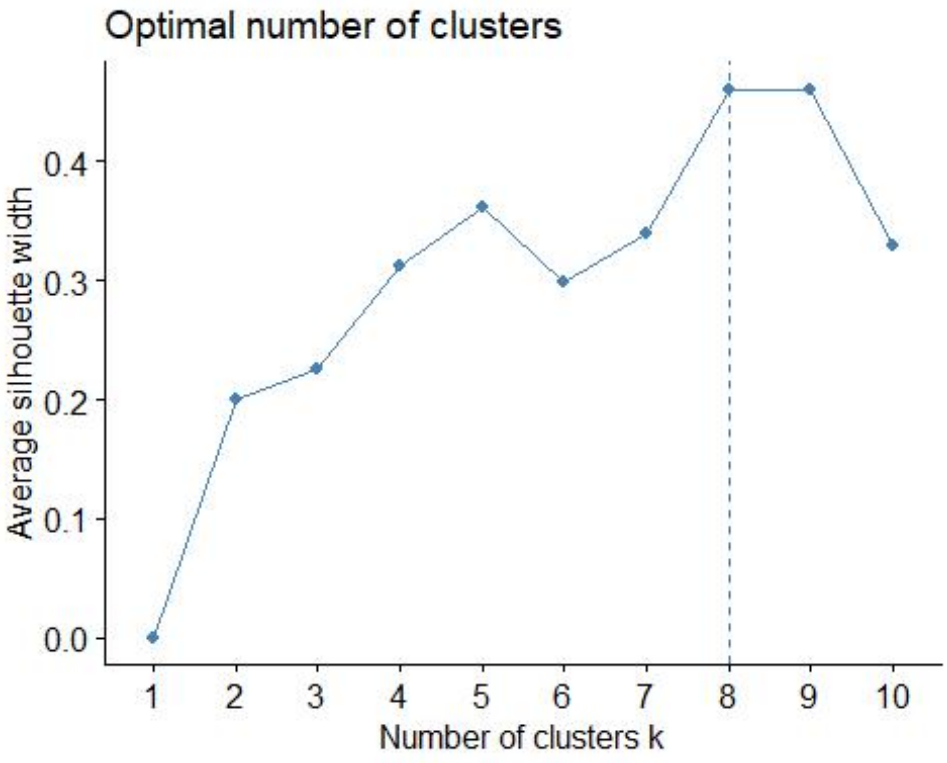
```r
normalize <- function(x){
  return ((x-min(x)) / (max(x)-min(x)))
}
normalized <- as.data.frame(apply(df,2, normalize))
head(normalized)
```

```
##   administrative administrative_duration informational informational
_duration
## 1     0.00000000              0.00000000             0
        0
## 2     0.00000000              0.00000000             0
        0
## 3     0.00000000              0.00000000             0
        0
## 4     0.00000000              0.00000000             0
        0
## 5     0.00000000              0.00000000             0
        0
## 6     0.08333333              0.02458066             0
        0
##   productrelated productrelated_duration bouncerates  exitrates page
values
## 1    0.004484305             0.000000000  1.00000000 1.00000000
     0
## 2    0.008968610             0.000784014  0.25000000 0.70000000
     0
## 3    0.044843049             0.184488284  0.10000000 0.25000000
     0
## 4    0.085201794             0.045340507  0.07894737 0.12280702
     0
## 5    0.071748879             0.119880634  0.09375000 0.12916667
     0
## 6    0.103139013             0.490484556  0.04166666 0.08156318
     0
##   specialday monthAug monthDec monthFeb monthJul monthJune monthMar
monthMay
## 1        0.0        0        0        1        0         0        0
       0
## 2        0.0        0        0        1        0         0        0
       0
## 3        0.0        0        0        1        0         0        0
       0
## 4        0.0        0        0        1        0         0        0
       0
## 5        0.4        0        0        1        0         0        0
       0
## 6        0.0        0        0        1        0         0        0
       0
##   monthNov monthOct monthSep operatingsystems    browser region traf
fictype
```

```
## 1        0        0        0        0.0000000 0.00000000  0.000   0.
0000000
## 2        0        0        0        0.2857143 0.08333333  0.125   0.
1578947
## 3        0        0        0        0.2857143 0.16666667  0.000   0.
1578947
## 4        0        0        0        0.1428571 0.08333333  0.000   0.
1052632
## 5        0        0        0        0.0000000 0.00000000  0.375   0.
1052632
## 6        0        0        0        0.0000000 0.00000000  1.000   0.
1052632
```

K-means clustering

```
#Clustering
set.seed(123)
#Determining the number of optimal clusters

#Determining optimal number of Clusters (Cluster silhoutte Method )
fviz_nbclust(normalized, FUN = kmeans, method = "silhouette")
```




```
result<- kmeans(normalized,8) #aplly k-means algorithm with no. of cent
roids(k)=8

result$size  # gives no. of records in each cluster

## [1] 2174 1356 1159 1423  565  704  347 1228
```

