Customer Segmentation

Dave Njoroge

2022-06-16

1. Defining the Question

#a) Specifying the Data Analytic Question

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.

#b) Defining the Metric for Success

- Perform clustering stating insights drawn from your analysis and visualizations.
- 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.

#c) Understanding the context

understanding customer behavior is relevant to a business since it helps in determining sales distributions

#d) 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

#e) Data Relevance

The dataset for this Independent project can be found here [http://bit.ly/EcommerceCustomersDataset

The dataset consists of 10 numerical and 8 categorical attributes.

"Administrative"
"Administrative_Duration" "Informational"
"Informational_Duration" "ProductRelated"
"ProductRelated_Duration" "BounceRates"
"ExitRates"
"PageValues"
"SpecialDay"
"Month"
"OperatingSystems"
"Browser"
"Region"
"TrafficType"
"VisitorType"
"Weekend"
"Revenue"

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

2. Reading the Data

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(ggplot2) #Plotting
library(dplyr)
```

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

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

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

```
library(tidyverse)
```

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

## v tibble  3.1.7     v purrr   0.3.4
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

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

```r
library(DataExplorer)
library(factoextra)
```

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

```r
library(cluster)
```

```r
online_shoppers_intention <- read.csv(file = 'online_shoppers_intention.csv')
```

```r
head(online_shoppers_intention)
```

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

```r
df <- data.frame(online_shoppers_intention)
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
```

3. Checking the Data

```
# Determining the no. of records in our dataset
dim(df)
```

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

```
# Checking whether each column has an appropriate datatype
str(df)
```

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

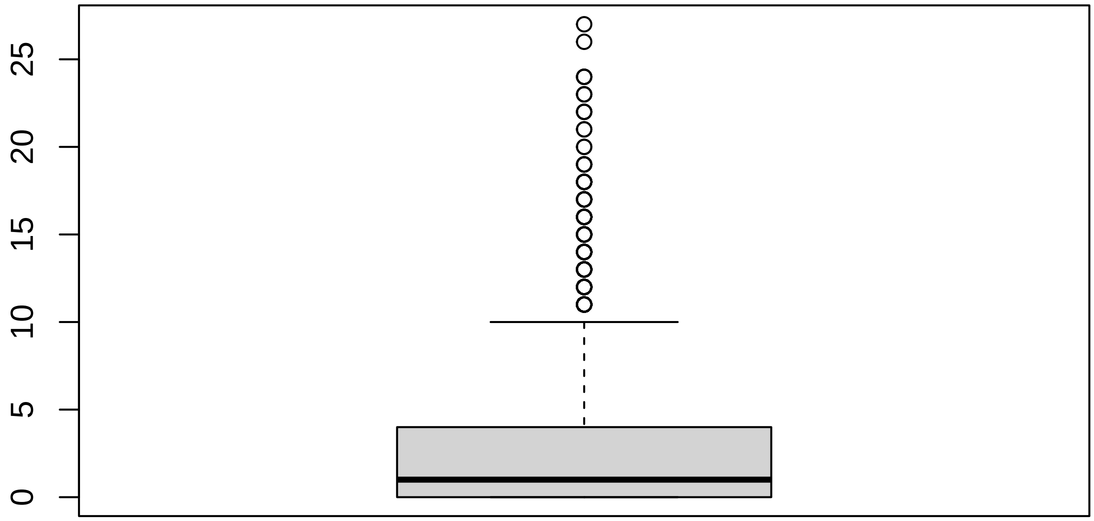
```
names(df)
```

