Cryptography

John Maronga

2022-03-29

R Project: Predict whether a user will Click an Ad or not

1.1.1 Defining the Question

Create a prediction model that more accurately predicts whether a user will click an Ad.

1.1.2 The Context

A Kenyan entrepreneur has created an online cryptography course and would want to advertise it on her blog. She currently targets audiences originating from various countries. In the past, she ran ads to advertise a related course on the same blog and collected data in the process. She would now like to employ your services as a Data Science Consultant to create a solution that would allow her to determine whether ads targeted to audiences of certain characteristics i.e. city, male country, ad topic, etc. would click on her ads.

1.1.4 Experimental Design Taken

Installing packages and loading libraries needed Loading the data Exploratory Data Analysis Data Cleaning Visualizations Conclusion

1.1.5 Appropriateness of the Data

Dataset link: link text The columns in the dataset include: Daily Time Spent on Site Age Area Income Daily Internet Usage Ad Topic Line City Male Country Timestamp Clicked on Ad

1.2 Installing & Loading Necessary Packages

```
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()
```

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

```
## Warning: package 'ggplot2' is in use and will not be installed
```

1.3 Loading the dataset

```
# Reading the csv url then loading it
ads<- read.csv("http://bit.ly/IPAdvertisingData")

#Invoke a spreadsheet stye data viewer in R
View(ads)
```

1.4 Data Understanding

```
#Viewing the top 6 observations
head(ads)
```

```
##   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
```

```
#Viewing the bottom 6 observations
tail(ads)
```

```
##      Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 995                     43.70  28    63126.96               173.01
## 996                     72.97  30    71384.57               208.58
## 997                     51.30  45    67782.17               134.42
## 998                     51.63  51    42415.72               120.37
## 999                     55.55  19    41920.79               187.95
## 1000                    45.01  26    29875.80               178.35
##                            Ad.Topic.Line          City Male
## 995         Front-line bifurcated ability  Nicholasland    0
## 996         Fundamental modular algorithm     Duffystad    1
## 997       Grass-roots cohesive monitoring   New Darlene    1
## 998          Expanded intangible solution South Jessica    1
## 999  Proactive bandwidth-monitored policy   West Steven    0
```

```
## 1000      Virtual 5thgeneration emulation   Ronniemouth    0
##                      Country           Timestamp Clicked.on.Ad
## 995                  Mayotte 2016-04-04 03:57:48             1
## 996                  Lebanon 2016-02-11 21:49:00             1
## 997  Bosnia and Herzegovina 2016-04-22 02:07:01             1
## 998                 Mongolia 2016-02-01 17:24:57             1
## 999                Guatemala 2016-03-24 02:35:54             0
## 1000                  Brazil 2016-06-03 21:43:21             1
```

```
#### Get high level overview of the dataset
summary(ads)
```

```
##  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
##  Length:1000        Min.   :0.0
##  Class :character   1st Qu.:0.0
##  Mode  :character   Median :0.5
##                     Mean   :0.5
##                     3rd Qu.:1.0
##                     Max.   :1.0
```

```
####checking the no. of rows and columns####
dim(ads)
```

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

The dataset had 1000 rows and 10 columns

1.5 Data Cleaning

```
#Checking the data types in each column
sapply(ads, 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"              "character"
##            Clicked.on.Ad
##                "integer"
```

The timestamp column should be changed to Date type

```
#changing the datatype
ads$Timestamp <- as.Date(ads$Timestamp)

#Confirming if the changes have takes effect
str(ads)

## 'data.frame':    1000 obs. of  10 variables:
##  $ Daily.Time.Spent.on.Site: num  69 80.2 69.5 74.2 68.4 ...
##  $ Age                     : int  35 31 26 29 35 23 33 48 30 20 ...
##  $ Area.Income             : num  61834 68442 59786 54806 73890 ...
##  $ Daily.Internet.Usage    : num  256 194 236 246 226 ...
##  $ Ad.Topic.Line           : chr  "Cloned 5thgeneration orchestration"
"Monitored national standardization" "Organic bottom-line service-desk"
"Triple-buffered reciprocal time-frame" ...
##  $ City                    : chr  "Wrightburgh" "West Jodi" "Davidton"
"West Terrifurt" ...
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Country                 : chr  "Tunisia" "Nauru" "San Marino" "Italy"
...
##  $ Timestamp               : Date, format: "2016-03-27" "2016-04-04" ...
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...

#Checking for missing values
colSums(is.na(ads))

## 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
```