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

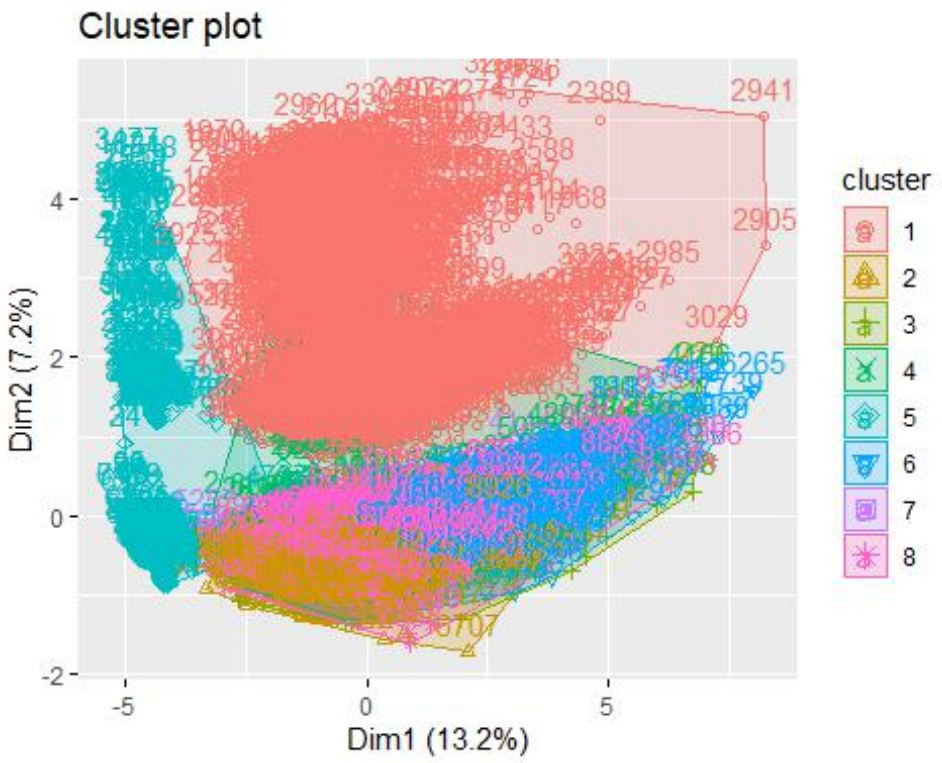
```
##   administrative administrative_duration informational informational
_duration
## 1    0.098781049              0.03872324  0.0309625115            0.
012817879
## 2    0.071994838              0.02768511  0.0202802360            0.
007889324
## 3    0.104903653              0.04274284  0.0336496980            0.
016478025
## 4    0.122042633              0.04373755  0.0296468728            0.
013716897
## 5    0.001032448              0.00000000  0.0006637168            0.
000000000
## 6    0.160511364              0.06365713  0.0566406250            0.
028297032
## 7    0.129082613              0.04744510  0.0326008646            0.
014256541
## 8    0.098873507              0.03854130  0.0315553746            0.
015187210
##   productrelated productrelated_duration bouncerates exitrates pagev
alues
## 1    0.134201179              0.30780772  0.05669666 0.1487563 0.020
13059
## 2    0.088601400              0.18895516  0.05045339 0.1512411 0.019
14955
## 3    0.111395706              0.29075906  0.03369860 0.1195598 0.016
02676
## 4    0.125182382              0.25453959  0.05528113 0.1476881 0.018
47761
## 5    0.007341561              0.00250979  0.99260079 0.9949917 0.000
00000
## 6    0.262408276              0.63642400  0.03220952 0.1028743 0.029
46729
## 7    0.145138988              0.27780662  0.06372485 0.1500044 0.018
13105
## 8    0.123395072              0.29306359  0.03929893 0.1335654 0.023
13437
##   specialday   monthAug  monthDec   monthFeb   monthJul  monthJune
monthMar
## 1 0.20505980 0.00000000 0.0000000 0.00000000 0.00000000 0.00000000 0.
0000000
## 2 0.00000000 0.00000000 0.0000000 0.00000000 0.00000000 0.00000000 0.
0000000
## 3 0.00000000 0.00000000 0.0000000 0.00000000 0.00000000 0.00000000 1.
0000000
## 4 0.01335207 0.00000000 0.0000000 0.05832748 0.22979621 0.14054814 0.
0000000
## 5 0.07256637 0.02123894 0.1415929 0.03362832 0.04070796 0.04778761 0.
```

```
1415929
## 6 0.00000000 0.00000000 0.0000000 0.00000000 0.00000000 0.00000000 0.
0000000
## 7 0.00000000 1.00000000 0.0000000 0.00000000 0.00000000 0.00000000 0.
0000000
## 8 0.00000000 0.00000000 1.0000000 0.00000000 0.00000000 0.00000000 0.
0000000
##    monthMay  monthNov   monthOct   monthSep operatingsystems    brow
ser
## 1 1.0000000 0.0000000 0.00000000 0.00000000        0.1625706 0.11645
201
## 2 0.0000000 1.0000000 0.00000000 0.00000000        0.1597134 0.10379
794
## 3 0.0000000 0.0000000 0.00000000 0.00000000        0.1559226 0.10648
548
## 4 0.0000000 0.0000000 0.31763879 0.25368939        0.1536994 0.10705
083
## 5 0.3150442 0.2336283 0.01238938 0.01238938        0.1694058 0.11312
684
## 6 0.0000000 1.0000000 0.00000000 0.00000000        0.1548295 0.09694
602
## 7 0.0000000 0.0000000 0.00000000 0.00000000        0.1490325 0.11239
193
## 8 0.0000000 0.0000000 0.00000000 0.00000000        0.1769428 0.12981
813
##      region traffictype
## 1 0.2709867   0.1759066
## 2 0.2927729   0.2022590
## 3 0.2535591   0.1156623
## 4 0.2822382   0.1413988
## 5 0.2887168   0.2100605
## 6 0.1942472   0.1536334
## 7 0.2766571   0.1258911
## 8 0.2908184   0.1538231

result$cluster #gives cluster vector showing the custer where

##    1    2    3    4    5    6    7    8    9   10   11   12   13   1
4   15   16
##    5    4    4    4    4    4    4    4    4    4    4    4    4
4    4    4
##   17   18   19   20   21   22   23   24   25   26   27   28   29   3
0   31   32
##    4    4    4    4    4    4    4    5    5    4    4    4    4
4    4    5
##   33   34   35   36   37   38   39   40   41   42   43   44   45   4
6   47   48
##    4    5    5    4    4    4    4    5    5    4    4    5    5
4    4    5
##   49   50   51   52   53   54   55   56   57   58   59   60   61   6
```

```
2   63   64
##    4    4    4    4    4    4    4    5    5    4    4    4    4
4    4    4
##   65   66   67   68   69   70   71   72   73   74   75   76   77   7
8   79   80
##    4    5    4    4    4    4    4    4    4    4    4    5    4
4    4    4
##   81   82   83   84   85   86   87   88   89   90   91   92   93   9
4   95   96
##    4    4    4    5    4    4    4    5    4    5    4    4    4
4    4    4
##   97   98   99  100  101  102  103  104  105  106  107  108  109  11
0  111  112
##    4    5    4    4    4    4    3    3    3    3    3    3    5
3    3    3
##  113  114  115  116  117  118  119  120  121  122  123  124  125  12
6  127  128
##    3    3    3    3    3    3    3    3    3    3    3    3    3
3    3    3
