week13IP

Alex

09/07/2021

IMPORTING OUR DATASET

```
library("data.table")
data <- fread("http://bit.ly/EcommerceCustomersDataset")
```

Previewing our data set

```
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
```

Checking data types

```
typeof(data)
```

```
## [1] "list"
```

Checking the data shape of our data set

```
nrow(data)
```

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

```
ncol(data)
```

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

```
#From our data set, we can see that we have a total of 12330 rows and a total of 18 columns.
```

Checking for missing values

```
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
```

When we check for null values in each column, we can see that there are no missing values in our data set.

Dropping Null values

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

Checking again for null values

```
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
```

Checking for duplicates

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

duplicated_rows
```

```
##      Administrative Administrative_Duration Informational
##   1:              0                       0             0
##   2:              0                       0             0
##   3:              0                       0             0
##   4:              0                       0             0
##   5:              0                       0             0
##  ---
## 113:              0                       0             0
## 114:              0                       0             0
## 115:              0                       0             0
## 116:              0                       0             0
## 117:              0                       0             0
##      Informational_Duration ProductRelated ProductRelated_Duration BounceRates
##   1:                      0              1                       0         0.2
##   2:                      0              1                       0         0.2
##   3:                      0              1                       0         0.2
##   4:                      0              1                       0         0.2
##   5:                      0              1                       0         0.2
##  ---
## 113:                      0              1                       0         0.2
## 114:                      0              1                       0         0.2
## 115:                      0              1                       0         0.2
## 116:                      0              1                       0         0.2
## 117:                      0              1                       0         0.2
##      ExitRates PageValues SpecialDay Month OperatingSystems Browser Region
##   1:       0.2          0          0   Feb                1       1      1
##   2:       0.2          0          0   Feb                3       2      3
##   3:       0.2          0          0   Mar                1       1      1
##   4:       0.2          0          0   Mar                2       2      4
##   5:       0.2          0          0   Mar                3       2      3
##  ---
## 113:       0.2          0          0   Dec                1       1      1
## 114:       0.2          0          0   Dec                1       1      4
## 115:       0.2          0          0   Dec                1       1      1
## 116:       0.2          0          0   Dec                1      13      9
## 117:       0.2          0          0   Dec                8      13      9
##      TrafficType       VisitorType Weekend Revenue
##   1:           3 Returning_Visitor   FALSE   FALSE
##   2:           3 Returning_Visitor   FALSE   FALSE
##   3:           1 Returning_Visitor    TRUE   FALSE
##   4:           1 Returning_Visitor   FALSE   FALSE
```

```
##   5:           1 Returning_Visitor   FALSE   FALSE
##  ---
## 113:           2       New_Visitor   FALSE   FALSE
## 114:           1 Returning_Visitor    TRUE   FALSE
## 115:           3 Returning_Visitor   FALSE   FALSE
## 116:          20 Returning_Visitor   FALSE   FALSE
## 117:          20             Other   FALSE   FALSE
```

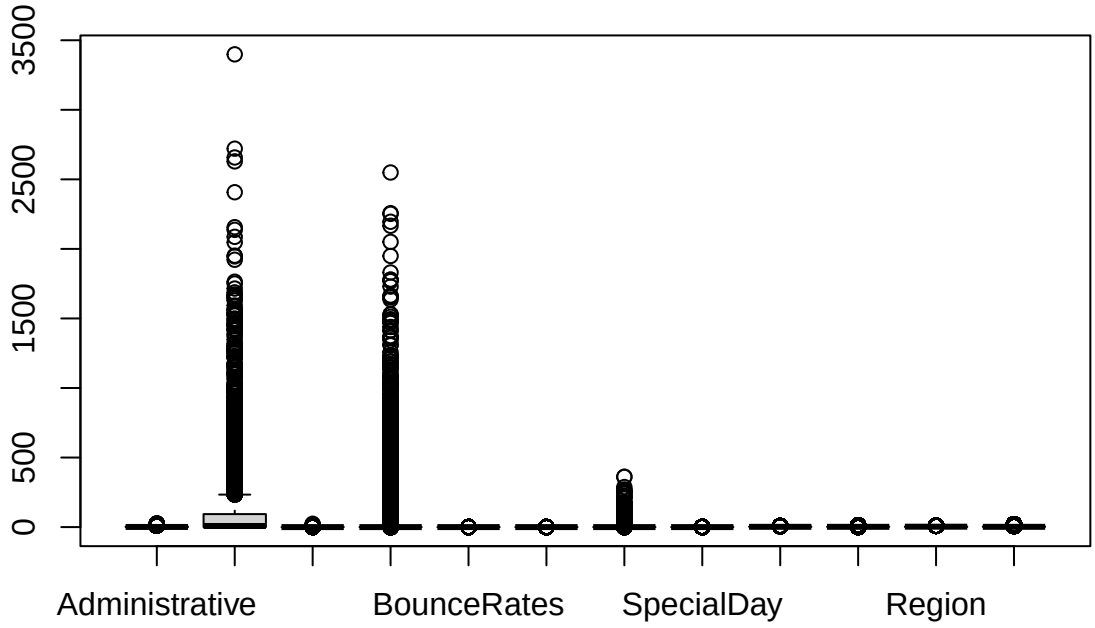
From the above code, we can clearly see the output which shows that there are no duplicates in our data set.

Checking for Outliers

