Ecommerce

Problem Definition

Kira Plastinina (Links to an external site.) 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.

Context

Knowing your customer is key for any business endeavor. Successful business owners understand what their customers want and the most effective way of making their product or service available. The depth of knowledge is also crucial – it requires knowing more than their names, ages and incomes. As a business owner, knowing your customer's hobbies, tastes and interests along with what they watch, listen to and read can be a profitable advantage.

Understanding your customer's buying behavior is also very important. As a business owner, you need to comprehend what type of person is most likely to need or want the product or service you provide.

Data Sourcing

The dataset was got from http://bit.ly/EcommerceCustomersDataset.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.

Installing and Loading the Packages

```
install.packages("plyr")
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
install.packages("dplyr")
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
install.packages("tidyr")
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
install.packages("DataExplorer")
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
install.packages("lubridate")
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
install.packages("ggbiplot")
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)

## Warning: package 'ggbiplot' is not available (for R version 3.6.3)
```

```
install.packages("devtools")
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)

## also installing the dependencies 'credentials', 'curl', 'gert', 'gh', 'openssl', 'xml2', 'usethis',

## Warning in install.packages("devtools"): installation of package 'curl' had non-
## zero exit status

## Warning in install.packages("devtools"): installation of package 'openssl' had
## non-zero exit status

## Warning in install.packages("devtools"): installation of package 'xml2' had non-
## zero exit status

## Warning in install.packages("devtools"): installation of package 'credentials'
## had non-zero exit status

## Warning in install.packages("devtools"): installation of package 'httr' had non-
## zero exit status
```

```
## Warning in install.packages("devtools"): installation of package 'roxygen2' had
## non-zero exit status

## Warning in install.packages("devtools"): installation of package 'rversions' had
## non-zero exit status

## Warning in install.packages("devtools"): installation of package 'gert' had non-
## zero exit status

## Warning in install.packages("devtools"): installation of package 'gh' had non-
## zero exit status

## Warning in install.packages("devtools"): installation of package 'usethis' had
## non-zero exit status

## Warning in install.packages("devtools"): installation of package 'devtools' had
## non-zero exit status
```

```
install.packages("Rtsne")
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
library(data.table)
library(dplyr)
```

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

## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

```
library(ggplot2)
library(tidyr)
library(DataExplorer)
tinytex::install_tinytex()
```

```
## tlmgr option sys_bin ~/bin
```

Data EXploration

```
df<-read.csv('http://bit.ly/EcommerceCustomersDataset')
head(df)
```

```
##   Administrative Administrative_Duration Informational Informational_Duration
## 1              0                       0             0                      0
## 2              0                       0             0                      0
## 3              0                      -1             0                     -1
## 4              0                       0             0                      0
## 5              0                       0             0                      0
## 6              0                       0             0                      0
##   ProductRelated ProductRelated_Duration BounceRates ExitRates PageValues
## 1              1                0.000000  0.20000000 0.2000000          0
## 2              2               64.000000  0.00000000 0.1000000          0
## 3              1               -1.000000  0.20000000 0.2000000          0
## 4              2                2.666667  0.05000000 0.1400000          0
## 5             10              627.500000  0.02000000 0.0500000          0
## 6             19              154.216667  0.01578947 0.0245614          0
##   SpecialDay Month OperatingSystems Browser Region TrafficType
## 1          0   Feb                1       1      1           1
## 2          0   Feb                2       2      1           2
## 3          0   Feb                4       1      9           3
## 4          0   Feb                3       2      2           4
## 5          0   Feb                3       3      1           4
## 6          0   Feb                2       2      1           3
##         VisitorType Weekend Revenue
## 1 Returning_Visitor   FALSE   FALSE
## 2 Returning_Visitor   FALSE   FALSE
## 3 Returning_Visitor   FALSE   FALSE
## 4 Returning_Visitor   FALSE   FALSE
## 5 Returning_Visitor    TRUE   FALSE
## 6 Returning_Visitor   FALSE   FALSE
```

```
dim(df)
```

I'll then check the dimensions of the dataset

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

The dataset has 12,330 rows and 18 columns

```
colnames(df)
```

I'll then check te column names on the dataset

```
##  [1] "Administrative"          "Administrative_Duration"
```

