R- project 26/03/2022

Stacy

2022-03-30

```
# Define the question :A Kenyan entrepreneur has created an online cryptography course and would want t
```

```
# The metric for success:Your findings should help inform the team in formulating the marketing and sal

# The context:1. Perform clustering stating insights drawn from your analysis and visualizations.
#            2. Upon implementation, provide comparisons between the approaches learned this week i.e. 

# Experimental design taken:
## 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

# The appropriateness of the available data to answer the given question:The data provided information 
```

```
# Load Libraries
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

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

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

```
library(ggplot2)
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.15.2
## Type citation(’dendextend’) for how to cite the package.
##
## Type browseVignettes(package = ’dendextend’) for the package vignette.
## 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 tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: ’dendextend’
```

```
## The following object is masked from ’package:stats’:
##
##     cutree
```

```
library(tidyverse)
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(numDeriv)
library(e1071) # package that holds the Naive Bayes function.
library (caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: ’caret’
```

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

```
library(moments)
```

```
##
## Attaching package: ’moments’

## The following objects are masked from ’package:e1071’:
##
##     kurtosis, moment, skewness
```

```
library(gridExtra)
```

```
##
## Attaching package: ’gridExtra’

## The following object is masked from ’package:dplyr’:
##
##     combine
```

```
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```
# Read the data
data<- read.csv('http://bit.ly/EcommerceCustomersDataset')
```

```
# Preview data
head(data)
```

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

```
# Column names
colnames(data)
```

```
##  [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"
```

```
#  sum of null values per column
colSums(is.na(data))
```

```
##          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
```

```
# fill the missing value in a column with the mean value of the column
data$Administrative[is.na(data$Administrative)] <- mean(data$Administrative, na.rm = TRUE)

data$Administrative_Duration[is.na(data$Administrative_Duration)] <- mean(data$Administrative_Duration,

data$ProductRelated[is.na(data$ProductRelated)] <- mean(data$ProductRelated, na.rm = TRUE)

data$Informational_Duration[is.na(data$Informational_Duration)] <- mean(data$Informational_Duration, na

data$ProductRelated_Duration[is.na(data$ProductRelated_Duration)] <- mean(data$ProductRelated_Duration,

data$BounceRates[is.na(data$BounceRates)] <- mean(data$BounceRates, na.rm = TRUE)

data$Informational[is.na(data$Informational)] <- mean(data$Informational, na.rm = TRUE)

data$ExitRates[is.na(data$ExitRates)] <- mean(data$ExitRates, na.rm = TRUE)

data$Revenue[is.na(data$Revenue)] <- mean(data$Revenue, na.rm = TRUE)
```

```
# r  piping to create a data frame for all numerical values

df = data %>% select(Administrative, 'Administrative_Duration', Informational, 'Informational_Duration'
'ProductRelated_Duration', BounceRates ,ExitRates, PageValues,  SpecialDay, OperatingSystems,
Browser, Region, TrafficType)
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 OperatingSystems Browser Region TrafficType
## 1          0                1       1      1           1
## 2          0                2       2      1           2
## 3          0                4       1      9           3
## 4          0                3       2      2           4
## 5          0                3       3      1           4
## 6          0                2       2      1           3
```

```
# check to confirm if the missing values are dealt with
colSums(is.na(data))
```

```
##          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
```

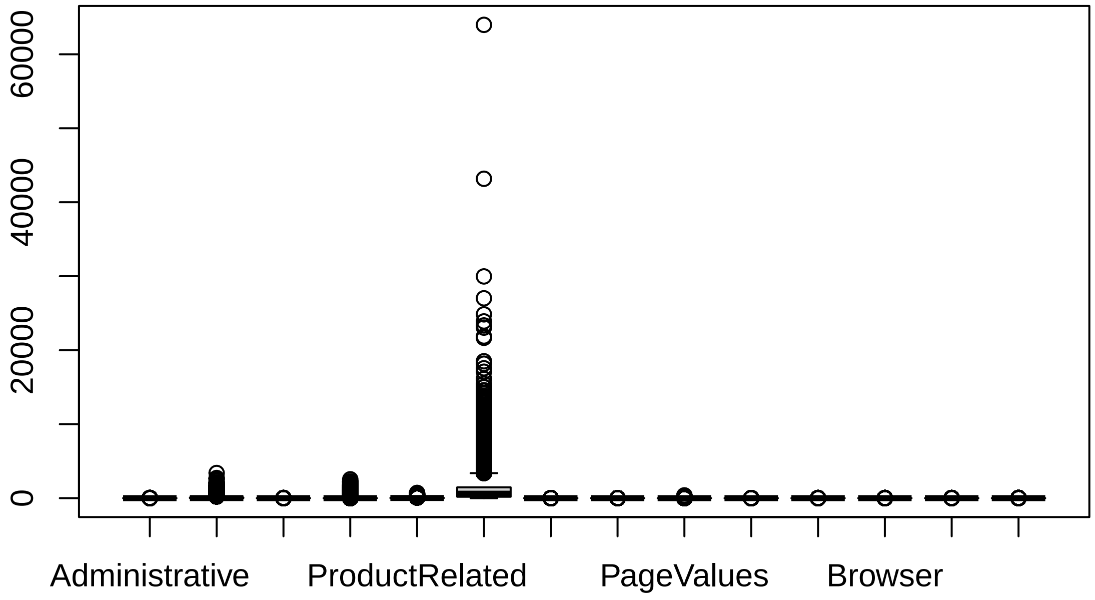
```
# Data types of each column}

str(data)
```

```
## 'data.frame':    12330 obs. of  18 variables:
##  $ Administrative         : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ Administrative_Duration: num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ Informational          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration : num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ ProductRelated         : num  1 2 1 2 10 19 1 1 2 3 ...
##  $ ProductRelated_Duration: num  0 64 -1 2.67 627.5 ...
##  $ BounceRates            : num  0.2 0 0.2 0.05 0.02 ...
##  $ ExitRates              : num  0.2 0.1 0.2 0.14 0.05 ...
##  $ PageValues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialDay             : num  0 0 0 0 0 0 0.4 0 0.8 0.4 ...
##  $ 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                : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
# Deal with Duplicated values
#unique_data <- unique(data)
dup<- data[duplicated(data),]
head(dup)
```

