Data Analysis

John Maronga

26/03/2022

Business Understanding

My-Duka is an online shop that recently launched their services. As a new company, they would like to use effective and strategic marketing techniques to reach their clientelle.

Specifying the analytic Question

My-duka would like to understand which customers are highly likely to click on an add ontheir site and vice-versa.

Define the Metric for Success

Thorough Data Cleaning Perform Univariate analysis Perform Bivariate

Experimental design

Data Understanding Univariate Analysis Bivariate Analysis Plotting the summaries

Data Cleaning

```
#specify the path where the file is located
library("data.table")
```

- obtaining the path to the working directrory

```
getwd()
```

```
## [1] "C:/Users/User/Desktop/MoringaExe/Advertisement-and-Marketing-models-in-R"
```

Loading the datasets

```
#library("readr")
df <- fread("http://bit.ly/IPAdvertisingData")
head(df)
```

```
##    Daily Time Spent on Site Age Area Income Daily Internet Usage
## 1:                    68.95  35    61833.90               256.09
## 2:                    80.23  31    68441.85               193.77
## 3:                    69.47  26    59785.94               236.50
## 4:                    74.15  29    54806.18               245.89
## 5:                    68.37  35    73889.99               225.58
## 6:                    59.99  23    59761.56               226.74
##                            Ad Topic Line           City Male    Country
## 1:    Cloned 5thgeneration orchestration    Wrightburgh    0    Tunisia
## 2:    Monitored national standardization      West Jodi    1      Nauru
## 3:      Organic bottom-line service-desk       Davidton    0 San Marino
## 4: Triple-buffered reciprocal time-frame West Terrifurt    1      Italy
## 5:         Robust logistical utilization   South Manuel    0    Iceland
## 6:       Sharable client-driven software      Jamieberg    1     Norway
##              Timestamp Clicked on Ad
## 1: 2016-03-27 00:53:11             0
## 2: 2016-04-04 01:39:02             0
## 3: 2016-03-13 20:35:42             0
## 4: 2016-01-10 02:31:19             0
## 5: 2016-06-03 03:36:18             0
## 6: 2016-05-19 14:30:17             0
```

Previewing the top of the dataset

```
advert_df <- data.frame(df)
head(advert_df)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    68.95  35    61833.90               256.09
## 2                    80.23  31    68441.85               193.77
## 3                    69.47  26    59785.94               236.50
## 4                    74.15  29    54806.18               245.89
## 5                    68.37  35    73889.99               225.58
## 6                    59.99  23    59761.56               226.74
##                         Ad.Topic.Line           City Male    Country
## 1    Cloned 5thgeneration orchestration    Wrightburgh    0    Tunisia
## 2    Monitored national standardization      West Jodi    1      Nauru
## 3      Organic bottom-line service-desk       Davidton    0 San Marino
## 4 Triple-buffered reciprocal time-frame West Terrifurt    1      Italy
## 5         Robust logistical utilization   South Manuel    0    Iceland
## 6       Sharable client-driven software      Jamieberg    1     Norway
##             Timestamp Clicked.on.Ad
## 1 2016-03-27 00:53:11             0
## 2 2016-04-04 01:39:02             0
## 3 2016-03-13 20:35:42             0
## 4 2016-01-10 02:31:19             0
```

```
## 5 2016-06-03 03:36:18             0
## 6 2016-05-19 14:30:17             0
```

Previewing the summary of the dataset

```
summary(advert_df)
```

```
##  Daily.Time.Spent.on.Site      Age          Area.Income    Daily.Internet.Usage
##  Min.   :32.60            Min.   :19.00   Min.   :13996   Min.   :104.8
##  1st Qu.:51.36            1st Qu.:29.00   1st Qu.:47032   1st Qu.:138.8
##  Median :68.22            Median :35.00   Median :57012   Median :183.1
##  Mean   :65.00            Mean   :36.01   Mean   :55000   Mean   :180.0
##  3rd Qu.:78.55            3rd Qu.:42.00   3rd Qu.:65471   3rd Qu.:218.8
##  Max.   :91.43            Max.   :61.00   Max.   :79485   Max.   :270.0
##  Ad.Topic.Line          City                Male         Country
##  Length:1000        Length:1000        Min.   :0.000   Length:1000
##  Class :character   Class :character   1st Qu.:0.000   Class :character
##  Mode  :character   Mode  :character   Median :0.000   Mode  :character
##                                        Mean   :0.481
##                                        3rd Qu.:1.000
##                                        Max.   :1.000
##    Timestamp                    Clicked.on.Ad
##  Min.   :2016-01-01 02:52:10   Min.   :0.0
##  1st Qu.:2016-02-18 02:55:42   1st Qu.:0.0
##  Median :2016-04-07 17:27:29   Median :0.5
##  Mean   :2016-04-10 10:34:06   Mean   :0.5
##  3rd Qu.:2016-05-31 03:18:14   3rd Qu.:1.0
##  Max.   :2016-07-24 00:22:16   Max.   :1.0
```

Properties of the dataset

```
length(advert_df)
```

```
## [1] 10
```

```
#dimensions
dim(advert_df)
```

```
## [1] 1000   10
```

```
# Column names
colnames(advert_df)
```

```
##  [1] "Daily.Time.Spent.on.Site" "Age"
##  [3] "Area.Income"              "Daily.Internet.Usage"
##  [5] "Ad.Topic.Line"            "City"
##  [7] "Male"                     "Country"
##  [9] "Timestamp"                "Clicked.on.Ad"
```

```r
# Column data types
sapply(advert_df, class)
```

```
## $Daily.Time.Spent.on.Site
## [1] "numeric"
##
## $Age
## [1] "integer"
##
## $Area.Income
## [1] "numeric"
##
## $Daily.Internet.Usage
## [1] "numeric"
##
## $Ad.Topic.Line
## [1] "character"
##
## $City
## [1] "character"
##
## $Male
## [1] "integer"
##
## $Country
## [1] "character"
##
## $Timestamp
## [1] "POSIXct" "POSIXt"
##
## $Clicked.on.Ad
## [1] "integer"
```

Data Cleaning

Missing values

```r
#Checking the sum of missing values per column
colSums(is.na(advert_df))
```

```
## Daily.Time.Spent.on.Site                      Age             Area.Income
##                        0                        0                        0
##     Daily.Internet.Usage           Ad.Topic.Line                     City
##                        0                        0                        0
##                     Male                  Country                Timestamp
##                        0                        0                        0
##            Clicked.on.Ad
##                        0
```

Duplicates

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

```
##  [1] Daily.Time.Spent.on.Site Age                      Area.Income
##  [4] Daily.Internet.Usage     Ad.Topic.Line            City
##  [7] Male                     Country                  Timestamp
## [10] Clicked.on.Ad
## <0 rows> (or 0-length row.names)
```

Assigning the appropriate datatypes for each column

- Changing the timestamp datatypes from factor to date_time

```
#changing the timestamp datatype from factor to date_time
advert_df$Timestamp <- as.Date(advert_df$Timestamp, format = "%Y-%m-%s-%h-%m-
%s")
#checking the new datatype for the Timestamp column
sapply(advert_df, class)
```

```
## Daily.Time.Spent.on.Site                      Age             Area.Income
##                "numeric"                "integer"               "numeric"
##     Daily.Internet.Usage            Ad.Topic.Line                     City
##                "numeric"              "character"             "character"
##                     Male                  Country                Timestamp
##                "integer"              "character"                   "Date"
##            Clicked.on.Ad
##                "integer"
```

Univariate Analysis

Daily Time Spent on Site

```
#This column represents the amount of time that a user spends on the website
# measures of central tendency
# mean
mean(advert_df$Daily.Time.Spent.on.Site)
```

```
## [1] 65.0002
```

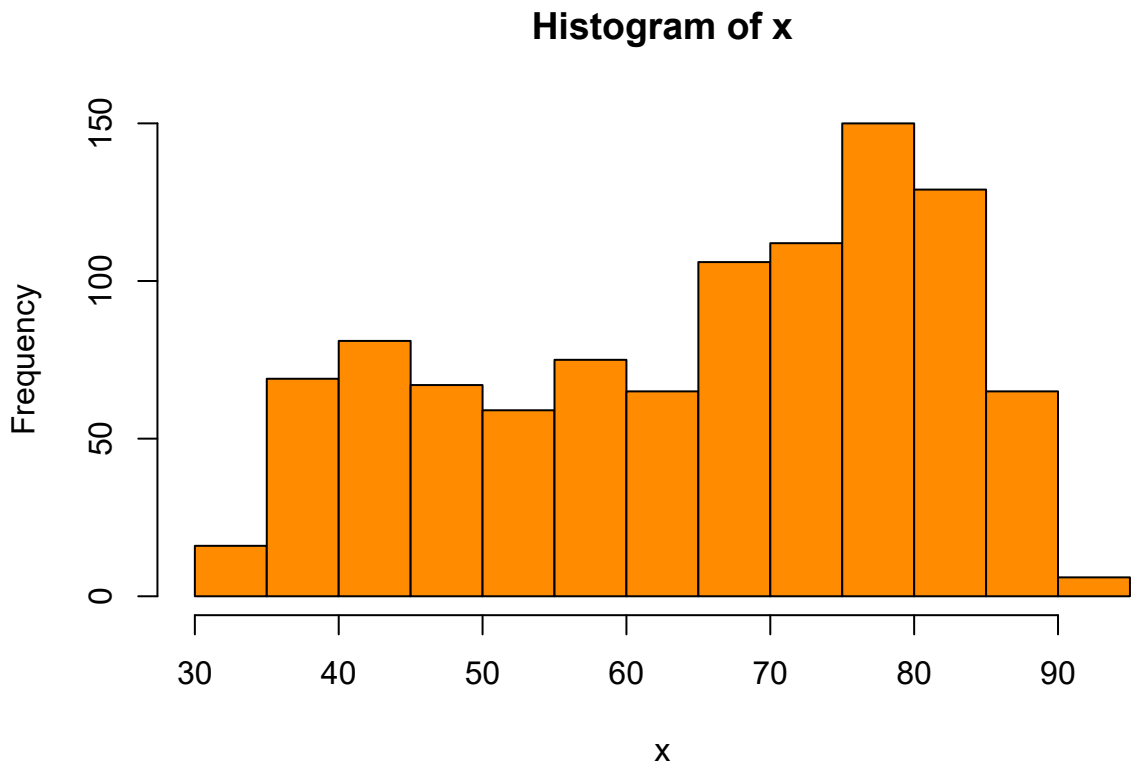
```
## [1] 65.0002
# median
median(advert_df$Daily.Time.Spent.on.Site)
```

```
## [1] 68.215
```

```
## [1] 68.215
# mode
x <- advert_df$Daily.Time.Spent.on.Site
#sort(x)
names(table(x))[table(x)==max(table(x))]
```

```
## [1] "62.26" "75.55" "77.05" "78.76" "84.53"
```

```
## [1] "62.26" "75.55" "77.05" "78.76" "84.53"
#each of the values printed below appear thrice in the dataset
#distribution
hist(x, col=c("darkorange"))
```



- The users spend an average 65.002 minutes on the website. - The modal time is “62.26” “75.55” “77.05” “78.76” “84.53” - The median time is 68.215. - The distribution above is left-skewed. - The highest frequency is 80 units of time(minutes).