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

Data Cleaning

```
str(df)
```

I'll then dispaly the internal structure of the dataset

```
## 'data.frame':    12330 obs. of  18 variables:
##  $ Administrative         : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Administrative_Duration: num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ Informational          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration : num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ ProductRelated         : int  1 2 1 2 10 19 1 1 2 3 ...
##  $ ProductRelated_Duration: num  0 64 -1 2.67 627.5 ...
##  $ BounceRates            : num  0.2 0 0.2 0.05 0.02 ...
##  $ ExitRates              : num  0.2 0.1 0.2 0.14 0.05 ...
##  $ PageValues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialDay             : num  0 0 0 0 0 0 0.4 0 0.8 0.4 ...
##  $ Month                  : Factor w/ 10 levels "Aug","Dec","Feb",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ OperatingSystems       : int  1 2 4 3 3 2 2 1 2 2 ...
##  $ Browser                : int  1 2 1 2 3 2 4 2 2 4 ...
##  $ Region                 : int  1 1 9 2 1 1 3 1 2 1 ...
##  $ TrafficType            : int  1 2 3 4 4 3 3 5 3 2 ...
##  $ VisitorType            : Factor w/ 3 levels "New_Visitor",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Weekend                : logi  FALSE FALSE FALSE FALSE TRUE FALSE ...
##  $ Revenue                : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

```
sum(is.na(df))
```

I'll then check for null values

```
## [1] 112
```

THe output shows their is 112 null values in the dataset.

```
colSums(is.na(df))
```

I'll then check which columns have the missing values.

```
##          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
```

```
df<-na.omit(df)
```

I'll then drop the missing values

```
sum(is.na(df))
```

I'll then confirm if they have been dropped

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

The output shows they have been dropped

```
colnames(df) <- tolower(colnames(df))
colnames(df)
```

I'll then convert the column names to lower case.

```
##  [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"
```

They have been converted

```
sum(duplicated(df))
```

I'll then check for duplicates in the dataset

```
## [1] 117
```

There is 117 duplicated values.I'll start dealing with them

```
df <- df[!duplicated(df,),]
```

```
sum(duplicated(df))
```

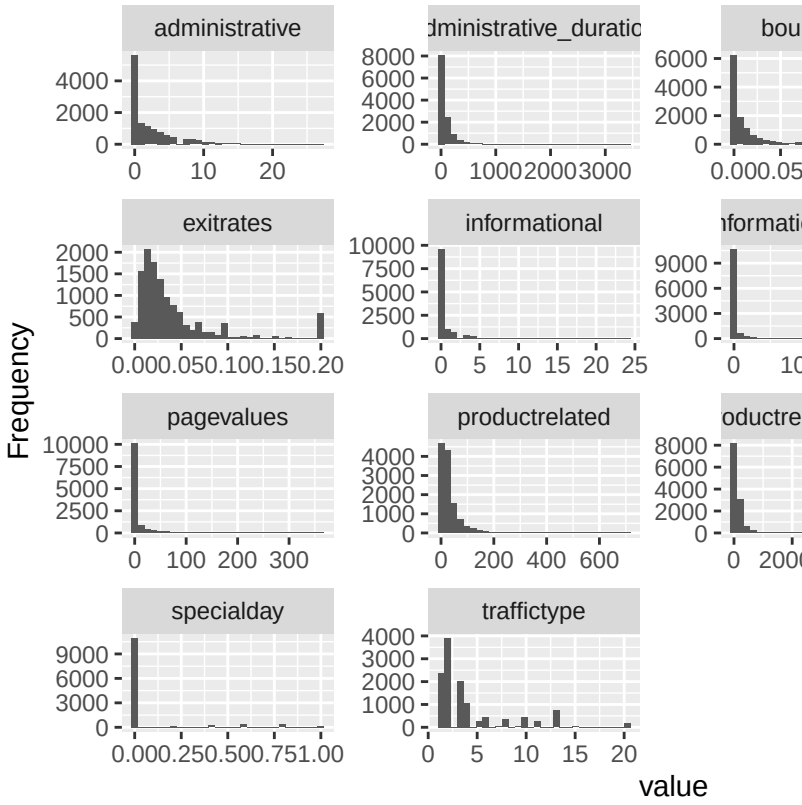
I'll then drop the duplicates in the dataset

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

The duplicated values have been dropped

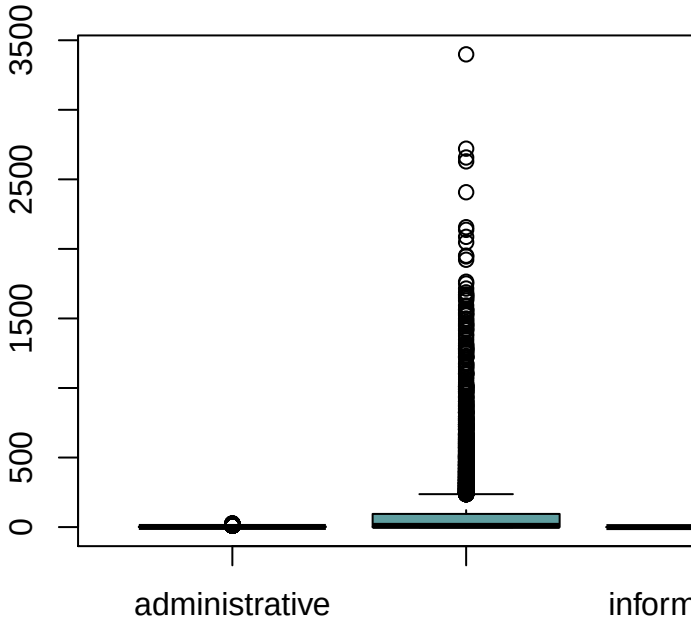
Univariate,Bivariate and Multivariate analysis