```
##     Administrative Administrative_Duration Informational Informational_Duration
## 159              0                       0             0                      0
## 179              0                       0             0                      0
## 419              0                       0             0                      0
## 457              0                       0             0                      0
## 484              0                       0             0                      0
## 513              0                       0             0                      0
##     ProductRelated ProductRelated_Duration BounceRates ExitRates PageValues
## 159              1                       0         0.2       0.2          0
## 179              1                       0         0.2       0.2          0
## 419              1                       0         0.2       0.2          0
## 457              1                       0         0.2       0.2          0
## 484              1                       0         0.2       0.2          0
## 513              1                       0         0.2       0.2          0
##     SpecialDay Month OperatingSystems Browser Region TrafficType
## 159          0   Feb                1       1      1           3
## 179          0   Feb                3       2      3           3
## 419          0   Mar                1       1      1           1
## 457          0   Mar                2       2      4           1
## 484          0   Mar                3       2      3           1
## 513          0   Mar                2       2      1           1
##           VisitorType Weekend Revenue
## 159 Returning_Visitor   FALSE       0
## 179 Returning_Visitor   FALSE       0
## 419 Returning_Visitor    TRUE       0
## 457 Returning_Visitor   FALSE       0
## 484 Returning_Visitor   FALSE       0
## 513 Returning_Visitor   FALSE       0
```

```
# outliers of numerical columns
boxplot(df)
```



```
# UNIVARIATE ANALYSIS
summary(data)
```

```
##  Administrative   Administrative_Duration Informational
##  Min.   : 0.000   Min.   :  -1.00         Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.:   0.00         1st Qu.: 0.000
##  Median : 1.000   Median :   8.00         Median : 0.000
##  Mean   : 2.318   Mean   :  80.91         Mean   : 0.504
##  3rd Qu.: 4.000   3rd Qu.:  93.26         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   Min.   :0.0000
##  1st Qu.: 2.00   Class :character   FALSE:9462      1st Qu.:0.0000
##  Median : 2.00   Mode  :character   TRUE :2868      Median :0.0000
##  Mean   : 4.07                                      Mean   :0.1547
##  3rd Qu.: 4.00                                      3rd Qu.:0.0000
##  Max.   :20.00                                      Max.   :1.0000
```

```
# Find the mean of numeric columns
colMeans(data[sapply(data, is.numeric)])
```

```
##          Administrative Administrative_Duration           Informational
##            2.317798e+00            8.090618e+01            5.039786e-01
##  Informational_Duration          ProductRelated ProductRelated_Duration
##            3.450639e+01            3.176388e+01            1.196037e+03
##             BounceRates               ExitRates              PageValues
##            2.215246e-02            4.300254e-02            5.889258e+00
##              SpecialDay        OperatingSystems                 Browser
##            6.142741e-02            2.124006e+00            2.357097e+00
##                  Region             TrafficType                 Revenue
##            3.147364e+00            4.069586e+00            1.547445e-01
```

```
# Standard Deviation
sapply(data, sd)
```

```
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion

##          Administrative Administrative_Duration           Informational
##            3.320867e+00            1.767600e+02            1.269980e+00
##  Informational_Duration          ProductRelated ProductRelated_Duration
##            1.407455e+02            4.446507e+01            1.913285e+03
##             BounceRates               ExitRates              PageValues
##            4.839963e-02            4.849974e-02            1.856844e+01
##              SpecialDay                   Month        OperatingSystems
##            1.989173e-01                      NA            9.113248e-01
##                 Browser                  Region             TrafficType
##            1.717277e+00            2.401591e+00            4.025169e+00
##             VisitorType                 Weekend                 Revenue
##                      NA            4.225086e-01            3.616756e-01
```

```
# Kurtosis
kurtosis(df, na.rm=FALSE)
```

```
##          Administrative Administrative_Duration           Informational
##                7.700779               53.547749               29.932131
##  Informational_Duration          ProductRelated ProductRelated_Duration
##               79.286648               34.217805              140.210828
##             BounceRates               ExitRates              PageValues
##               10.762924                7.045883               68.608594
##              SpecialDay        OperatingSystems                 Browser
##               12.909153               13.452116               15.741078
##                  Region             TrafficType
##                2.850893                6.477813
```

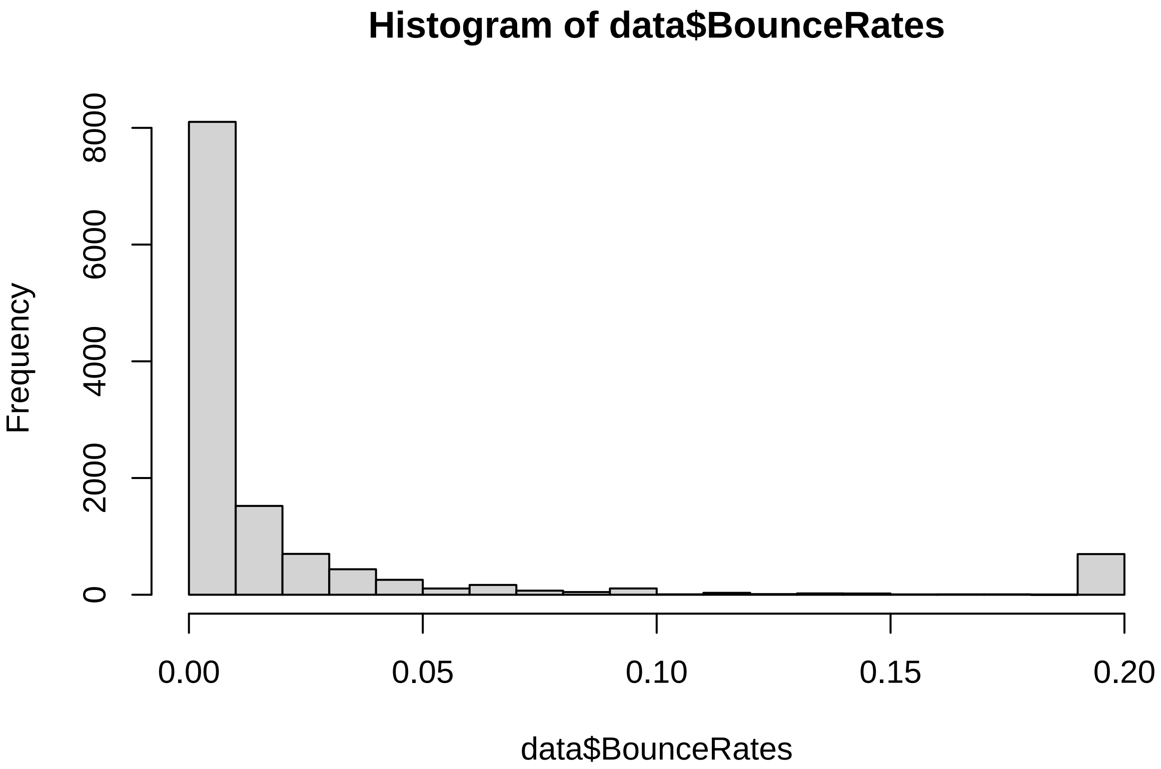
```
# skewness OF NUMERICAL COLUMNS
skewness(df, na.rm=FALSE)
```