```
##  [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"
```

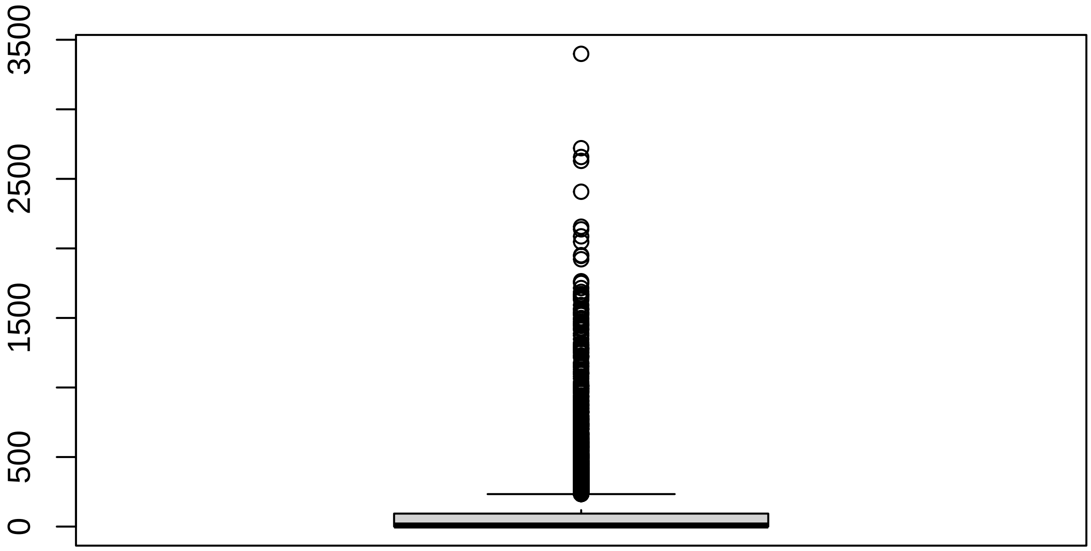
5. Tidying the Dataset

Checking for Outliers

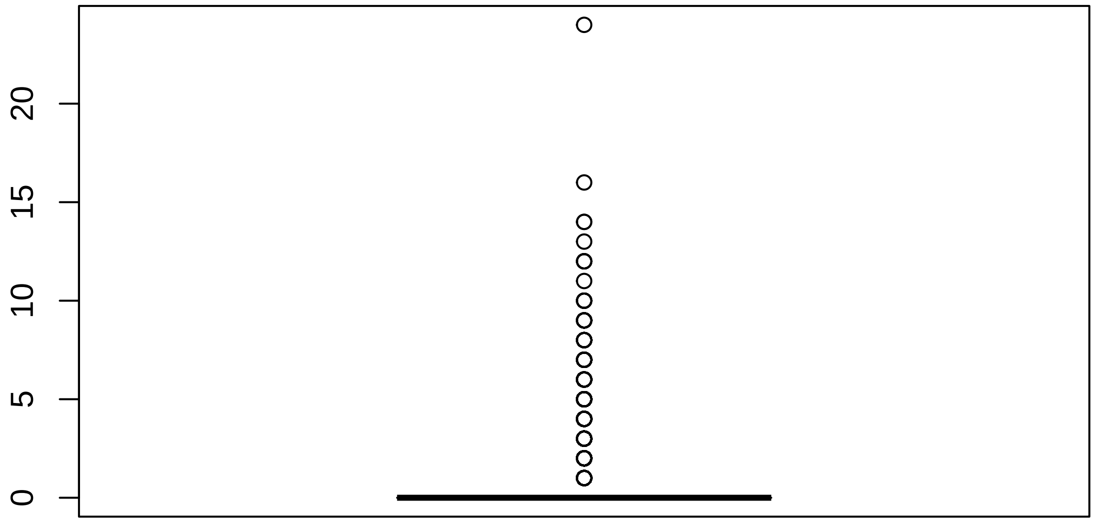
```
boxplot(df$Administrative)
```



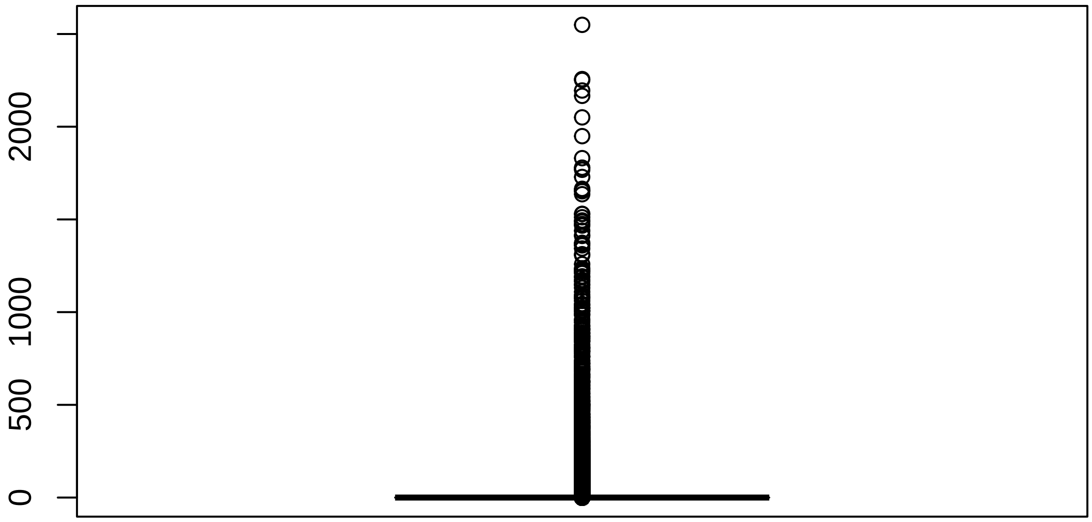
```
boxplot(df$Administrative_Duration)
```



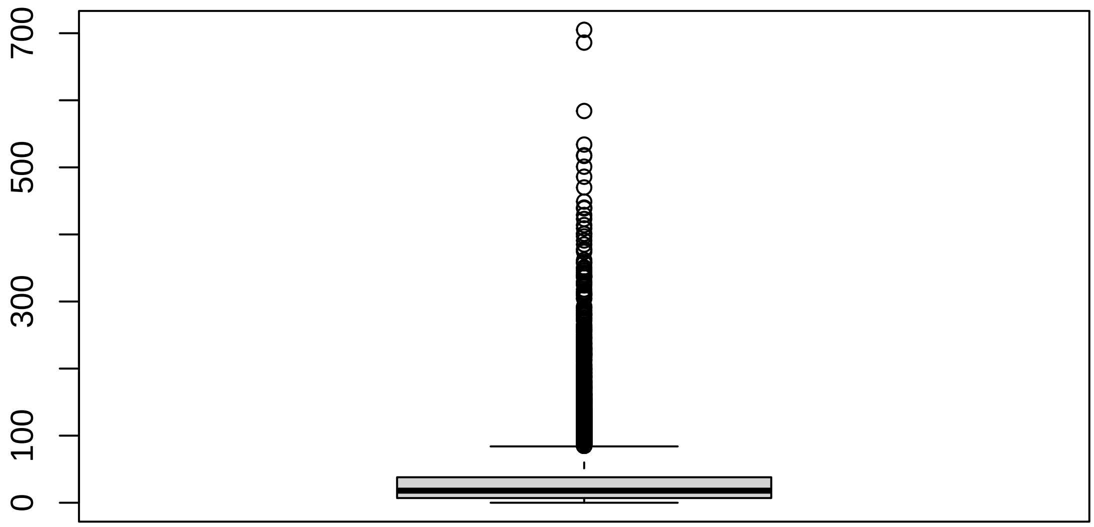
```
boxplot(df$Informational)
```



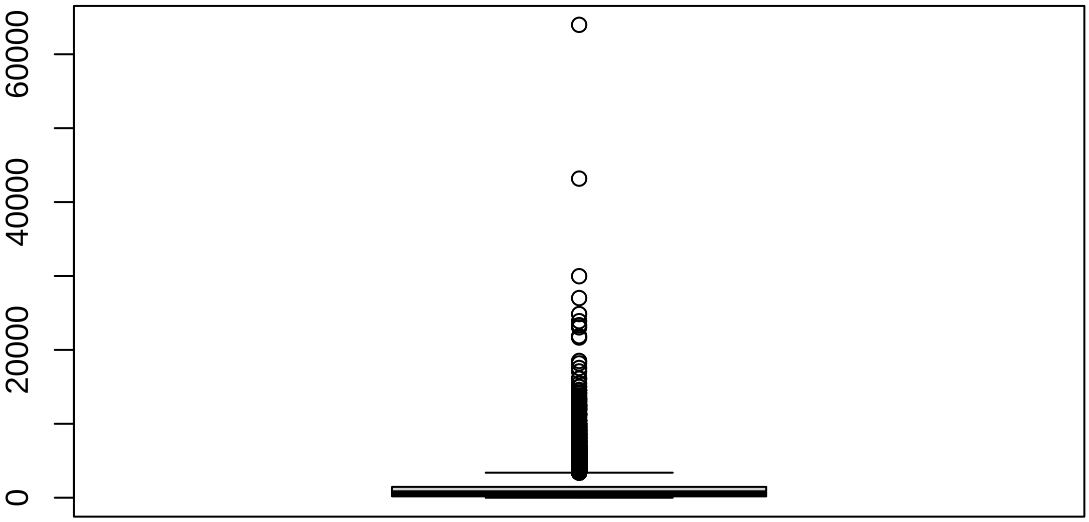
```
boxplot(df$Informational_Duration)
```



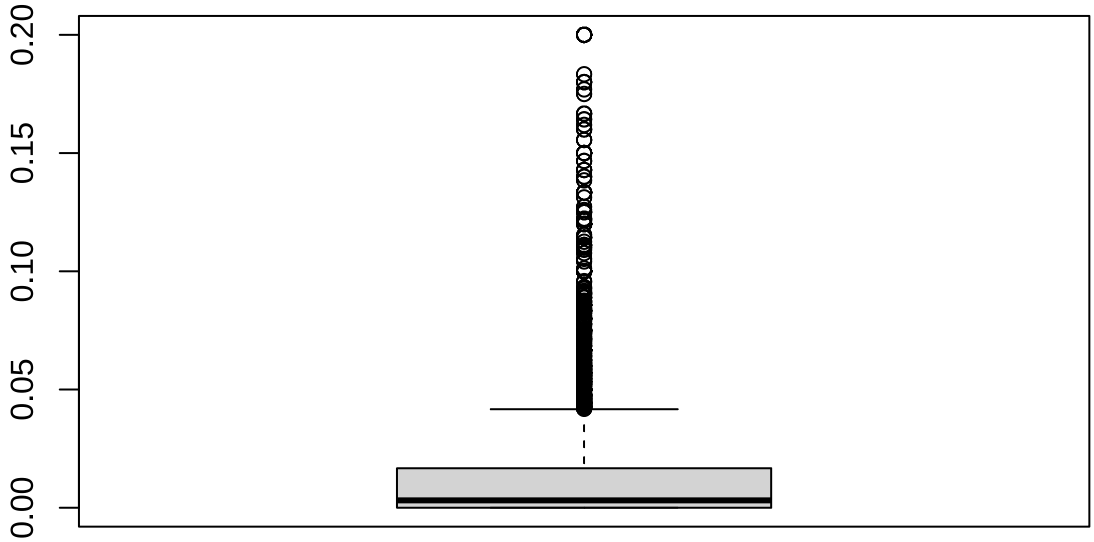
```
boxplot(df$ProductRelated)