Age

```
# Age of the user
#This column represents the Age of the user
# measures of central tendency
# mean
mean(advert_df$Age)
```

```
## [1] 36.009
```

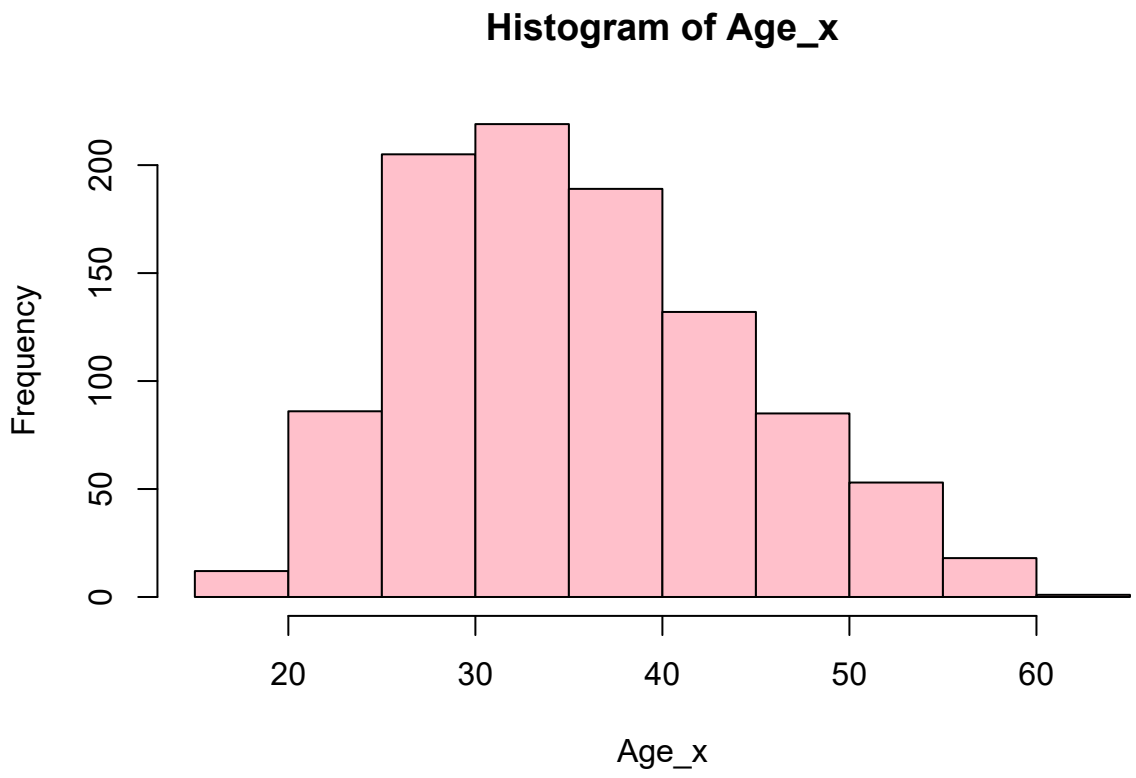
```
## [1] 36.009
# median
median(advert_df$Age)
```

```
## [1] 35
```

```
## [1] 35
# mode
Age_x <- advert_df$Age
#sort(Age_x)
names(table(Age_x))[table(Age_x)==max(table(Age_x))]
```

```
## [1] "31"
```

```
## [1] "31"
#each of the values printed below appear thrice in the dataset
#distribution
hist(Age_x, col = c("pink"))
```



- The age distribution is right skewed - The respondents on the website are mostly 25-40 years old. - The mean age is 36. - The median age is 35

Area Income

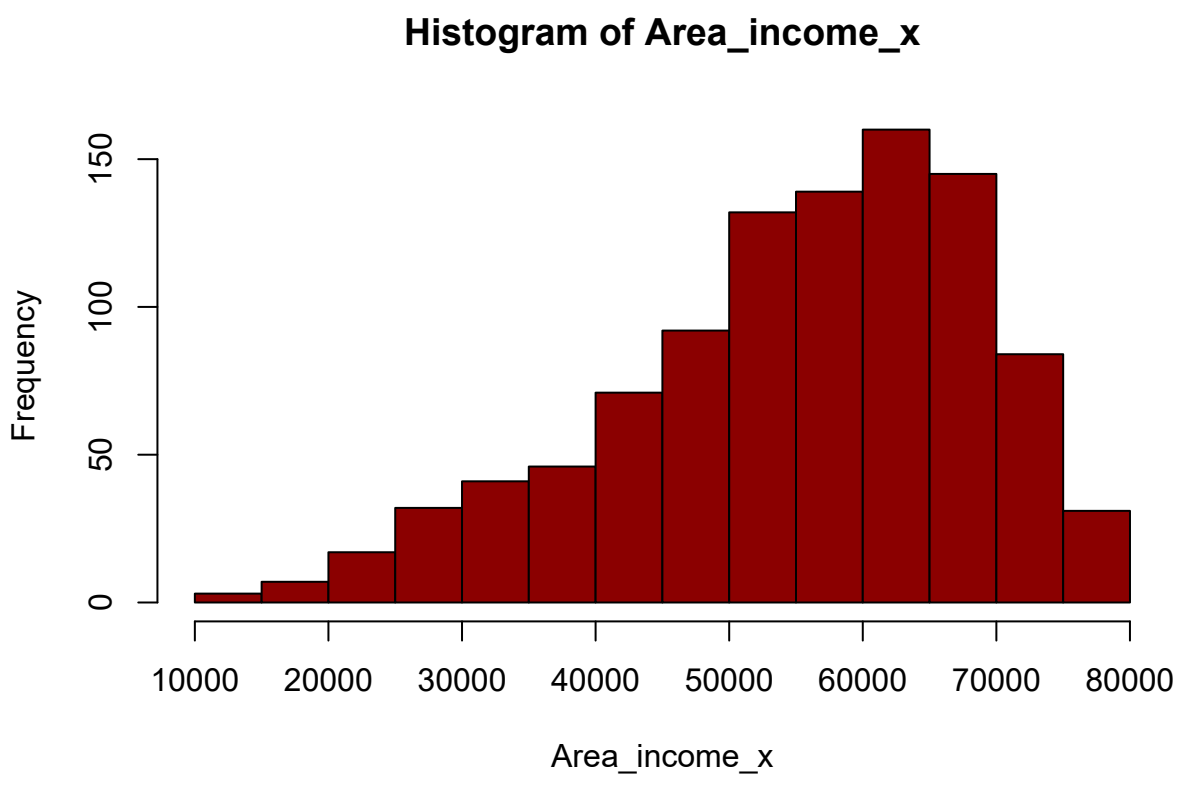
```
#income
# mean
mean(advert_df$Area.Income)
```

```
## [1] 55000
```

```
## [1] 55000
# median
median(advert_df$Area.Income)
```

```
## [1] 57012.3
```

```
## [1] 57012.3
# mode
Area_income_x <- advert_df$Area.Income
#sort(Daily.Internet.Usage_x)
#names(table(Area_income_x))[table(Area_income_x)==max(table(Area_income_x))]
#each of the values printed below appear thrice in the dataset
#distribution
hist(Area_income_x, col = c('darkred'))
```



- The income distribution is left skewed - The respondents on the website mostly earn between 55,000 to 70,000. - The mean income is 55,000. - The median income is 57,012.

Daily Internet Usage

```
#This column represents the amount of data that the user consumers in a day
# measures of central tendency
# mean
mean(advert_df$Daily.Internet.Usage)
```

```
## [1] 180.0001
```

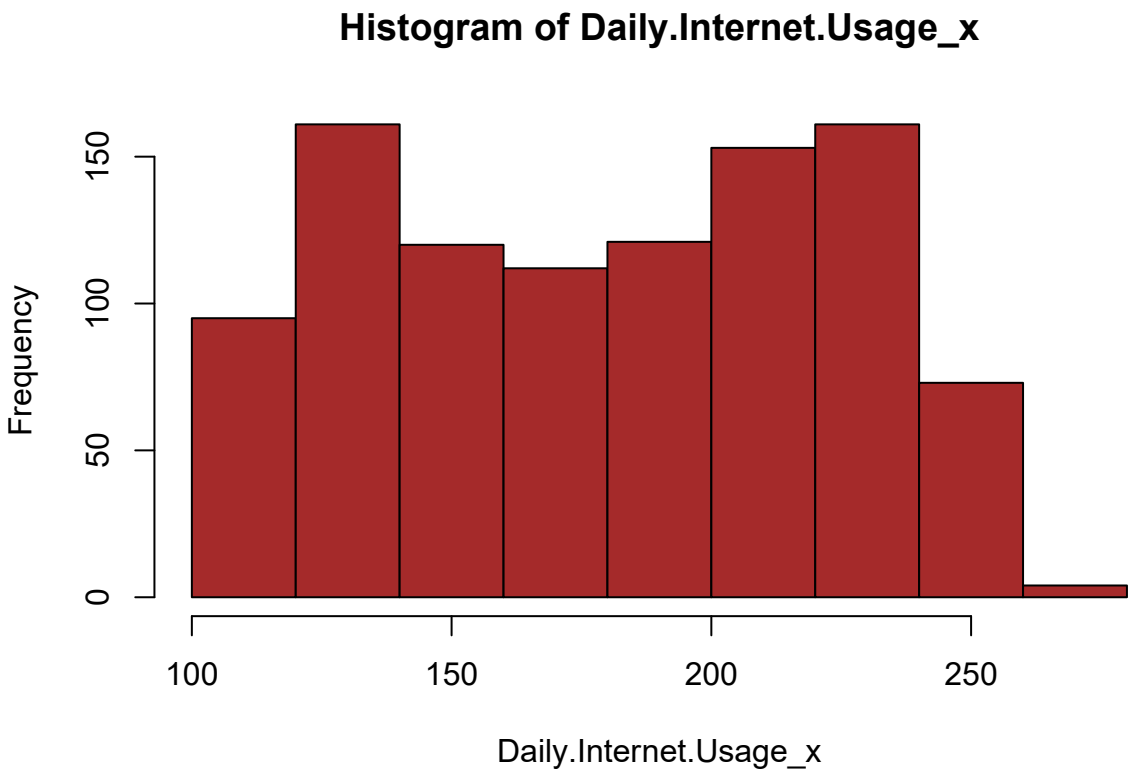
```
## [1] 180.0001
# median
median(advert_df$Daily.Internet.Usage)
```

```
## [1] 183.13
```

```
## [1] 183.13
# mode
Daily.Internet.Usage_x <- advert_df$Daily.Internet.Usage
#sort(Daily.Internet.Usage_x)
names(table(Daily.Internet.Usage_x))[table(Daily.Internet.Usage_x)==max(table
(Daily.Internet.Usage_x))]
```

```
##  [1] "113.53" "115.91" "117.3"  "119.3"  "120.06" "125.45" "132.38" "135.24"
##  [9] "136.18" "138.35" "158.22" "161.16" "162.44" "164.25" "167.22" "169.4"
## [17] "178.75" "182.65" "190.95" "194.23" "201.15" "211.87" "214.42" "215.18"
## [25] "219.72" "222.11" "223.16" "228.81" "230.36" "234.75" "235.28" "236.96"
## [33] "247.05" "256.4"
```

```
#each of the values printed below appear thrice in the dataset
#distribution
hist(Daily.Internet.Usage_x, col = c('brown'))
```