```
##          Administrative Administrative_Duration           Informational
##               1.9597507               5.6154663               4.0366232
##  Informational_Duration          ProductRelated ProductRelated_Duration
##               7.5781629               4.3421597               7.2649327
##             BounceRates               ExitRates              PageValues
##               2.9537840               2.1537141               6.3821877
##              SpecialDay        OperatingSystems                 Browser
##               3.3022649               2.0660337               3.2419552
##                  Region             TrafficType
##               0.9834295               1.9627479
```

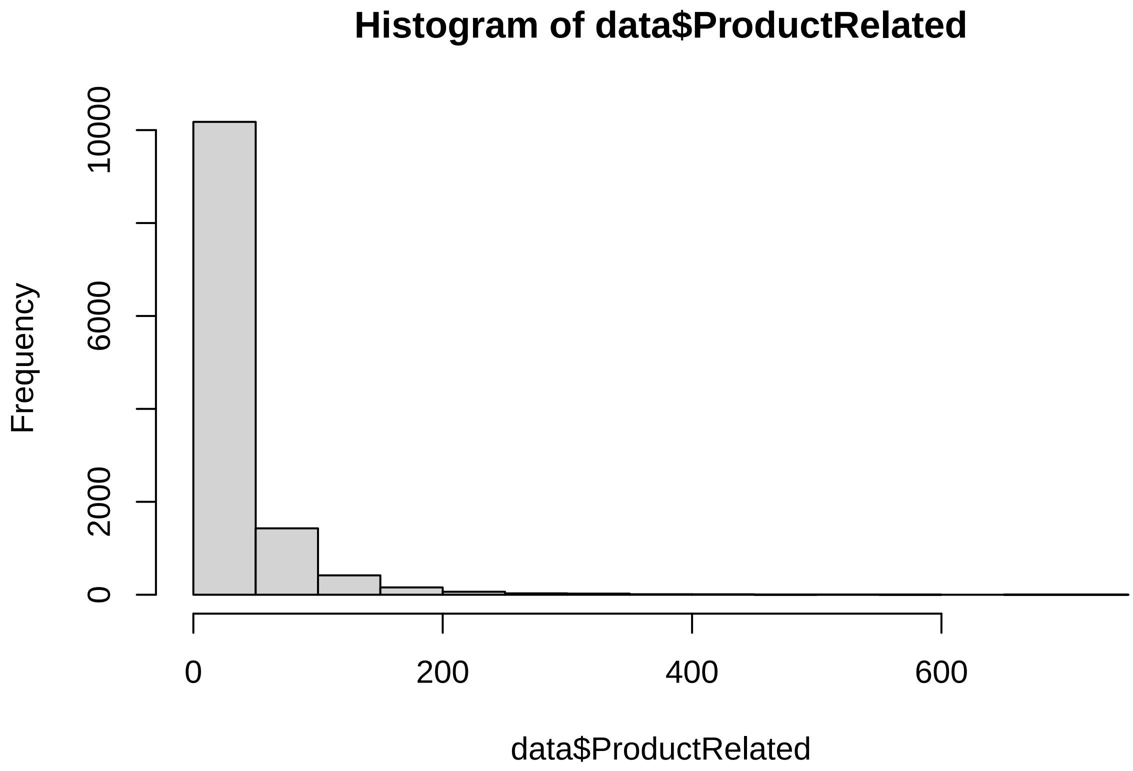
```
#data types
str(data)
```