##  129  130  131  132  133  134  135  136  137  138  139  140  141  14
2  143  144
##    3    3    3    3    3    3    3    3    3    3    3    3    3
3    3    3
##  145  146  147  148  149  150  151  152  153  154  155  156  157  15
8  159  160
##    3    3    3    3    3    3    3    3    3    3    3    5    3
3    3    3
##  161  162  163  164  165  166  167  168  169  170  171  172  173  17
4  175  176
##    3    3    5    3    3    3    3    3    3    3    3    3    5
3    3    3
##  177  178  179  180  181  182  183  184  185  186  187  188  189  19
0  191  192
##    3    3    5    3    3    3    3    5    3    3    3    5    3
3    3    3
##  193  194  195  196  197  198  199  200  201  202  203  204  205  20
6  207  208
##    3    3    3    3    3    3    3    3    3    3    5    3    3
3    3    5
##  209  210  211  212  213  214  215  216  217  218  219  220  221  22
2  223  224
##    3    3    5    3    3    3    3    3    3    3    3    5    3
3    5    3
##  225  226  227  228  229  230  231  232  233  234  235  236  237  23
8  239  240
##    3    3    3    3    3    3    3    3    3    3    3    3    3
3    3    3
##  241  242  243  244  245  246  247  248  249  250  251  252  253  25
4  255  256
##    3    3    3    5    3    3    3    3    3    3    3    3    3
```

```
    6    6    2
## 8657 8658 8659 8660 8661 8662 8663 8664 8665 8666 8667 8668 8669 867
0 8671 8672
##    8    6    6    6    8    8    6    5    2    8    2    8    8
8    2    6
## 8673 8674 8675 8676 8677 8678 8679 8680 8681 8682 8683 8684 8685 868
6 8687 8688
##    6    8    2    8    8    8    2    8    5    8    8    2    8
2    2    2
## 8689 8690 8691 8692 8693 8694 8695 8696 8697 8698 8699 8700 8701 870
2 8703 8704
##    2    6    2    2    2    8    2    2    2    2    2    6    8
2    2    6
## 8705 8706 8707 8708 8709 8710 8711 8712 8713 8714 8715 8716 8717 871
8 8719 8720
##    5    2    8    8    2    2    2    8    2    8    8    6    2
8    5    6
## 8721 8722 8723 8724 8725 8726 8727 8728 8729 8730 8731 8732 8733 873
4 8735 8736
##    2    5    2    8    5    2    6    6    8    2    2    8    2
8    8    2
## 8737 8738 8739 8740 8741 8742 8743 8744 8745 8746 8747 8748 8749 875
0 8751 8752
##    8    8    5    8    2    2    2    8    6    2    2    8    6
8    2    2
## 8753 8754 8755 8756 8757 8758 8759 8760 8761 8762 8763 8764 8765 876
6 8767 8768
##    8    8    2    6    6    2    2    2    8    8    8    8    8
2    2    8
## 8769 8770 8771 8772 8773 8774 8775 8776 8777 8778 8779 8780 8781 878
2 8783 8784
##    2    8    8    8    8    6    2    5    2    2    6    8    2
8    2    8
## 8785 8786 8787 8788 8789 8790 8791 8792 8793 8794 8795 8796 8797 879
8 8799 8800
##    8    5    8    8    2    5    2    8    8    8    2    8    2
2    6    8
## 8801 8802 8803 8804 8805 8806 8807 8808 8809 8810 8811 8812 8813 881
4 8815 8816
##    8    2    8    6    2    8    8    6    8    8    2    8    8
5    8    6
## 8817 8818 8819 8820 8821 8822 8823 8824 8825 8826 8827 8828 8829 883
0 8831 8832
##    2    2    8    6    2    6    8    2    6    2    2    2    2
2    8    2
## 8833 8834 8835 8836 8837 8838 8839 8840 8841 8842 8843 8844 8845 884
6 8847 8848
##    8    6    8    8    8    6    2    8    8    8    8    8    8
6    2    6
## 8849 8850 8851 8852 8853 8854 8855 8856 8857 8858 8859 8860 8861 886
```

```
2 8863 8864
##    6    2    8    8    6    2    5    5    2    8    6    8    8
8    2    2
## 8865 8866 8867 8868 8869 8870 8871 8872 8873 8874 8875 8876 8877 887
8 8879 8880
##    6    5    5    8    2    2    8    2    2    8    8    8    8
6    6    8
## 8881 8882 8883 8884 8885 8886 8887 8888 8889 8890 8891 8892 8893 889
4 8895 8896
##    8    8    5    2    2    8    8    8    8    6    2    8    6
2    8    6
## 8897 8898 8899 8900 8901 8902 8903 8904 8905 8906 8907 8908 8909 891
0 8911 8912
##    2    2    2    2    8    6    8    8    8    8    8    2    6
2    2    6
## 8913 8914 8915 8916 8917 8918 8919 8920 8921 8922 8923 8924 8925 892
6 8927 8928
##    8    6    8    8    6    8    6    2    2    2    8    2    6
2    8    8
## 8929 8930 8931 8932 8933 8934 8935 8936 8937 8938 8939 8940 8941 894
2 8943 8944
##    2    8    2    8    2    2    2    5    2    8    2    8    8
8    2    8
## 8945 8946 8947 8948 8949 8950 8951 8952 8953 8954 8955 8956
##    8    8    8    8    2    5    8    2    2    8    2    2

result$betweenss / result$totss  #accuracy of the model

## [1] 0.6215534

fviz_cluster(result, normalized)
```