```



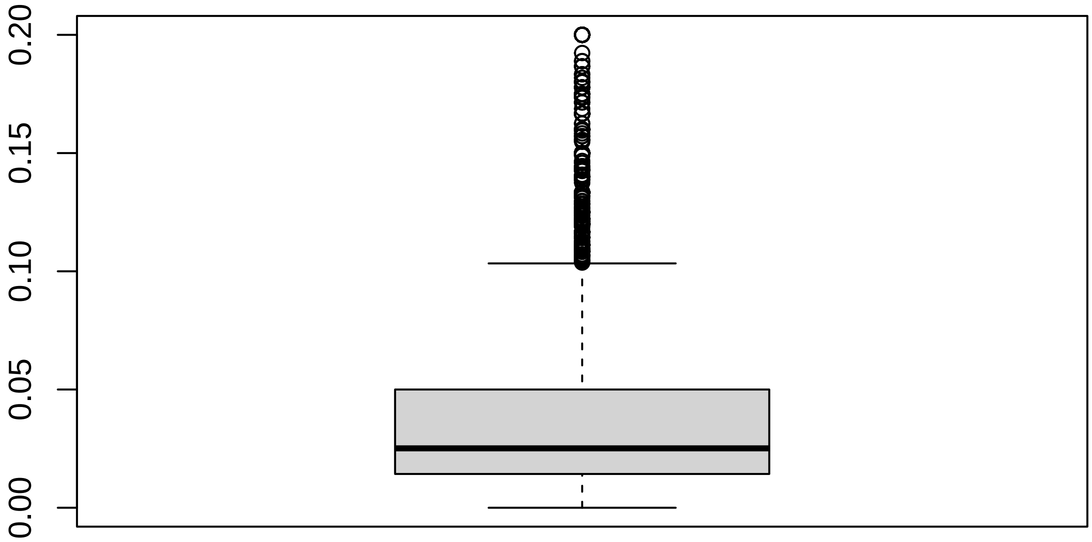
```
boxplot(df$ProductRelated_Duration)
```



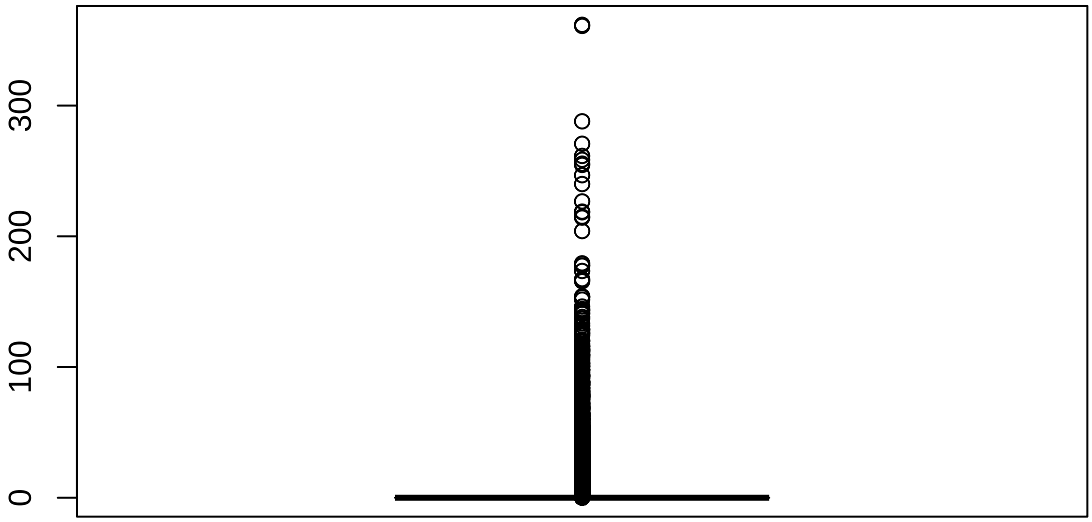
```
boxplot(df$BounceRates)
```



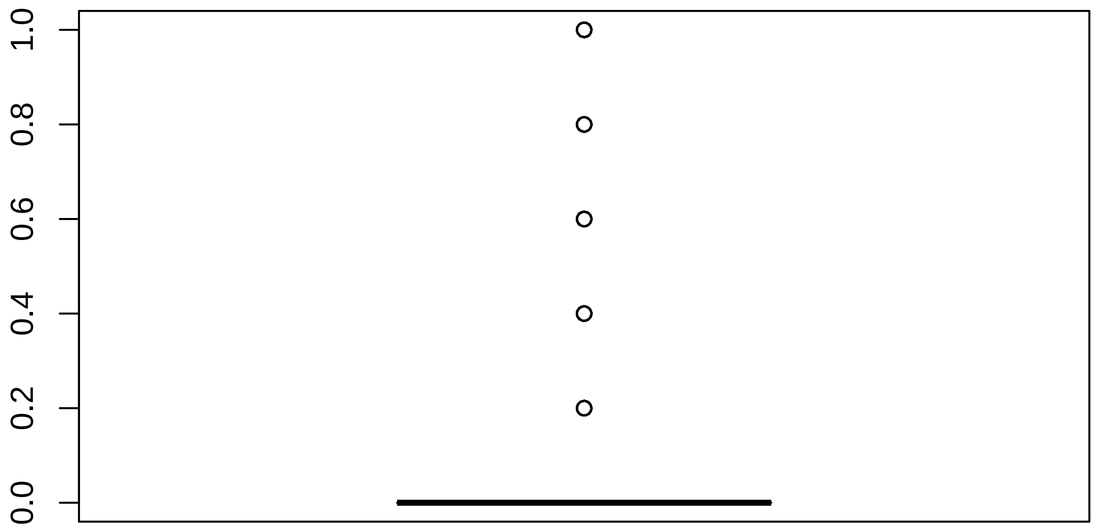
```
boxplot(df$ExitRates)
```



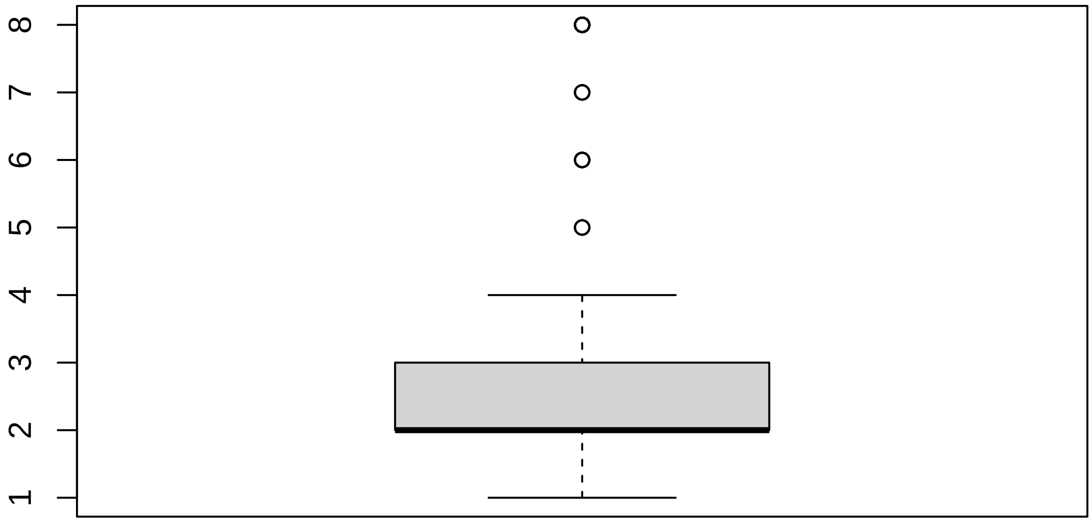
```
boxplot(df$PageValues)
```



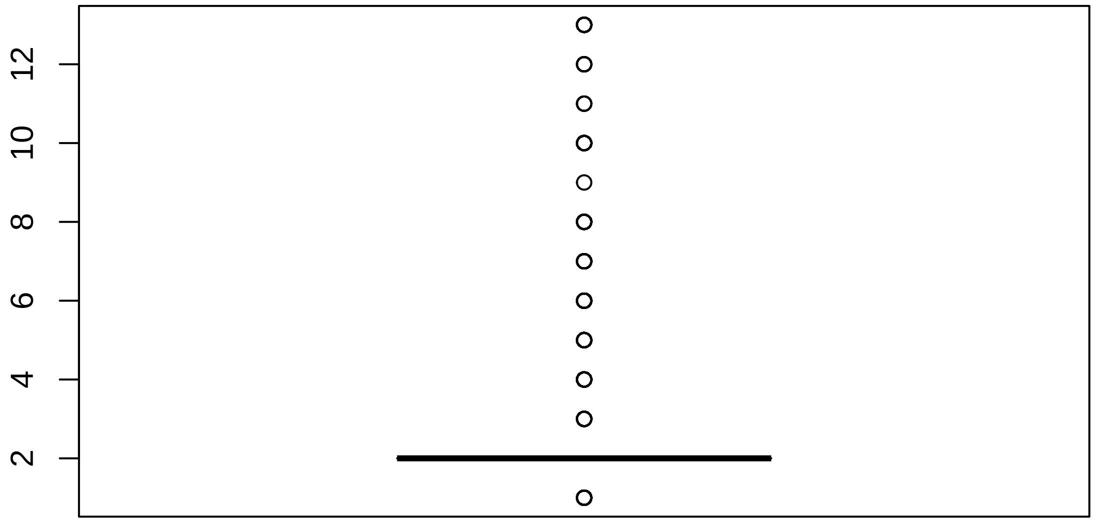
```
boxplot(df$SpecialDay)
```



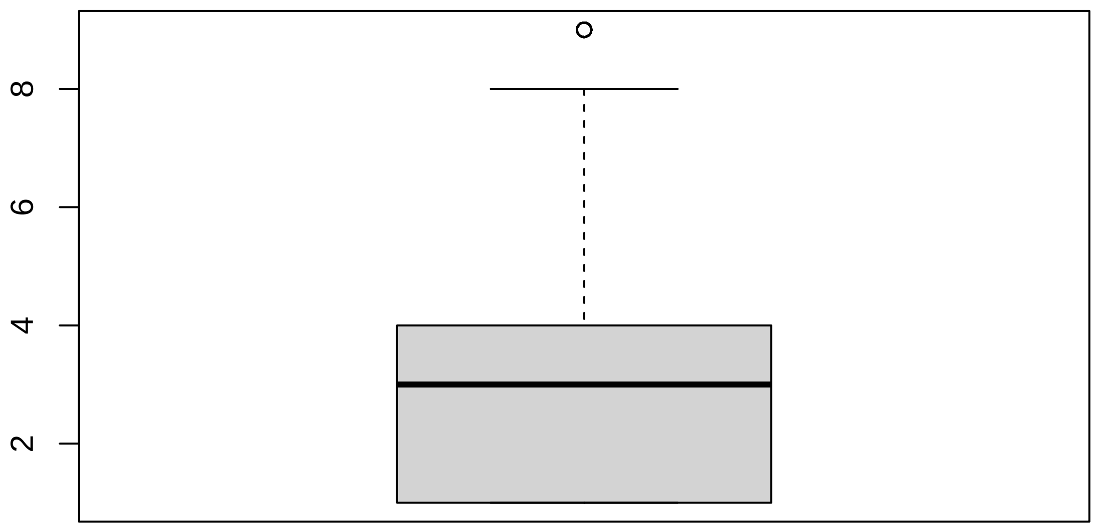
```
boxplot(df$OperatingSystems)
```



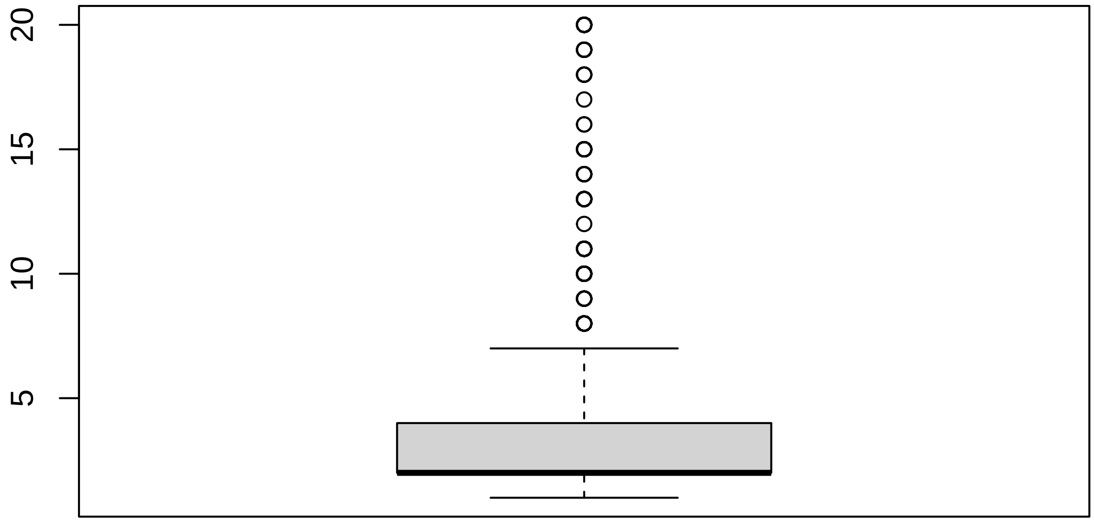
```
boxplot(df$Browser)
```