```
## ’data.frame':    12330 obs. of  18 variables:
##  $ Administrative         : num  0 0 0 0 0 0 0 1 0 0 ...
##  $ Administrative_Duration: num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ Informational          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Informational_Duration : num  0 0 -1 0 0 0 -1 -1 0 0 ...
##  $ ProductRelated         : num  1 2 1 2 10 19 1 1 2 3 ...
##  $ ProductRelated_Duration: num  0 64 -1 2.67 627.5 ...
##  $ BounceRates            : num  0.2 0 0.2 0.05 0.02 ...
##  $ ExitRates              : num  0.2 0.1 0.2 0.14 0.05 ...
##  $ PageValues             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ SpecialDay             : num  0 0 0 0 0 0 0.4 0 0.8 0.4 ...
##  $ 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                : num  0 0 0 0 0 0 0 0 0 0 ...
```

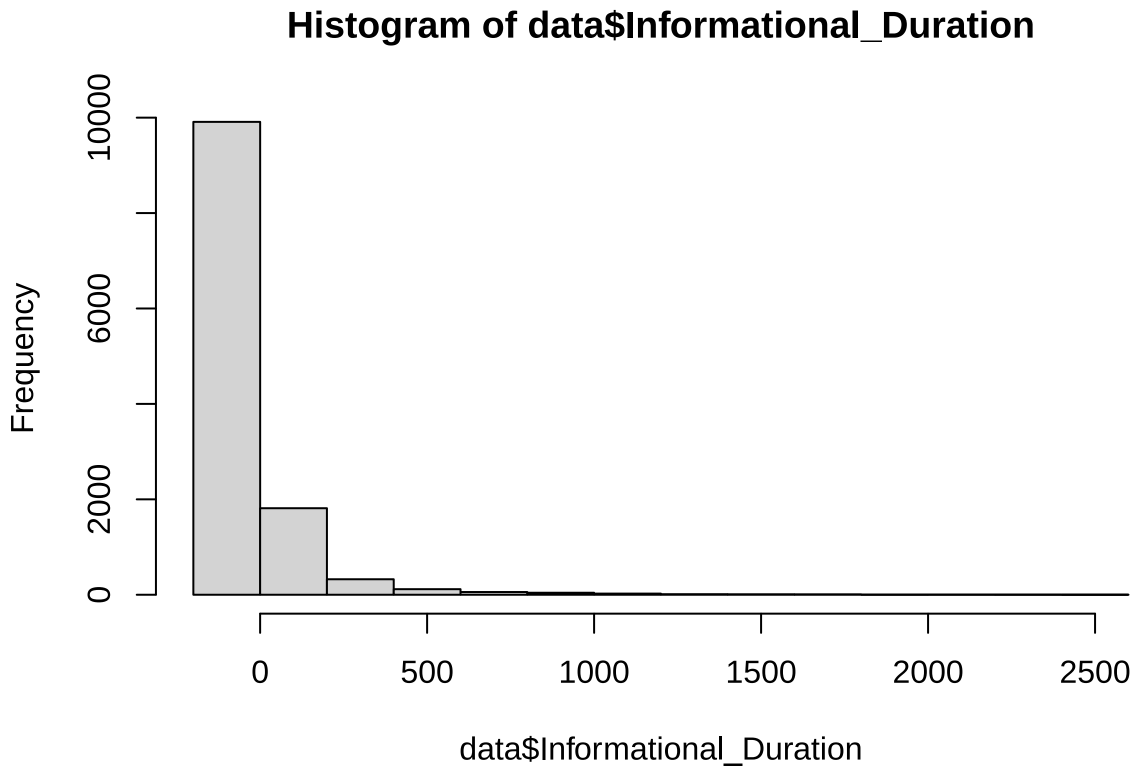
```
# data vs BounceRates
hist(data$BounceRates)
```



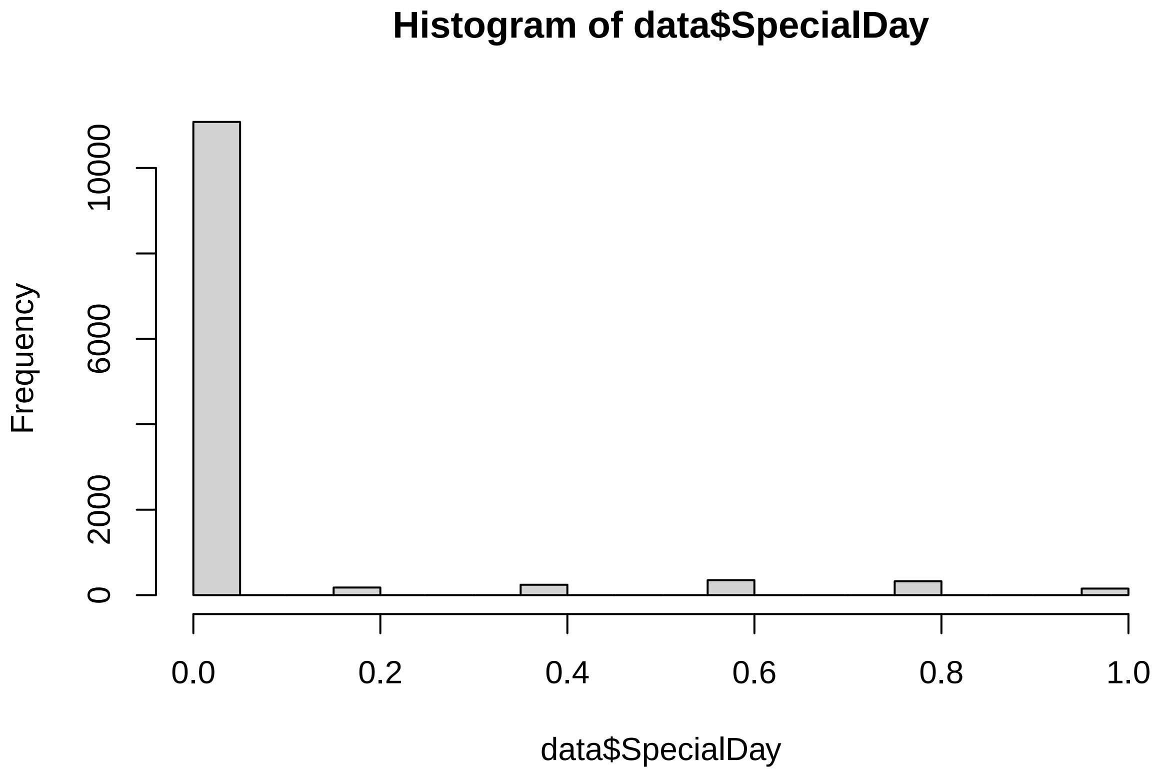
```
# data vs ProductRelated
hist(data$ProductRelated)
```



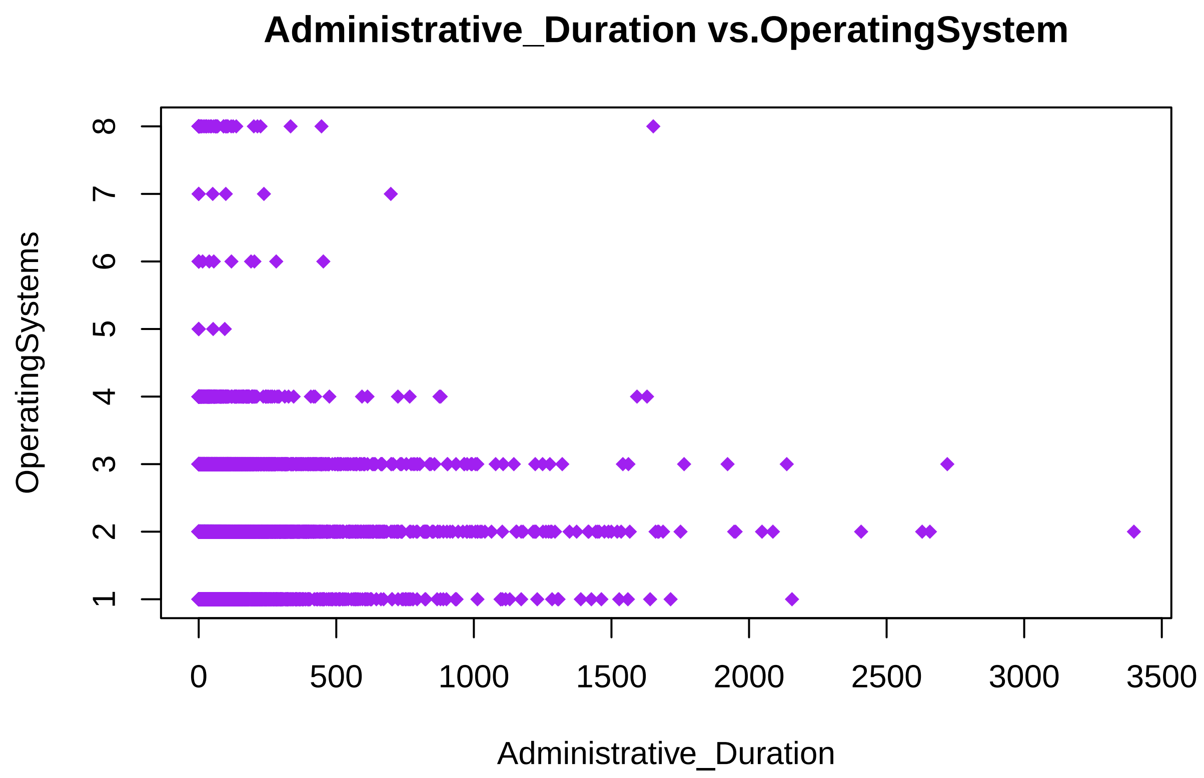
```
#data vs Informational_Duration

