R Project - Identifying individuals most likely to click an ad

Geoffrey Chege

2022-06-04

1. Introduction

1.1 Defining the question

- Determine which individuals are most likely to click on an ad using supervised learning prediction models.

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 my services as a Data Science Consultant to help her identify which individuals are most likely to click on her ads.

1.3 Metric for success

- Accuracy score of 85% and above.

1.4 Experimental Design Taken

- Installing packages and loading libraries needed
- Loading the data
- Data Cleaning
- Exploratory Data Analysis:
  - Univariate Analysis
  - Bivariate Analysis
- Modelling
- Predictions and evaluation of the model
- Conclusion

1.5 Appropriateness of the available data

- 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

2. Installing and loading Necessary Packages

3. Loading the Data

```
ad <- read.csv("C:/Users/user/Downloads/advertising.csv") #Loading the dataset
head(ad) #previewing the first 5 elements of the data
```

```
##   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
```

4. Data Cleaning

4.1 Checking the attribute types

```
## 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 attribute types in the data are: numeric, integer and character.

4.2 converting time variable from character to date and time (POSIXct) format

```
ad$Timestamp <- as.POSIXct(ad$Timestamp, "%Y-%m-%d %H:%M:%S",tz = "GMT")
```

4.3 Checking for duplicates

```
duplicates <- ad[duplicated(ad),] #storing duplicates in a table called "duplicates"
duplicates #previewing the table
```

```
##  [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)
```

- The duplicates table is empty. This means that there are no duplicates in the dataset.

4.4 checking for null values

```
colSums(is.na(ad)) #Checking the total number of null values in each column
```

```
## 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 null values in the dataset

4.5 checking column names

```
names(ad) #Displaying column names
```

```
## [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"
```

- The data set has the above column names. Columns with more than one word have periods "." separating the words. I will replace the periods "." with underscores "_"

```r
names(ad) <- gsub("[.]", "_", names(ad)) #Replacing "." with "_"
```

- The above code replaces the periods "." with underscores "_".

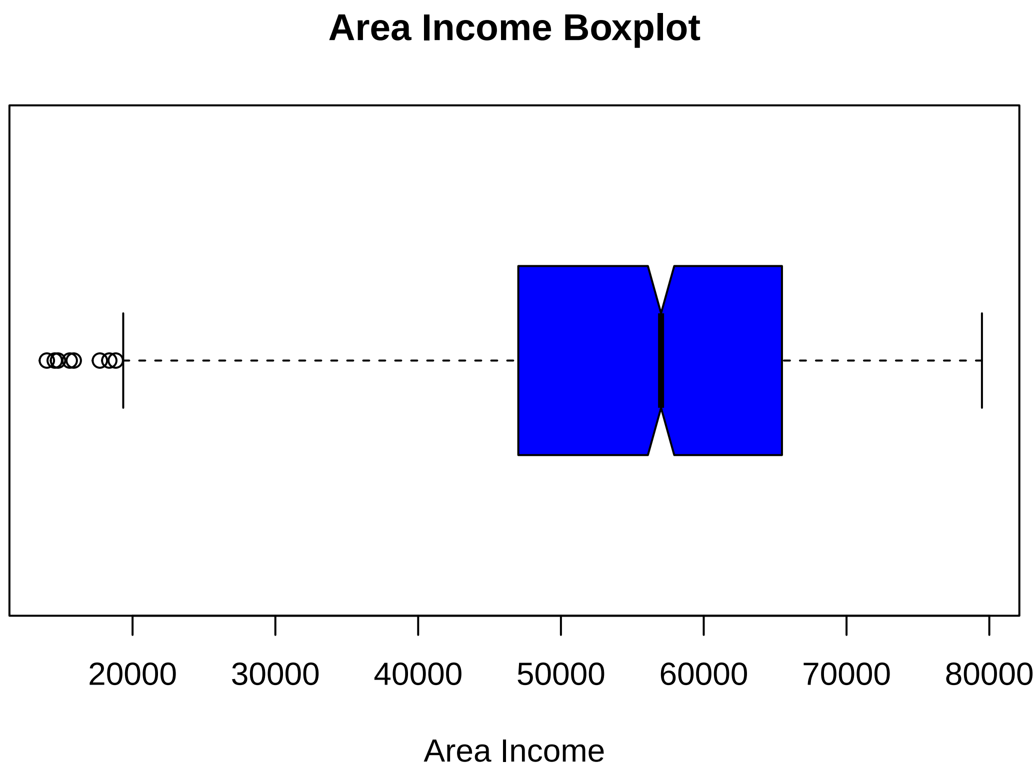
```r
names(ad) #Displaying column names
```

```
## [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"
```

