

#1. Defining the question

1.1 Specifying the data analytic objective

1. Perform clustering stating insights drawn from your analysis and visualizations.
2. Upon implementation, provide comparisons between the approaches learned this week i.e. K-Means clustering vs Hierarchical clustering highlighting the strengths and limitations of each approach in the context of your analysis.

1.2 Defining the metric of success

Clustering the data to maximize reaching the target audience

1.3 Understanding the context

Kira Plastinina is a Russian brand that is sold through a defunct chain of retail stores in Russia, Ukraine, Kazakhstan, Belarus, China, Philippines, and Armenia. The brand's Sales and Marketing team would like to understand their customer's behavior from data that they have collected over the past year. More specifically, they would like to learn the characteristics of customer groups.

1.4 Recording the Experimental Design

1. Loading the data
2. Checking the data
3. Tidying the data
4. Univariate Analysis
5. Bivariate Analysis
6. Challenging the solution
7. Recommendations
8. Follow up questions

2. Loading the libraries

Installing packages that we have not.

```
#library(devtools)
#install_github("vqv/ggbiplot", force = TRUE)
#install.packages("DataExplorer")
#install.packages("Hmisc")
#install.packages("pastecs")
#install.packages("psych")
#install.packages("corrplot")
#install.packages("factoextra")
#install.packages("Rtsne")
#install.packages("caret")
```

Loading Libraries necessary

```
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages ----- tidyverse
## 1.3.0 --

## v ggplot2 3.3.3      v purrr  0.3.4
## v tibble  3.1.0      v dplyr  1.0.5
## v tidyr   1.1.3      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.1

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

library(magrittr)

##
## Attaching package: 'magrittr'

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

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

library(warn = -1)

library(ggbiplot)

## Loading required package: plyr

## -----
##
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first,
## then dplyr:
## library(plyr); library(dplyr)

## -----
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize

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

## Loading required package: scales

##
## Attaching package: 'scales'

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

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

## Loading required package: grid

library(RColorBrewer)
library(ggplot2)
library(lattice)
library(corrplot)

## Warning: package 'corrplot' was built under R version 4.0.5

## corrplot 0.84 loaded

library(DataExplorer)

## Warning: package 'DataExplorer' was built under R version 4.0.5

library(Hmisc)

## Warning: package 'Hmisc' was built under R version 4.0.5

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:plyr':
##
##   is.discrete, summarize
```

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

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

library(pastecs)

## Warning: package 'pastecs' was built under R version 4.0.5

##
## Attaching package: 'pastecs'

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

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

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

library(psych)

## Warning: package 'psych' was built under R version 4.0.5

##
## Attaching package: 'psych'

## The following object is masked from 'package:Hmisc':
##
##   describe

## The following objects are masked from 'package:scales':
##
##   alpha, rescale

## The following objects are masked from 'package:ggplot2':
##
##   %+%, alpha

library(factoextra)

## Warning: package 'factoextra' was built under R version 4.0.5

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

library(Rtsne)
```

```
## Warning: package 'Rtsne' was built under R version 4.0.5

library(caret)

## Warning: package 'caret' was built under R version 4.0.5

##
## Attaching package: 'caret'

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

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

3. Loading Data

```
customers = read_csv('http://bit.ly/EcommerceCustomersDataset')

##
## -- Column specification -----
## -----
## 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()
## )

head(customers)

## # A tibble: 6 x 18
##   Administrative Administrative_D~ Informational Informational_D~
##   ProductRelated
##           <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>, BounceRates
<dbl>,
## #   ExitRates <dbl>, PageValues <dbl>, SpecialDay <dbl>, Month <chr>,
## #   OperatingSystems <dbl>, Browser <dbl>, Region <dbl>, TrafficType
<dbl>,
## #   VisitorType <chr>, Weekend <lgl>, Revenue <lgl>

tail(customers)

## # A tibble: 6 x 18
##   Administrative Administrative_D~ Informational Informational_D~
ProductRelated
##           <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>, BounceRates
<dbl>,
## #   ExitRates <dbl>, PageValues <dbl>, SpecialDay <dbl>, Month <chr>,
## #   OperatingSystems <dbl>, Browser <dbl>, Region <dbl>, TrafficType
<dbl>,
## #   VisitorType <chr>, Weekend <lgl>, Revenue <lgl>

```

structure of the data

```
str(customers)
```

```
## spec_tbl_df[,18] [12,330 x 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"
"Returning_Visitor" "Returning_Visitor" "Returning_Visitor" ...
## $ Weekend             : logi [1:12330] FALSE FALSE FALSE FALSE TRUE
FALSE ...
## $ Revenue             : logi [1:12330] FALSE FALSE FALSE FALSE FALSE
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()
## .. )
```

The data has 12330 obs. of 18 variables

Checking for the summary description of our data

Checking for the summary description of our data

```
summary(customers)
```

```
## 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. DATA CLEANING

```
colSums(is.na(customers))
```

```
## Administrative Administrative_Duration Informational
## 14 14 14
## Informational_Duration ProductRelated ProductRelated_Duration
## 14 14 14
## BounceRates ExitRates PageValues
```



```
##           14           14           0
##      SpecialDay      Month      OperatingSystems
##           0           0           0
##      Browser      Region      TrafficType
##           0           0           0
##      VisitorType      Weekend      Revenue
##           0           0           0
```

We have missing values in 8 columns. Since we have quite a number of rows, we will go ahead and drop these missing values as we will be left with enough data for our analysis

creating a new data frame that does not have missing values

```
customers1 <- na.omit(customers)
head(customers1)

## # A tibble: 6 x 18
##   Administrative Administrative_D~ Informational Informational_D~
##   ProductRelated
##           <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>, BounceRates
<dbl>,
## #   ExitRates <dbl>, PageValues <dbl>, SpecialDay <dbl>, Month <chr>,
## #   OperatingSystems <dbl>, Browser <dbl>, Region <dbl>, TrafficType
<dbl>,
## #   VisitorType <chr>, Weekend <lgl>, Revenue <lgl>
```

Confirming that we have no null values

```
sum(colSums(is.na(customers1)))