hist(data$Informational_Duration)
```



```
# data vs SpecialDay
hist(data$SpecialDay)
```



```
# Scatter plot :Administrative_Duration vs.OperatingSystem
plot(data$Administrative_Duration, data$OperatingSystems, pch=18, col='purple',
     main='Administrative_Duration vs.OperatingSystem',
     xlab='Administrative_Duration', ylab='OperatingSystems')
```

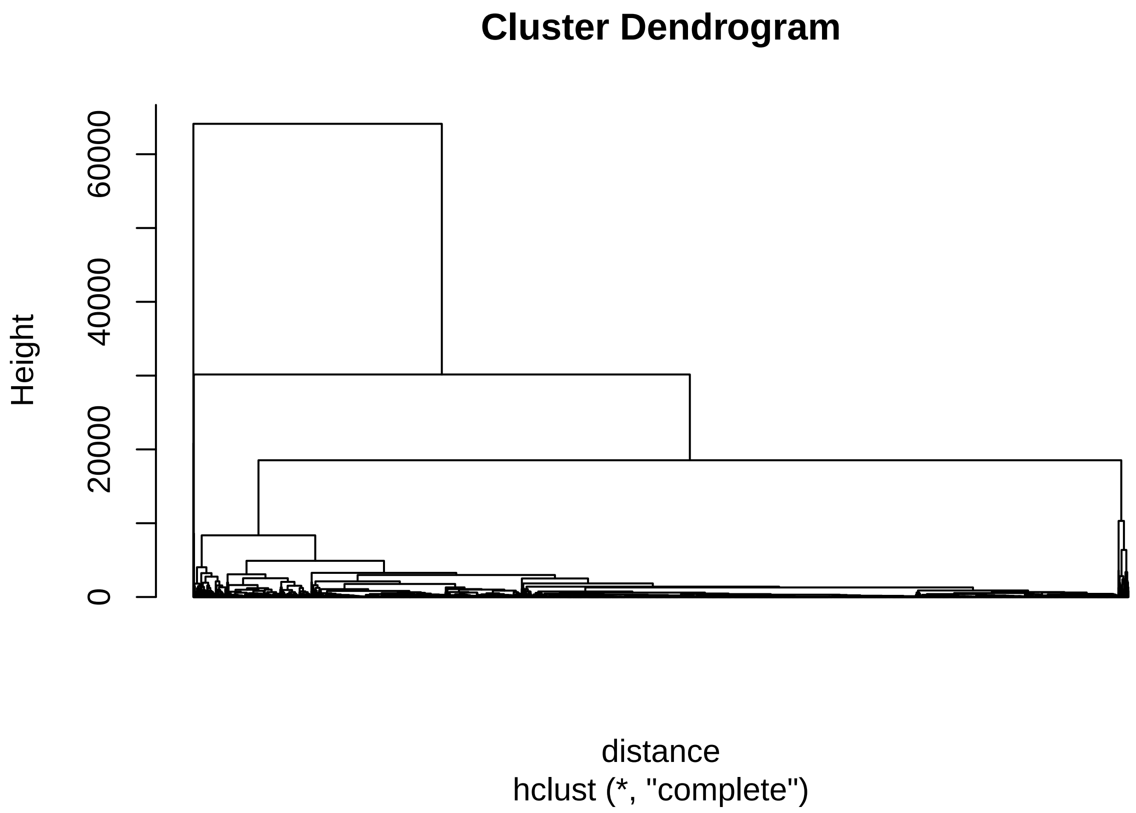


```
# MULTIVARIATE ANALYSIS OF NUMERICAL COLUMNS
corrr::correlate(df, method = "pearson")
```

```
##
## Correlation method: 'pearson'
## Missing treated using: 'pairwise.complete.obs'