There are no missing values in the dataset

```
#Checking for duplicates in the dataset
duplicated_rows <- ads[duplicated(ads),]
print(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)
```

There are no duplicates

```
# Changing the male column into factor
ads$"Clicked.on.Ad" <- as.factor(ads$"Clicked.on.Ad")
ads$"Male" <- as.factor(ads$"Male")
```

1.6 Exploratory Data Analysis

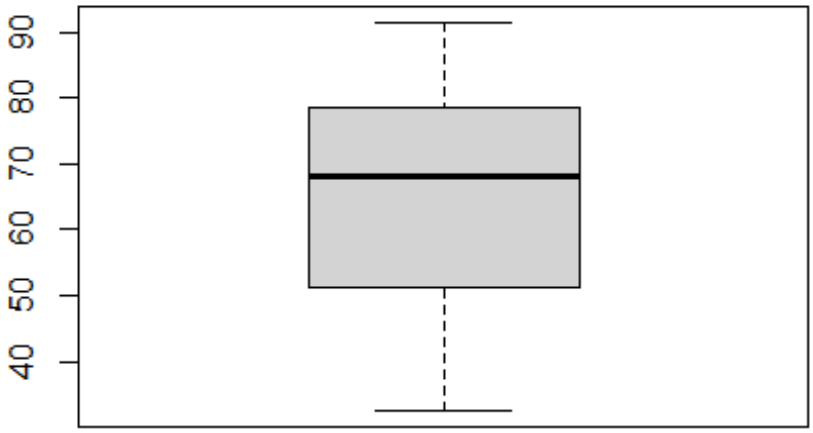
1.6.1 Univariate Analysis

```
#Finding the measures of central tendency
summary(ads)
```

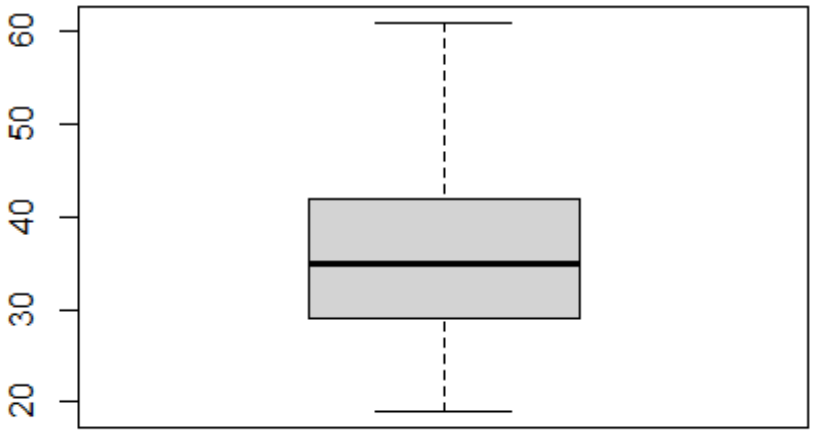
```
##  Daily.Time.Spent.on.Site      Age         Area.Income
Daily.Internet.Usage
##  Min.   :32.60            Min.   :19.00   Min.   :13996   Min.   :104.8
##  1st Qu.:51.36            1st Qu.:29.00   1st Qu.:47032   1st Qu.:138.8
##  Median :68.22            Median :35.00   Median :57012   Median :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        0:519   Length:1000
##  Class :character   Class :character   1:481   Class :character
##  Mode  :character   Mode  :character           Mode  :character
##
##
##
##    Timestamp          Clicked.on.Ad
##  Min.   :2016-01-01   0:500
##  1st Qu.:2016-02-17   1:500
##  Median :2016-04-07
##  Mean   :2016-04-09
##  3rd Qu.:2016-05-31
##  Max.   :2016-07-24
```