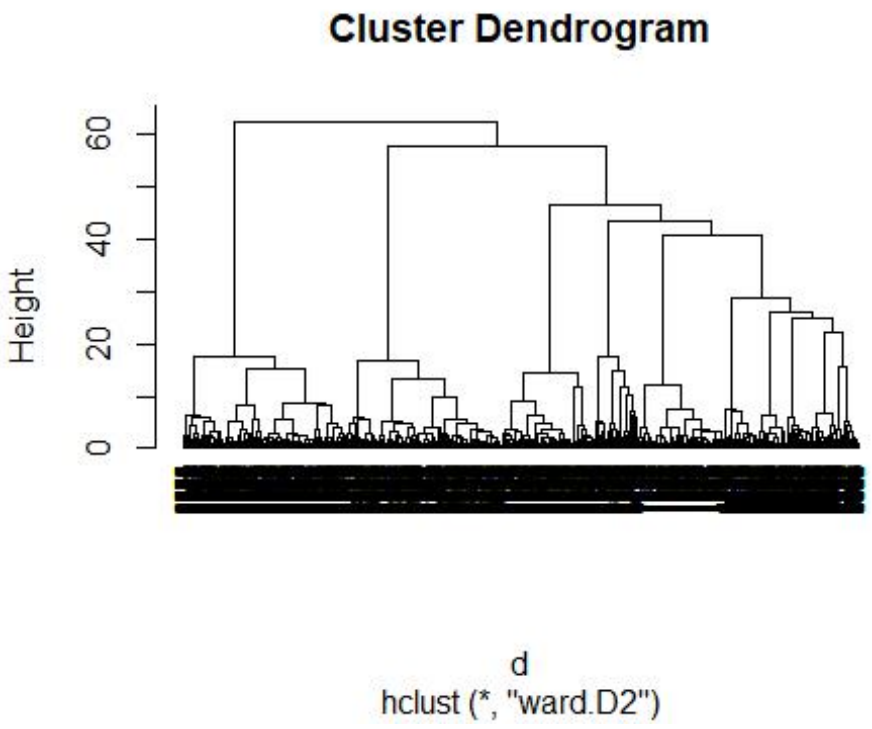

```
table(result$cluster, shoppers1$revenue)
```

```
##
##      FALSE TRUE
##    1  1860  314
##    2  1086  270
##    3   996  163
##    4  1173  250
##    5   562    3
##    6   458  246
##    7   282   65
##    8  1046  182
```