```
plot_histogram(df)
```



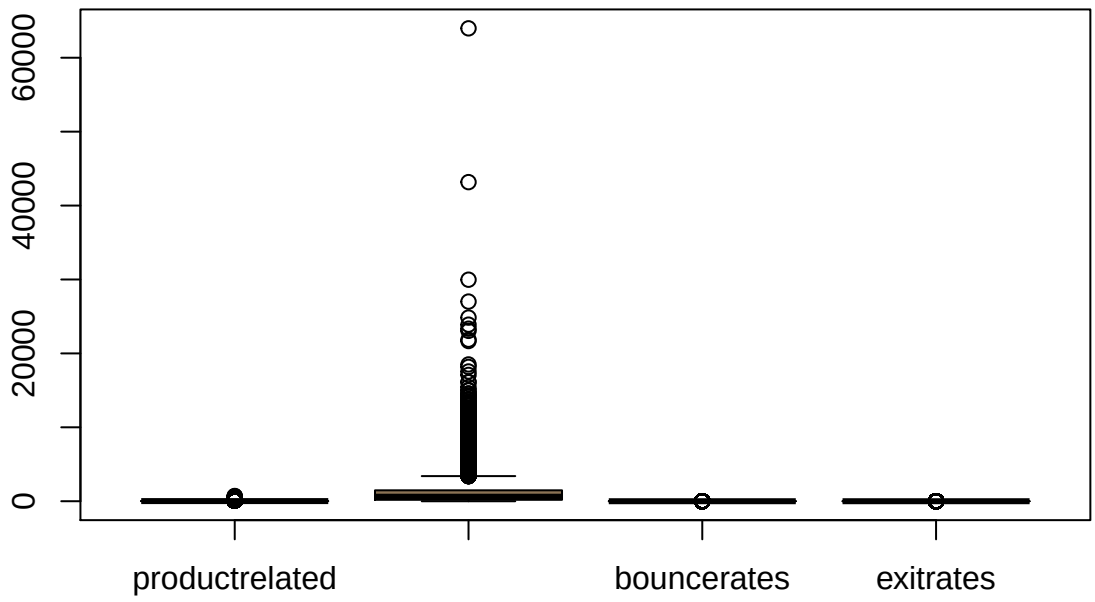
I'll then plot a distribition plot of the variables