```
boxplot(df$Region)
```



```
boxplot(df$TrafficType)
```



###we have outliers in our dataset but we won't deal with them now since the data maybe relevant

Identifying the Missing Data

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

```
##          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 some missing data in some columns

```
#we fill the missing with their mean
df$Administrative[is.na(df$Administrative)] <- mean(df$Administrative, na.rm = TRUE)
```

```r
df$Administrative_Duration[is.na(df$Administrative_Duration)] <- mean(df$Administrative_Duration, na.rm
df$Informational[is.na(df$Informational)] <- mean(df$Informational, na.rm = TRUE)
df$Informational_Duration[is.na(df$Informational_Duration)] <- mean(df$Informational_Duration, na.rm =
df$ProductRelated[is.na(df$ProductRelated)] <- mean(df$ProductRelated, na.rm = TRUE)
df$ProductRelated_Duration[is.na(df$ProductRelated_Duration)] <- mean(df$ProductRelated_Duration, na.rm
df$BounceRates[is.na(df$BounceRates)] <- mean(df$BounceRates, na.rm = TRUE)
df$ExitRates[is.na(df$ExitRates)] <- mean(df$ExitRates, na.rm = TRUE)
```

```r
#confirming the missing data
colSums(is.na(df))
```

```
##          Administrative Administrative_Duration           Informational
##                       0                       0                       0
##  Informational_Duration          ProductRelated ProductRelated_Duration
##                       0                       0                       0
##             BounceRates               ExitRates              PageValues
##                       0                       0                       0
##              SpecialDay                   Month        OperatingSystems
##                       0                       0                       0
##                 Browser                  Region             TrafficType
##                       0                       0                       0
##             VisitorType                 Weekend                 Revenue
##                       0                       0                       0
```

Checking statistical summary of the dataset

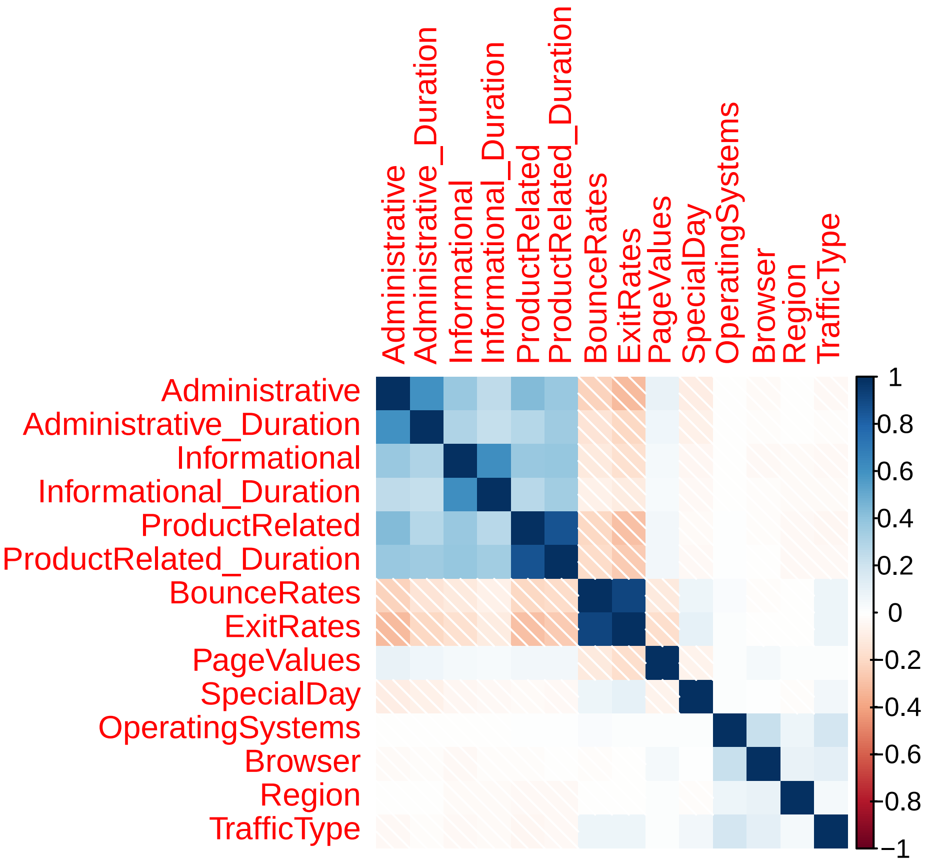
```r
summary(df)
```

```
##  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.:   93.26        3rd Qu.: 0.000
##  Max.   :27.000   Max.   :3398.75         Max.   :24.000
##  Informational_Duration ProductRelated   ProductRelated_Duration
##  Min.   :  -1.00        Min.   :  0.00   Min.   :   -1.0
##  1st Qu.:   0.00        1st Qu.:  7.00   1st Qu.:  185.3
##  Median :   0.00        Median : 18.00   Median :  601.1
##  Mean   :  34.51        Mean   : 31.76   Mean   : 1196.0
##  3rd Qu.:   0.00        3rd Qu.: 38.00   3rd Qu.: 1464.2
##  Max.   :2549.38        Max.   :705.00   Max.   :63973.5
##   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 :0.003125   Median :0.02516   Median :  0.000   Median :0.00000
##  Mean   :0.022152   Mean   :0.04300   Mean   :  5.889   Mean   :0.06143
##  3rd Qu.:0.016941   3rd Qu.:0.05000   3rd Qu.:  0.000   3rd Qu.:0.00000
##  Max.   :0.200000   Max.   :0.20000   Max.   :361.764   Max.   :1.00000
##     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
```

checking for

```
numeric_data = df[, sapply(df, is.numeric)]
```

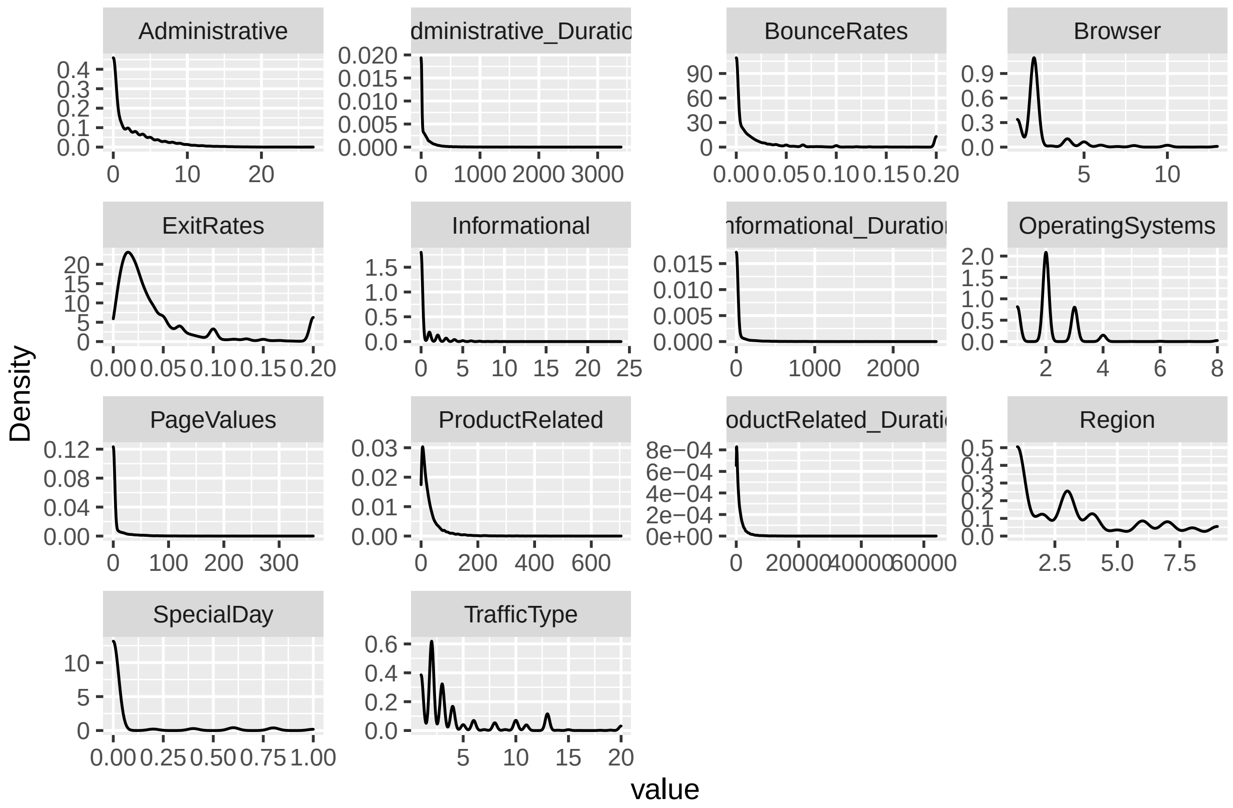
```
corrplot(cor(numeric_data), method = 'shade')
```