## # A tibble: 14 x 15
##    term           Administrative Administrative_~ Informational Informational_D~
##    <chr>                   <dbl>            <dbl>         <dbl>            <dbl>
##  1 Administrative       NA                0.601         0.377            0.256
##  2 Administrativ~        0.601           NA             0.303            0.238
##  3 Informational         0.377            0.303        NA                0.619
##  4 Informational~        0.256            0.238         0.619           NA
##  5 ProductRelated        0.431            0.289         0.374            0.280
##  6 ProductRelate~        0.374            0.355         0.387            0.347
##  7 BounceRates          -0.223           -0.144        -0.116           -0.0741
##  8 ExitRates            -0.316           -0.206        -0.164           -0.105
##  9 PageValues            0.0988           0.0675        0.0485           0.0308
## 10 SpecialDay           -0.0950          -0.0735       -0.0483          -0.0307
## 11 OperatingSyst~       -0.00646         -0.00742      -0.00943         -0.00959
## 12 Browser              -0.0252          -0.0155       -0.0383          -0.0193
## 13 Region               -0.00568         -0.00568      -0.0294          -0.0272
## 14 TrafficType          -0.0337          -0.0145       -0.0345          -0.0247
## # ... with 10 more variables: ProductRelated <dbl>,
## #   ProductRelated_Duration <dbl>, BounceRates <dbl>, ExitRates <dbl>,
## #   PageValues <dbl>, SpecialDay <dbl>, OperatingSystems <dbl>, Browser <dbl>,
## #   Region <dbl>, TrafficType <dbl>
```

```
# UNSUPERVISED LEARNING: Hierarchical clustering
## Get Euclidean distance between players
distance  <-  dist(df,method = "euclidean")
```

```
# Hierarchical clustering
## Perform Hierarchical clustering
## Input : Distance Matrix and linkage method.
hclust_df  <-  hclust(distance,method = "complete")

# Plot the obtained dendrogram
plot(hclust_df, cex = 0.02, hang = -20)
```

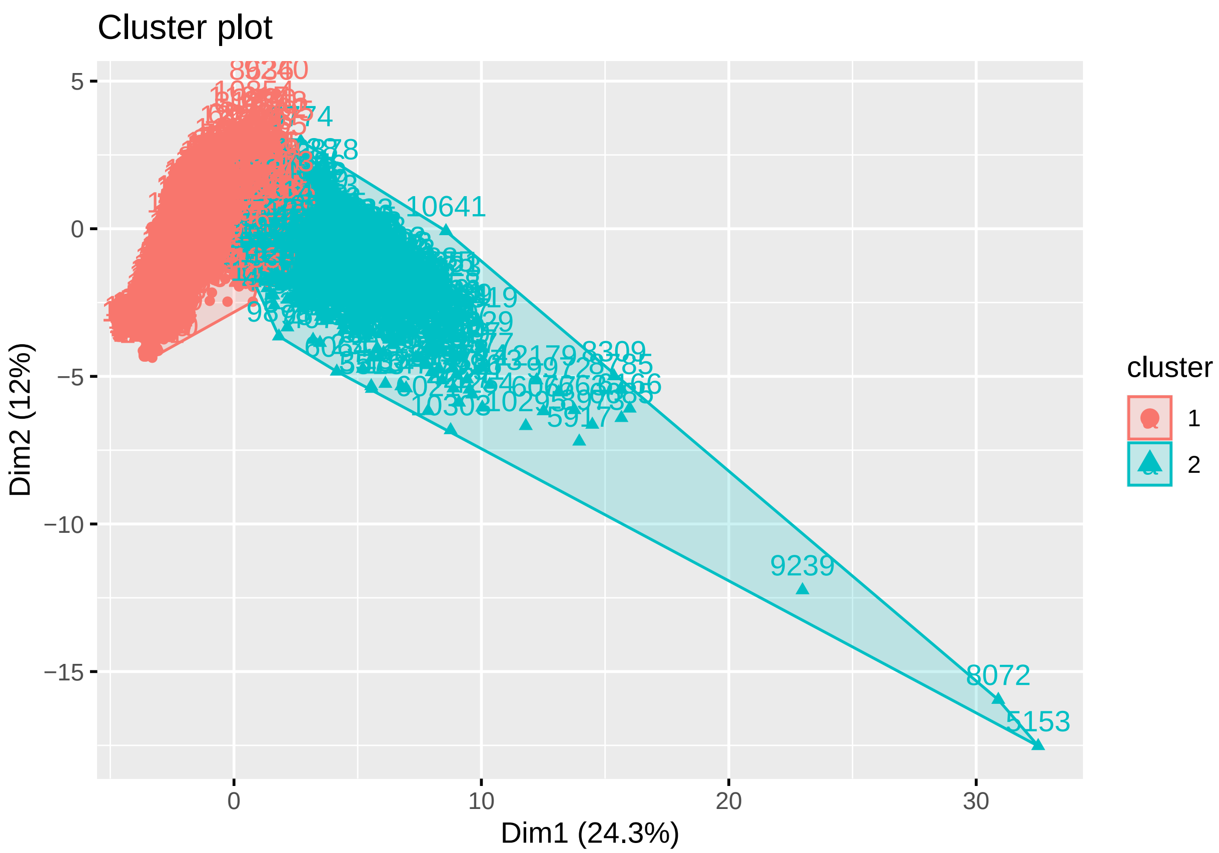


```
# Scale
df <- scale(df)
```

```
# Computing k-means clustering in R
k2 <- kmeans(df, centers = 2, nstart = 25)
str(k2)
```