## [1] 0
```

Checking for Duplicates

```
customer <- customers1[duplicated(customers1),]
dim(customer)
```

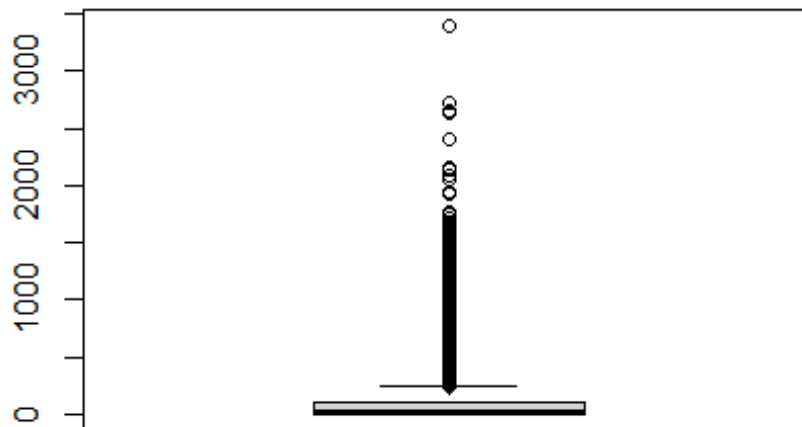
```
## [1] 117  18

# Removing these duplicated rows in the dataset
cust <- customers1[!duplicated(customers1), ]
dim(cust)

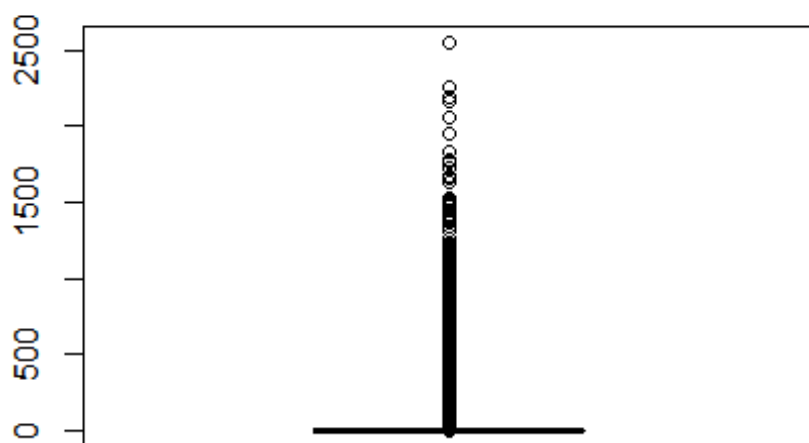
## [1] 12199  18
```

Checking for outliers

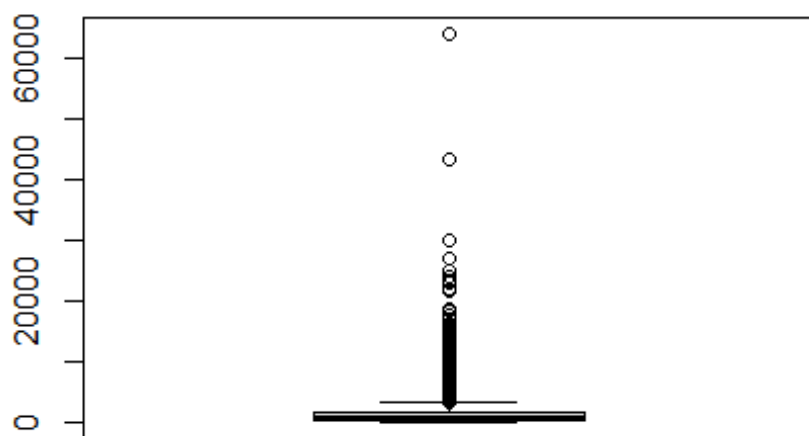
```
# Administrative_Duration
# Plot a boxplot to help us visualise any existing outliers
boxplot(cust$Administrative_Duration)
```



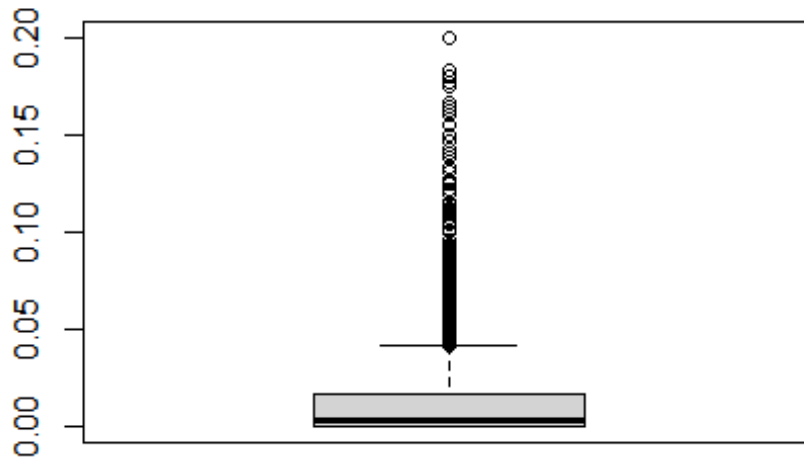
```
# Informational_Duration
boxplot(cust$Informational_Duration)
```



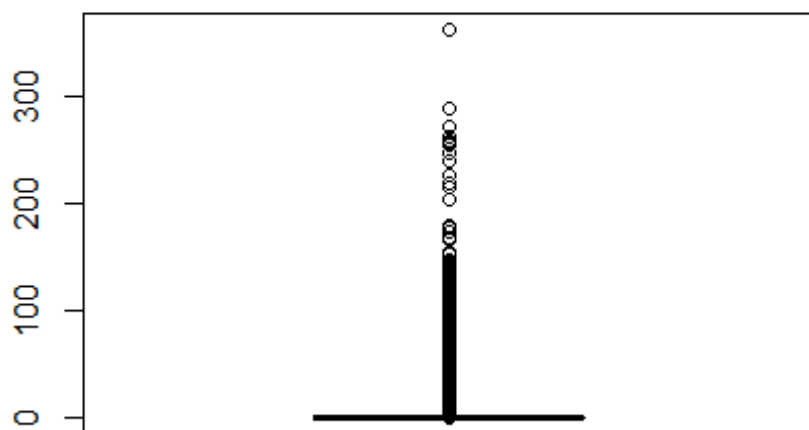
```
# ProductRelated_Duration  
boxplot(cust$ProductRelated_Duration)
```



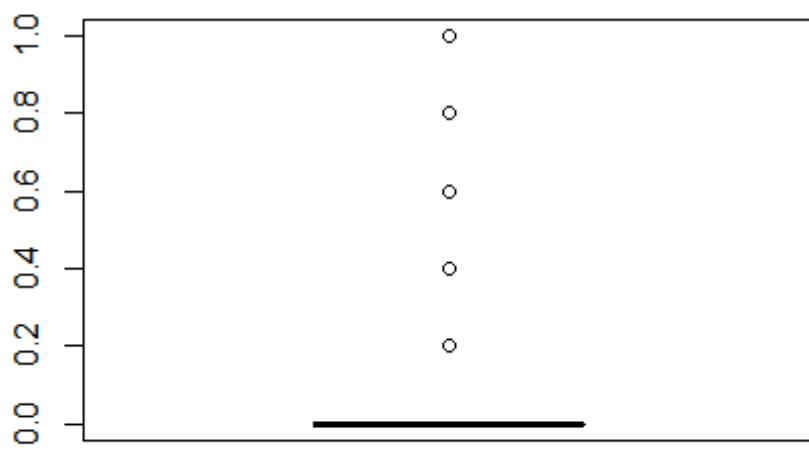
```
# BounceRates  
boxplot(cust$BounceRates)
```