####Olotting Boxplots

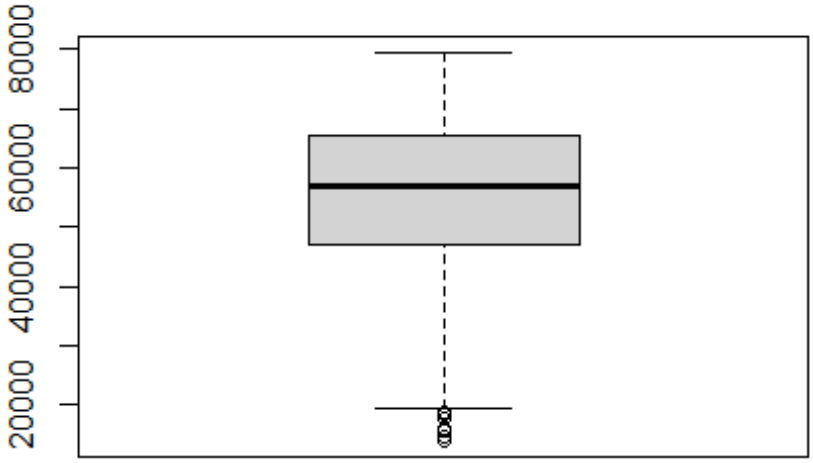
```
#Plotting
boxplot(ads$Daily.Time.Spent.on.Site)
```



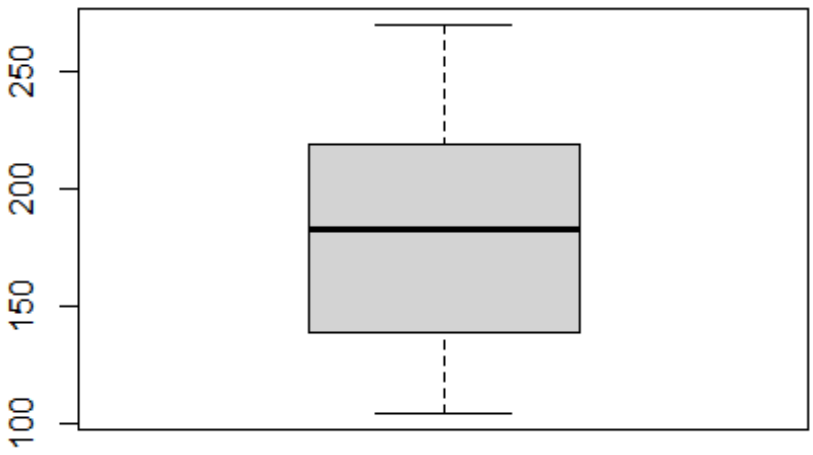

```
boxplot(ads$Age)
```




```
boxplot(ads$Area.Income)
```

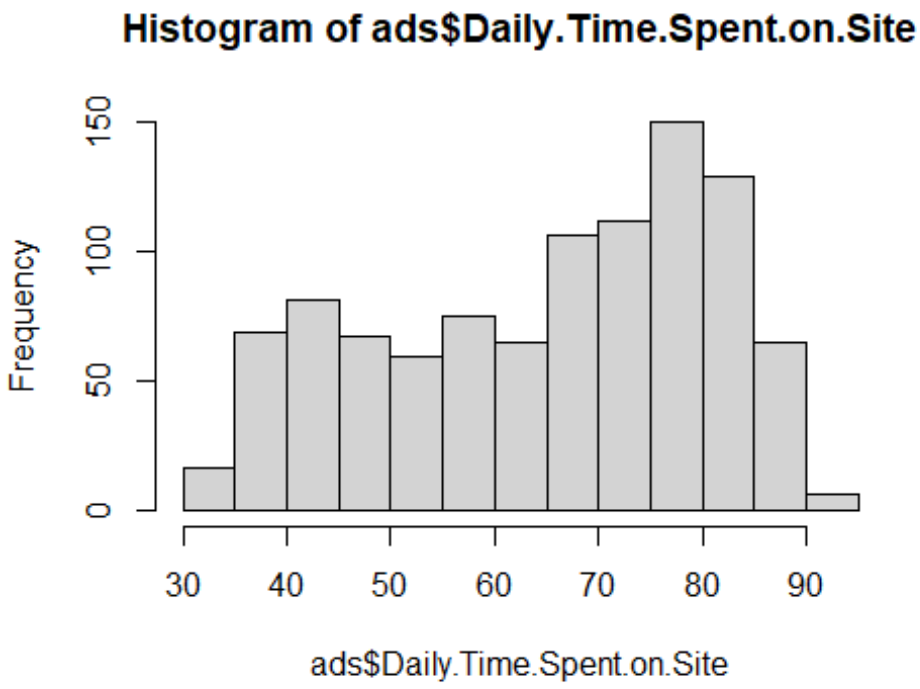



```
boxplot(ads$Daily.Internet.Usage)
```