```
options(repr.plot.width=10, repr.plot.height=5)
boxplot(df[, c(1:4)], col="cadetblue")
```

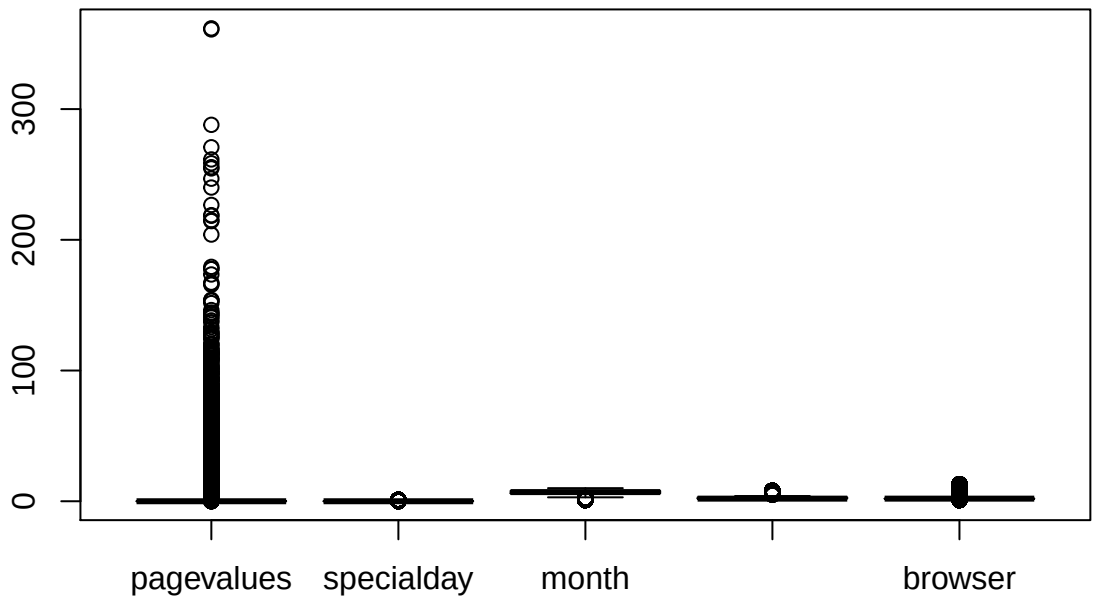


I'll then check for outliers of some of the variables

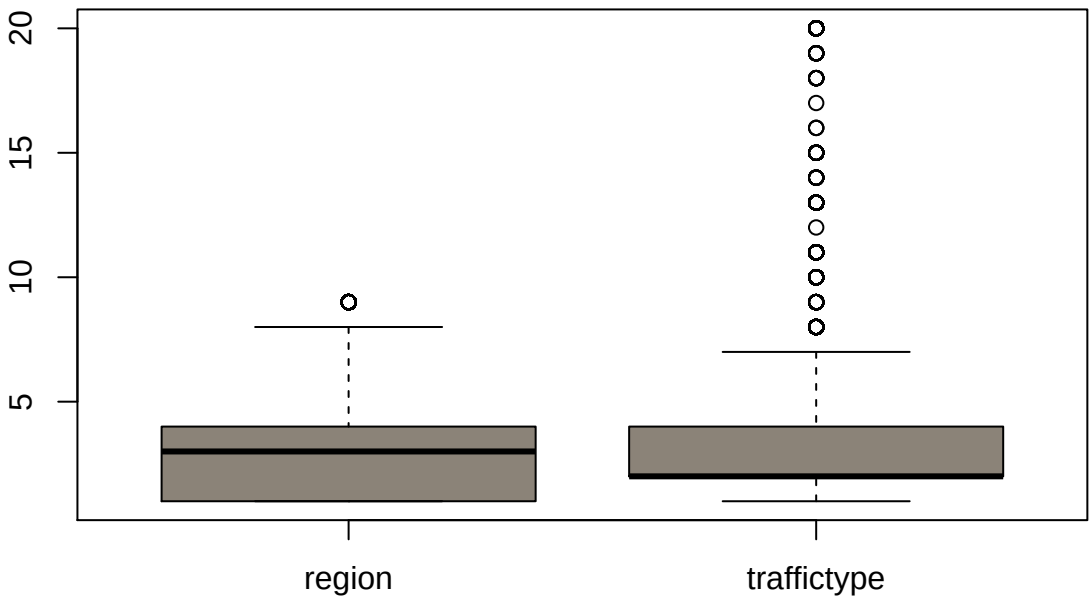
```
boxplot(df[, c(5:8)], col="burlywood4")
```