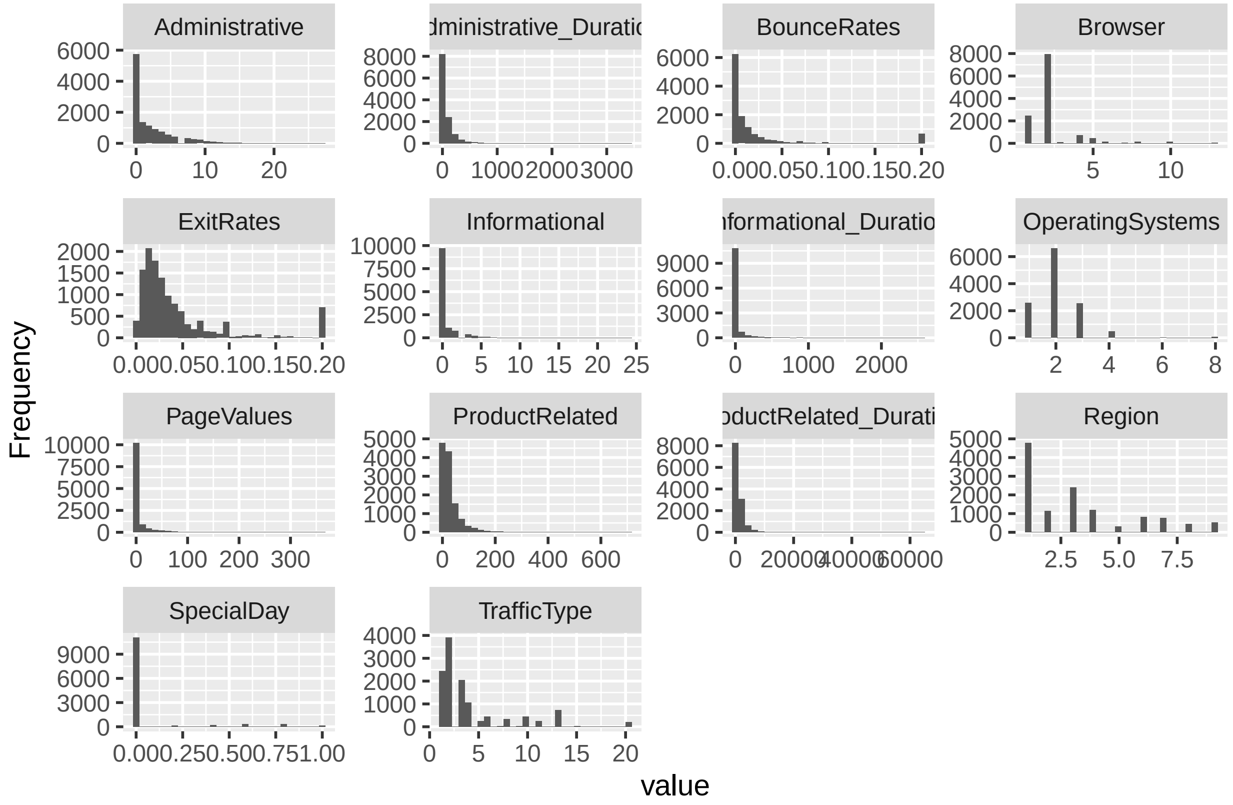
6. Analysis

#univariate

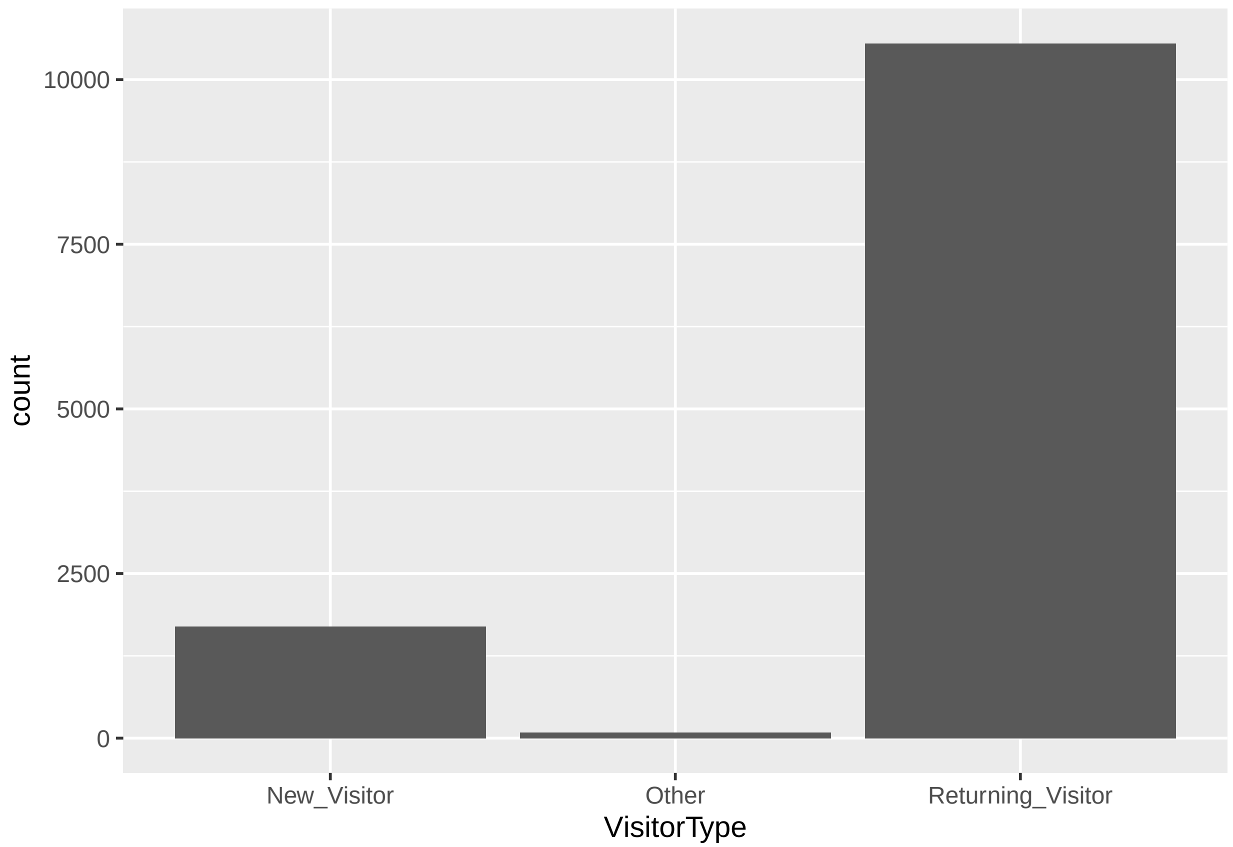
```
plot_density(df)
```



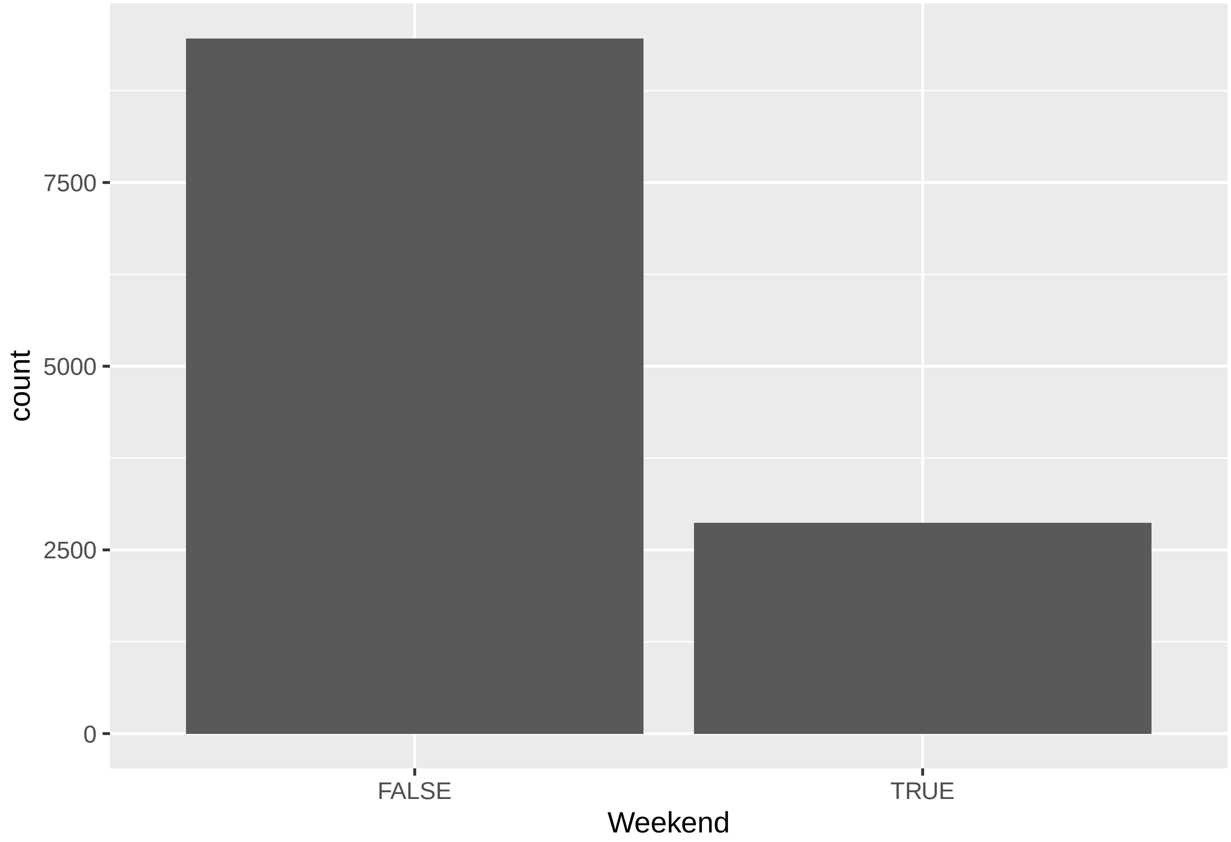
```
plot_histogram(df,ncol = 4L)
```



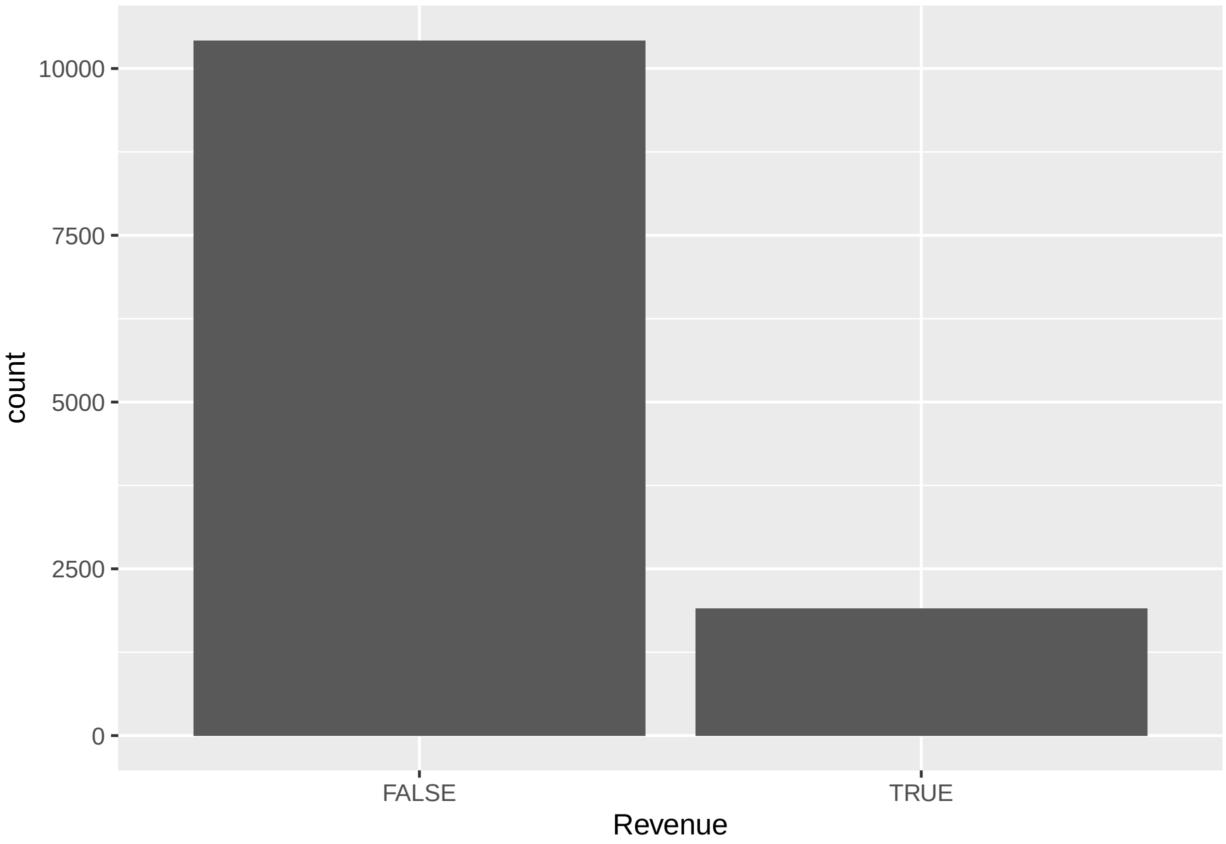
```r
ggplot(data = df) +
  geom_bar(mapping = aes(x = VisitorType))
```



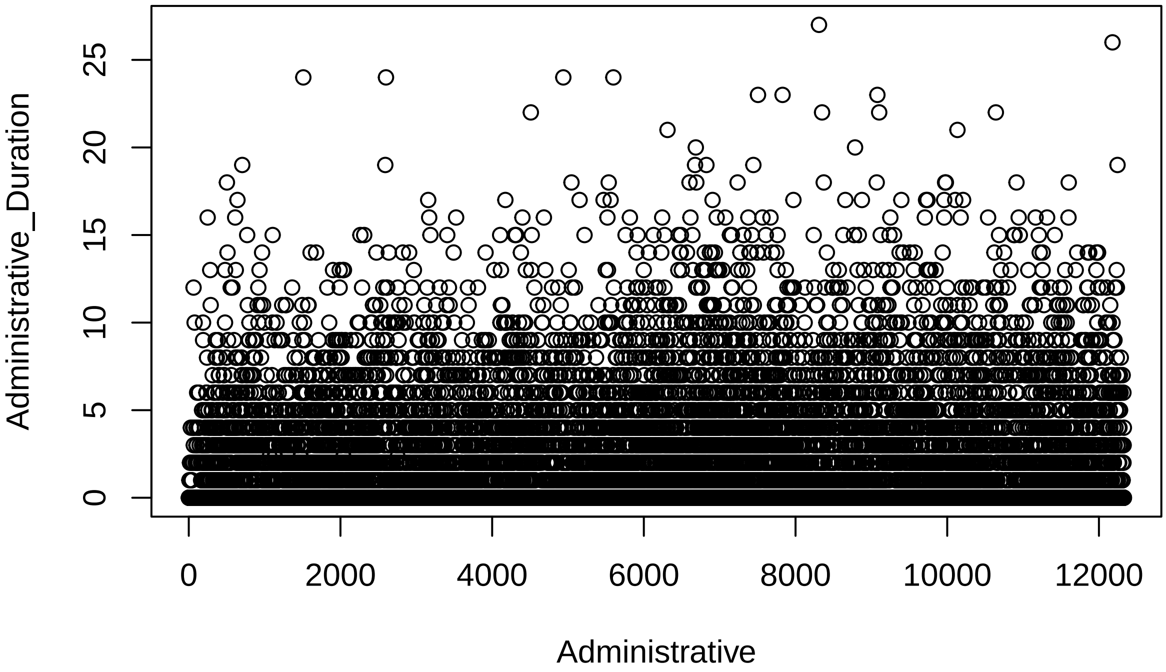