- The mean data usage is 180 units.
- The median data usage is 183.13 units .

City

```
#city where the user is located
# measures of central tendency
length(levels(advert_df$City))
```

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

```
## [1] 969
#there are 969 unique cities in the dataset
# mode
City_x <- advert_df$City
#sort(City_x) #this code gives an ordered list of all the elements in the cities column
#The modal cities in the dataset
names(table(City_x))[table(City_x)==max(table(City_x))]
```

```
## [1] "Lisamouth"    "Williamsport"
```

Male

```
#gender of the user
#1 indicates that the user is male while 0 indicates that they are female
#obtaining the unique levels in the gender(Male column)
unique(factor(advert_df$Male))
```

```
## [1] 0 1
## Levels: 0 1
```

```
Male_x <- table(advert_df$Male)
#distribution
barplot(Male_x, main="Gender Distribution",col=c("darkgreen"),xlab="Gender")
```

Country

```
#country where the user belongs
# measures of central tendency
# mode
Country_x <- advert_df$Country
#levels(Country_x) #this code gives the names of the countries
#There are 237 unique countries represented in the dataset
length(levels(Country_x))
```

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

```
## [1] 237
#the modal countries in the dataset
names(table(Country_x))[table(Country_x)==max(table(Country_x))]
```

```
## [1] "Czech Republic" "France"
```

Clicked on Ad

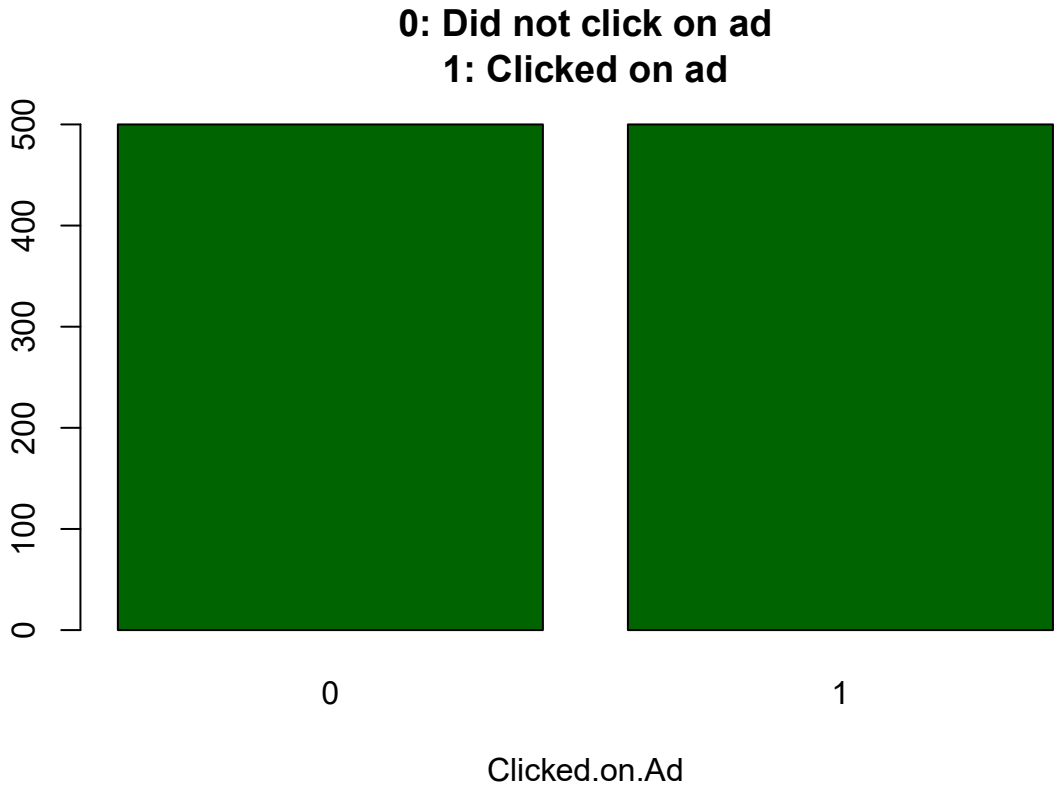
```
#zero indicates that a user did not click on an add while 1 indicates that a user clicked on an add
# measures of central tendency
#levels(advert_df$Clicked.on.Ad) #this code does not work
unique(factor(advert_df$Clicked.on.Ad))
```

```
## [1] 0 1
## Levels: 0 1
```

```
Clicked.on.Ad_x <- table(advert_df$Clicked.on.Ad)
#sort(Daily.Internet.Usage_x)
names(table(Clicked.on.Ad_x))[table(Clicked.on.Ad_x)==max(table(Clicked.on.Ad_x))]
```

```
## [1] "500"
```

```
#distribution
barplot(Clicked.on.Ad_x, main="0: Did not click on ad
 1: Clicked on ad " , col=c("darkgreen"),xlab="Clicked.on.Ad")
```



Bivariate Analysis and Multivariate Graphical Data analysis

```
advert_df2 <- subset(advert_df, select = c(Daily.Time.Spent.on.Site,
Age,Area.Income,Daily.Internet.Usage,Male,Clicked.on.Ad ))
head(advert_df2)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage Male
## 1                    68.95  35    61833.90               256.09    0
## 2                    80.23  31    68441.85               193.77    1
## 3                    69.47  26    59785.94               236.50    0
## 4                    74.15  29    54806.18               245.89    1
## 5                    68.37  35    73889.99               225.58    0
## 6                    59.99  23    59761.56               226.74    1
##   Clicked.on.Ad
## 1             0
## 2             0
## 3             0
## 4             0
## 5             0
## 6             0
```

Correlation

```
#The default method is Pearson, but we can also compute Spearman or Kendall coefficients.
mydata = cor(advert_df2, method = c("spearman"))
mydata1= cor(advert_df2, method = c("kendall"))
mydata2= cor(advert_df2, method = c("pearson"))
mydata #spearman
```

```
##                           Daily.Time.Spent.on.Site         Age Area.Income
## Daily.Time.Spent.on.Site               1.00000000 -0.31686155  0.28313439
## Age                                   -0.31686155  1.00000000 -0.13595396
## Area.Income                            0.28313439 -0.13595396  1.00000000
## Daily.Internet.Usage                   0.51410805 -0.37086395  0.33916021
## Male                                  -0.01592213 -0.02315468 -0.01436909
## Clicked.on.Ad                         -0.74487253  0.48633733 -0.46722440
##                           Daily.Internet.Usage        Male Clicked.on.Ad
## Daily.Time.Spent.on.Site           0.51410805 -0.01592213   -0.74487253
## Age                               -0.37086395 -0.02315468    0.48633733
## Area.Income                        0.33916021 -0.01436909   -0.46722440
## Daily.Internet.Usage               1.00000000  0.02820432   -0.77660702
## Male                               0.02820432  1.00000000   -0.03802747
## Clicked.on.Ad                     -0.77660702 -0.03802747    1.00000000
```

- Using the 3 correlation coefficients to get the correlation between the features, we can see that the correlation is very low and negative in most cases.
- This means that most of the variables are NOT dependent of each other
- Significance levels (p-values) can also be generated using the rcorr function which is found in the Hmisc package.
- install the required package and load the library.

```
#install_version("latticeExtra")
#install.packages("Hmisc", dependencies = T)
library("Hmisc")
```

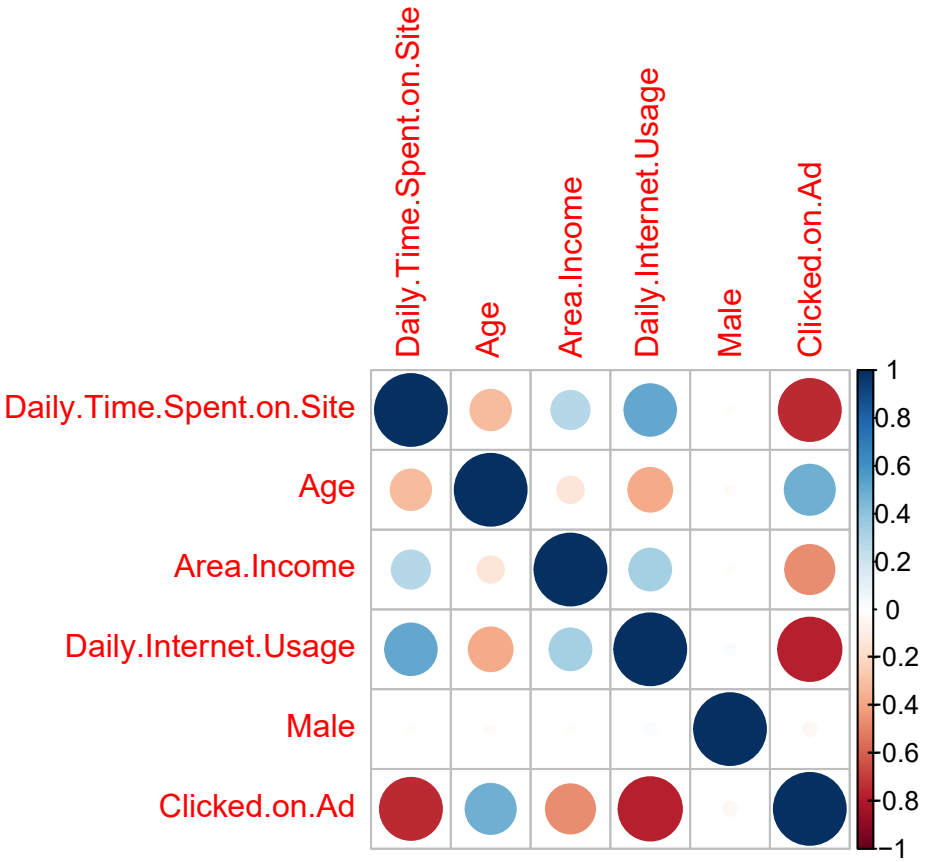
```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

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

- This generates one table of correlation coefficients (the correlation matrix) and another table of the p-values. By default, the correlations and p-values are stored in an object of class type rcorr.