Hierachical clustering

```
d <- dist(normalized, method = "euclidean")

# We then hierarchical clustering using the Ward's method
# ---
#
res.hc <- hclust(d, method = "ward.D2" )

# We then hierarchical clustering using the Ward's method
# ---
#
res.hc <- hclust(d, method = "ward.D2" )
```

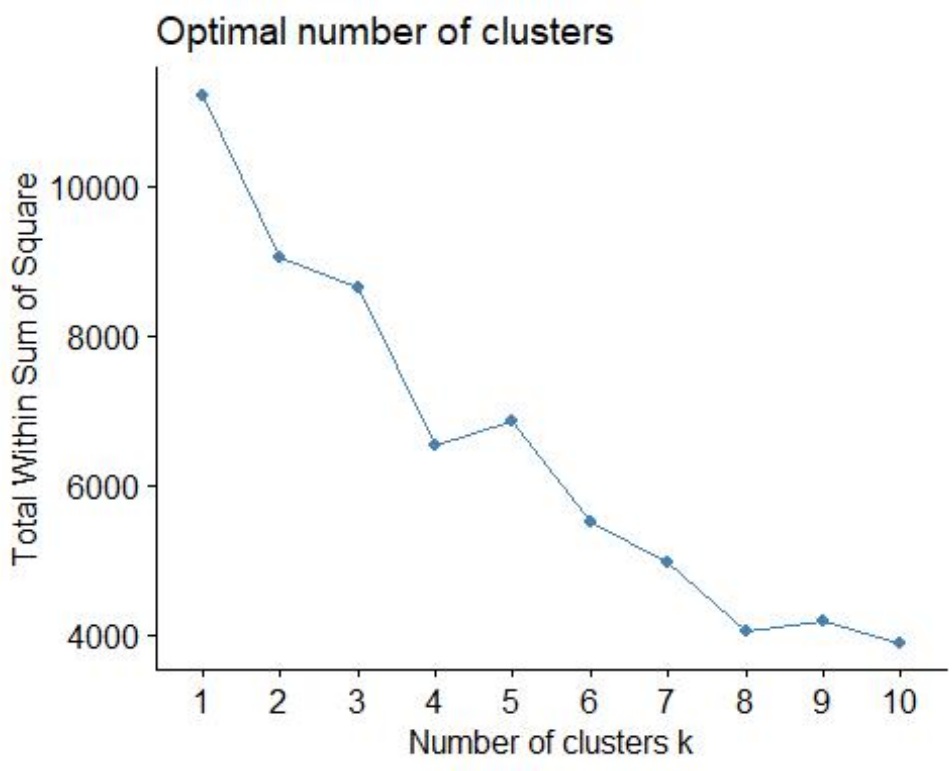
```
options(repr.plot.width=35 ,repr.plot.height=20 )
plot(res.hc, cex = 0.6, hang = -1)
```