```
#creating a variable with only numeric columns
library(tidyverse)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## 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::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```
data2 <-data %>% select(1,2,3,4,7,8,9,10,12,13,14,15)

#Previewing outliers for numeric columns using box plots
boxplot(data2)
```



We can observe some outliers but they are the actual representation of the data so we will not drop them

Exploratory Data Analysis

Univariate Analysis

Finding the mean

```
lapply(data2,FUN=mean)
```

```
## $Administrative
## [1] 2.317798
##
## $Administrative_Duration
## [1] 80.90618
##
## $Informational
## [1] 0.5039786
##
## $Informational_Duration
## [1] 34.50639
##
## $BounceRates
## [1] 0.02215246
```

```
##
## $ExitRates
## [1] 0.04300254
##
## $PageValues
## [1] 5.895952
##
## $SpecialDay
## [1] 0.06149724
##
## $OperatingSystems
## [1] 2.124147
##
## $Browser
## [1] 2.357584
##
## $Region
## [1] 3.148019
##
## $TrafficType
## [1] 4.070477
```

Finding the median

```
lapply(data2,FUN=median)
```

```
## $Administrative
## [1] 1
##
## $Administrative_Duration
## [1] 8
##
## $Informational
## [1] 0
##
## $Informational_Duration
## [1] 0
##
## $BounceRates
## [1] 0.003119412
##
## $ExitRates
## [1] 0.02512449
##
## $PageValues
## [1] 0
##
## $SpecialDay
## [1] 0
##
## $OperatingSystems
## [1] 2
##
```

```
## $Browser
## [1] 2
##
## $Region
## [1] 3
##
## $TrafficType
## [1] 2
```

Finding the mode

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

lapply(data,FUN=getmode)
```

```
## $Administrative
## [1] 0
##
## $Administrative_Duration
## [1] 0
##
## $Informational
## [1] 0
##
## $Informational_Duration
## [1] 0
##
## $ProductRelated
## [1] 1
##
## $ProductRelated_Duration
## [1] 0
##
## $BounceRates
## [1] 0
##
## $ExitRates
## [1] 0.2
##
## $PageValues
## [1] 0
##
## $SpecialDay
## [1] 0
##
## $Month
## [1] "May"
##
## $OperatingSystems
## [1] 2
```

```
##
## $Browser
## [1] 2
##
## $Region
## [1] 1
##
## $TrafficType
## [1] 2
##
## $VisitorType
## [1] "Returning_Visitor"
##
## $Weekend
## [1] FALSE
##
## $Revenue
## [1] FALSE
```

Measures of Dispersion

Minimum

```
lapply(data2,FUN=min)
```

```
## $Administrative
## [1] 0
##
## $Administrative_Duration
## [1] -1
##
## $Informational
## [1] 0
##
## $Informational_Duration
## [1] -1
##
## $BounceRates
## [1] 0
##
## $ExitRates
## [1] 0
##
## $PageValues
## [1] 0
##
## $SpecialDay
## [1] 0
##
## $OperatingSystems
## [1] 1
##
## $Browser
## [1] 1
```

```
##
## $Region
## [1] 1
##
## $TrafficType
## [1] 1
```

Maximum

```
lapply(data2,FUN=max)
```

```
## $Administrative
## [1] 27
##
## $Administrative_Duration
## [1] 3398.75
##
## $Informational
## [1] 24
##
## $Informational_Duration
## [1] 2549.375
##
## $BounceRates
## [1] 0.2
##
## $ExitRates
## [1] 0.2
##
## $PageValues
## [1] 361.7637
##
## $SpecialDay
## [1] 1
##
## $OperatingSystems
## [1] 8
##
## $Browser
## [1] 13
##
## $Region
## [1] 9
##
## $TrafficType
## [1] 20
```

Range

```
lapply(data2,FUN=range)
```

```
## $Administrative
```

```
## [1]  0 27
##
## $Administrative_Duration
## [1]   -1.00 3398.75
##
## $Informational
## [1]  0 24
##
## $Informational_Duration
## [1]   -1.000 2549.375
##
## $BounceRates
## [1] 0.0 0.2
##
## $ExitRates
## [1] 0.0 0.2
##
## $PageValues
## [1]   0.0000 361.7637
##
## $SpecialDay
## [1] 0 1
##
## $OperatingSystems
## [1] 1 8
##
## $Browser
## [1]  1 13
##
## $Region
## [1] 1 9
##
## $TrafficType
## [1]  1 20
```