```
# PageValues  
boxplot(cust$PageValues)
```



```
# SpecialDay  
boxplot(cust$SpecialDay)
```



We have outliers in several of our numerical columns. We shall not delete the outliers as they will result in us losing so much customer data which could alter our analysis.

5. Exploratory Data Analysis

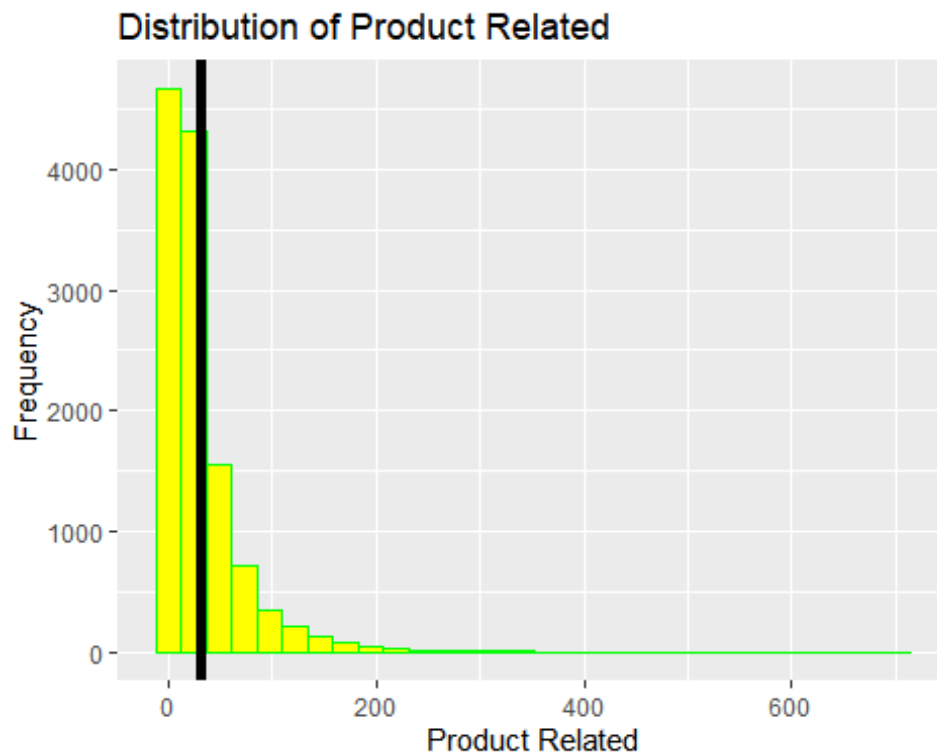
Univariate Analysis