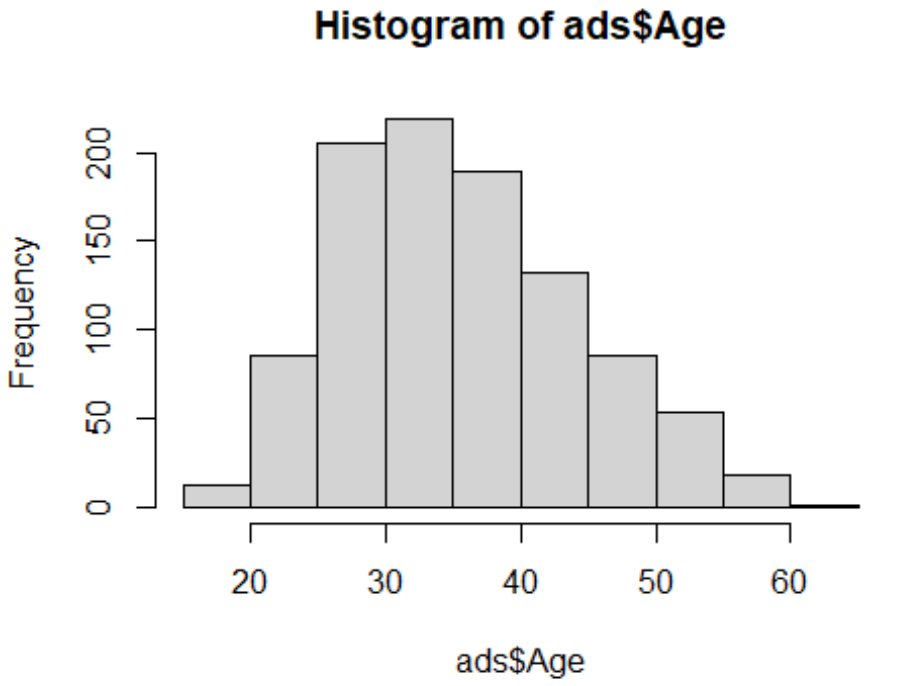


Plotting Histograms

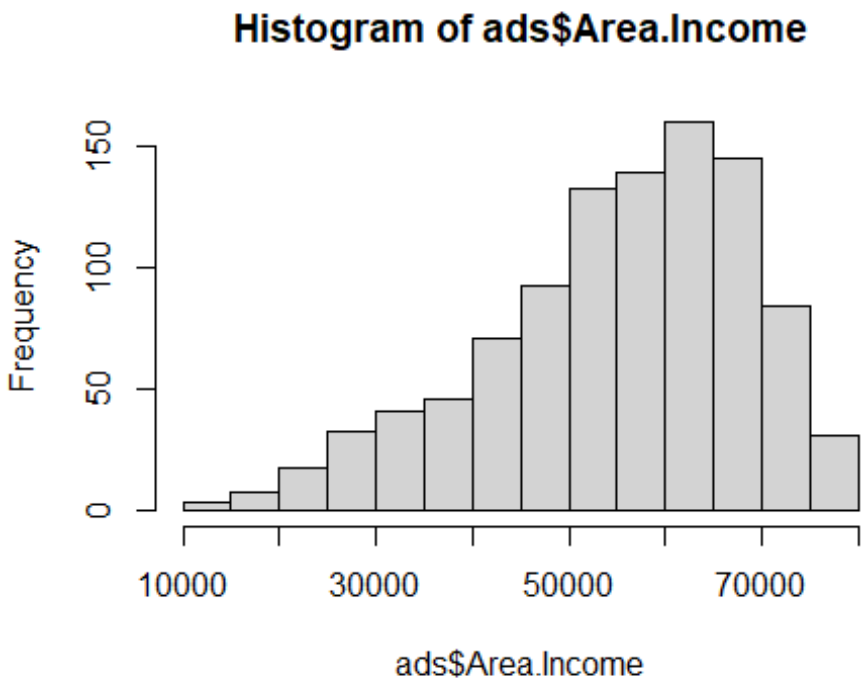
```
hist(ads$Daily.Time.Spent.on.Site)
```