Quantile

```
lapply(data2,FUN=quantile)
```

```
## $Administrative
##   0%  25%  50%  75% 100%
##    0    0    1    4   27
##
## $Administrative_Duration
##      0%     25%     50%     75%    100%
##   -1.00    0.00    8.00   93.50 3398.75
##
## $Informational
##   0%  25%  50%  75% 100%
##    0    0    0    0   24
##
## $Informational_Duration
##       0%      25%      50%      75%     100%
```

```
##   -1.000    0.000    0.000    0.000 2549.375
##
## $BounceRates
##          0%         25%         50%         75%        100%
## 0.000000000 0.000000000 0.003119412 0.016683674 0.200000000
##
## $ExitRates
##         0%        25%        50%        75%       100%
## 0.00000000 0.01428571 0.02512449 0.05000000 0.20000000
##
## $PageValues
##       0%      25%      50%      75%     100%
##   0.0000   0.0000   0.0000   0.0000 361.7637
##
## $SpecialDay
##   0%  25%  50%  75% 100%
##    0    0    0    0    1
##
## $OperatingSystems
##   0%  25%  50%  75% 100%
##    1    2    2    3    8
##
## $Browser
##   0%  25%  50%  75% 100%
##    1    2    2    2   13
##
## $Region
##   0%  25%  50%  75% 100%
##    1    1    3    4    9
##
## $TrafficType
##   0%  25%  50%  75% 100%
##    1    2    2    4   20
```

Variance

```
lapply(data2,FUN=var)
```

```
## $Administrative
## [1] 11.04069
##
## $Administrative_Duration
## [1] 31279.61
##
## $Informational
## [1] 1.614682
##
## $Informational_Duration
## [1] 19831.82
##
## $BounceRates
## [1] 0.002345187
##
```

```
## $ExitRates
## [1] 0.002354899
##
## $PageValues
## [1] 345.1393
##
## $SpecialDay
## [1] 0.03960877
##
## $OperatingSystems
## [1] 0.8309524
##
## $Browser
## [1] 2.95162
##
## $Region
## [1] 5.770618
##
## $TrafficType
## [1] 16.19739
```

Standard Deviation

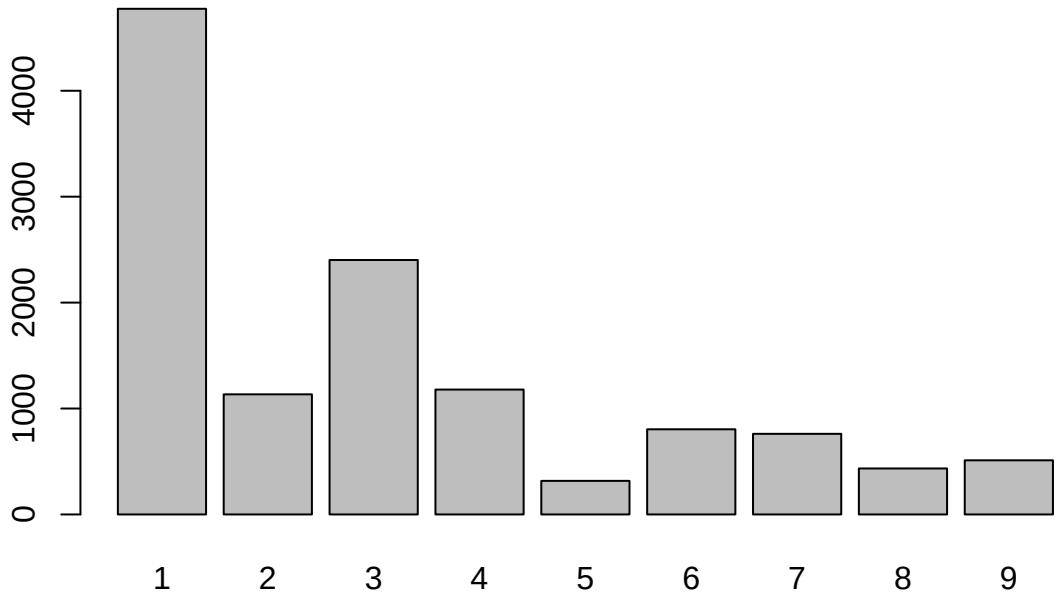
```
lapply(data2,FUN=sd)
```

```
## $Administrative
## [1] 3.322754
##
## $Administrative_Duration
## [1] 176.8604
##
## $Informational
## [1] 1.270701
##
## $Informational_Duration
## [1] 140.8255
##
## $BounceRates
## [1] 0.04842713
##
## $ExitRates
## [1] 0.0485273
##
## $PageValues
## [1] 18.57793
##
## $SpecialDay
## [1] 0.1990195
##
## $OperatingSystems
## [1] 0.9115659
##
## $Browser
```

```
## [1] 1.718028
##
## $Region
## [1] 2.402211
##
## $TrafficType
## [1] 4.024598
```