4.6 Outliers

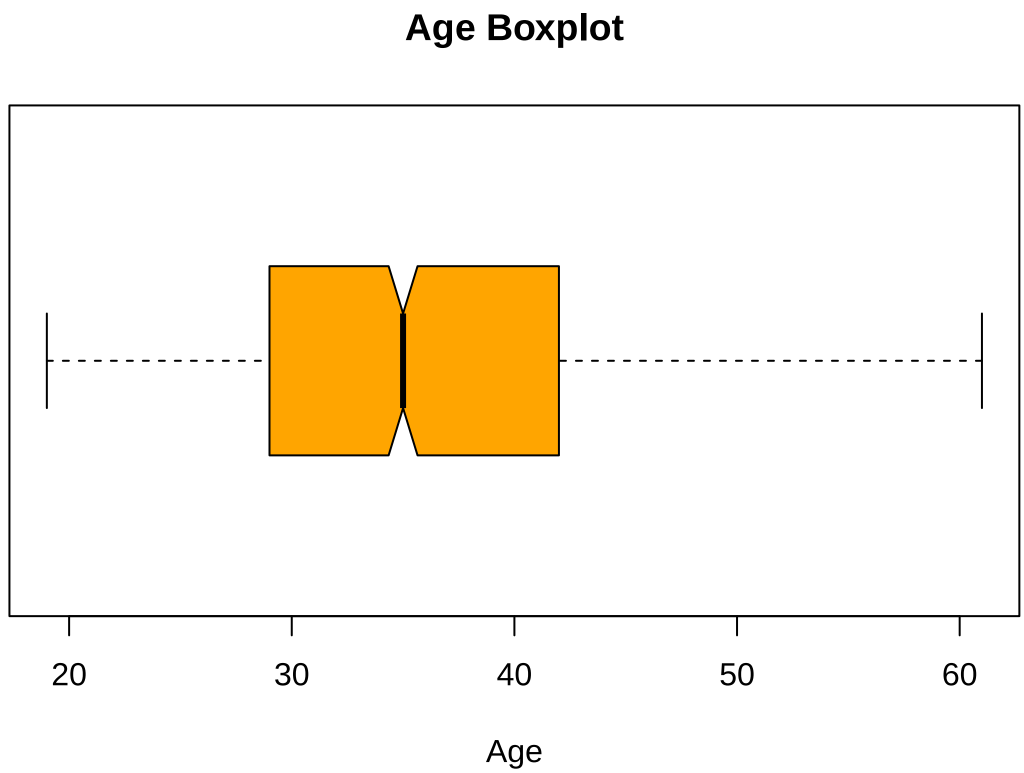
- I will use boxplots to check for outliers.

Boxplot for "Area_Income"