```
ggplot(data = df) +
  geom_bar(mapping = aes(x = Weekend))
```



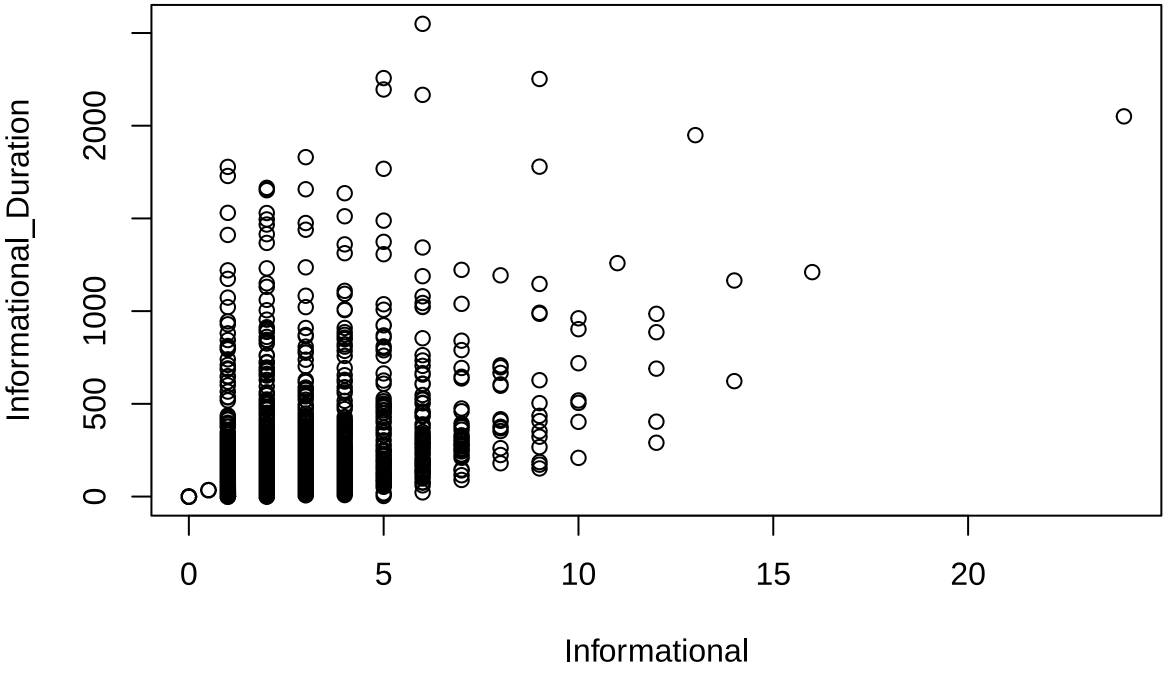
```
ggplot(data = df) +
  geom_bar(mapping = aes(x = Revenue))
```



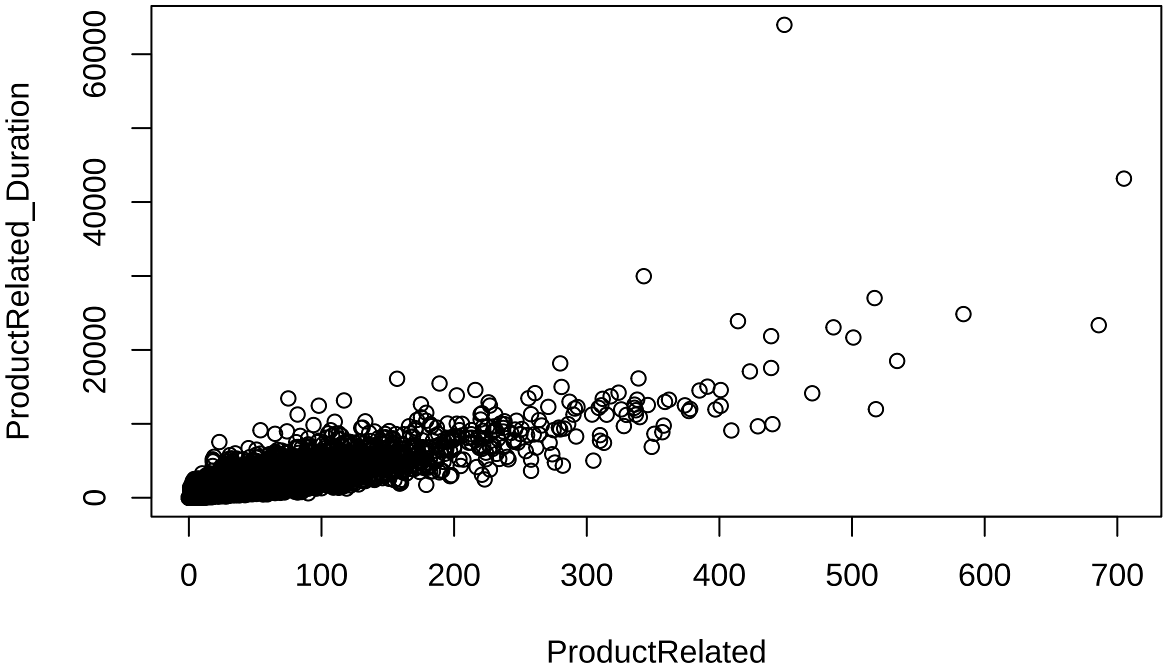
```
plot(df$Administrative,df$Daily.Administrative_Duration, xlab='Administrative',ylab='Administrative_Dura
```