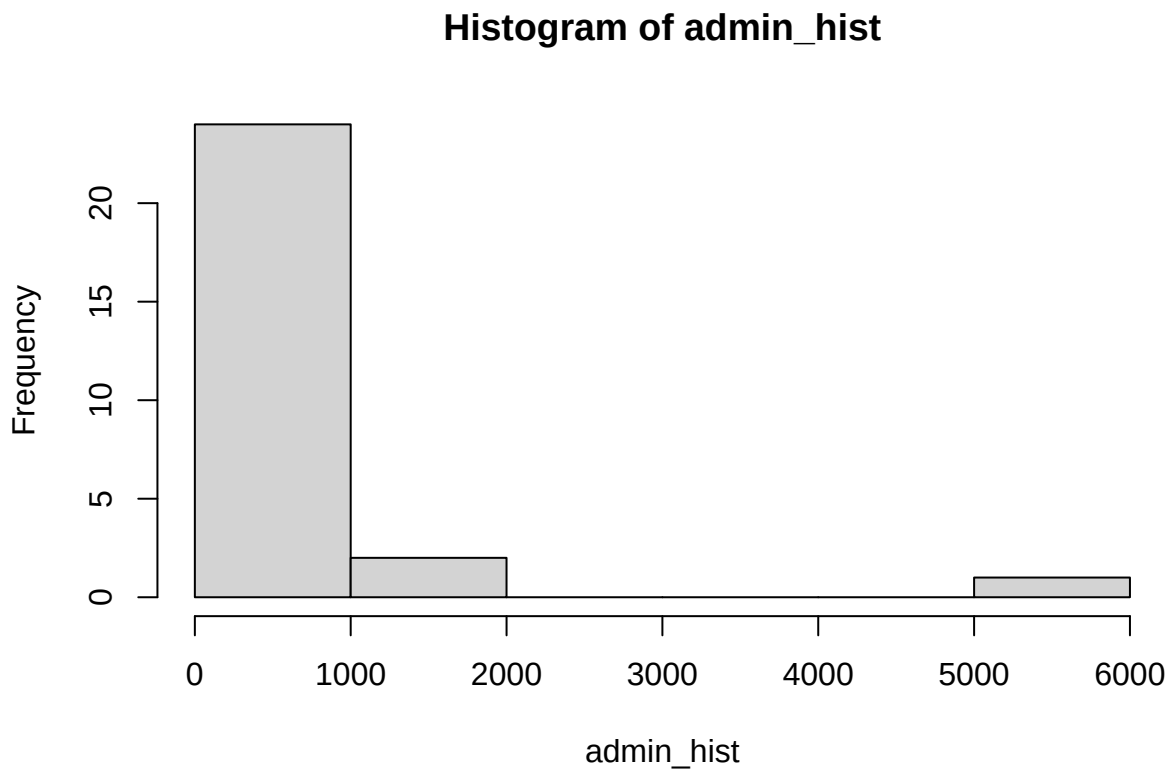
Univariate Graphical

```
region_frequency <- table(data$`Region`)

barplot(region_frequency)
```

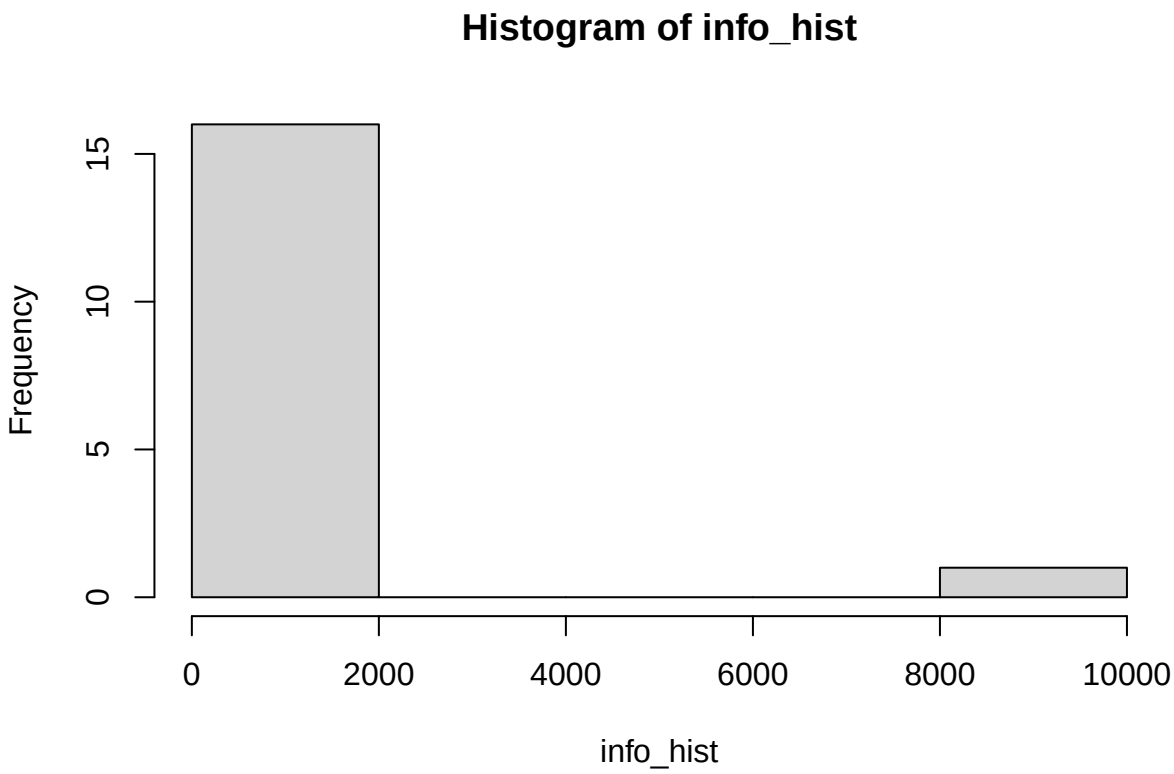


From the plot, we can see that majority are from the region 1

```
admin_hist <- table(data$`Administrative`)

hist(admin_hist)
```




```
info_hist <- table(data$Informational)

hist(info_hist)
```