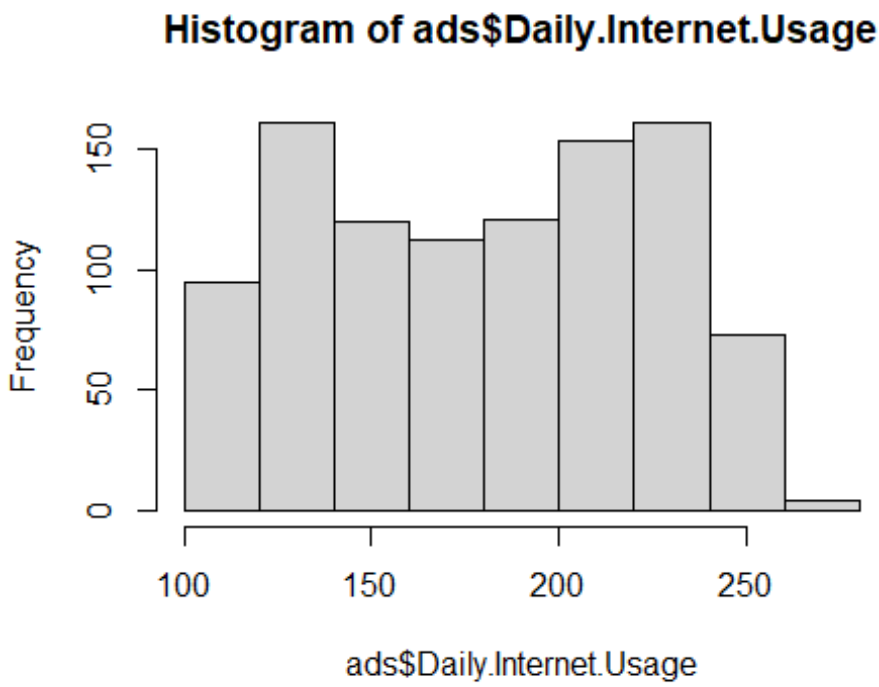

```
hist(ads$Age)
```



```
hist(ads$Area.Income)
```




```
hist(ads$Daily.Internet.Usage)
```



Plotting

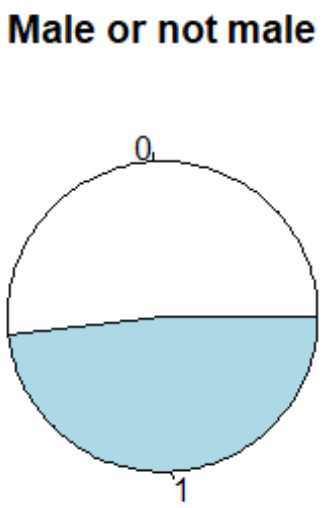
Frequency Tables

```
#checking frequency table of the male column
males<-table(ads$Male)
print(males)
```

```
##
##   0   1
## 519 481
```

519 were not males and 481 were males

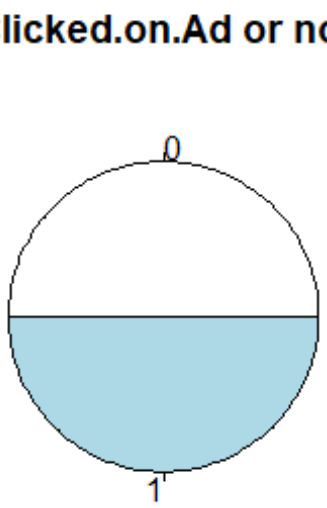
```
#plotting a pie chart for the male column
pie(table(ads$Male), main="Male or not male")
```




```
#checking frequency table of the clicked an ad column
clicked<-table(ads$Clicked.on.Ad)
print(clicked)

##
##   0   1
## 500 500
```