```
## List of 9
##  $ cluster     : int [1:12330] 1 1 1 1 1 1 1 1 1 1 ...
##  $ centers     : num [1:2, 1:14] -0.262 1.474 -0.216 1.215 -0.254 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:14] "Administrative" "Administrative_Duration" "Informational" "Informational_Dura
##  $ totss       : num 172606
##  $ withinss    : num [1:2] 102363 46582
##  $ tot.withinss: num 148944
##  $ betweenss   : num 23662
##  $ size        : int [1:2] 10468 1862
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
# Cluster plot
fviz_cluster(k2, data = df)
```

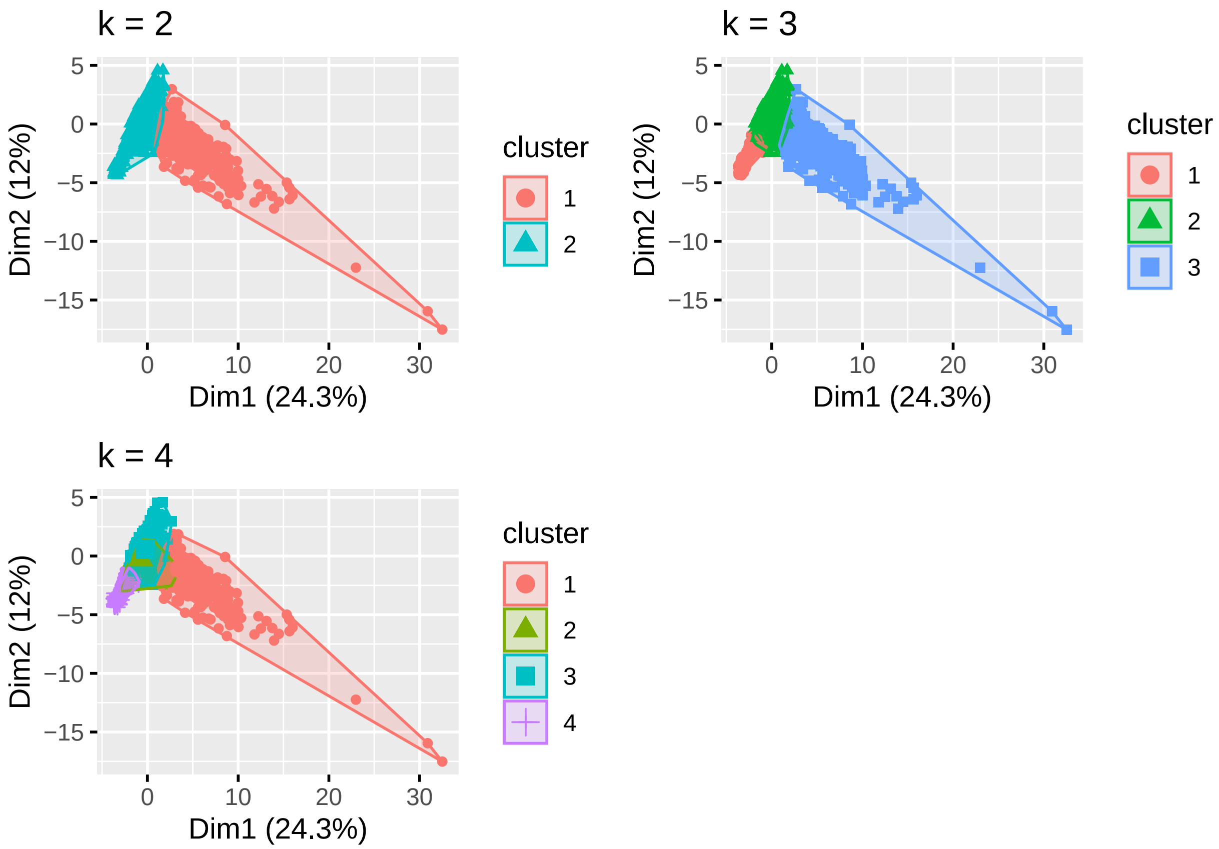


```
k2 <- kmeans(df, centers = 2, nstart = 25)
k3 <- kmeans(df, centers = 3, nstart = 25)
```

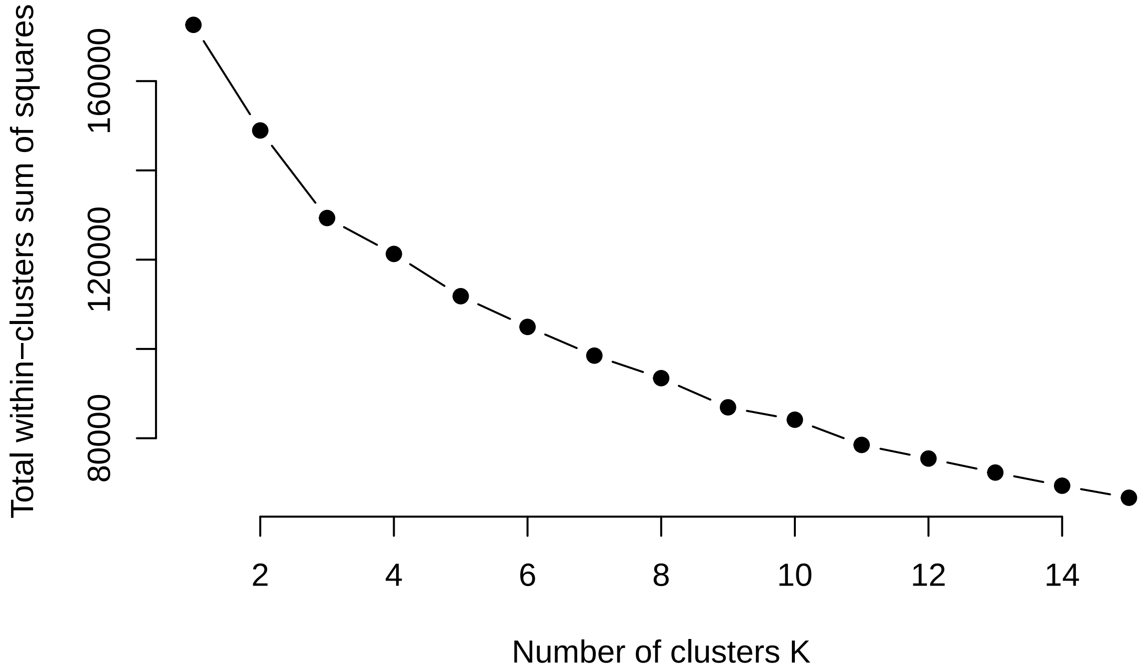
```
k4 <- kmeans(df, centers = 4, nstart = 25)
```

```
# plots to compare
p1 <- fviz_cluster(k2, geom = "point", data = df) + ggtitle("k = 2")
p2 <- fviz_cluster(k3, geom = "point",  data = df) + ggtitle("k = 3")
```