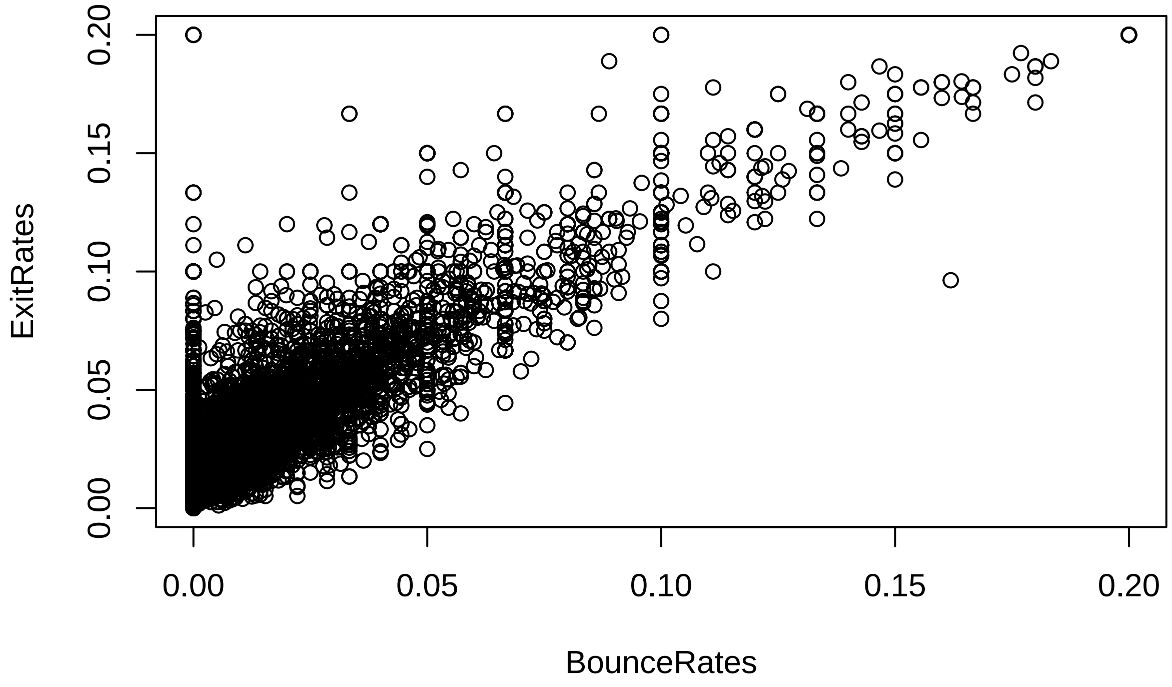
```
plot(df$Informational,df$Informational_Duration, xlab='Informational',ylab='Informational_Duration')
```



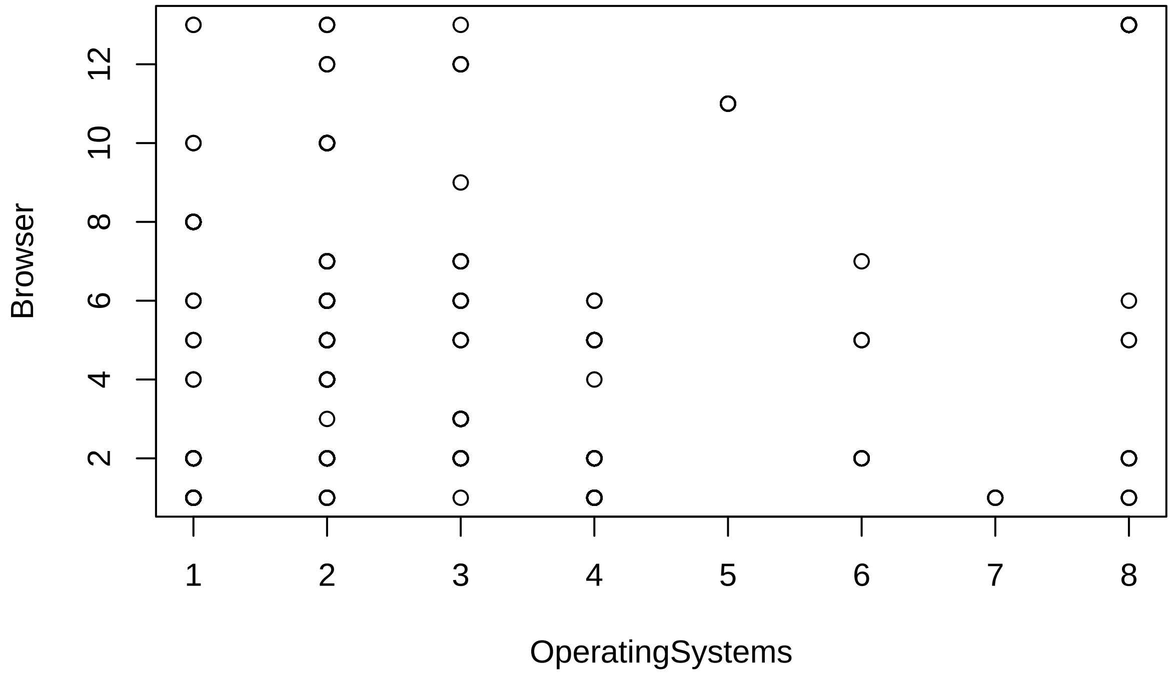
```
plot(df$ProductRelated,df$ProductRelated_Duration, xlab='ProductRelated',ylab='ProductRelated_Duration')
```



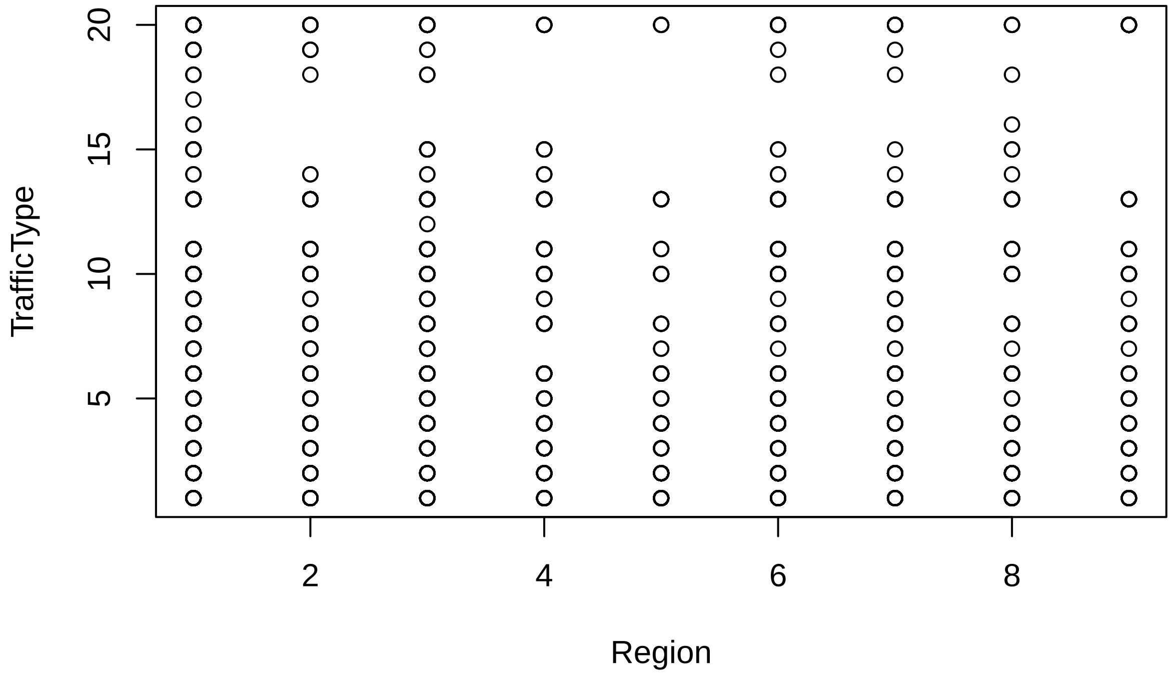
```
plot(df$BounceRates,df$ExitRates, xlab='BounceRates',ylab='ExitRates')
```



```
plot(df$OperatingSystems,df$Browser, xlab='OperatingSystems',ylab='Browser')
```



```
plot(df$Region,df$TrafficType, xlab='Region',ylab='TrafficType')
```