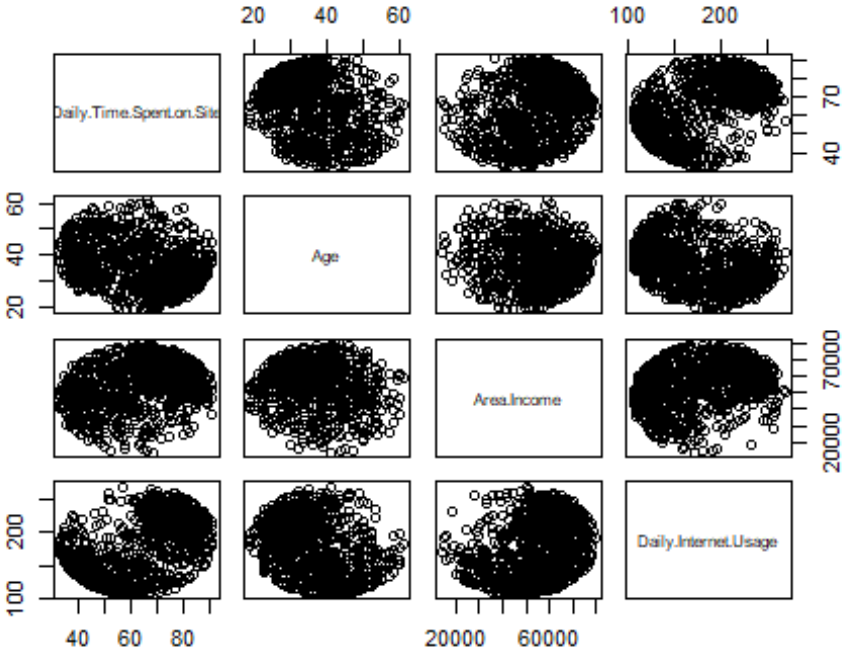
500 people didn't click the ads while 500 people clicked the ads

```
#plotting a pie chart for the male column
pie(table(ads$Clicked.on.Ad), main="Clicked.on.Ad or not")
```



1.6.2 Bivariate Analysis

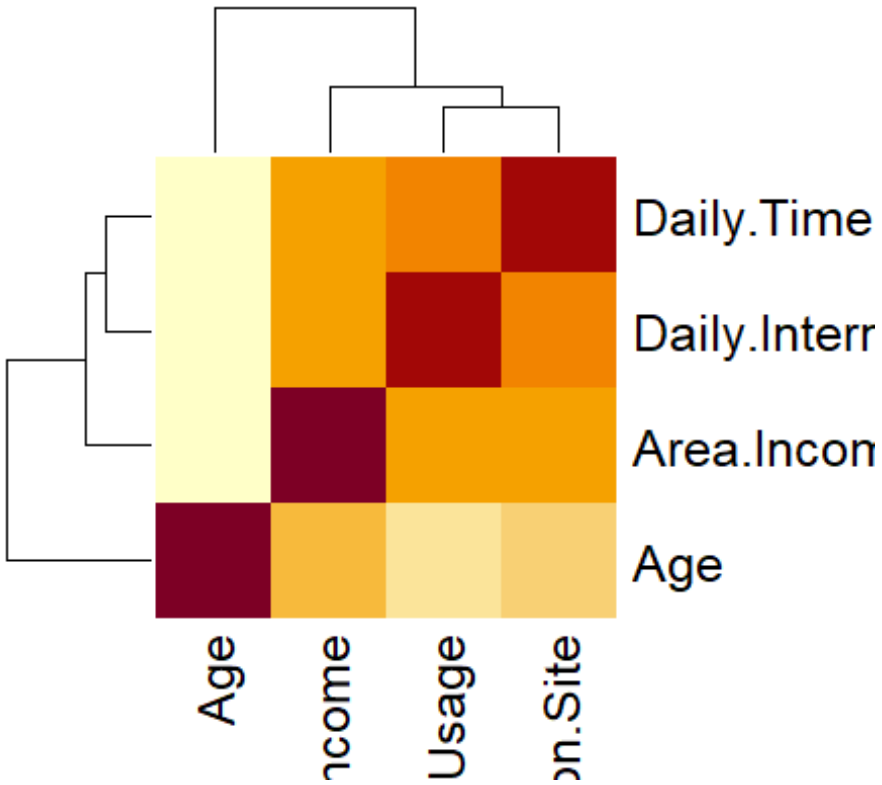
```
#scatterplot
plot(ads[, 1:4])
```



```
cor(ads[, 1:4])
```

```
##                           Daily.Time.Spent.on.Site        Age Area.Income
## Daily.Time.Spent.on.Site                1.0000000 -0.3315133   0.3109544
## Age                                    -0.3315133  1.0000000  -0.1826050
## Area.Income                             0.3109544 -0.1826050   1.0000000
## Daily.Internet.Usage                    0.5186585 -0.3672086   0.3374955
##                           Daily.Internet.Usage
## Daily.Time.Spent.on.Site            0.5186585
## Age                                -0.3672086
## Area.Income                         0.3374955
## Daily.Internet.Usage                1.0000000
```

```
#heatmap
heatmap(cor(ads[, 1:4]))
```