Bivariate Analysis

Finding the covariance between Product related and product duration

```
prel <- (data$ProductRelated)


pdur  <- (data$ProductRelated)

cov(pdur, prel)
```

```
## [1] 1979.39
```

Finding the covariance between Bounce rates and Exit Rates

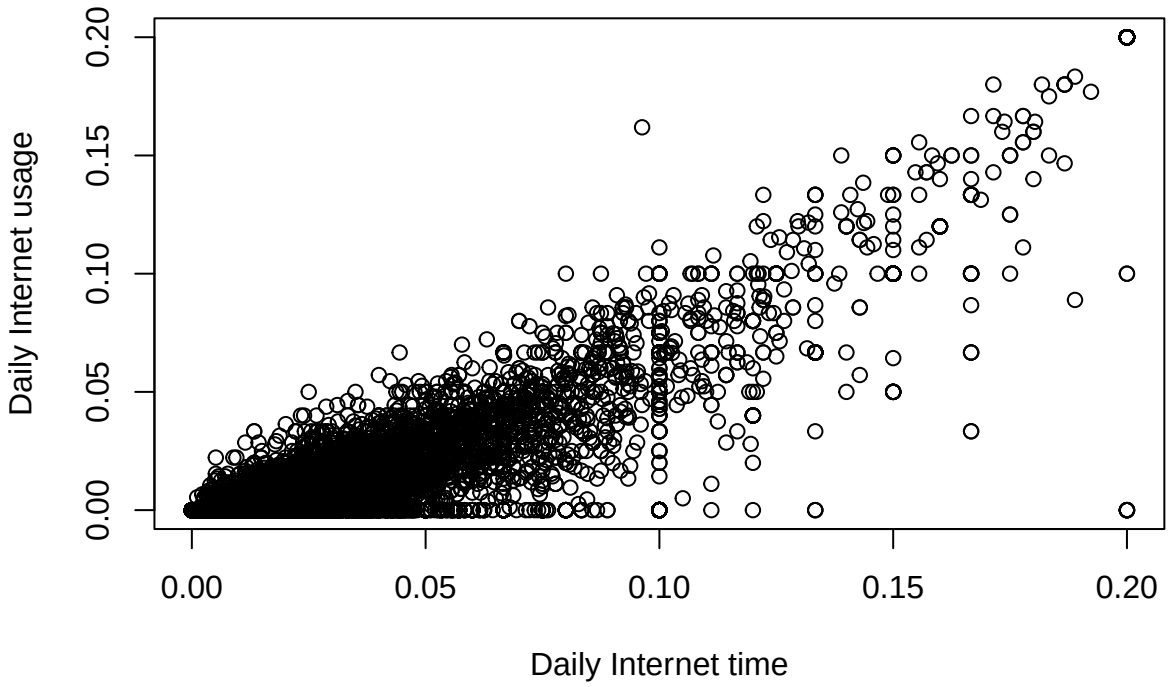
```
brate <- (data$BounceRates)


erate <- (data$ExitRates)

cov(brate, erate)
```

```
## [1] 0.00214661
```

Graphical Techniques

Scatter plot of the covariance between Bounce rates and Exit Rates

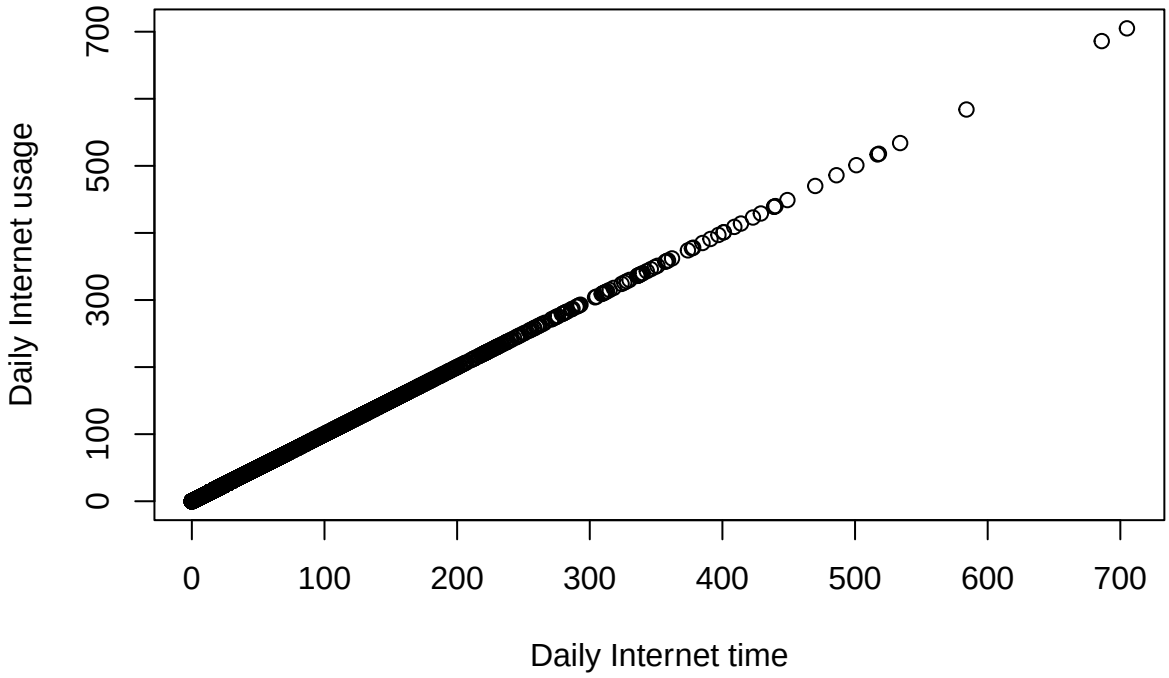
```
plot(erate, brate, xlab="Daily Internet time", ylab="Daily Internet usage")
```