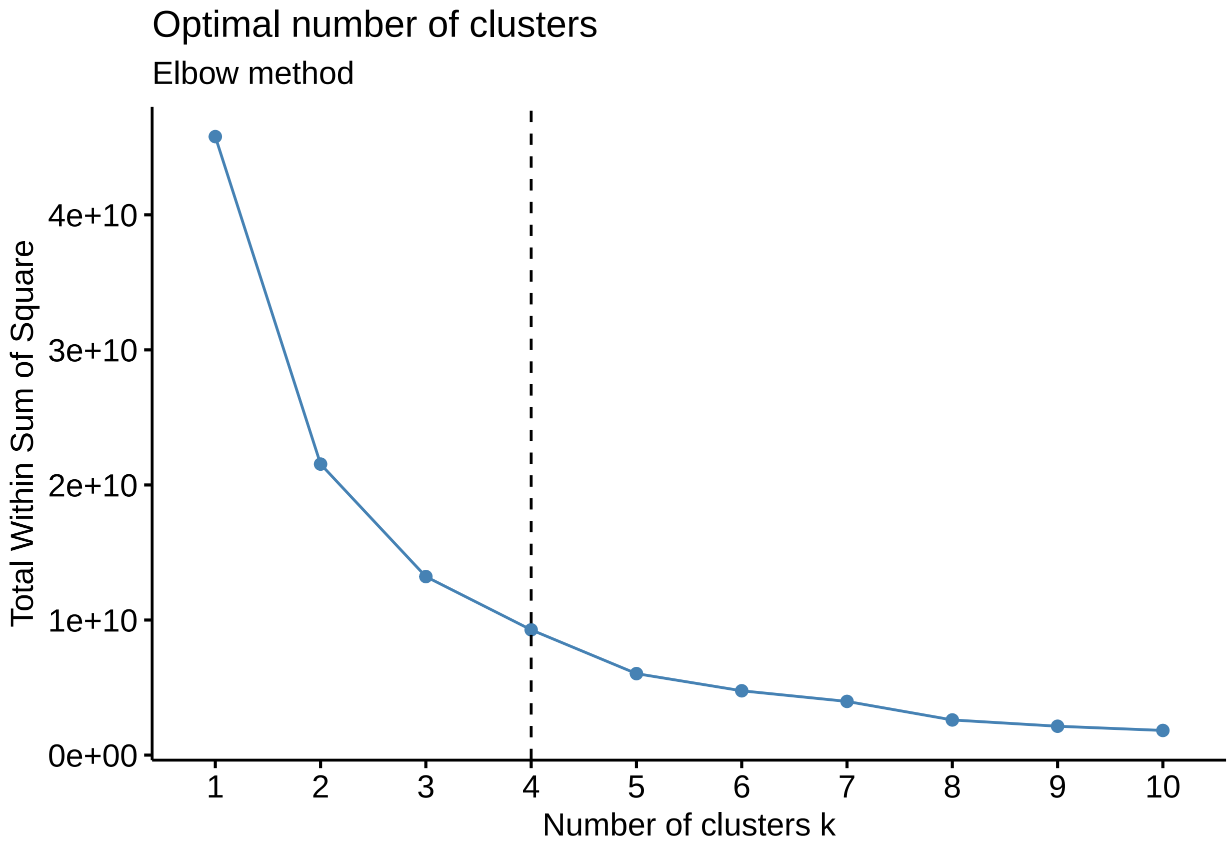
changing categorical values to numbers

```
Month_enc = data.frame(model.matrix(~0+df$Month))
VisitorType_enc = data.frame(model.matrix(~0+df$VisitorType))
Weekend_enc = data.frame(model.matrix(~0+df$Weekend))
Revenue_enc = data.frame(model.matrix(~0+df$Revenue))
# Dropping non numerical columns
drop_cols = c('Month', 'VisitorType','Weekend','Revenue')
df_customers = select(data.frame(cbind(df,Month_enc, VisitorType_enc,Weekend_enc,Revenue_enc)), -drop_c
```

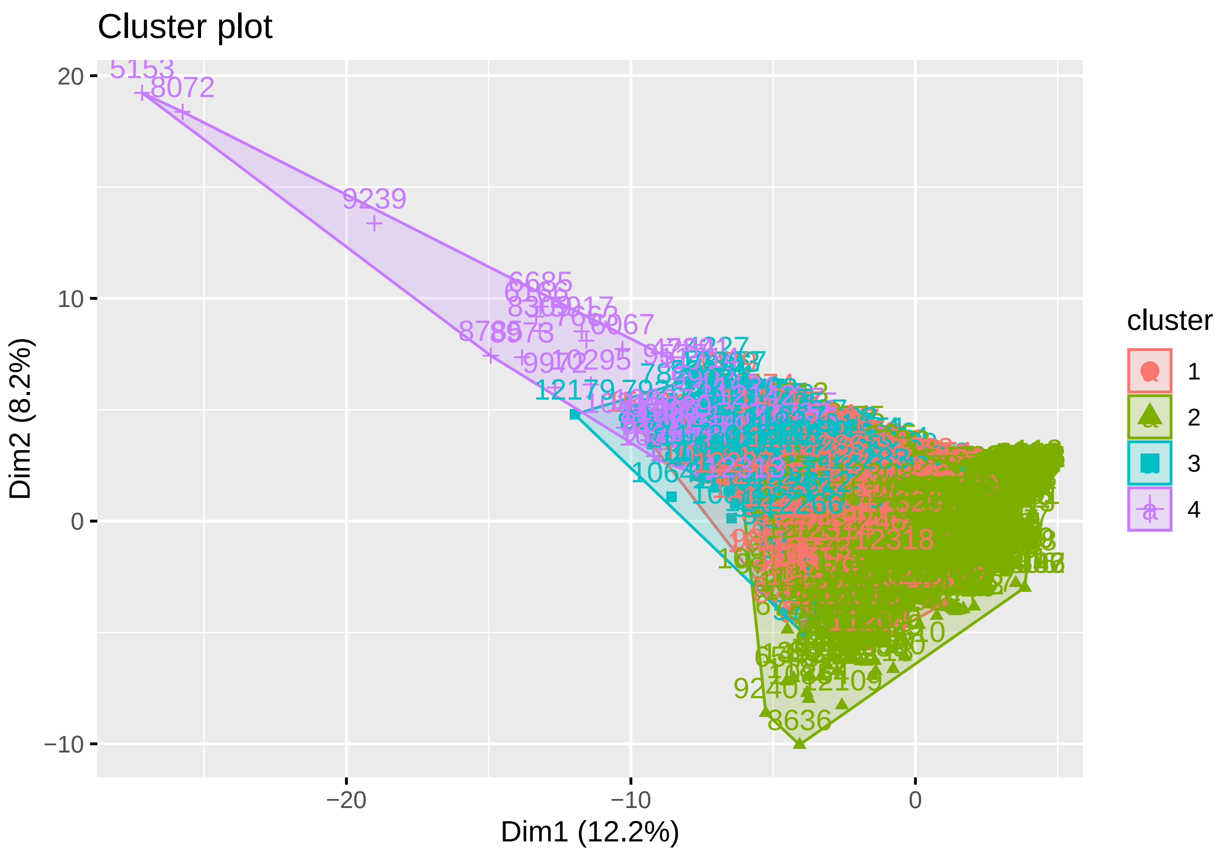
```
## Note: Using an external vector in selections is ambiguous.
## i Use ‘all_of(drop_cols)‘ instead of ‘drop_cols‘ to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

MODELING #K-MEANS CLUSTERING

```
fviz_nbclust(df_customers, kmeans, method = "wss") +
    geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
```



```
kmeans_model = kmeans(df_customers, 4)
fviz_cluster(kmeans_model, df_customers)
```



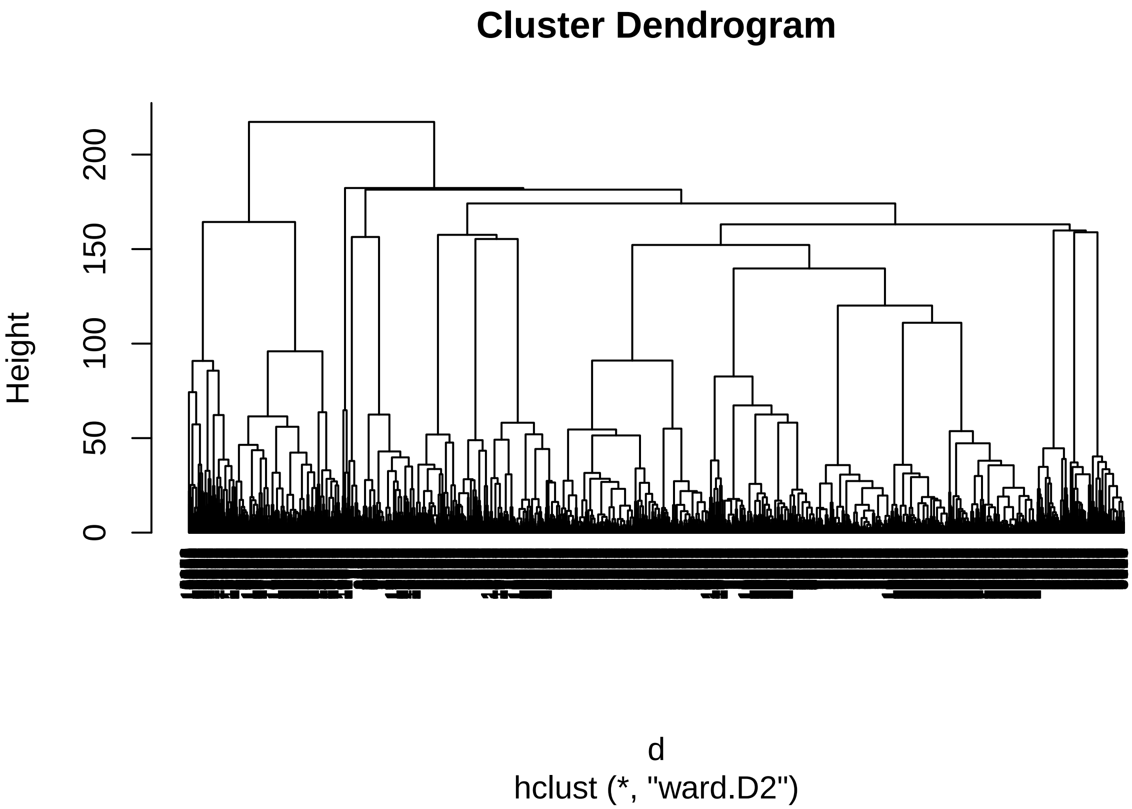
#HIERACHICAL CLUSTERING

```
df_H <- scale(df_customers)
```

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

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

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



We were not able to draw meaningful insights from the dendogram above.

Challenging the Solution Our Hierachical Clustering Method did not perform very well. This might have been caused by the big number of columns which could have reduced using the Principal Component Analysis.