Modelling # K Nearest Neighbours

```
#Creating a function for normalization of our data
normalize <- function(x) {
return ((x - min(x)) / (max(x) - min(x)))
}
#Normalizing the numeric columns
ads$Daily.Time.Spent.on.Site<-normalize(ads$Daily.Time.Spent.on.Site)
ads$Age<-normalize(ads$Daily.Internet.Usage)
ads$Area.Income<-normalize(ads$Area.Income)
ads$Daily.Internet.Usage<-normalize(ads$Daily.Internet.Usage)
```

```
library(dplyr)

#Selecting the columns we'll use for modelling.
cols = c('Daily.Time.Spent.on.Site', 'Age', 'Area.Income',
'Daily.Internet.Usage', 'Male', 'Clicked.on.Ad')
ad = select(ads, cols)

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

head(ad)

##   Daily.Time.Spent.on.Site       Age Area.Income Daily.Internet.Usage Male
## 1                0.6178820 0.9160310   0.7304725            0.9160310    0
## 2                0.8096209 0.5387456   0.8313752            0.5387456    1
## 3                0.6267211 0.7974331   0.6992003            0.7974331    0
## 4                0.7062723 0.8542802   0.6231599            0.8542802    1
## 5                0.6080231 0.7313234   0.9145678            0.7313234    0
## 6                0.4655788 0.7383460   0.6988280            0.7383460    1
##   Clicked.on.Ad
## 1             0
## 2             0
## 3             0
## 4             0
## 5             0
## 6             0

# Lets now create test and train data sets
#Extracting the training set
ad_train <- ad[1:800,]
##Extracting the testing set
ad_test <- ad[801:1000,]
train_sp <- ad[1:800,5]
test_sp <- ad[801:1000,5]

# Split into Train and Validation sets
# Training Set : Validation Set = 70 : 30 (random)
set.seed(100)
train <- sample(nrow(ads), 0.7*nrow(ads), replace = FALSE)
TrainSet <- ads[train,]
ValidSet <- ads[-train,]
summary(TrainSet)

##  Daily.Time.Spent.on.Site      Age          Area.Income
##  Min.   :0.0000           Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.3102           1st Qu.:0.2130   1st Qu.:0.5115
##  Median :0.5944           Median :0.4714   Median :0.6631
##  Mean   :0.5445           Mean   :0.4566   Mean   :0.6306
##  3rd Qu.:0.7770           3rd Qu.:0.6921   3rd Qu.:0.7917
```

```
##  Max.   :0.9990           Max.   :1.0000   Max.   :0.9977
##  Daily.Internet.Usage Ad.Topic.Line          City           Male
##  Min.   :0.0000       Length:700         Length:700         0:360
##  1st Qu.:0.2130       Class :character   Class :character   1:340
##  Median :0.4714       Mode  :character   Mode  :character
##  Mean   :0.4566
##  3rd Qu.:0.6921
##  Max.   :1.0000
##    Country            Timestamp          Clicked.on.Ad
##  Length:700         Min.   :2016-01-01   0:347
##  Class :character   1st Qu.:2016-02-15   1:353
##  Mode  :character   Median :2016-04-06
##                     Mean   :2016-04-08
##                     3rd Qu.:2016-05-29
##                     Max.   :2016-07-23

summary(ValidSet)

##  Daily.Time.Spent.on.Site      Age            Area.Income
##  Min.   :0.00408          Min.   :0.001332   Min.   :0.07364
##  1st Qu.:0.33006          1st Qu.:0.185994   1st Qu.:0.46494
##  Median :0.63989          Median :0.477388   Median :0.65419
##  Mean   :0.56533          Mean   :0.452448   Mean   :0.61575
##  3rd Qu.:0.78336          3rd Qu.:0.681030   3rd Qu.:0.76426
##  Max.   :1.00000          Max.   :0.945877   Max.   :1.00000
##  Daily.Internet.Usage Ad.Topic.Line          City           Male
##  Min.   :0.001332     Length:300         Length:300         0:159
##  1st Qu.:0.185994     Class :character   Class :character   1:141
##  Median :0.477388     Mode  :character   Mode  :character
##  Mean   :0.452448
##  3rd Qu.:0.681030
##  Max.   :0.945877
##    Country            Timestamp          Clicked.on.Ad
##  Length:300         Min.   :2016-01-01   0:153
##  Class :character   1st Qu.:2016-02-29   1:147
##  Mode  :character   Median :2016-04-13
##                     Mean   :2016-04-14
##                     3rd Qu.:2016-06-03
##                     Max.   :2016-07-24

#We'll now use the K-NN algorithm but first we'll call the "class" package
which contains the K-NN algorithm.
library(class)
require(class)
model <- knn(train= ad_train,test=ad_test,cl= train_sp,k=5)
table(factor(model))

##
##   0   1
##  99 101
```