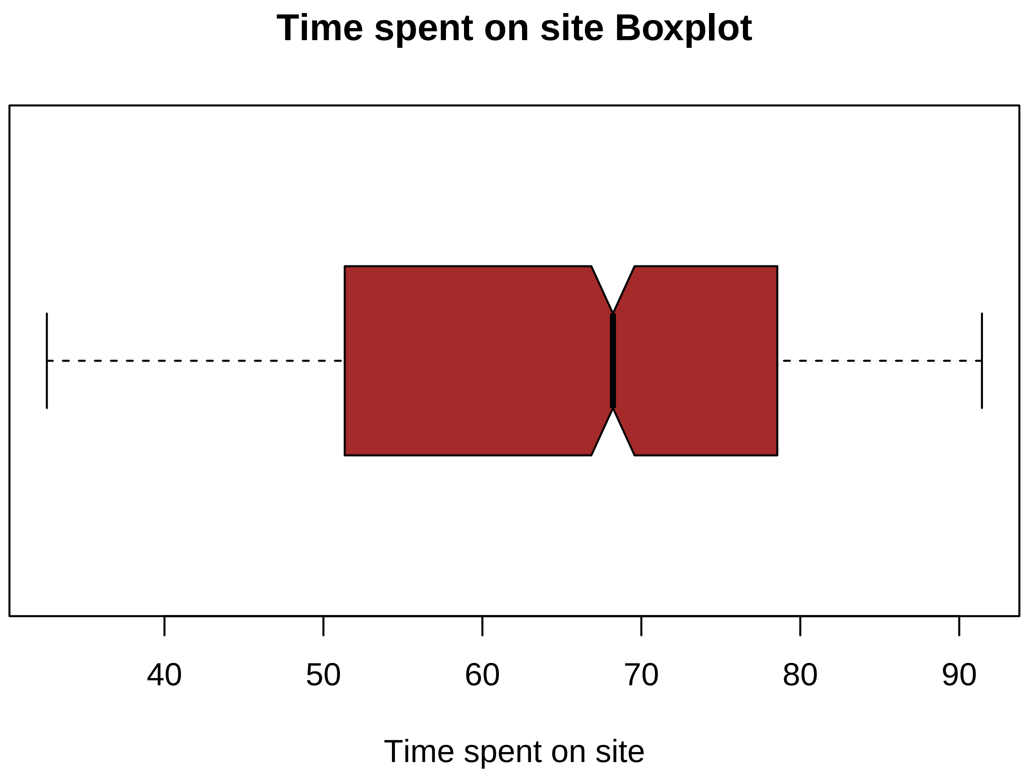
- There are few outliers in the "Area_Income" column. I will not remove them because they will be relevant in the analysis.

Boxplot for "Age"



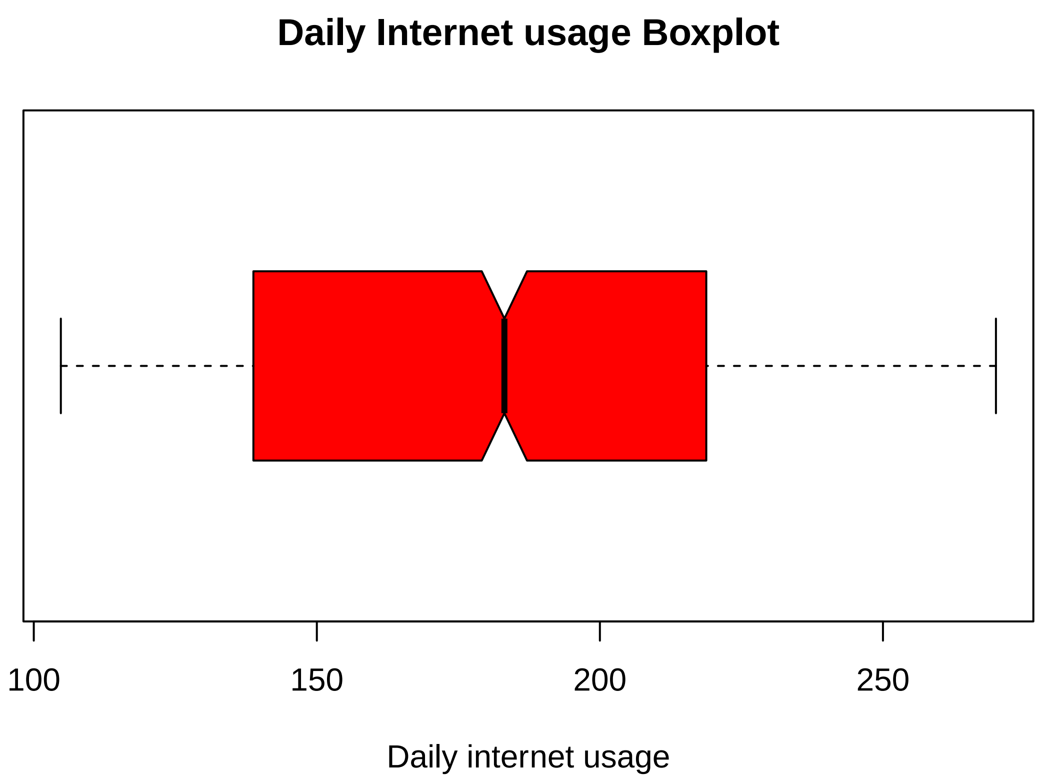
- There are no outliers in the "Age" column.