Challenge the Solution

K-means clustering

```
#Determining optimal number of Clusters (Cluster elbow method)
fviz_nbclust(normalized, FUN = kmeans, method = "wss")
```




```
result1<- kmeans(normalized,4) #aplly k-means algorithm with no. of cen
troids(k)=4

result1$size  # gives no. of records in each cluster

## [1] 2352 2192 3104 1308

result1$centers # gives value of cluster center datapoint

##   administrative administrative_duration informational informational
_duration
## 1     0.09148243              0.03579266    0.02867241             0.
01184782
## 2     0.09612606              0.03757100    0.03085082             0.
01396854
## 3     0.10954951              0.04131473    0.02980026             0.
01403486
## 4     0.09288991              0.03618403    0.02962538             0.
01425833
##   productrelated productrelated_duration bouncerates exitrates pagev
alues
## 1      0.1246015               0.2847000  0.12745545 0.2127895 0.018
60710
## 2      0.1394717               0.3214738  0.10172915 0.1867882 0.021
31011
## 3      0.1157150               0.2564683  0.10064552 0.1849474 0.016
48200
```

```
## 4      0.1161702               0.2752108  0.09792998 0.1864491 0.021
71942
##    specialday  monthAug monthDec   monthFeb  monthJul  monthJune  mo
nthMar
## 1 0.204336735 0.0000000        0 0.00000000 0.0000000 0.00000000 0.0
000000
## 2 0.000000000 0.0000000        0 0.00000000 0.0000000 0.00000000 0.0
000000
## 3 0.008118557 0.1156572        0 0.03286082 0.1127577 0.07313144 0.3
991624
## 4 0.000000000 0.0000000        1 0.00000000 0.0000000 0.00000000 0.0
000000
##   monthMay monthNov  monthOct  monthSep operatingsystems   browser
  region
## 1        1        0 0.0000000 0.0000000        0.1624150 0.1152920 0.
2727997
## 2        0        1 0.0000000 0.0000000        0.1610401 0.1032543 0.
2627737
## 3        0        0 0.1478737 0.1185567        0.1536267 0.1072541 0.
2685245
## 4        0        0 0.0000000 0.0000000        0.1768239 0.1298420 0.
2931957
##   traffictype
## 1   0.1802721
## 2   0.1908855
## 3   0.1304768
## 4   0.1559231

result1$cluster #gives cluster vector showing the custer where

##    1    2    3    4    5    6    7    8    9   10   11   12   13   1
4   15   16
##    3    3    3    3    3    3    3    3    3    3    3    3    3
3    3    3
##   17   18   19   20   21   22   23   24   25   26   27   28   29   3
0   31   32
##    3    3    3    3    3    3    3    3    3    3    3    3    3
3    3    3
##   33   34   35   36   37   38   39   40   41   42   43   44   45   4
6   47   48
##    3    3    3    3    3    3    3    3    3    3    3    3    3
3    3    3
##   49   50   51   52   53   54   55   56   57   58   59   60   61   6
2   63   64
##    3    3    3    3    3    3    3    3    3    3    3    3    3
3    3    3
##   65   66   67   68   69   70   71   72   73   74   75   76   77   7
8   79   80
##    3    3    3    3    3    3    3    3    3    3    3    3    3
3    3    3
```