```
p3 <- fviz_cluster(k4, geom = "point",  data = df) + ggtitle("k = 4")
grid.arrange(p1, p2, p3, nrow = 2)
```

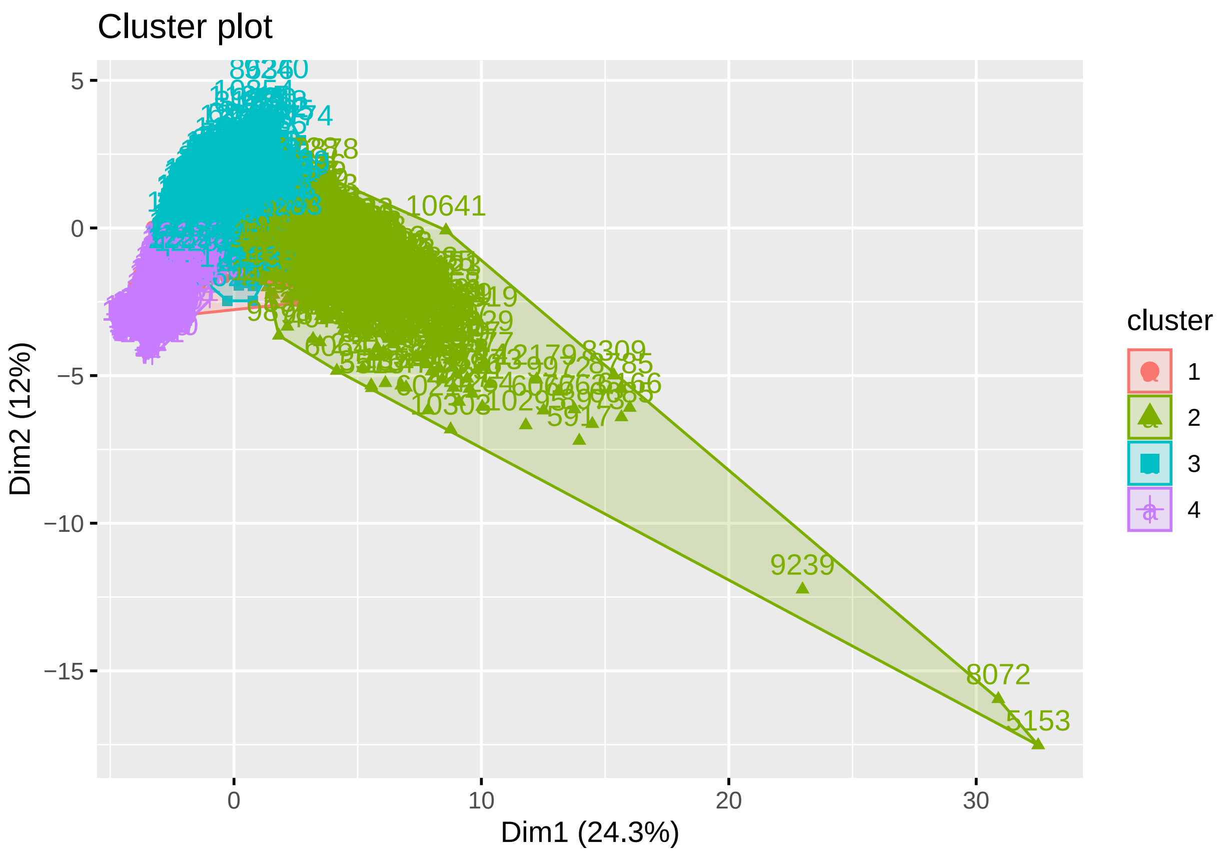


```
# SELCTING THE BEST VALUE OF K
set.seed(123)

# function to compute total within-cluster sum of square
wss <- function(k) {
  kmeans(df, k, nstart = 10 )$tot.withinss}

# Compute and plot wss for k = 1 to k = 15
k.values <- 1:15

# extract wss for 2-15 clusters
wss_values <- map_dbl(k.values, wss)

plot(k.values, wss_values,
       type="b", pch = 19, frame = FALSE,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares")
```



```
# Compute k-means clustering with k = 4
set.seed(123)
final <- kmeans(df, 4, nstart = 25)
```

```
fviz_cluster(final, data = df)
```



```
# Supervised Learning  SUPPORT VECTOR MACHINE
# Splitting data into training and test data sets

indxTrain <- createDataPartition(y =data$Revenue,p = 0.75,list = FALSE)
training <- data[indxTrain,]
testing <- data[-indxTrain,]
```

```
# Checking dimensions of the split
dim(training)
```

```
## [1] 9248   18
```

```
dim(testing)
```

```
## [1] 3082   18
```

```
training[["Revenue"]] = factor(training[["Revenue"]])
```

```
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
```

```
svm_Linear <- train(Revenue ~., data = training, method = "svmLinear",
trControl=trctrl,
preProcess = c("center", "scale"),
tuneLength = 10)
```

```
svm_Linear
```

```
## Support Vector Machines with Linear Kernel
##
## 9248 samples
##   17 predictor
##    2 classes: '0', '1'
##
## Pre-processing: centered (26), scaled (26)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 8323, 8324, 8324, 8323, 8322, 8323, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.8853834  0.4628049
##
## Tuning parameter 'C' was held constant at a value of 1
```

```
# We can use the predict() method for predicting results as shown below.
# We pass 2 arguements, our trained model and our testing data frame.
# ---
#
test_pred <- predict(svm_Linear, newdata = testing)
```

```
confusionMatrix(table(test_pred, testing$Revenue))
```

```
## Confusion Matrix and Statistics
##
##
## test_pred    0    1
##         0 2552  301
##         1   55  174
##
##                Accuracy : 0.8845
##                  95% CI : (0.8727, 0.8956)
##     No Information Rate : 0.8459
##     P-Value [Acc > NIR] : 4.311e-10
##
##                   Kappa : 0.438
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9789
##             Specificity : 0.3663
##          Pos Pred Value : 0.8945
##          Neg Pred Value : 0.7598
##              Prevalence : 0.8459
##          Detection Rate : 0.8280
##    Detection Prevalence : 0.9257
##       Balanced Accuracy : 0.6726
##
##        'Positive' Class : 0
##
```