Boxplot for "Daily__Time__Spent__on__Site"



- There are no outliers in the "Time__Spent__on__Site" column.

Boxplot for "Daily__Internet__Usage"



- There are no outliers in the "Daily__Internet__Usage" column.

5. Exploratory Data Analysis

5.1 Univariate Analysis

- Summary statistics of the dataset

```
summary(ad)
```

```
##  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
```

- Using "describe()" function to get range, skewness, kurtosis and standard deviation. The "summary()" function does not give us this information.

```
describe(ad)
```

```
## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf

##                          vars    n     mean       sd   median  trimmed      mad
## Daily_Time_Spent_on_Site    1 1000    65.00    15.85    68.22    65.74    17.92
## Age                         2 1000    36.01     8.79    35.00    35.51     8.90
## Area_Income                 3 1000 55000.00 13414.63 57012.30 56038.94 13316.62
## Daily_Internet_Usage        4 1000   180.00    43.90   183.13   179.99    58.61
## Ad_Topic_Line*              5 1000   500.50   288.82   500.50   500.50   370.65
## City*                       6 1000   487.32   279.31   485.50   487.51   356.57
## Male                        7 1000     0.48     0.50     0.00     0.48     0.00
## Country*                    8 1000   116.41    69.94   114.50   115.82    89.70
## Timestamp                   9 1000      NaN       NA       NA      NaN       NA
## Clicked_on_Ad              10 1000     0.50     0.50     0.50     0.50     0.74
##                               min      max    range  skew kurtosis     se
## Daily_Time_Spent_on_Site    32.60    91.43    58.83 -0.37    -1.10   0.50
## Age                         19.00    61.00    42.00  0.48    -0.41   0.28
## Area_Income              13996.50 79484.80 65488.30 -0.65    -0.11 424.21
## Daily_Internet_Usage       104.78   269.96   165.18 -0.03    -1.28   1.39
## Ad_Topic_Line*               1.00  1000.00   999.00  0.00    -1.20   9.13
## City*                        1.00   969.00   968.00  0.00    -1.19   8.83
## Male                         0.00     1.00     1.00  0.08    -2.00   0.02
## Country*                     1.00   237.00   236.00  0.08    -1.23   2.21
## Timestamp                     Inf     -Inf     -Inf    NA       NA     NA
## Clicked_on_Ad                0.00     1.00     1.00  0.00    -2.00   0.02
```

From the "summary()" and "describe()" functions, the following measures of central tendency can be gathered:

Daily_Time_Spent_on_Site:

- mean: 65
- median: 68.22
- maximum: 91.43
- minimum: 32.60

- range: 58.83
- skew: -0.37
- kurtosis: -1.10

Age:

- mean: 36.01
- median: 35
- maximum: 61
- minimum: 19
- range: 42
- skew: 0.48
- kurtosis: -0.41

Area Income:

- mean: 55,000
- median: 57,012
- maximum: 79,484.8
- minimum: 13,996.5
- range: 65,488.30
- skew: -0.65
- kurtosis: -0.11

Daily_Internet_Usage:

- mean: 180
- median: 183.1
- maximum: 269.96
- minimum: 104.78
- range: 165.18
- skew: -0.03
- kurtosis: -1.28

Mode

- A function to determine the mode:

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

The most recurrent Ad Topic Line:

```
## [1] "Cloned 5thgeneration orchestration"
```

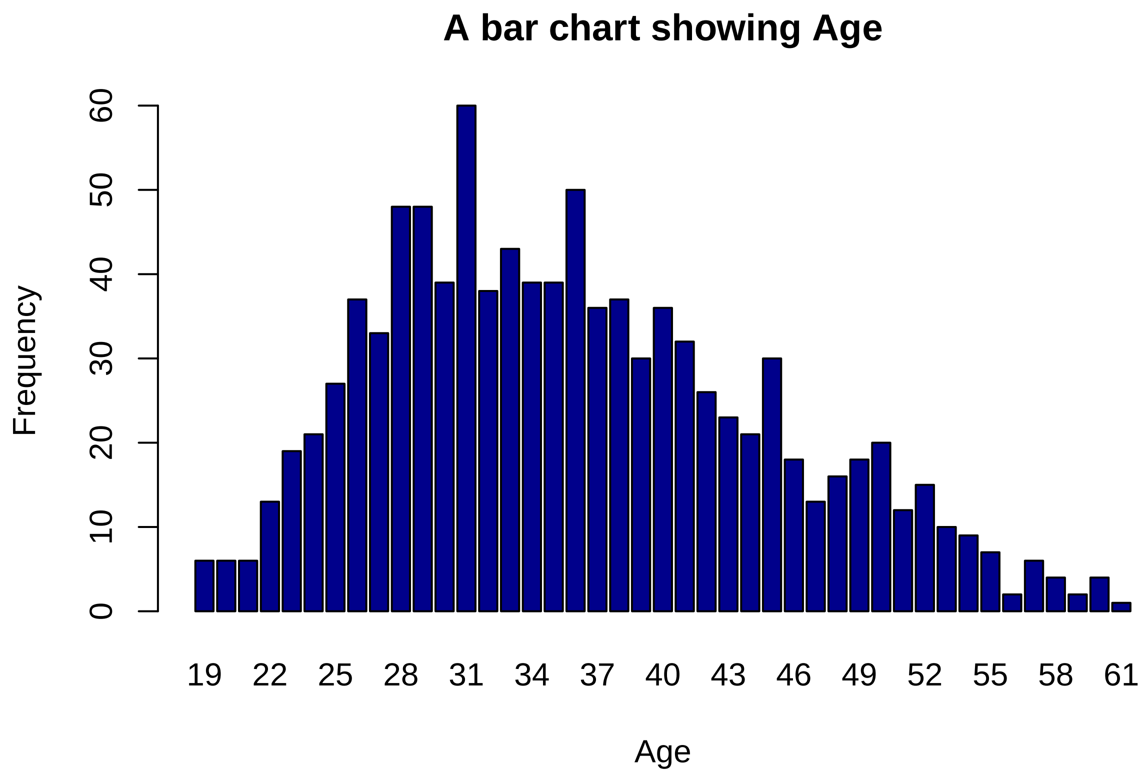
The most recurrent City:

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

The most recurrent Country:

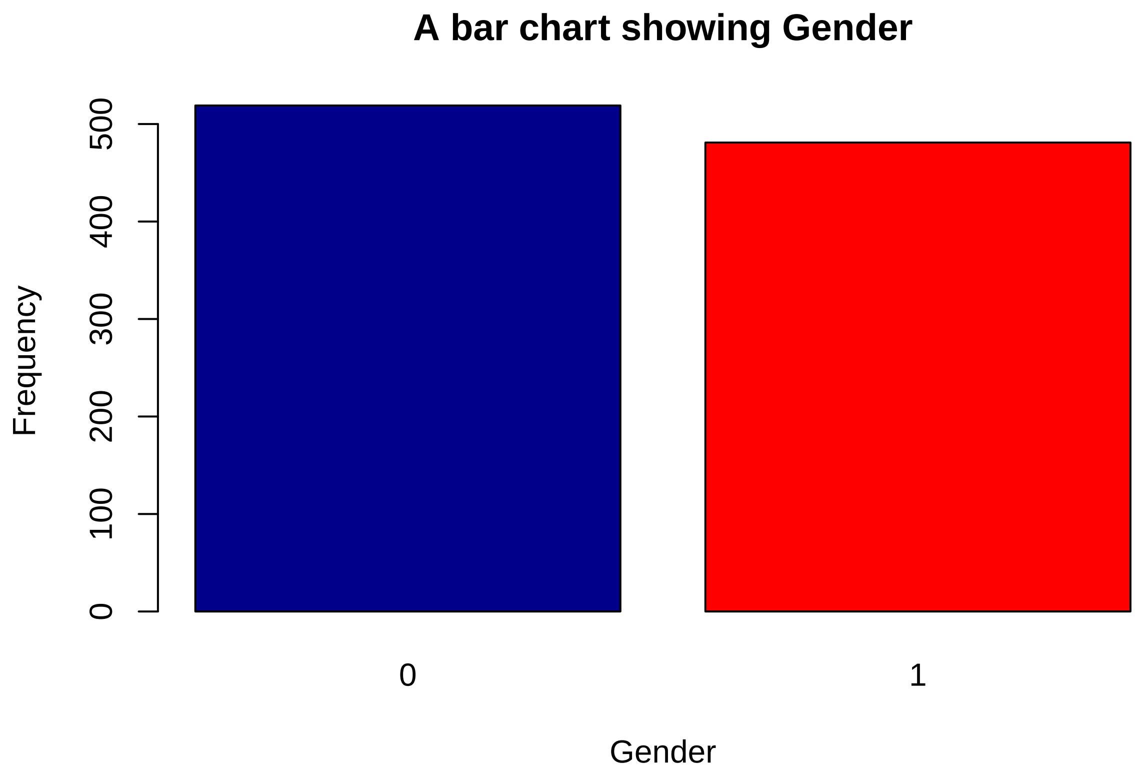
```
## [1] "Czech Republic"
```

- Checking the modal age using a barplot:



- From the plot, the modal age is 31.
- Checking the distribution in terms of gender where 1 is Male and 0 is Female:

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

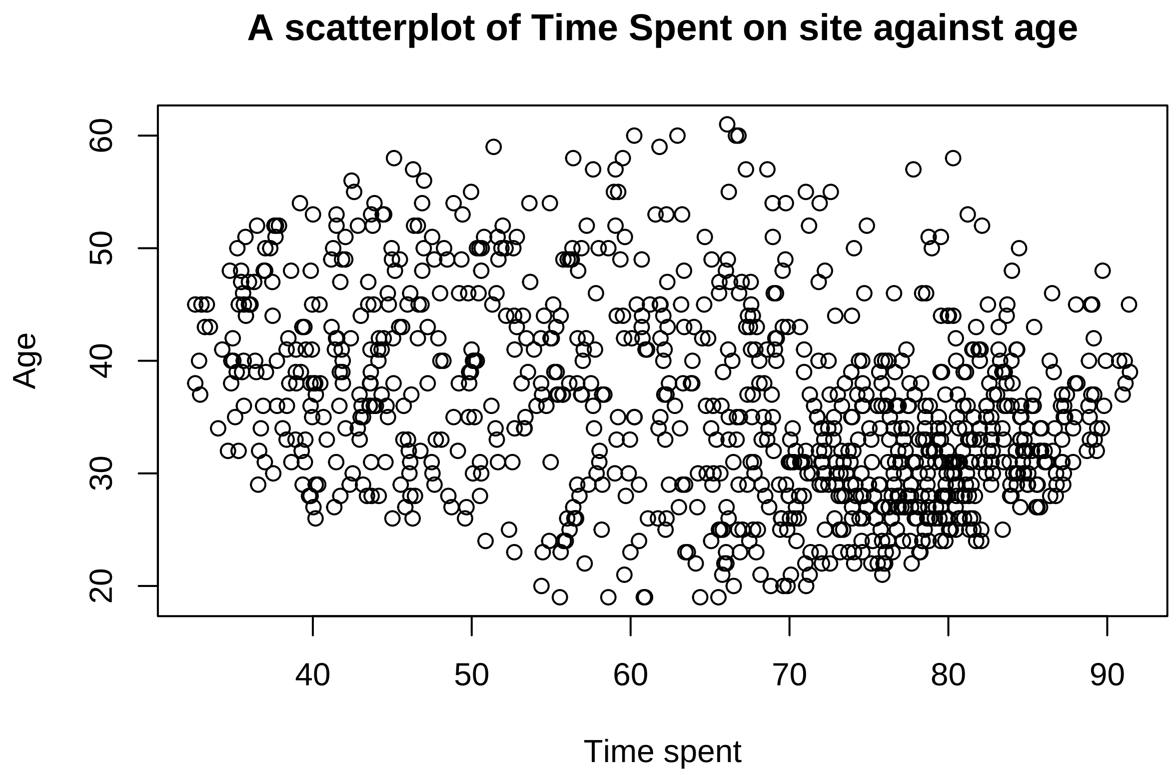


From this, there are More women than men, making female the modal gender.