From the above scatter plot, we can see that there is a positive correlation between bounce rate and Exit rate

Scatter plot of the covariance between Product related and Product Duration

```
plot(pdur, prel, xlab="Daily Internet time", ylab="Daily Internet usage")
```



From the above scatter plot, we can see that there is a positive correlation between Product related and product duration

```
data$Month <- as.integer(as.factor(data$Month))
data$VisitorType <- as.integer(as.factor(data$VisitorType))
data$Weekend <- as.integer(as.factor(data$Weekend))
data$Revenue <- as.integer(as.factor(data$Revenue))
```

```
#Previewing the data set
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 VisitorType
```

```
## 1:          0     3                1       1      1           1           3
## 2:          0     3                2       2      1           2           3
## 3:          0     3                4       1      9           3           3
## 4:          0     3                3       2      2           4           3
## 5:          0     3                3       3      1           4           3
## 6:          0     3                2       2      1           3           3
##    Weekend Revenue
## 1:       1       1
## 2:       1       1
## 3:       1       1
## 4:       1       1
## 5:       2       1
## 6:       1       1
```

```
data$Revenue <- as.integer(as.factor(data$Revenue))
```

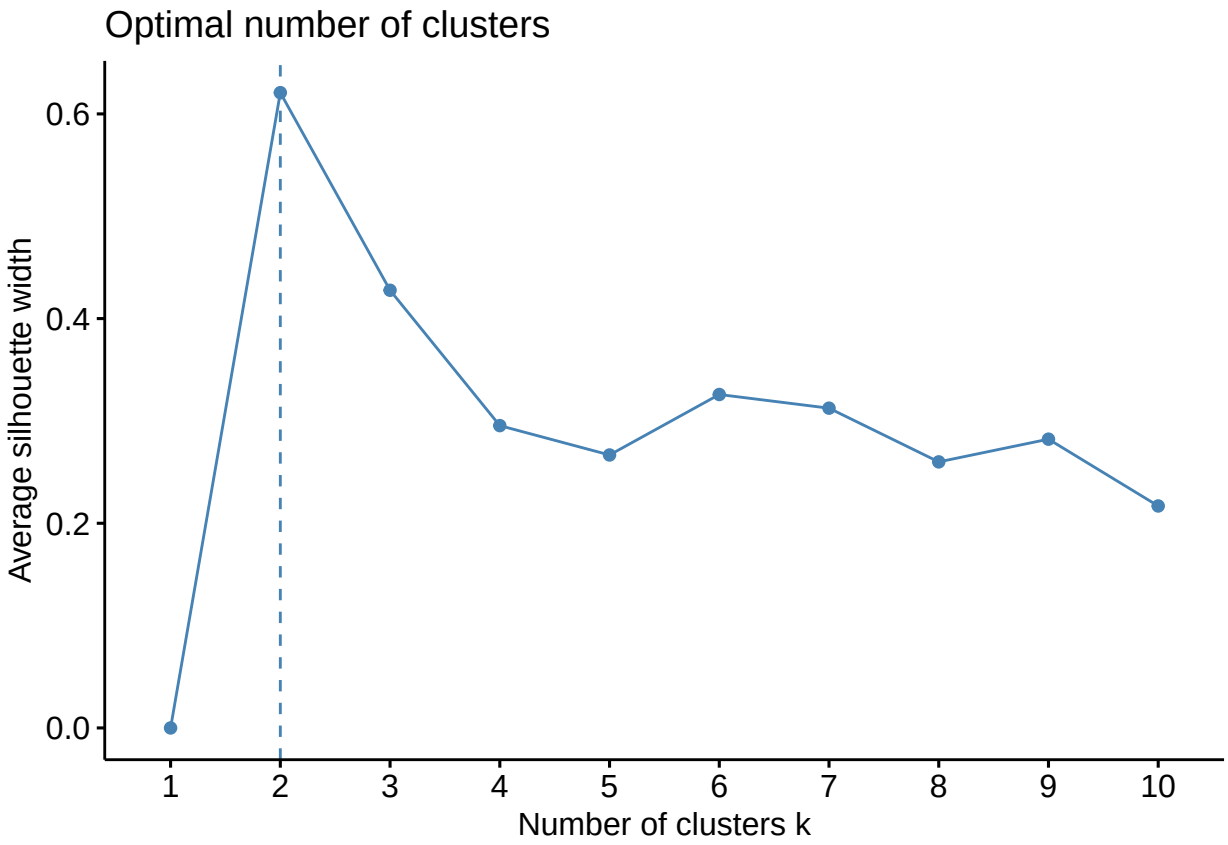
```
normalize <- function(x){
  return ((x-min(x)) / (max(x)-min(x)))}

data$Administrative <- normalize(data$Administrative)
data$Administrative_Duration <- normalize(data$Administrative_Duration)
data$Informational <- normalize(data$Informational)
data$Informational_Duration <- normalize(data$Informational_Duration)
data$ProductRelated <- normalize(data$ProductRelated)
data$ProductRelated_Duration <- normalize(data$ProductRelated_Duration)
data$BounceRates <- normalize(data$BounceRates)
data$ExitRates <- normalize(data$ExitRates)
data$PageValues <- normalize(data$PageValues)
data$SpecialDay <- normalize(data$SpecialDay)
data$OperatingSystems <- normalize(data$OperatingSystems)
data$Browser <- normalize(data$Browser)
data$Region <- normalize(data$Region)
data$TrafficType <- normalize(data$TrafficType)
```

```
library(factoextra)
```