```
boxplot(df[, c(9:13)], col="coral4")
```

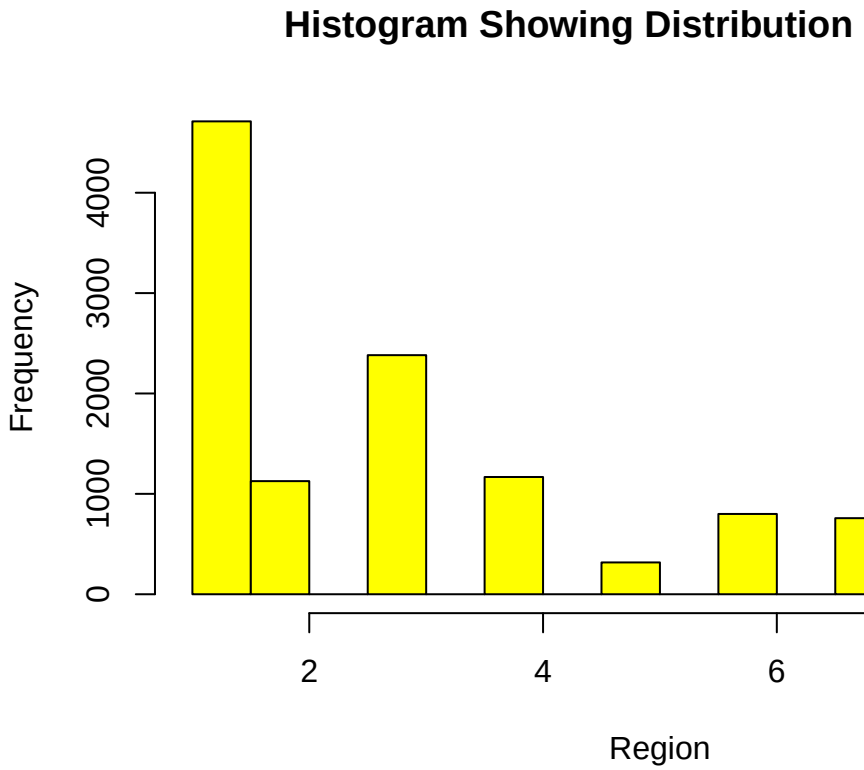



```
boxplot(df[, c(14,15)], col="antiquewhite4")
```



There is presence of outliers in the dataset though i will not drop them

```
hist(df$region, col = "yellow", main = "Histogram Showing Distribution of Region", xlab = "Region")
```



I'll then plot the distribution of the region

It shows that region 1 has the most visitors

```
num <- data.matrix(data.frame(unclass(df)))
head(num)
```

I'll then convert categorical variable to numerical

```
##      administrative administrative_duration informational
## [1,]              0                       0             0
## [2,]              0                       0             0
## [3,]              0                      -1             0
## [4,]              0                       0             0
## [5,]              0                       0             0
## [6,]              0                       0             0
##      informational_duration productrelated productrelated_duration bouncerates
## [1,]                      0              1                 0.000000  0.20000000
## [2,]                      0              2                64.000000  0.00000000
## [3,]                     -1              1                -1.000000  0.20000000
## [4,]                      0              2                 2.666667  0.05000000
## [5,]                      0             10               627.500000  0.02000000
## [6,]                      0             19               154.216667  0.01578947
##      exitrates pagevalues specialday month operatingsystems browser region
```