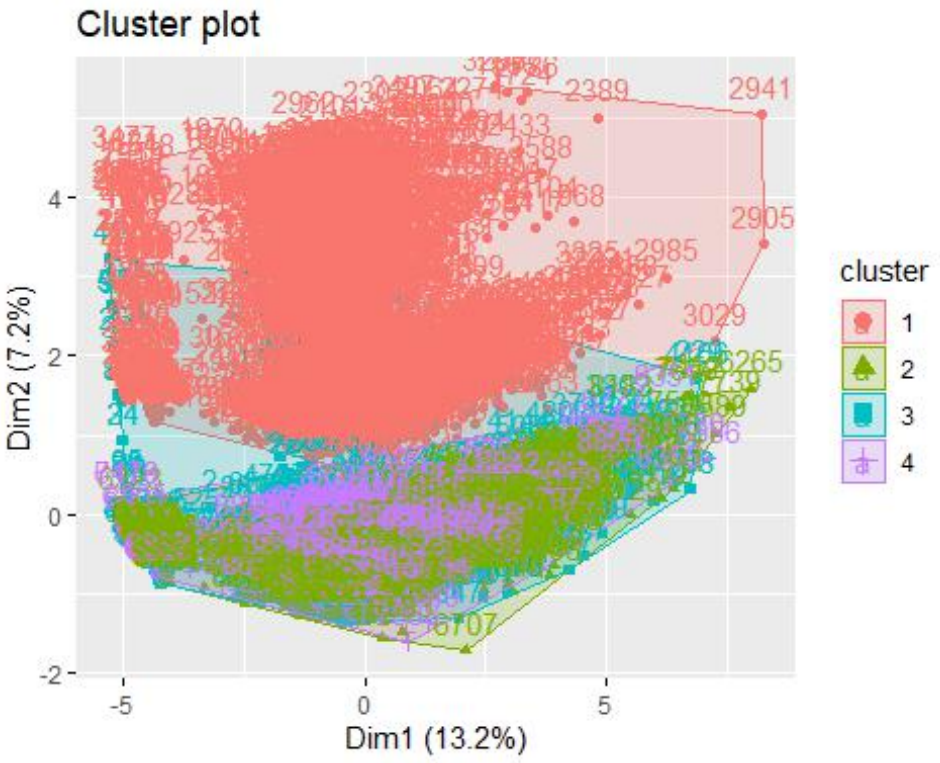
```
##    2    4    2    4    4    4    2    4    2    4    4    2    4
2    2    2
## 8689 8690 8691 8692 8693 8694 8695 8696 8697 8698 8699 8700 8701 870
2 8703 8704
##    2    2    2    2    2    4    2    2    2    2    2    2    4
2    2    2
## 8705 8706 8707 8708 8709 8710 8711 8712 8713 8714 8715 8716 8717 871
8 8719 8720
##    2    2    4    4    2    2    2    4    2    4    4    2    2
4    4    2
## 8721 8722 8723 8724 8725 8726 8727 8728 8729 8730 8731 8732 8733 873
4 8735 8736
##    2    4    2    4    2    2    2    2    4    2    2    4    2
4    4    2
## 8737 8738 8739 8740 8741 8742 8743 8744 8745 8746 8747 8748 8749 875
0 8751 8752
##    4    4    4    4    2    2    2    4    2    2    2    4    2
4    2    2
## 8753 8754 8755 8756 8757 8758 8759 8760 8761 8762 8763 8764 8765 876
6 8767 8768
##    4    4    2    2    2    2    2    2    4    4    4    4    4
2    2    4
## 8769 8770 8771 8772 8773 8774 8775 8776 8777 8778 8779 8780 8781 878
2 8783 8784
##    2    4    4    4    4    2    2    2    2    2    2    4    2
4    2    4
## 8785 8786 8787 8788 8789 8790 8791 8792 8793 8794 8795 8796 8797 879
8 8799 8800
##    4    2    4    4    2    2    2    4    4    4    2    4    2
2    2    4
## 8801 8802 8803 8804 8805 8806 8807 8808 8809 8810 8811 8812 8813 881
4 8815 8816
##    4    2    4    2    2    4    4    2    4    4    2    4    4
2    4    2
## 8817 8818 8819 8820 8821 8822 8823 8824 8825 8826 8827 8828 8829 883
0 8831 8832
##    2    2    4    2    2    2    4    2    2    2    2    2    2
2    4    2
## 8833 8834 8835 8836 8837 8838 8839 8840 8841 8842 8843 8844 8845 884
6 8847 8848
##    4    2    4    4    4    2    2    4    4    4    4    4    4
2    2    2
## 8849 8850 8851 8852 8853 8854 8855 8856 8857 8858 8859 8860 8861 886
2 8863 8864
##    2    2    4    4    2    2    4    4    2    4    2    4    4
4    2    2
## 8865 8866 8867 8868 8869 8870 8871 8872 8873 8874 8875 8876 8877 887
8 8879 8880
##    2    2    4    4    2    2    4    2    2    4    4    4    4
```

```
2    2    4
## 8881 8882 8883 8884 8885 8886 8887 8888 8889 8890 8891 8892 8893 889
4 8895 8896
##    4    4    2    2    2    4    4    4    4    2    2    4    2
2    4    2
## 8897 8898 8899 8900 8901 8902 8903 8904 8905 8906 8907 8908 8909 891
0 8911 8912
##    2    2    2    2    4    2    4    4    4    4    4    2    2
2    2    2
## 8913 8914 8915 8916 8917 8918 8919 8920 8921 8922 8923 8924 8925 892
6 8927 8928
##    4    2    4    4    2    4    2    2    2    2    4    2    2
2    4    4
## 8929 8930 8931 8932 8933 8934 8935 8936 8937 8938 8939 8940 8941 894
2 8943 8944
##    2    4    2    4    2    2    2    2    2    4    2    4    4
4    2    4
## 8945 8946 8947 8948 8949 8950 8951 8952 8953 8954 8955 8956
##    4    4    4    4    2    2    4    2    2    4    2    2
```

```
result1$betweenss / result$totss
```

```
## [1] 0.4513823
```

```
fviz_cluster(result1, normalized)
```