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

```
library(NbClust)
```

```
fviz_nbclust(data, kmeans, method = "silhouette")
```



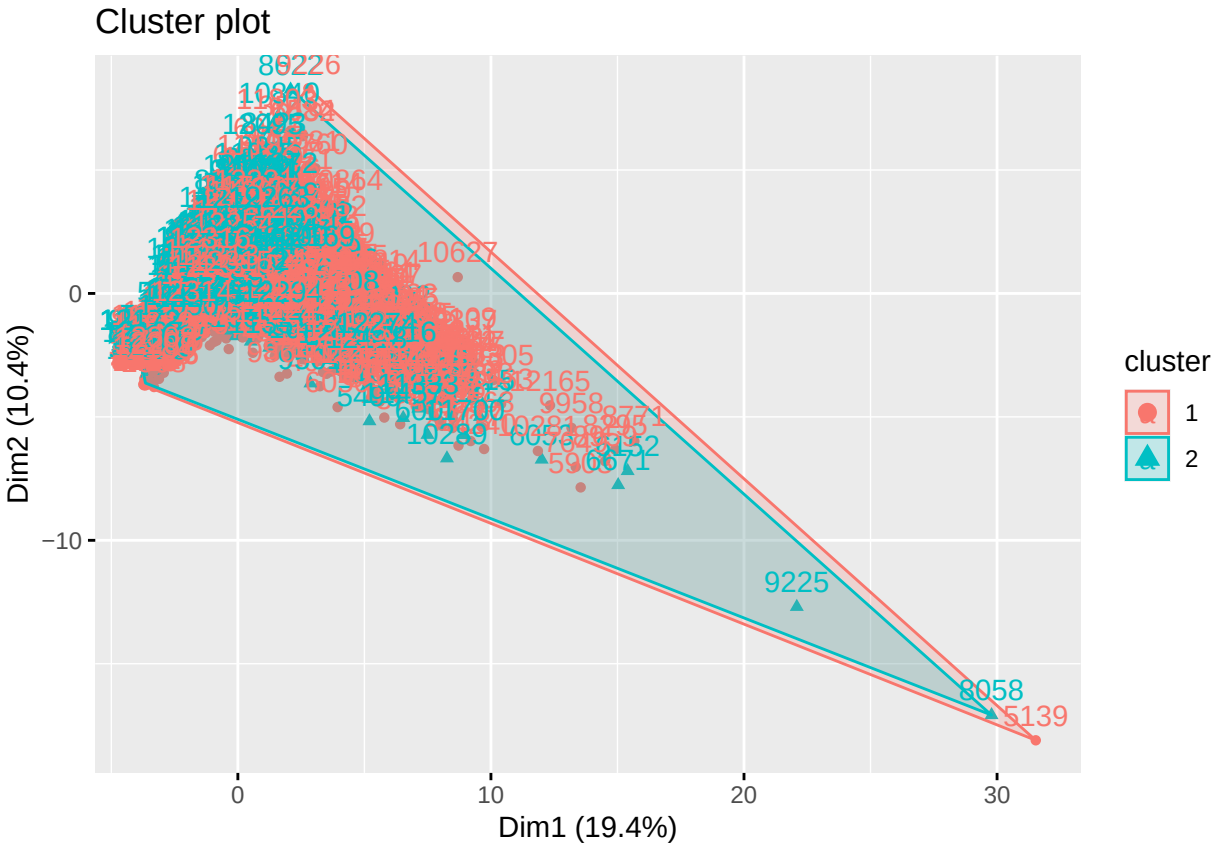
Performing clustering with a k value of 2

```
kmeans_model = kmeans(data, 2)

# Checking the cluster centers for each attribute

kmeans_model$centers
```

```
##   Administrative Administrative_Duration Informational Informational_Duration
## 1     0.08637317              0.02425905    0.02117400             0.01369911
## 2     0.08402711              0.02351715    0.02039805             0.01468809
##   ProductRelated ProductRelated_Duration BounceRates ExitRates PageValues
## 1     0.04586972              0.01909943   0.1105004 0.2142179 0.01631461
## 2     0.04225581              0.01737680   0.1116625 0.2177441 0.01624001
##   SpecialDay    Month OperatingSystems   Browser    Region TrafficType
## 1 0.07488470 7.311321        0.1583408 0.1093990 0.2620283   0.1651992
## 2 0.01548991 2.221542        0.1683306 0.1259606 0.2907511   0.1492492
##   VisitorType  Weekend  Revenue
## 1    2.739308 1.238050 1.162159
## 2    2.644813 1.213977 1.130043
```