```
#mydata.coeff = mydata.rcorr$r
#mydata.p = mydata.rcorr$P
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
## corrplot 0.84 loaded
corrplot(mydata)
```



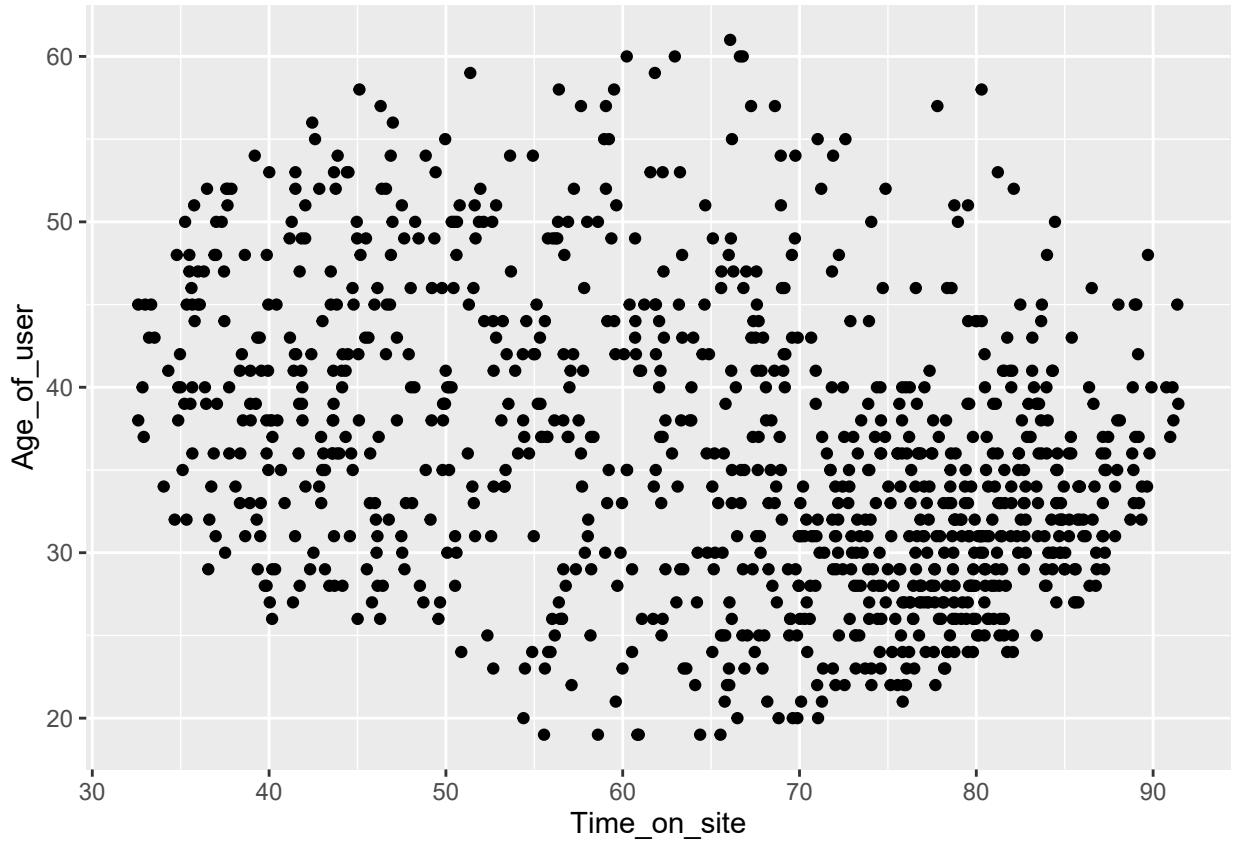
- A default correlation matrix plot (called a Correlogram) is generated. Positive correlations are displayed in a blue scale while negative correlations are displayed in a red scale - There is very minimal positive correlation between the variables in the data

The Plots below are scatterplots of a few pairs of variables

Time spent on the site vs age of teh user

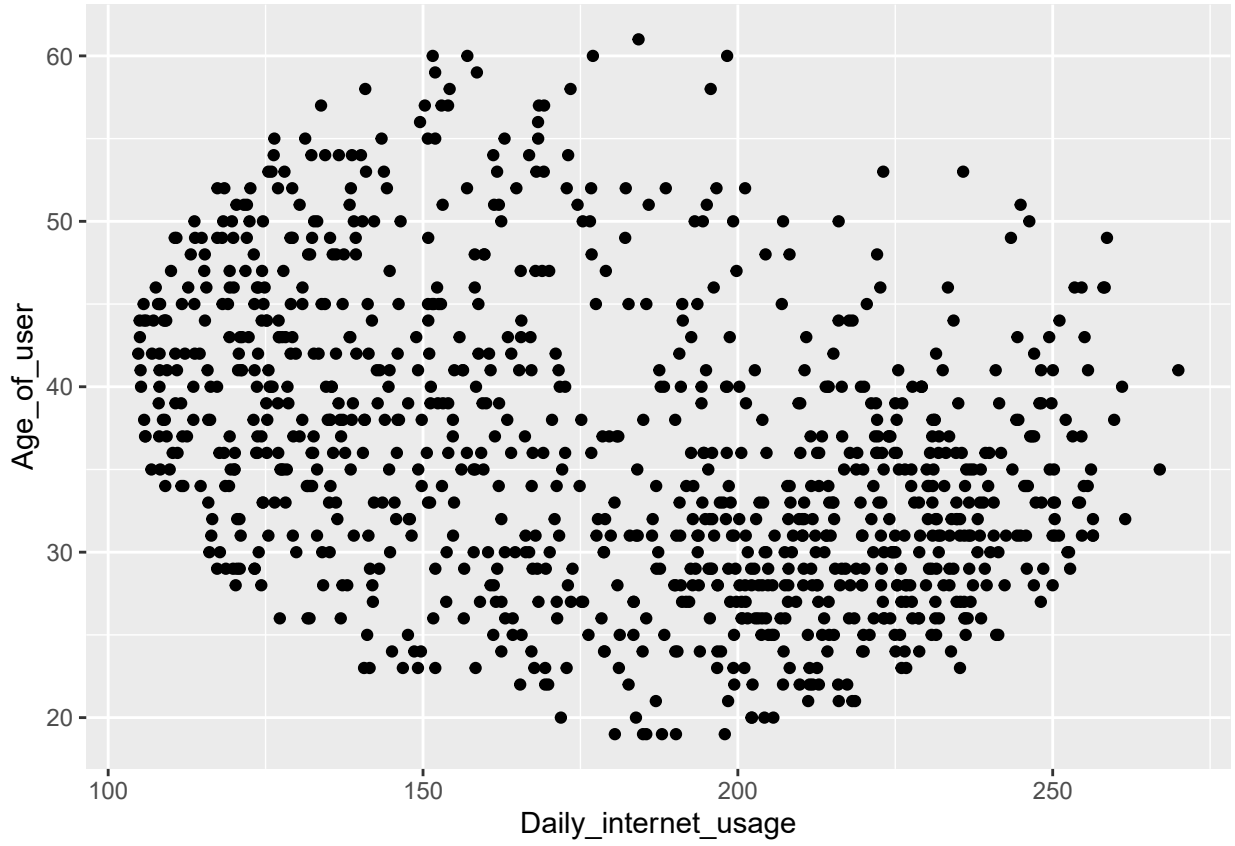
```
#Time spent on the site vs age of the user
# Libraries
library(ggplot2)
# create data
Time_on_site <- advert_df$Daily.Time.Spent.on.Site
Age_of_user <- advert_df$Age
```