```
## [1,] 0.2000000          0          0     3                1       1       1
## [2,] 0.1000000          0          0     3                2       2       1
## [3,] 0.2000000          0          0     3                4       1       9
## [4,] 0.1400000          0          0     3                3       2       2
## [5,] 0.0500000          0          0     3                3       3       1
## [6,] 0.0245614          0          0     3                2       2       1
##      traffictype visitortype weekend revenue
## [1,]           1           3       0       0
## [2,]           2           3       0       0
## [3,]           3           3       0       0
## [4,]           4           3       0       0
## [5,]           4           3       1       0
## [6,]           3           3       0       0
```

They have been converted

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

I'll then check the coorelation of the variables

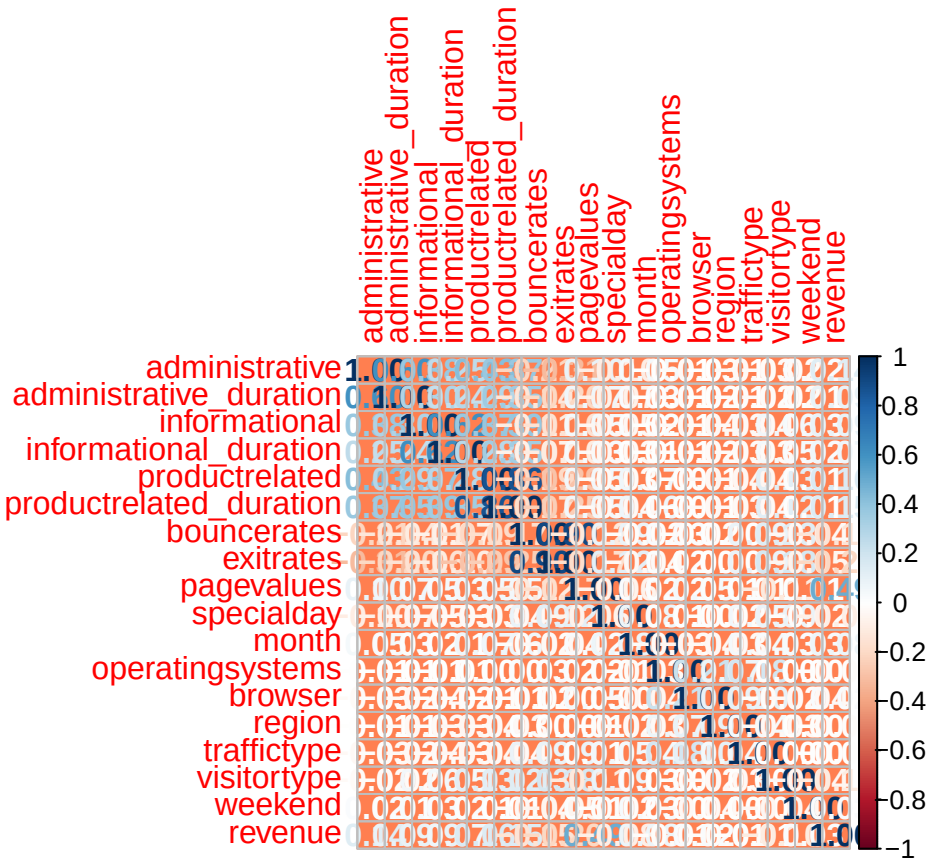
```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
correlation <- cor(num, method = "pearson")

options(repr.plot.width=12, repr.plot.height=12)
corrplot(correlation, diag=TRUE, method="number", bg="coral",)
```



Modelling

```
# I'll define the feautres to be clustered
df.1<-num[, c(1:17)]
df.2 <-num[, "revenue"]
#
head(df.1)
```

K Means-Clustering

```
##      administrative administrative_duration informational
## [1,]              0                       0             0
## [2,]              0                       0             0
## [3,]              0                      -1             0
## [4,]              0                       0             0
## [5,]              0                       0             0
## [6,]              0                       0             0
##      informational_duration productrelated productrelated_duration bouncerates
## [1,]                      0              1                 0.000000  0.20000000
## [2,]                      0              2                64.000000  0.00000000
## [3,]                     -1              1                -1.000000  0.20000000
## [4,]                      0              2                 2.666667  0.05000000
```

```
## [5,]                     0            10              627.500000  0.02000000
## [6,]                     0            19              154.216667  0.01578947
##      exitrates pagevalues specialday month operatingsystems browser region
## [1,] 0.2000000          0          0     3                 1       1      1
## [2,] 0.1000000          0          0     3                 2       2      1
## [3,] 0.2000000          0          0     3                 4       1      9
## [4,] 0.1400000          0          0     3                 3       2      2
## [5,] 0.0500000          0          0     3                 3       3      1
## [6,] 0.0245614          0          0     3                 2       2      1
##      traffictype visitortype weekend
## [1,]           1           3       0
## [2,]           2           3       0
## [3,]           3           3       0
## [4,]           4           3       0
## [5,]           4           3       1
## [6,]           3           3       0
```