Visualizing Kmeans

```
fviz_cluster(kmeans_model, data)
```



##Advantages and Disadvantages of K-Means Clustering

###Advantages

Easy to implement With a large number of variables, K-Means may be computationally faster than hierarchical clustering (if K is small). k-Means may produce Highter clusters than hierarchical clustering An instance can change cluster (move to another cluster) when the centroids are recomputed.

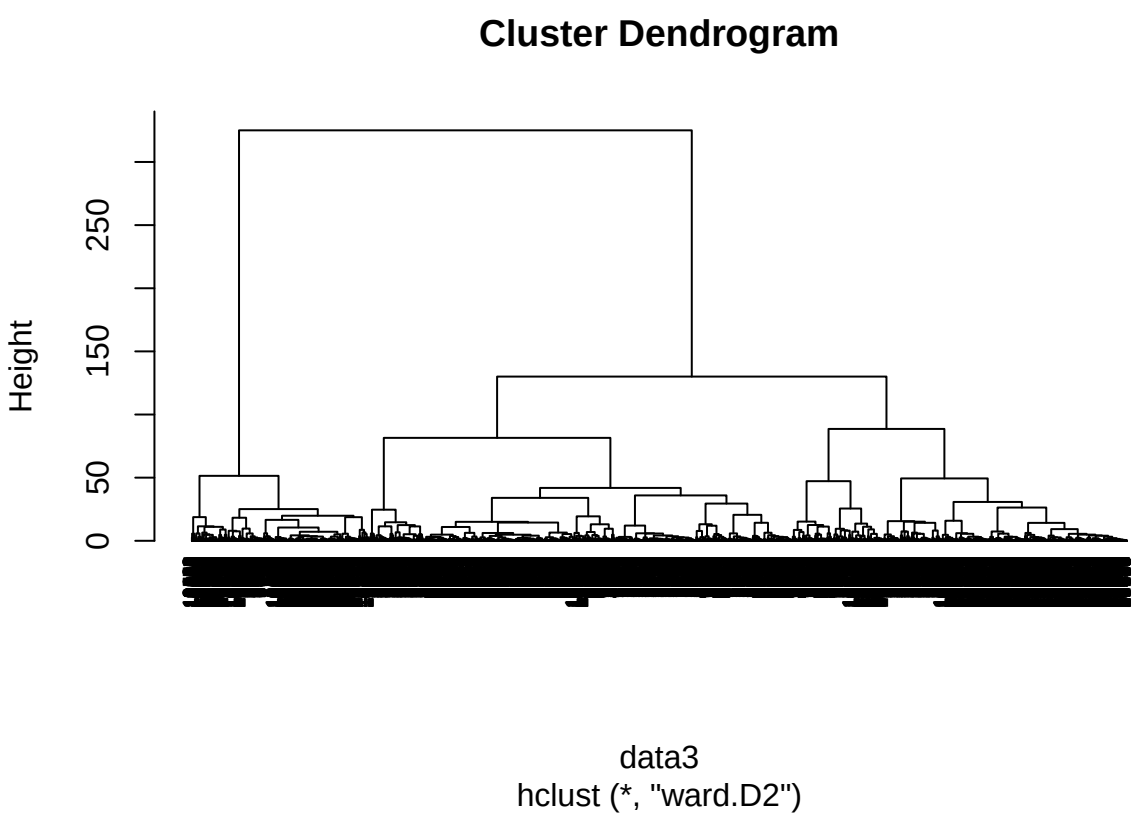
###Disadvantages Difficult to predict the number of clusters (K-Value) Initial seeds have a strong impact on the final results The order of the data has an impact on the final results Sensitive to scale: rescaling your datasets (normalization or standardization) will completely change results.

Hierarchical Clust

```
data3 <- dist(data, method = "euclidean")
```

```
res.hc <- hclust(data3, method = "ward.D2" )
```

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



The three main advantages of using the Hierarchical clustering techniques are as follows;

We do not need to specify the number of clusters required for the algorithm. Hierarchical clustering outputs a hierarchy, ie a structure that is more informative than the unstructured set of flat clusters returned by k-means. It is also easy to implement. Below are the limitations of the hierarchical clustering technique;

There is no mathematical objective for Hierarchical clustering.

All the approaches to calculate the similarity between clusters has its own disadvantages. High space and time complexity for Hierarchical clustering. Hence this clustering algorithm cannot be used when we have huge data.

Challenging the solution

There are a few challenges that come along while working with the K-means clustering algorithm. One of those challenges is that it makes clusters of the same size. The other challenge is that it decides the number of clusters at the beginning of the algorithm and thus we would not know how many clusters we should have while working with the algorithm. Therefore, Hierarchical Clustering is the best option