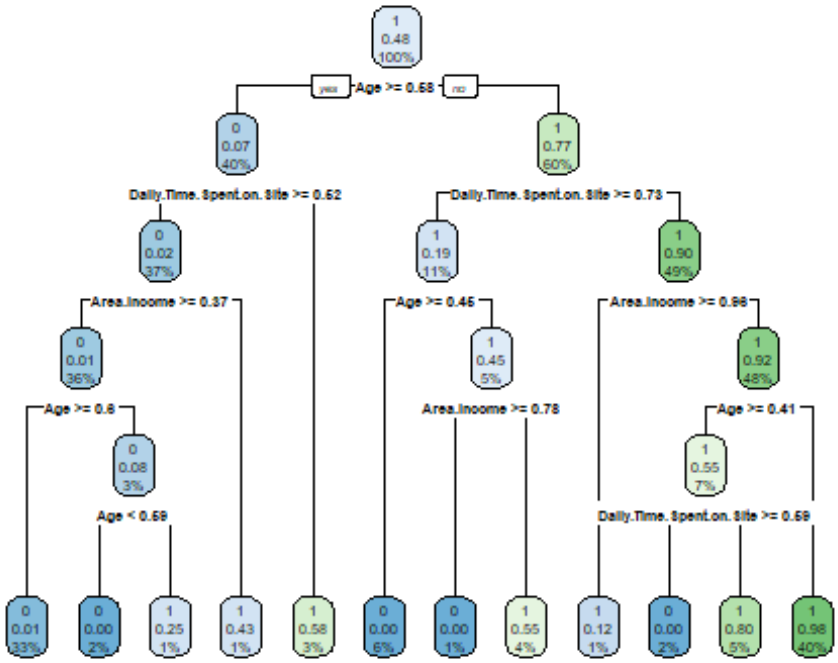
```
#Evaluating our model using a confusion matrix
tab <- table(test_sp,model)

#Calculating the accuracy score of our model
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(tab)

## [1] 100
```

Challenging the model

```
#Training our model with k of 3
model <- knn(train= ad_train,test=ad_test,cl= train_sp,k=3)
tab <- table(test_sp,model)
#Calculating the accuracy score of our model
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(tab)

## [1] 100
```

Decision Trees

```
library(caret)

## Loading required package: lattice

##
## Attaching package: 'caret'

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

library(mlbench)
library(rpart)
library("rpart.plot")

cols = c('Daily.Time.Spent.on.Site', 'Age', 'Area.Income',
'Daily.Internet.Usage', 'Male', 'Clicked.on.Ad')
ad = select(ads, cols)
#Lets now create test and train data sets
#Extracting the training set

ad_train <- ad[1:800,]
##Extracting the testing set
ad_test <- ad[801:1000,]
#Penalty matrix
penalty.matrix <- matrix(c(0,1,10,0), byrow=TRUE, nrow=2)
#Building the classification tree with rpart
tree <- rpart(Clicked.on.Ad~.,data=ad_train, parms = list(loss =
penalty.matrix), method = "class")
#Visulaizing the tree
rpart.plot(tree)
```




```
#Evaluating our model using a confusion matrix
p <- predict(tree, ad, type = "class")
a <- table(p, ad$Clicked.on.Ad)
a

##
## p     0   1
##   0 428   7
##   1  72 493

#Evaluating our model using the accuracy score
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(a)

## [1] 92.1
```

OUr model achieved an accuracy score of 92.1 % which is a good model