```
data <- data.frame(Time_on_site,Age_of_user)
# Plot
ggplot(data, aes(x=Time_on_site, y=Age_of_user)) + geom_point()
```



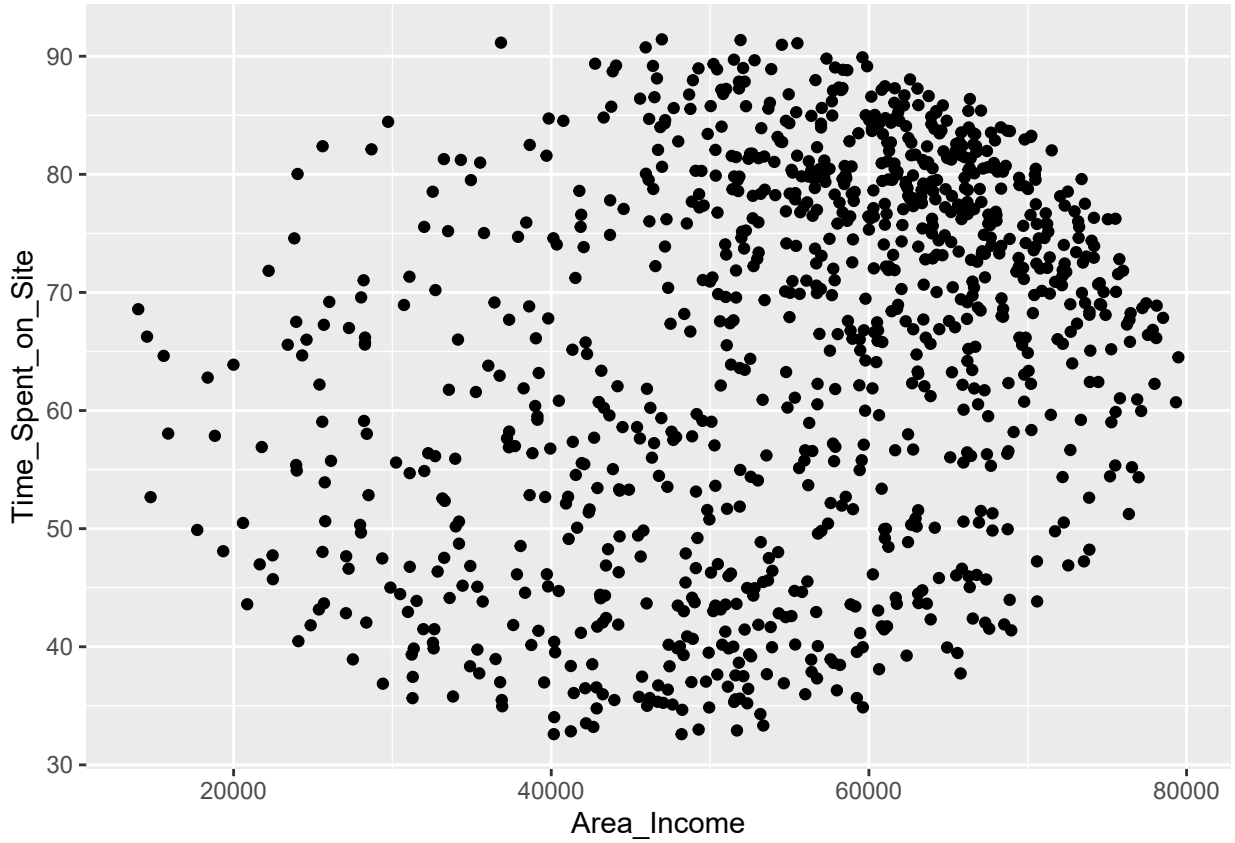
Age of the user vs daily internet usage

```
Daily_internet_usage <- advert_df$Daily.Internet.Usage
Age_of_user <- advert_df$Age
data1 <- data.frame(Daily_internet_usage,Age_of_user)
# Plot
ggplot(data1, aes(x=Daily_internet_usage, y=Age_of_user)) + geom_point()
```



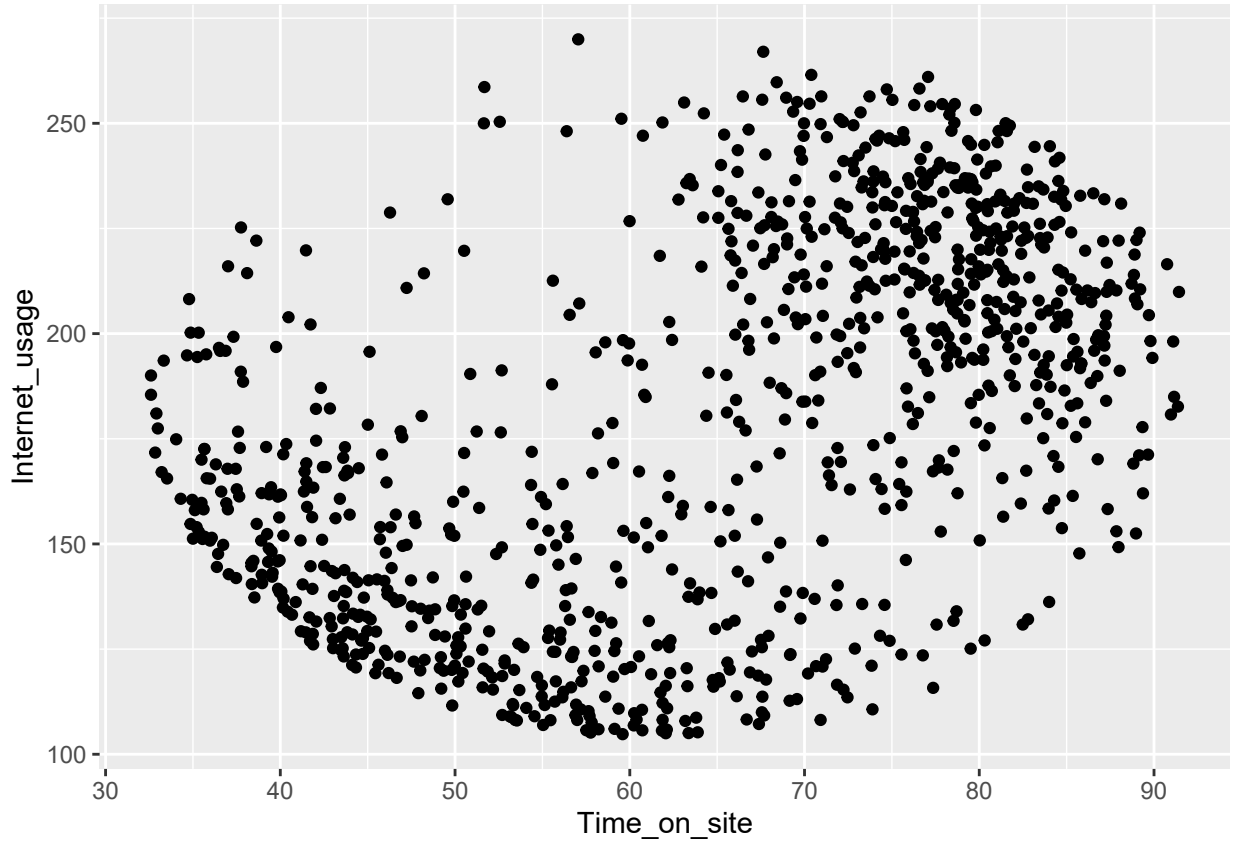
Time spenton the site vs area.income

```
Area_Income <- advert_df$Area.Income
Time_Spent_on_Site <- advert_df$Daily.Time.Spent.on.Site
data2 <- data.frame(Area_Income,Time_Spent_on_Site)
# Plot
ggplot(data2, aes(x=Area_Income, y=Time_Spent_on_Site)) + geom_point()
```



Time spent on the site vs daily internet usage

```
Time_on_site <- advert_df$Daily.Time.Spent.on.Site
Internet_usage <- advert_df$Daily.Internet.Usage
data3 <- data.frame(Time_on_site,Internet_usage)
# Plot
ggplot(data3, aes(x=Time_on_site, y=Internet_usage)) + geom_point()
```



seperating the data Clicked and Gender columns

```
#creating a new column with null values
advert_df2["Female"] <- NA
dim(advert_df2)
```

```
## [1] 1000    7
```

```
#populating the column with false values from the male column
advert_df2$Female <- advert_df2$Male == 0
dim(advert_df2)
```

```
## [1] 1000    7
```

```
#converting the column to nu,eric
dim(advert_df2 <- apply(advert_df2, 2, as.numeric))
```

```
## [1] 1000    7
```

Gender Vs Clicked on Ad

```
library(tidyverse)
```

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

## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1
## v purrr   0.3.4

## -- 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 dplyr::src()       masks Hmisc::src()
## x dplyr::summarize() masks Hmisc::summarize()
## x purrr::transpose() masks data.table::transpose()
```

```
#Male respondents who clicked on an add
dim(advert_df%>% filter(Male == 1 , Clicked.on.Ad == 1))
```

```
## [1] 231  10
```

```
#Male respondents did not click on an add
dim(advert_df%>% filter(Male == 1, Clicked.on.Ad == 0))
```

```
## [1] 250  10
```

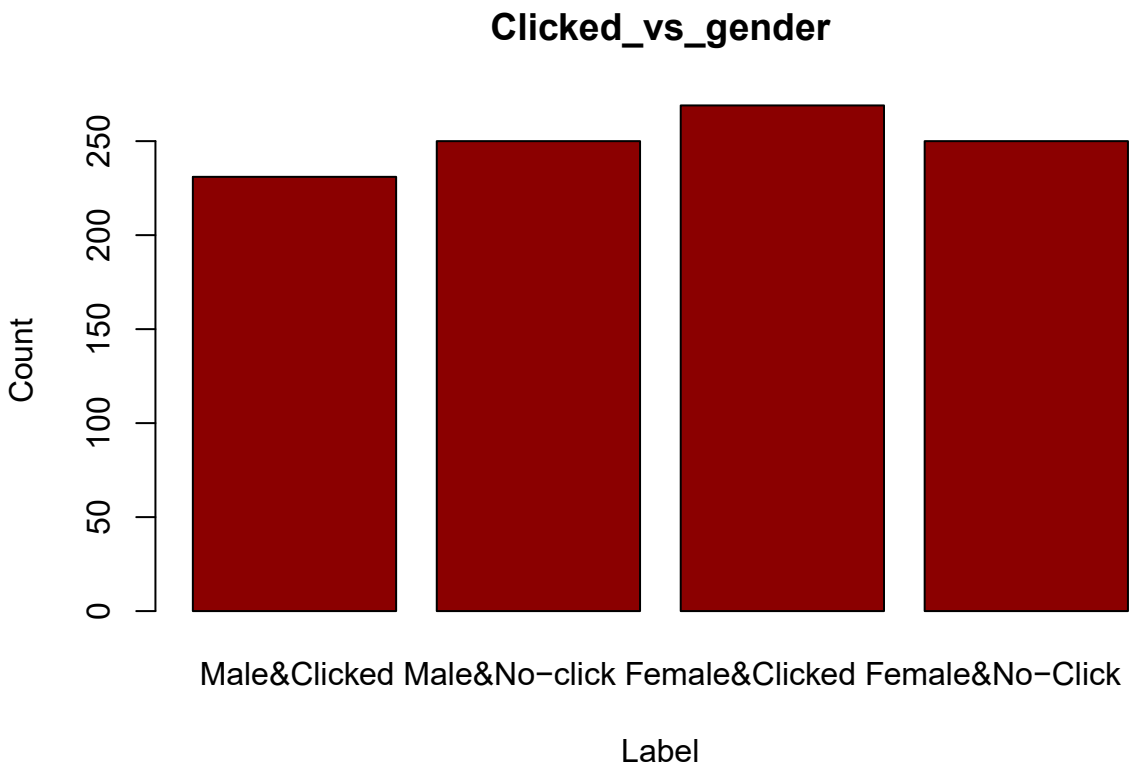
```
#Female respondents who clicked on an add
dim(advert_df%>% filter(Male == 0 , Clicked.on.Ad == 1))
```

```
## [1] 269  10
```

```
#Female respondents who clicked did not on an add
dim(advert_df%>% filter(Male == 0, Clicked.on.Ad == 0))
```

```
## [1] 250  10
```

```
Clicked_vs_gender <- c( 231 , 250 , 269 , 250 )

# barchart with added parameters
barplot(Clicked_vs_gender, main = " Clicked_vs_gender " , xlab = " Label ",
ylab = " Count ",
names.arg = c("Male&Clicked Male&No-click Female&Clicked Female&No-Click"),
col = "darkred",
horiz = FALSE)
```