```
# Plotting a histogram using ggplots
```

```
#  
#
```

```
cust %>%  
  ggplot(aes(ProductRelated)) +  
  geom_histogram(color = "Green", fill = "yellow") +  
  geom_vline(xintercept = mean(cust$ProductRelated), lwd = 2) +  
  labs(title = "Distribution of Product Related",  
       x = "Product Related",  
       y = "Frequency")
```

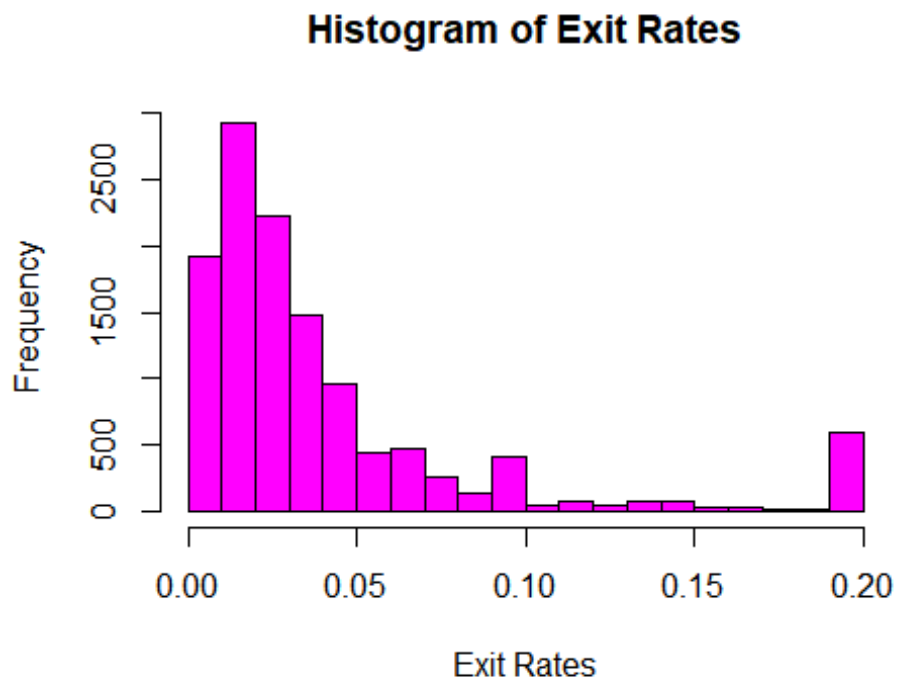
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



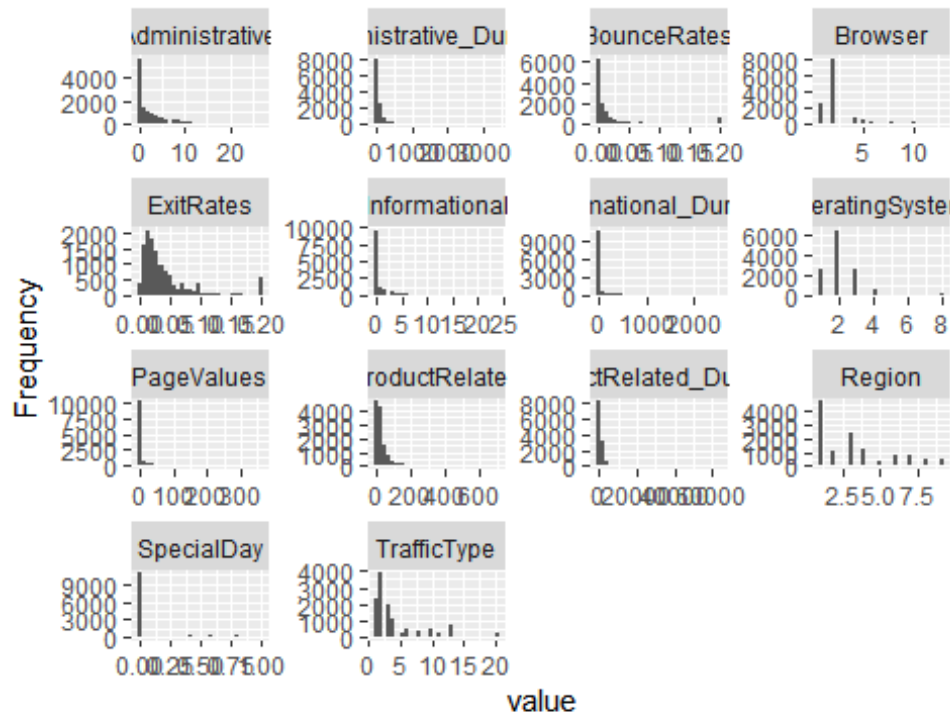
```
# plotting a histogram of Exit Rates
```

```
hist(cust$ExitRates,  
     main = "Histogram of Exit Rates",
```

```
xlab = "Exit Rates",  
col = "magenta")
```



```
# Plotting all histograms in the continuous variables in our data  
plot_histogram(cust)
```



From the histograms, most of our variables are positively skewed.

```
# Bar plots of the categorical/factor modes variables

#par(mfrow=c(4,1))

#for(i in 11:16) {

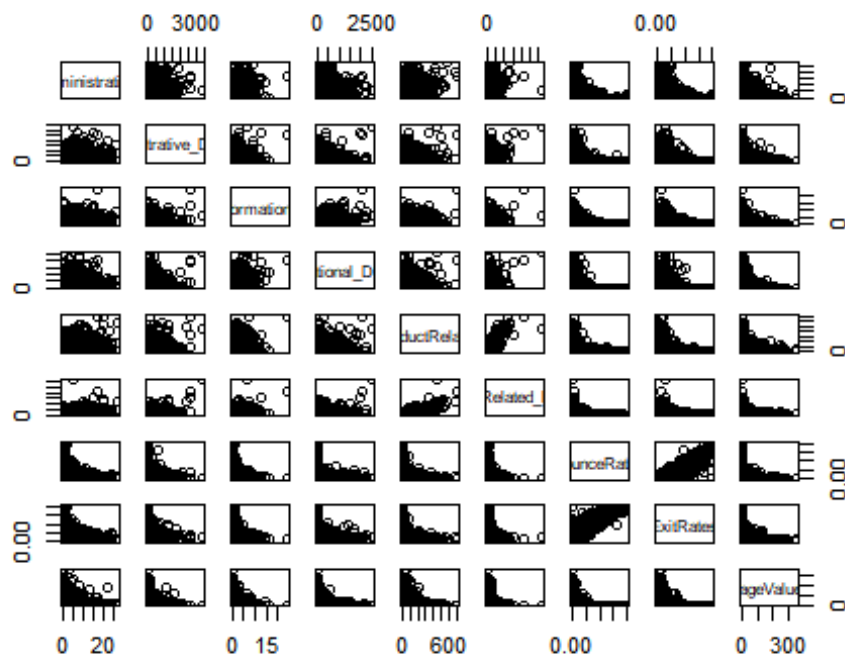
#   counts <- table(cust[,i])

#   name <- names(cust)[i]
#   barplot(counts, main=name, col = heat.colors(20))}
```

May and November were busy months receiving high traffic, Feb received the least traffic of customers. Most visitors were returning type. Traffic mode number 2, 1 and 3 were heavily used in that order. Region number 1 had the most activity, region 5 was less active. Browser 2 and 1 were the most commonly used for browsing. Operating systems 2, 1 and 3 were mostly used by customers.

Bivariate Analysis

```
# Pair plots for the continuous variables
pairs(cust[,1:9])
```

Correlations

subsetting our data frame to get the numeric variables
 numerics <- cust[, c(1:10)]

getting the correlation between these numeric variables
 numerics.cor <- cor(numerics)
 numerics.cor

##	Administrative	Administrative_Duration	
Informational			
## Administrative	1.00000000	0.60040965	
0.37528761			
## Administrative_Duration	0.60040965	1.00000000	
0.30143630			
## Informational	0.37528761	0.30143630	
1.00000000			
## Informational_Duration	0.25478602	0.23718986	
0.61867795			
## ProductRelated	0.42819151	0.28678391	
0.37260472			
## ProductRelated_Duration	0.37102722	0.35351379	
0.38608372			
## BounceRates	-0.21366664	-0.13733340	-
0.10950530			
## ExitRates	-0.31127413	-0.20202445	-
0.15956681			