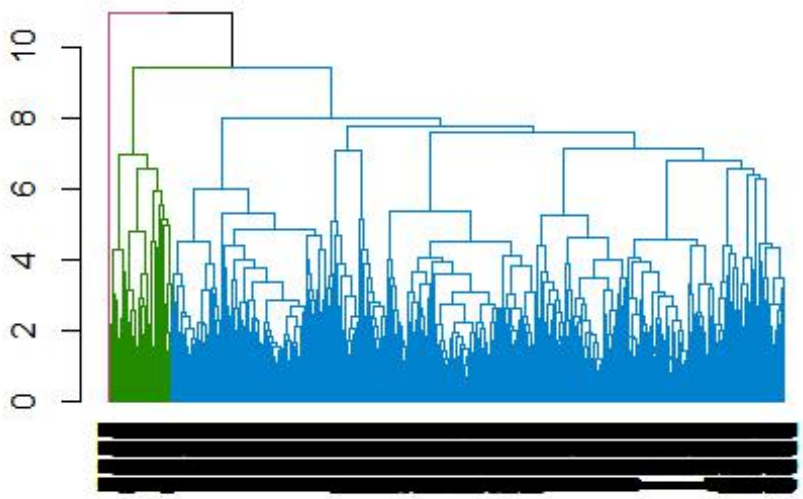



```
table(result1$cluster, shoppers1$revenue)
```

```
##
##     FALSE TRUE
##   1  2038  314
##   2  1674  518
##   3  2625  479
##   4  1126  182
```

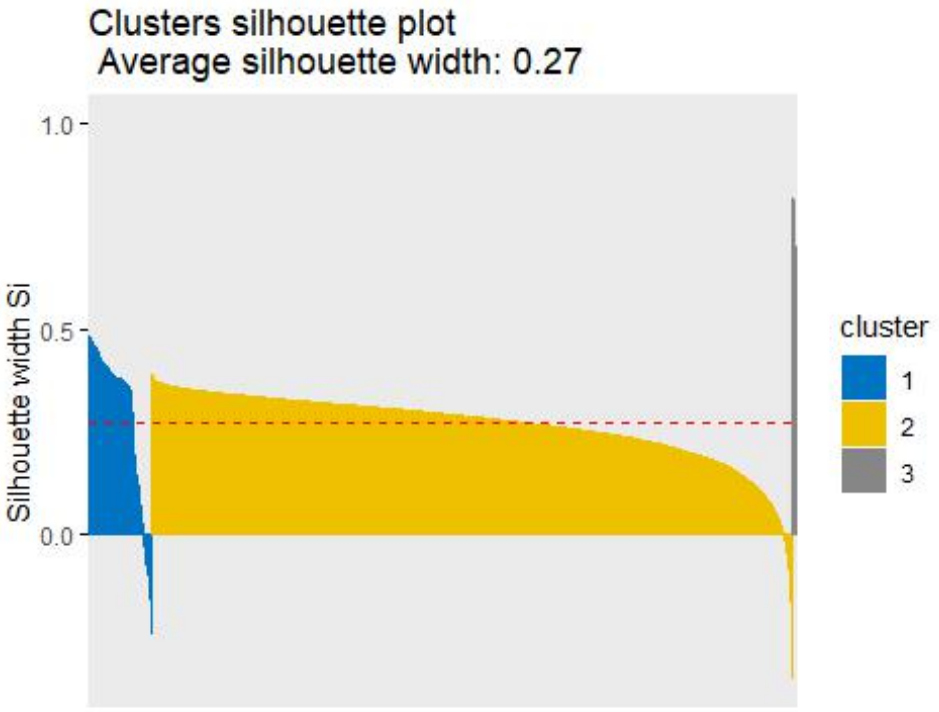
Hierarchical Clustering

```
#calculate manhattan distance

data2di <- dist(normalized, method = "man")

## We then hierarchical clustering using the complete method
data2hc <- hclust(data2di, method = "complete")
data2as <- cutree(data2hc, k = 3)

dend_data <- as.dendrogram(data2hc)
cc <- color_branches(dend_data, k=3)

options(repr.plot.width=40 ,repr.plot.height=20 )
plot(cc)
```




```
sil <- silhouette(data2as, data2di)

fviz_silhouette(sil,palette= "jco",ggtheme = theme_minimal())
```

```
##   cluster size ave.sil.width
## 1       1  801          0.29
## 2       2 8114          0.27
## 3       3   41          0.70
```



We could also do PCA (Principal component analysis) for dimensional reduction and then do k-means and hierarchical clustering

Summary

k-means may be computationally faster than hierarchical clustering if the number of clusters is small.

Hierarchical clustering is also easy to implement. Required less codes to implement

On the other hand, the result of a hierarchical clustering is a structure that is more informative and interpretable than the unstructured set of flat clusters returned by k-means. Therefore, it is easier to determine the optimal number of clusters by looking at the dendrogram of a hierarchical clustering than trying to predict this optimal number in advance in case of k-means.

High space and time complexity for Hierarchical clustering. Hence this clustering algorithm is difficult to use when we have huge data.

Follow up Questions

Are there more models to improve the getting of characteristics