Multivariate Analysis

```
# A glimpse of the data
library(dplyr)
glimpse(advert_df2)
```

```
##  num [1:1000, 1:7] 69 80.2 69.5 74.2 68.4 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:7] "Daily.Time.Spent.on.Site" "Age" "Area.Income" "Daily.Internet.Usage" ...
```

```
# One hot encoding of the factor variables.
# dummify the data
library(caret)
```

```
##
## Attaching package: 'caret'
```

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

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

```
dmy <- dummyVars(" ~ .", data = advert_df2)
dummy_df <- data.frame(predict(dmy, newdata = advert_df2))
#print(dummy_df)
glimpse(dummy_df)
```

```
## Rows: 1,000
## Columns: 7
## $ Daily.Time.Spent.on.Site <dbl> 68.95, 80.23, 69.47, 74.15, 68.37, 59.99, 88.~
## $ Age                      <dbl> 35, 31, 26, 29, 35, 23, 33, 48, 30, 20, 49, 3~
## $ Area.Income              <dbl> 61833.90, 68441.85, 59785.94, 54806.18, 73889~
## $ Daily.Internet.Usage     <dbl> 256.09, 193.77, 236.50, 245.89, 225.58, 226.7~
## $ Male                     <dbl> 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, ~
## $ Clicked.on.Ad            <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, ~
## $ Female                   <dbl> 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, ~
```

```
sapply(dummy_df, class)
```

```
## Daily.Time.Spent.on.Site                      Age              Area.Income
##                "numeric"                "numeric"                "numeric"
##     Daily.Internet.Usage                     Male             Clicked.on.Ad
##                "numeric"                "numeric"                "numeric"
##                   Female
##                "numeric"
```

```
#removing the revenue column from the data
#we select all the column indexes before 30
dummy_df2 <-
subset(dummy_df, select = -Clicked.on.Ad)
dim(dummy_df2)
```

```
## [1] 1000    6
```

```
dummy_df.class<- advert_df2[, "Clicked.on.Ad"]
```

SCALING VS NORMALIZATION

Scaling

- In this step the data is transformed to fit within the range between 0 and 1

```
dummy_df2_scaled <- scale(dummy_df2)
summary(dummy_df2_scaled)
```

```
##  Daily.Time.Spent.on.Site      Age           Area.Income
##  Min.   :-2.0437          Min.   :-1.9360   Min.   :-3.0566
##  1st Qu.:-0.8604          1st Qu.:-0.7978   1st Qu.:-0.5940
##  Median : 0.2028          Median :-0.1148   Median : 0.1500
##  Mean   : 0.0000          Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.8545          3rd Qu.: 0.6819   3rd Qu.: 0.7805
##  Max.   : 1.6671          Max.   : 2.8446   Max.   : 1.8252
##  Daily.Internet.Usage      Male             Female
##  Min.   :-1.71335     Min.   :-0.9622   Min.   :-1.0382
##  1st Qu.:-0.93777     1st Qu.:-0.9622   1st Qu.:-1.0382
##  Median : 0.07129     Median :-0.9622   Median : 0.9622
##  Mean   : 0.00000     Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.88361     3rd Qu.: 1.0382   3rd Qu.: 0.9622
##  Max.   : 2.04909     Max.   : 1.0382   Max.   : 0.9622
```

Normalizing

- Normalization is a technique often applied to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values.

```
dummy_df2_norm <- as.data.frame(apply(dummy_df2, 2, function(x) (x -
min(x))/(max(x)-min(x))))
summary(dummy_df2_norm)
```

```
##  Daily.Time.Spent.on.Site      Age           Area.Income
##  Min.   :0.0000           Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.3189           1st Qu.:0.2381   1st Qu.:0.5044
##  Median :0.6054           Median :0.3810   Median :0.6568
##  Mean   :0.5507           Mean   :0.4050   Mean   :0.6261
##  3rd Qu.:0.7810           3rd Qu.:0.5476   3rd Qu.:0.7860
##  Max.   :1.0000           Max.   :1.0000   Max.   :1.0000
##  Daily.Internet.Usage      Male           Female
##  Min.   :0.0000       Min.   :0.000   Min.   :0.000
##  1st Qu.:0.2061       1st Qu.:0.000   1st Qu.:0.000
##  Median :0.4743       Median :0.000   Median :1.000
##  Mean   :0.4554       Mean   :0.481   Mean   :0.519
##  3rd Qu.:0.6902       3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :1.0000       Max.   :1.000   Max.   :1.000
```

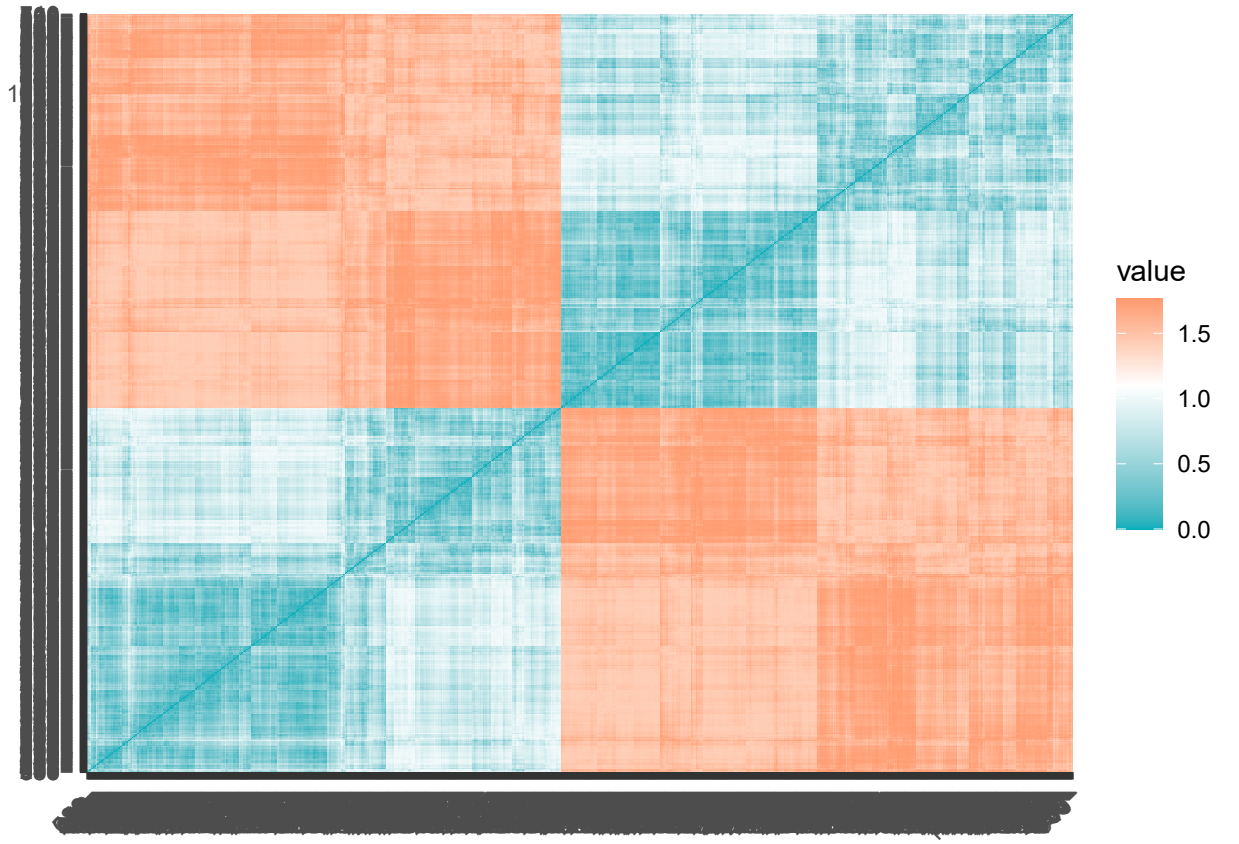
The distance Matrix

- How the elements are represented in the Euclidean space
- There are 4 distinct quarters which means that four of the elements in the data explian a great percentage of the variance.

```
library(factoextra)
```