```

## PageValues          0.09692097          0.06616837
0.04739015
## SpecialDay          -0.09707210          -0.07473689    -
0.04937677
##                      Informational_Duration ProductRelated
## Administrative      0.25478602          0.42819151
## Administrative_Duration 0.23718986          0.28678391
## Informational        0.61867795          0.37260472
## Informational_Duration 1.00000000          0.27906195
## ProductRelated      0.27906195          1.00000000
## ProductRelated_Duration 0.34658069          0.86030819
## BounceRates         -0.07015947          -0.19351577
## ExitRates           -0.10293268          -0.28616321
## PageValues          0.03006416          0.05411549
## SpecialDay          -0.03129304          -0.02593062
##                      ProductRelated_Duration BounceRates ExitRates
## Administrative      0.37102722 -0.21366664 -0.3112741
## Administrative_Duration 0.35351379 -0.13733340 -0.2020245
## Informational        0.38608372 -0.10950530 -0.1595668
## Informational_Duration 0.34658069 -0.07015947 -0.1029327
## ProductRelated      0.86030819 -0.19351577 -0.2861632
## ProductRelated_Duration 1.00000000 -0.17437550 -0.2453340
## BounceRates         -0.17437550  1.00000000  0.9033582
## ExitRates           -0.24533401  0.90335819  1.0000000
## PageValues          0.05084062 -0.11599198 -0.1735715
## SpecialDay          -0.03821065  0.08783999  0.1167838
##                      PageValues SpecialDay
## Administrative      0.09692097 -0.09707210
## Administrative_Duration 0.06616837 -0.07473689
## Informational        0.04739015 -0.04937677
## Informational_Duration 0.03006416 -0.03129304
## ProductRelated      0.05411549 -0.02593062
## ProductRelated_Duration 0.05084062 -0.03821065
## BounceRates         -0.11599198  0.08783999
## ExitRates           -0.17357154  0.11678376
## PageValues          1.00000000 -0.06453271
## SpecialDay          -0.06453271  1.00000000

```

installing packages that we shall use to plot the correlation plots

```
install.packages("Hmisc")
```

```
## Warning: package 'Hmisc' is in use and will not be installed
```

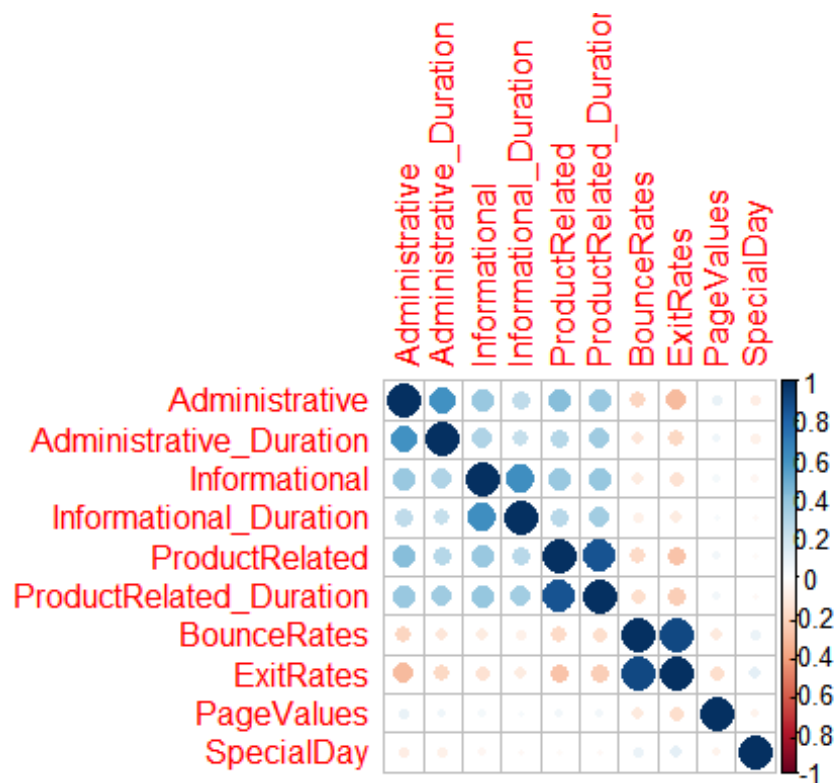
```
library("Hmisc")
```

```
install.packages("corrplot")
```

```
## Warning: package 'corrplot' is in use and will not be installed
```

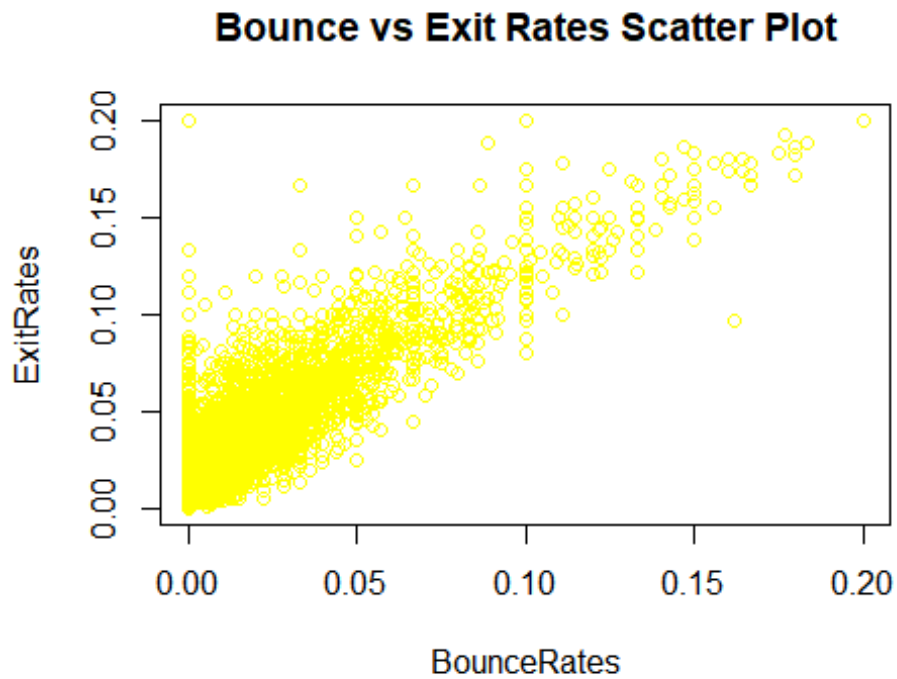
```
library(corrplot)