```
normalize <- function(x){
  return ((x - min(x)) / (max(x) - min(x)))
}
# scaling the variables to mitigate bias towards higher values
nom.col <- c(1:17)

for (col in nom.col){
  df.1[, col] <- normalize(df.1[, col])
}

head(df.1)
```

```
##      administrative administrative_duration informational
## [1,]              0            0.0002941393             0
## [2,]              0            0.0002941393             0
## [3,]              0            0.0000000000             0
## [4,]              0            0.0002941393             0
## [5,]              0            0.0002941393             0
## [6,]              0            0.0002941393             0
##      informational_duration productrelated productrelated_duration bouncerates
## [1,]           0.0003920992    0.001418440            1.563122e-05  1.00000000
## [2,]           0.0003920992    0.002836879            1.016029e-03  0.00000000
## [3,]           0.0000000000    0.001418440            0.000000e+00  1.00000000
## [4,]           0.0003920992    0.002836879            5.731448e-05  0.25000000
## [5,]           0.0003920992    0.014184397            9.824223e-03  0.10000000
## [6,]           0.0003920992    0.026950355            2.426226e-03  0.07894737
##      exitrates pagevalues specialday     month operatingsystems    browser
## [1,]  1.000000          0          0 0.2222222        0.0000000 0.00000000
## [2,]  0.500000          0          0 0.2222222        0.1428571 0.08333333
## [3,]  1.000000          0          0 0.2222222        0.4285714 0.00000000
## [4,]  0.700000          0          0 0.2222222        0.2857143 0.08333333
## [5,]  0.250000          0          0 0.2222222        0.2857143 0.16666667
## [6,]  0.122807          0          0 0.2222222        0.1428571 0.08333333
##      region traffictype visitortype weekend
## [1,]  0.000  0.00000000           1       0
## [2,]  0.000  0.05263158           1       0
## [3,]  1.000  0.10526316           1       0
```

```
## [4,]  0.125  0.15789474           1       0
## [5,]  0.000  0.15789474           1       1
## [6,]  0.000  0.10526316           1       0
```

```
# I'll define seed to ensure similar results are produced
set.seed(123)

# I'll  cluster the features using kmeans and 3 clusters
kres <- kmeans(df.1, 3, nstart=1)

# previewing the no. of records in each cluster
kres$size
```

```
## [1] 9560  916 1723
```

```
install.packages("cluster")
```

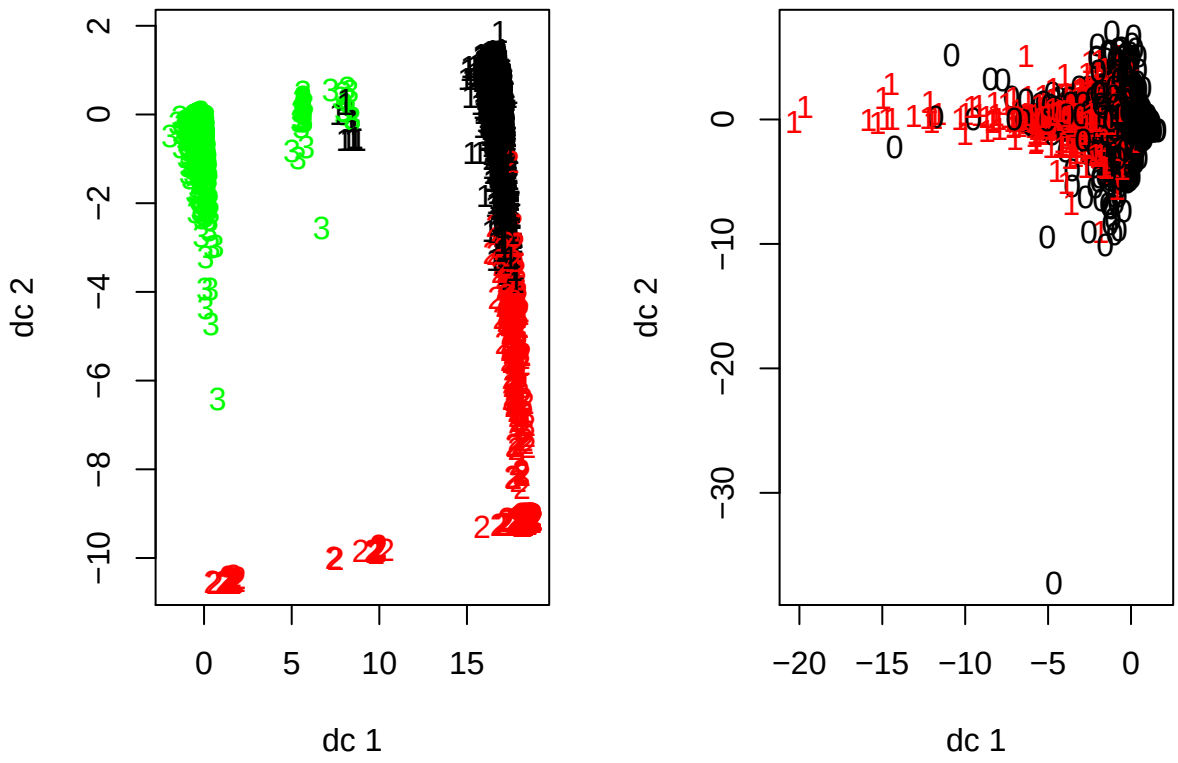
```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
install.packages("fpc")
```