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

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
distance <- get_dist(dummy_df2_norm)
fviz_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high =
"#FC4E07"))
```

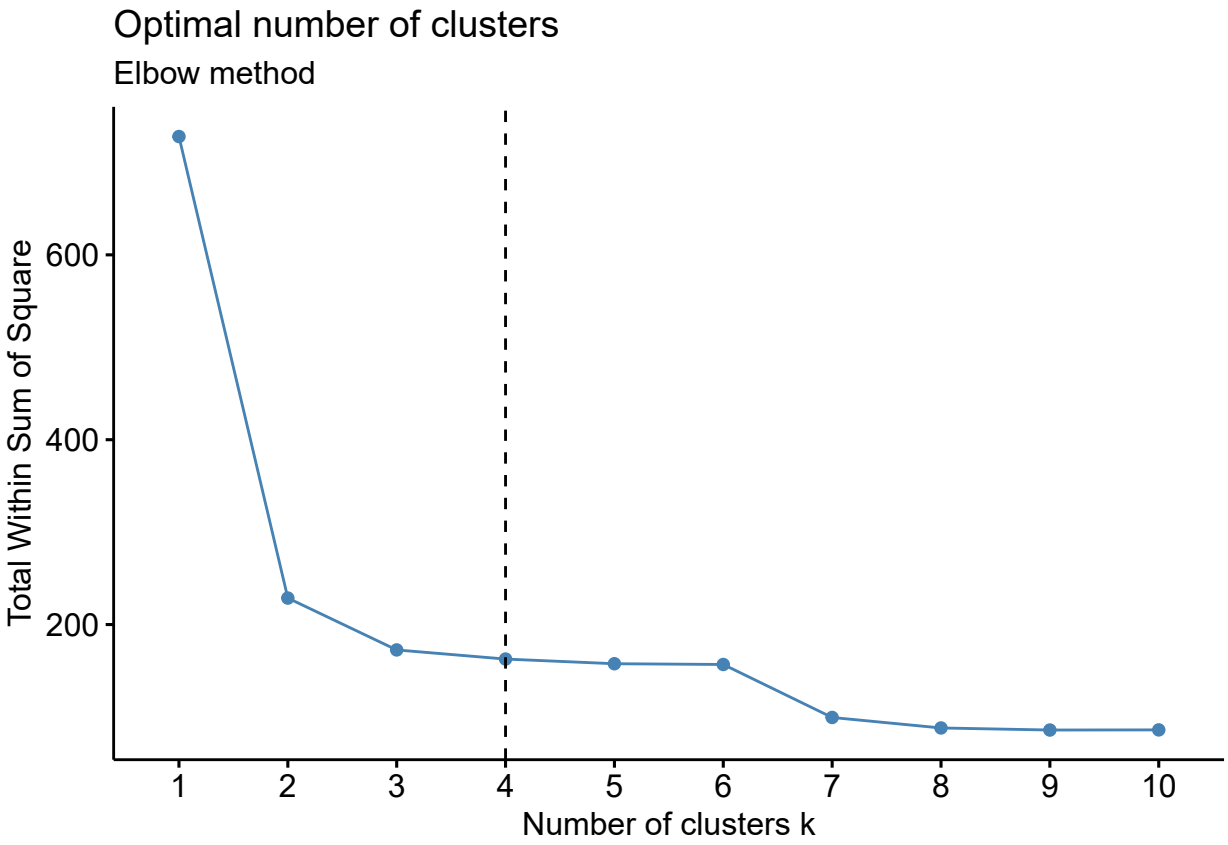


- The normalized dataset has a smaller range for the values which are between 0 and 1 unlike the standardized dataset which has values ranging from -2 to 2.9

Finding the Optimal number of Clusters

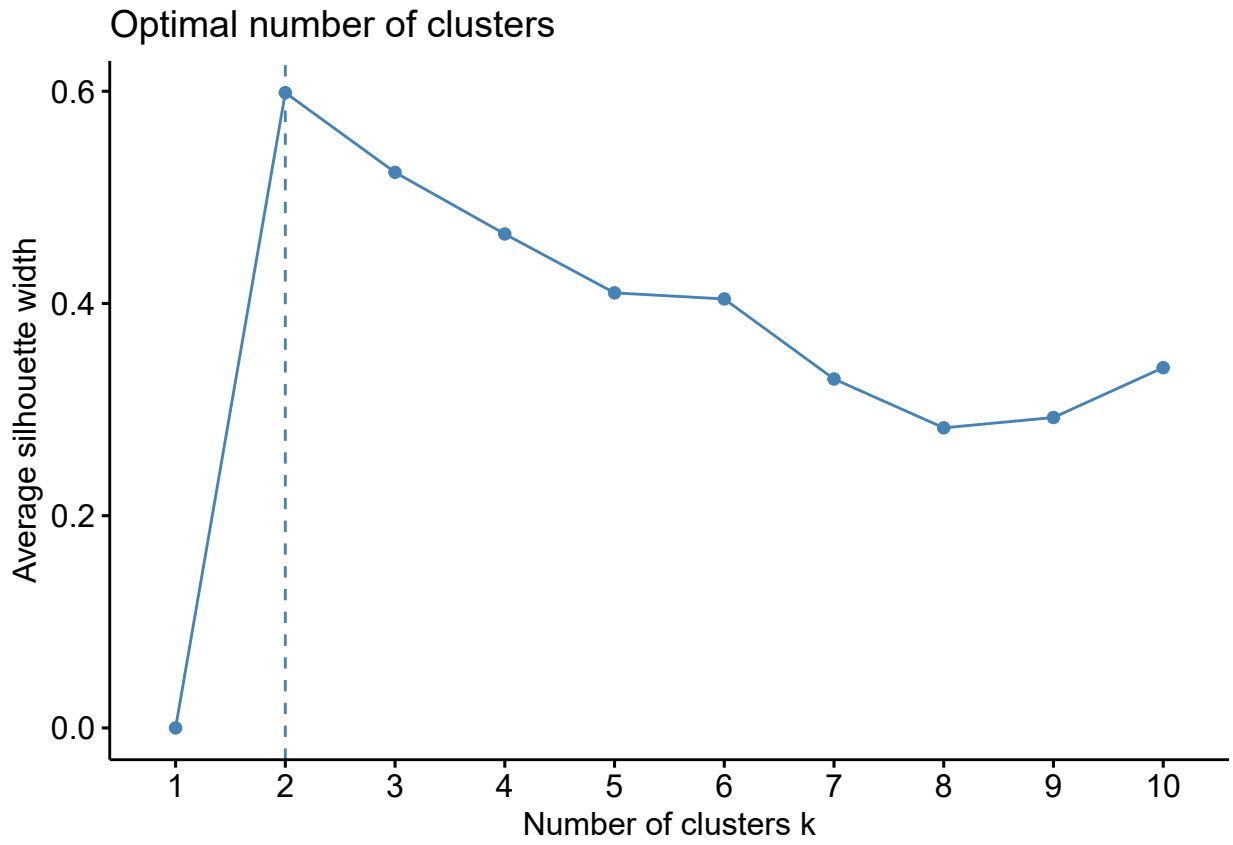
Method 1: Elbow method

```
# Searching for the optimal number of clusters
# # Elbow method
# Searching for the optimal number of clusters
# # Elbow method
library(factoextra)
fviz_nbclust(dummy_df2_norm, kmeans, method = "wss") + geom_vline(xintercept = 4, linetype = 2) + labs(
```



Method 2: Silhouette

```
library(cluster)
fviz_nbclust(dummy_df2_norm, kmeans, method = "silhouette")
```



Implementing the Solution

K-MEANS CLUSTERING

Using 4 clusters [Elbow Method]

```
outputk <- kmeans(dummy_df2_norm, 4)
```

Results

```
# Previewing the number of records in each cluster
outputk$size
```

```
## [1] 139 135 207 519
```

The cluster center datapoints Per attribute

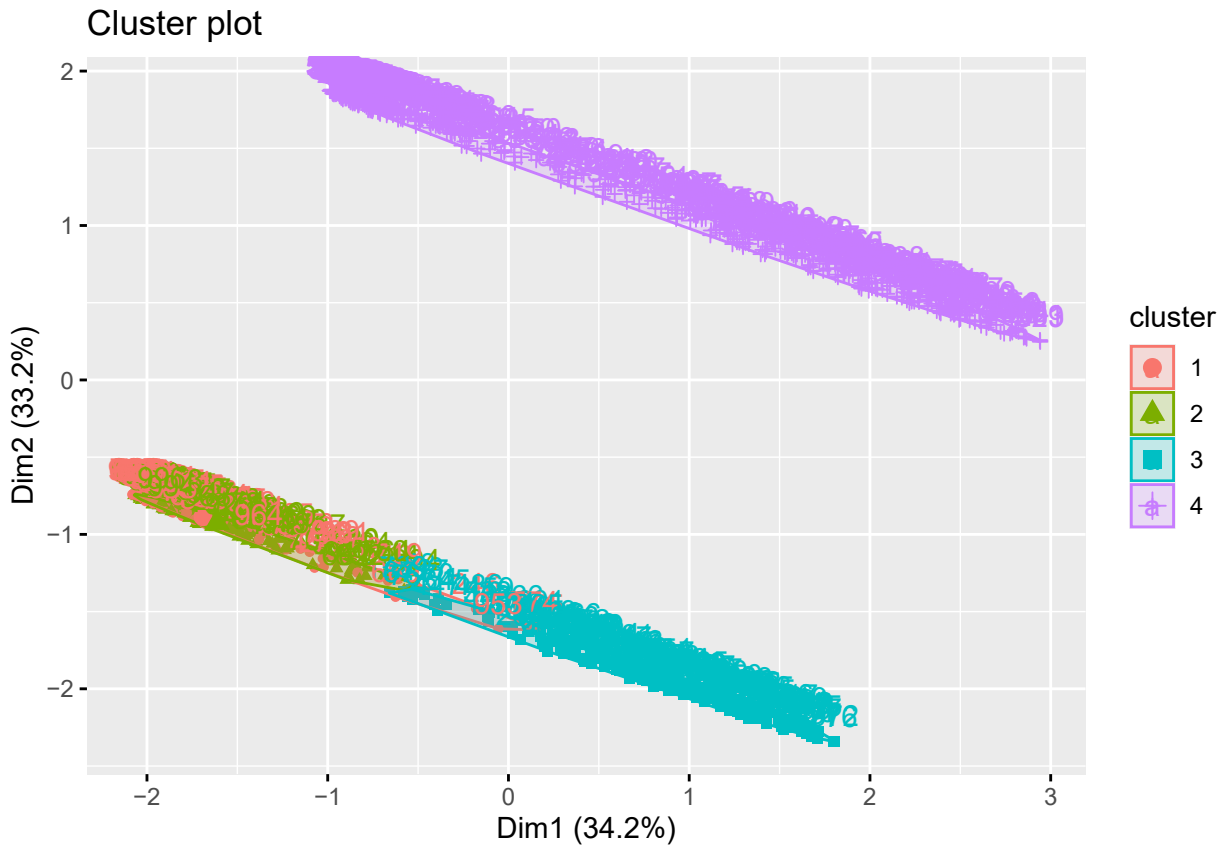
```
outputk$centers
```

```
##   Daily.Time.Spent.on.Site       Age Area.Income Daily.Internet.Usage Male
```

```
## 1               0.7548295 0.3979102   0.7024404            0.7530751    1
## 2               0.7226522 0.2000000   0.7026312            0.5443059    1
## 3               0.2892638 0.5327812   0.5256247            0.2154511    1
## 4               0.5556568 0.4092119   0.6258588            0.4482187    0
##   Female
## 1      0
## 2      0
## 3      0
## 4      1
```

Visualising the clusters of the whole dataset