corrplot(numerics.cor)
```



```
# Plotting a scatter plot
```

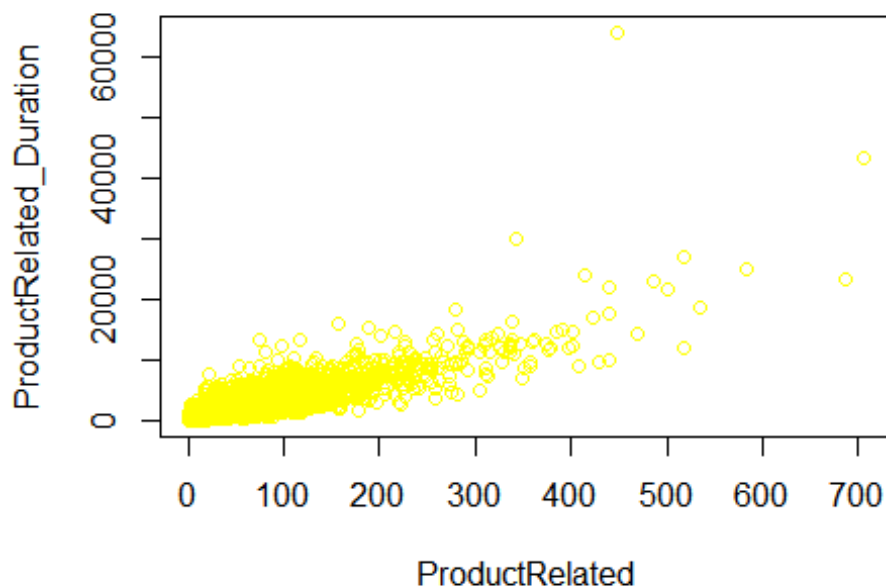
```
plot(ExitRates ~ BounceRates, data = cust,
     col = "yellow",
     main = "Bounce vs Exit Rates Scatter Plot")
```



There is a strong positive correlation between bounce rates and exit rates

```
plot(ProductRelated_Duration ~ ProductRelated, data = cust,  
      col = "yellow",  
      main = "Product related vs Product related durations Scatter Plot")
```

Product related vs Product related durations Scatter



There is a positive correlation between product relation and product relation duration

K-MEAN CLUSTERING

converting some of our columns into numerical data types by one hot encoding

```
dmy = dummyVars(" ~ .", data = cust)
```

```
df4 = data.frame(predict(dmy, newdata = cust))
```

Checking the data types of each attribute

```
sapply(df4, class)
```

##	Administrative	Administrative_Duration
##	"numeric"	"numeric"
##	Informational	Informational_Duration
##	"numeric"	"numeric"
##	ProductRelated	ProductRelated_Duration
##	"numeric"	"numeric"
##	BounceRates	ExitRates
##	"numeric"	"numeric"
##	PageValues	SpecialDay
##	"numeric"	"numeric"
##	MonthAug	MonthDec
##	"numeric"	"numeric"
##	MonthFeb	MonthJul

```
##          "numeric"          "numeric"
##      MonthJune          MonthMar
##          "numeric"          "numeric"
##      MonthMay          MonthNov
##          "numeric"          "numeric"
##      MonthOct          MonthSep
##          "numeric"          "numeric"
##      OperatingSystems      Browser
##          "numeric"          "numeric"
##          Region          TrafficType
##          "numeric"          "numeric"
##      VisitorTypeNew_Visitor      VisitorTypeOther
##          "numeric"          "numeric"
##      VisitorTypeReturning_Visitor      WeekendFALSE
##          "numeric"          "numeric"
##      WeekendTRUE          RevenueFALSE
##          "numeric"          "numeric"
##      RevenueTRUE
##          "numeric"
```

Confirming changes

```
glimpse(df4)
```

```
## Rows: 12,199
## Columns: 31
## $ Administrative      <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
## 0, 0,~
## $ Administrative_Duration <dbl> 0, 0, -1, 0, 0, 0, -1, -1, 0, 0, 0,
## 0, 0,~
## $ Informational      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## 0, 0,~
## $ Informational_Duration <dbl> 0, 0, -1, 0, 0, 0, -1, -1, 0, 0, 0,
## 0, 0,~
## $ ProductRelated      <dbl> 1, 2, 1, 2, 10, 19, 1, 1, 2, 3, 3,
## 16, 7,~
## $ ProductRelated_Duration <dbl> 0.000000, 64.000000, -1.000000,
## 2.666667,~
## $ BounceRates      <dbl> 0.2000000000, 0.0000000000,
## 0.2000000000, 0.~
## $ ExitRates      <dbl> 0.2000000000, 0.1000000000,
## 0.2000000000, 0.~
## $ PageValues      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
## 0, 0,~
## $ SpecialDay      <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.4,
## 0.0, 0~
## $ MonthAug      <dbl> 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,~
## $ MonthFeb      <dbl> 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,~
## $ MonthJune <dbl> 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,~
## $ MonthMay <dbl> 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,~
## $ MonthOct <dbl> 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,~
## $ OperatingSystems <dbl> 1, 2, 4, 3, 3, 2, 2, 1, 2, 2, 1, 1,
1, 2,~
## $ Browser <dbl> 1, 2, 1, 2, 3, 2, 4, 2, 2, 4, 1, 1,
1, 5,~
## $ Region <dbl> 1, 1, 9, 2, 1, 1, 3, 1, 2, 1, 3, 4,
1, 1,~
## $ TrafficType <dbl> 1, 2, 3, 4, 4, 3, 3, 5, 3, 2, 3, 3,
3, 3,~
## $ VisitorTypeNew_Visitor <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,~
## $ VisitorTypeOther <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,~
## $ VisitorTypeReturning_Visitor <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1,~
## $ WeekendFALSE <dbl> 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1,
1, 1,~
## $ WeekendTRUE <dbl> 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,
0, 0,~
## $ RevenueFALSE <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1,~
## $ RevenueTRUE <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,~

```

*# We are instructed to use Revenue as the class Label,
Hence we will remove it and store it in another variable*