```
## Installing package into '/home/greg/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

```
library(cluster)
library(fpc)

# plotting graphs to display the clusters
par(mfrow = c(1,2), mar=c(5,4,2,2))
plotcluster(df.1, kres$cluster)
plotcluster(df.1, df.2)
```




```
table(kres$cluster, df.2)
```

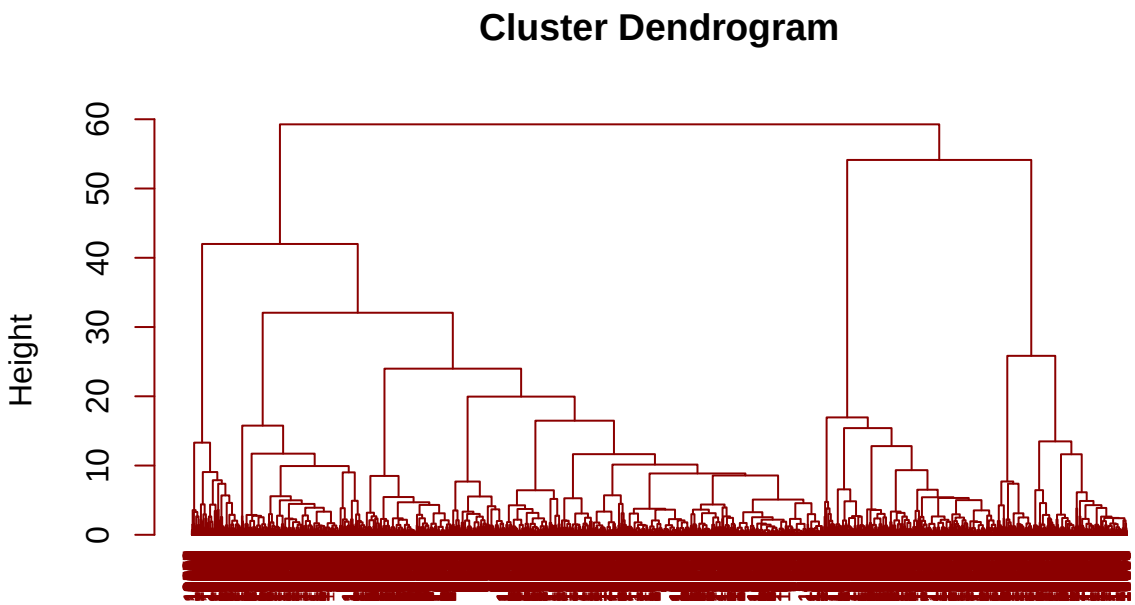
```
##    df.2
##        0    1
##   1 8095 1465
##   2  910    6
##   3 1286  437
```

Hierarchical Clustering

```
# defining the distance metric to be used (i.e. euclidean)
dis <- dist(df.1, method="euclidean")

# training the hierarchical clustering
hclus <- hclust(dis, method="ward.D2")
```

```
# plotting the dendrogram of the hierarchical cluster
options(repr.plot.width=10, repr.plot.height=9)
plot(hclus, cex=0.6, hang=-2, col="dark red")
```


Cluster Dendrogram
Height
0
10
20
30
40
50
60
dis
hclust (*, "ward.D2")