```
options(repr.plot.width = 11, repr.plot.height = 6)
fviz_cluster(outputk, dummy_df2_norm)
```

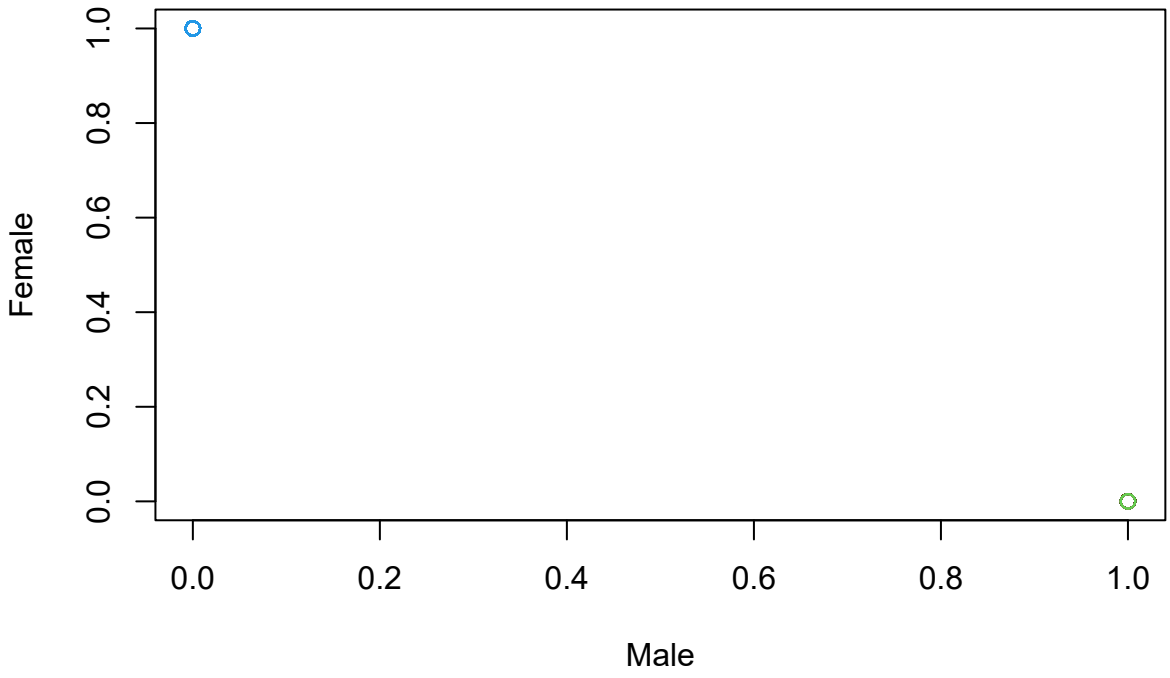


-while using four points, we can see that the data is divided into two distinct clusters first then two more clusters from the two.

Visualizing variable datatypes on a scatter plot

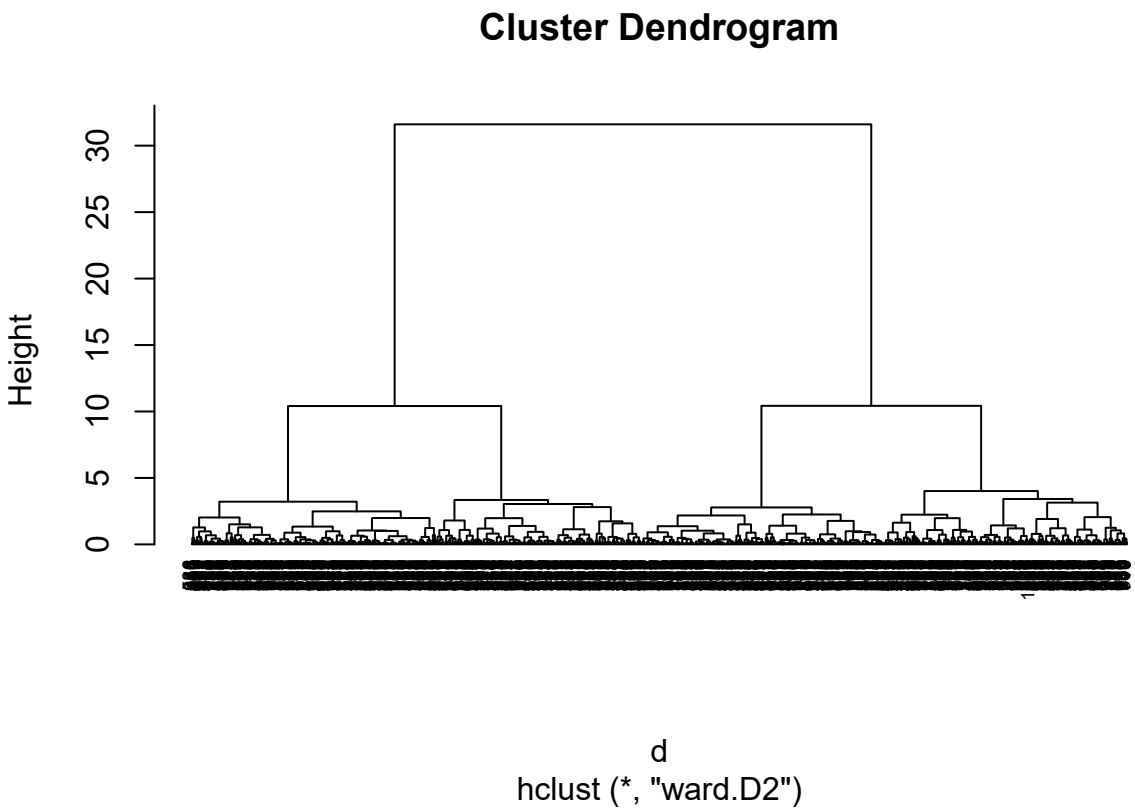
```
# Plotting two variables to see how their data points
# have been distributed in the cluster
```

```r
# Product Related, vs Product Related Duration
plot(dummy_df2_norm[, 5:6], col = outputk$cluster)
```



HIERACHICAL CLUSTERING

```r
d <- dist(dummy_df2_norm, method = "euclidean")
# We then apply hierarchical clustering using the Ward's method
res.hc <- hclust(d, method = "ward.D2")
# Lastly we plot the obtained dendrogram
#--
plot(res.hc, cex = 0.6, hang = -1)
```



Challenging the solution

K-MEANS CLUSTERING

- Using a different number of clusters 2 clusters using the silhouette method

Using 2 clusters [Silhouette Method]

```
outputs <- kmeans(dummy_df2_norm, 2)
```

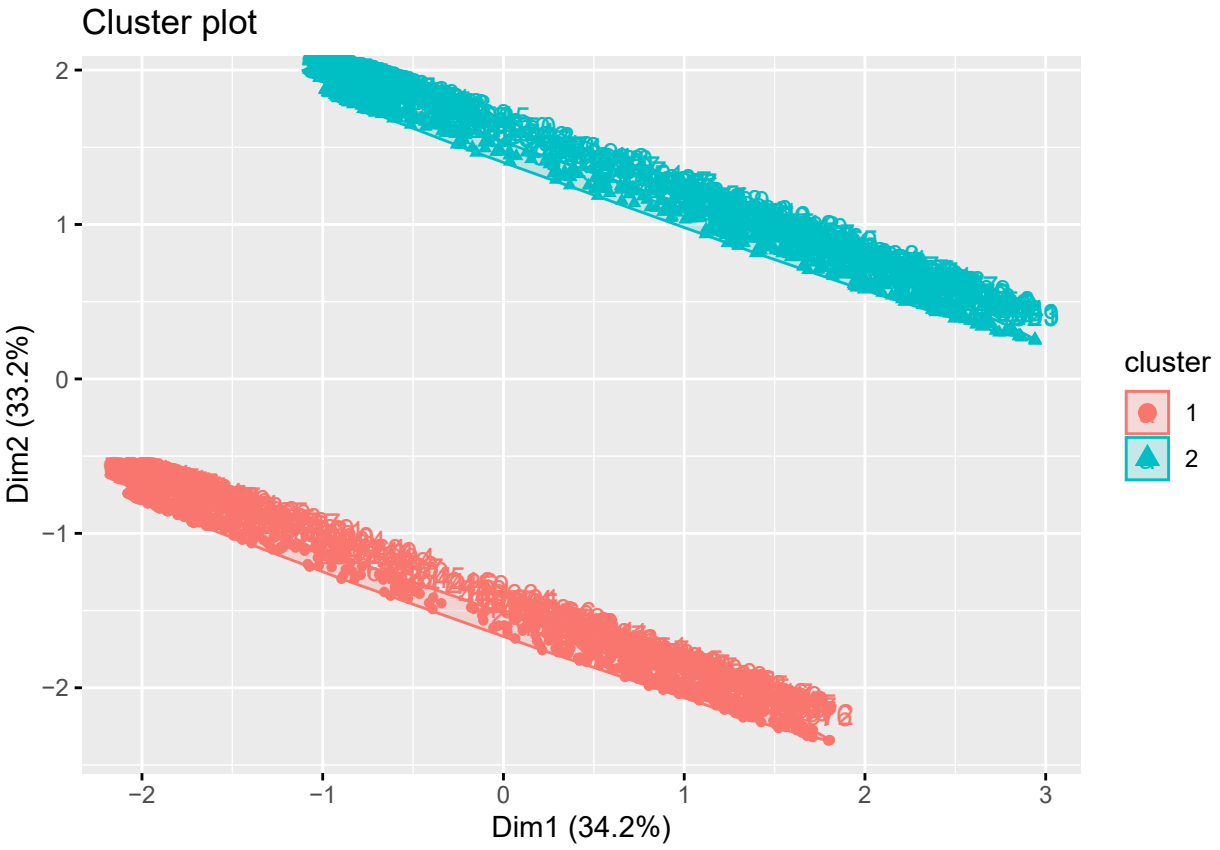
Results

```
# Previewing the number of records in each cluster
outputs$size
```

```
## [1] 481 519
```

Visualising the clusters of the whole dataset

```
options(repr.plot.width = 11, repr.plot.height = 6)
fviz_cluster(outputs, dummy_df2_norm)
```



Summary

Compasiron Between K-MEANS and HIERACHICAL clustering From the Analysis, we can identify that:

1. K-means Cluster Analysis performs much better in identyfing patterns as compared to Hierrachical clustering.
2. Since the dataset is large, visualizing hierrachical clusters is abit cumbersome as compared to K-means clustering.
3. K-means clustering yields better reults using the optimal number of clusters which can be determined by Elbow and Silhouette Methods
4. Clicking on an add is dependent on the gender of the respondent
5. We can conclude that,The order of the factors that affect if a respondent clicks on an ad is:

a) Gender
b) Daily Internet Usage
c) Daily time spent on the site
d) Income