```

df4_copy <- df4[, -c(30:31)]
cust.class<- cust[, "Revenue"]

df4_copy_copy <- df4[, -c(30,31)]

# Previewing the class column

head (cust.class)

```

```
## # A tibble: 6 x 1
##   Revenue
##   <lgl>
## 1 FALSE
## 2 FALSE
## 3 FALSE
## 4 FALSE
## 5 FALSE
## 6 FALSE
```

```
# Previewing the copy dataset with dummies
head(df4_copy)
```

```
##   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 MonthAug MonthDec MonthFeb MonthJul MonthJune MonthMar
MonthMay
## 1           0           0           0           1           0           0
0
## 2           0           0           0           1           0           0
0
## 3           0           0           0           1           0           0
0
## 4           0           0           0           1           0           0
0
## 5           0           0           0           1           0           0
0
## 6           0           0           0           1           0           0
0
##   MonthNov MonthOct MonthSep OperatingSystems Browser Region TrafficType
## 1           0           0           0           1           1           1
1
```



```
## 2      0      0      0      2      2      1      2
## 3      0      0      0      4      1      9      3
## 4      0      0      0      3      2      2      4
## 5      0      0      0      3      3      1      4
## 6      0      0      0      2      2      1      3
## VisitorTypeNew_Visitor VisitorTypeOther VisitorTypeReturning_Visitor
## 1      0      0      1
## 2      0      0      1
## 3      0      0      1
## 4      0      0      1
## 5      0      0      1
## 6      0      0      1
## WeekendFALSE WeekendTRUE
## 1      1      0
## 2      1      0
## 3      1      0
## 4      1      0
## 5      0      1
## 6      1      0
```

Normalizing the a copy of the original data

```
df_norm <- as.data.frame(apply(df4_copy, 2, function(x) (x - min(x))/(max(x)-min(x))))
```

Applying K-Means Clustering algorithm

Using 3 centroids as K=3

```
result <- kmeans(df_norm, 10)
```

Previewing the number of records in each cluster

```
result$size
```

```
## [1] 1165 1949 2118 165 2192 1768 714 507 422 1199
```

Viewing the cluster center data points by each attribute

```
result$centers
```

```
## Administrative Administrative_Duration Informational
Informational_Duration
## 1 0.0684469878 0.0210925574 0.0180972818
0.0131467053
## 2 0.0946544287 0.0263836156 0.0236018471
0.0171064737
## 3 0.0951631518 0.0264185030 0.0256728045
0.0174008444
## 4 0.0006734007 0.0003137486 0.0015151515
0.0003849701
## 5 0.0733306299 0.0208918777 0.0182291667
```

```

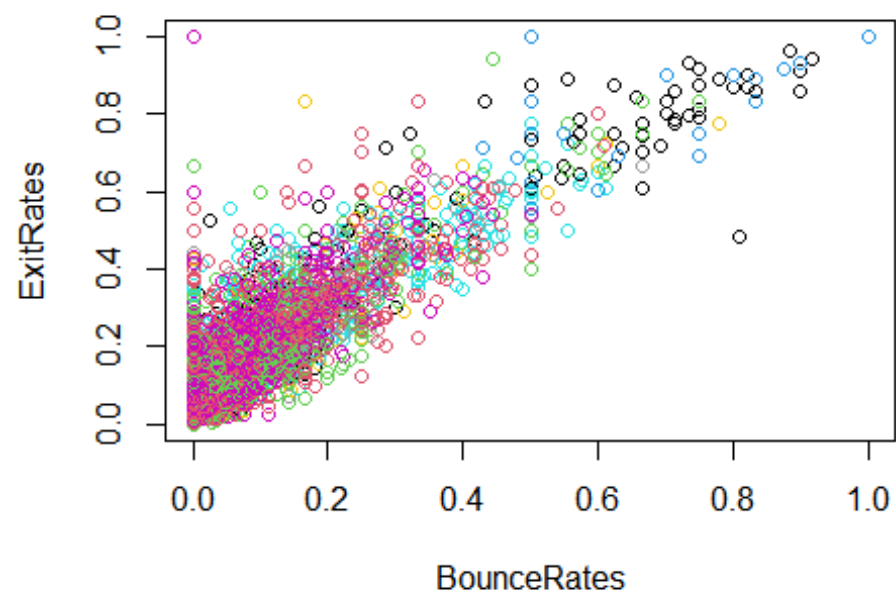
0.0112642652
## 6 0.1065233786 0.0294866865 0.0294117647
0.0185956406
## 7 0.0858491545 0.0239718096 0.0218253968
0.0139072327
## 8 0.1300314121 0.0324702143 0.0247370151
0.0168324044
## 9 0.0008776549 0.0003273868 0.0001974724
0.0003697997
## 10 0.0956661415 0.0275529126 0.0139699750
0.0080522520
## ProductRelated ProductRelated_Duration BounceRates ExitRates
PageValues
## 1 0.030198764 0.0140726725 0.05199285 0.18584136
0.009093501
## 2 0.051603113 0.0217891088 0.05554100 0.17315715
0.016814774
## 3 0.049094891 0.0200242048 0.07299160 0.16957559
0.017334502
## 4 0.002699334 0.0007492500 0.92219617 0.95609447
0.000000000
## 5 0.040694078 0.0167713765 0.07504985 0.19937798
0.012855139
## 6 0.074741664 0.0317968108 0.04881432 0.15342458
0.018541801
## 7 0.042467767 0.0172678189 0.09917428 0.20233652
0.018464178
## 8 0.059647209 0.0224895266 0.06696784 0.17272254
0.016314769
## 9 0.003465430 0.0004198304 0.94181813 0.96465536
0.000000000
## 10 0.025996841 0.0101417848 0.01546754 0.09405906
0.031883153
## SpecialDay MonthAug MonthDec MonthFeb MonthJul MonthJune
## 1 0.000000000 0.00000000 0.0000000 0.0000000000 0.00000000 0.00000000
## 2 0.014161108 0.14109800 0.5259107 0.0636223704 0.13648025 0.00000000
## 3 0.004627007 0.04532578 0.1704438 0.0132200189 0.04910293 0.02219075
## 4 0.000000000 0.00000000 0.0000000 0.0000000000 0.00000000 0.00000000
## 5 0.223631387 0.00000000 0.0000000 0.0000000000 0.00000000 0.00000000
## 6 0.000000000 0.00000000 0.0000000 0.0000000000 0.00000000 0.00000000
## 7 0.208683473 0.00000000 0.0000000 0.0000000000 0.00000000 0.00000000
## 8 0.000000000 0.00000000 0.0000000 0.0000000000 0.00000000 0.37475345
## 9 0.127488152 0.03317536 0.1350711 0.0687203791 0.04976303 0.05687204
## 10 0.021351126 0.04003336 0.2193495 0.0008340284 0.03419516 0.02001668
## MonthMar MonthMay MonthNov MonthOct MonthSep OperatingSystems
## 1 1.00000000 0.00000000 0.0000000 0.000000000 0.00000000 0.1543838
## 2 0.00000000 0.00000000 0.0000000 0.000000000 0.13288866 0.1704171
## 3 0.2242682 0.0000000 0.3621341 0.067988669 0.04532578 0.1634966
## 4 0.00000000 0.0000000 1.0000000 0.000000000 0.00000000 0.1991342
## 5 0.00000000 1.0000000 0.0000000 0.000000000 0.00000000 0.1629953

```

```
## 6 0.0000000 0.0000000 1.0000000 0.000000000 0.00000000 0.1588559
## 7 0.0000000 1.0000000 0.0000000 0.000000000 0.00000000 0.1520608
## 8 0.0000000 0.0000000 0.0000000 0.625246548 0.00000000 0.1597633
## 9 0.1706161 0.4597156 0.0000000 0.009478673 0.01658768 0.1668923
## 10 0.1175980 0.1901585 0.2360300 0.070058382 0.07172644 0.1419040
##      Browser      Region TrafficType VisitorTypeNew_Visitor VisitorTypeOther
## 1 0.1135908 0.2653433 0.1040434 0.00000000 0.000000000
## 2 0.1336155 0.2948307 0.1531690 0.00000000 0.025141098
## 3 0.1017863 0.2638692 0.1539685 0.18555241 0.002832861
## 4 0.1227273 0.2886364 0.2822967 0.06060606 0.042424242
## 5 0.1173206 0.2616902 0.1849068 0.00000000 0.000000000
## 6 0.1050151 0.2451923 0.1696535 0.00000000 0.007918552
## 7 0.1045752 0.2794118 0.1744066 0.11904762 0.000000000
## 8 0.1050296 0.2573964 0.1708710 0.00000000 0.001972387
## 9 0.1149289 0.2787322 0.2055375 0.01421801 0.009478673
## 10 0.1107868 0.2823186 0.1488521 1.00000000 0.000000000
##      VisitorTypeReturning_Visitor WeekendFALSE WeekendTRUE
## 1 1.0000000 1.0000000 0.0000000
## 2 0.9748589 1.0000000 0.0000000
## 3 0.8116147 0.0000000 1.0000000
## 4 0.8969697 0.8545455 0.1454545
## 5 1.0000000 1.0000000 0.0000000
## 6 0.9920814 1.0000000 0.0000000
## 7 0.8809524 0.0000000 1.0000000
## 8 0.9980276 1.0000000 0.0000000
## 9 0.9763033 1.0000000 0.0000000
## 10 0.0000000 1.0000000 0.0000000
```

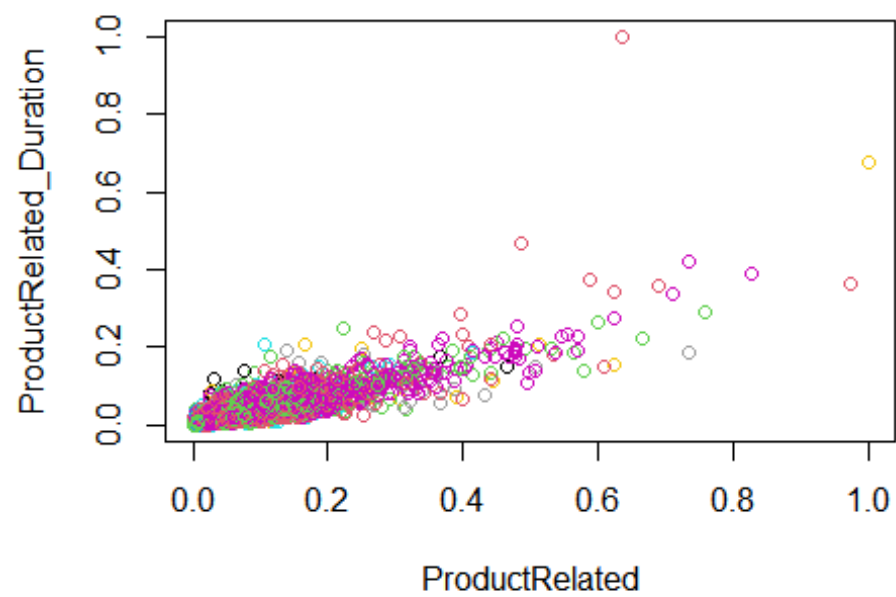
Plotting to see how exit rates and bounce rates data points have been distributed in clusters

```
plot(df_norm[c(7,8)], col = result$cluster)
```



```
# Product Related, vs Product Related Duration
```

```
plot(df_norm[, 5:6], col = result$cluster)
```



Hierarchical Clustering

```
# We use R function hclust()
# For hierarchical clustering
# First we use the dist() to compute the Euclidean distance btwn obs
# d will be the first argument in the hclust() dissimilarity matrix
#

d <- dist(df_norm, method = "euclidean")

# We then apply hierarchical clustering using the Ward's method

res.hc <- hclust(d, method = "ward.D2")

# Lastly we plot the obtained dendrogram
#--

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