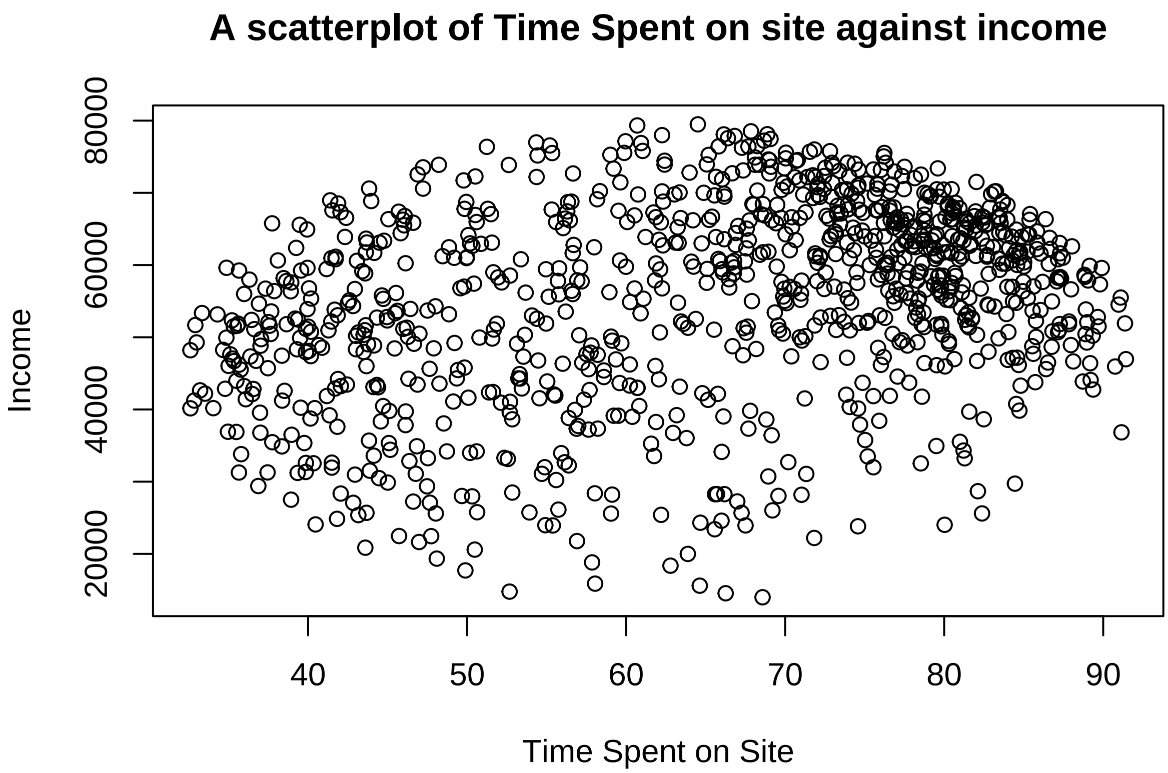
5.2 Bivariate Analysis

Scatterplots

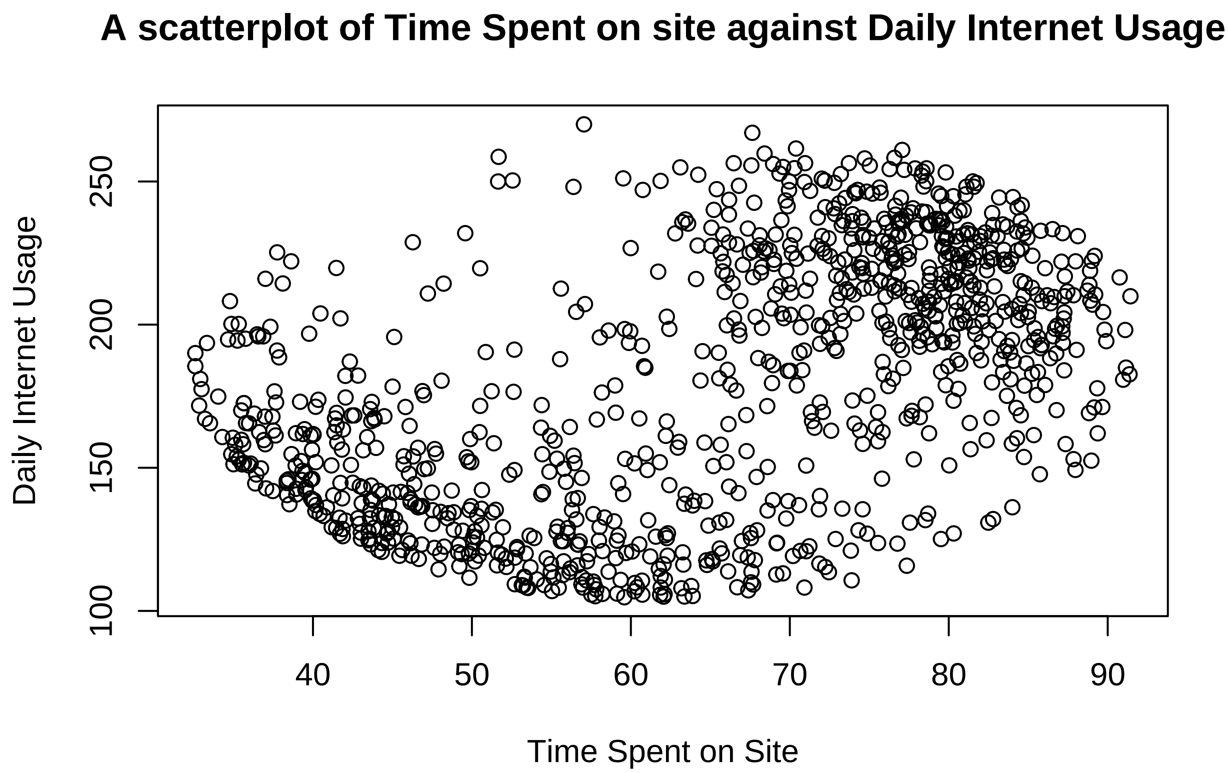
```
# scatterplot
plot((ad$Daily_Time_Spent_on_Site), (ad$Age),
     main = "A scatterplot of Time Spent on site against age",
     xlab = 'Time spent',
     ylab = 'Age')
```




```
# scatterplot of Time on site vs income
plot((ad$Daily_Time_Spent_on_Site), (ad$Area_Income),
     main = "A scatterplot of Time Spent on site against income",
     xlab = 'Time Spent on Site',
     ylab = 'Income')
```

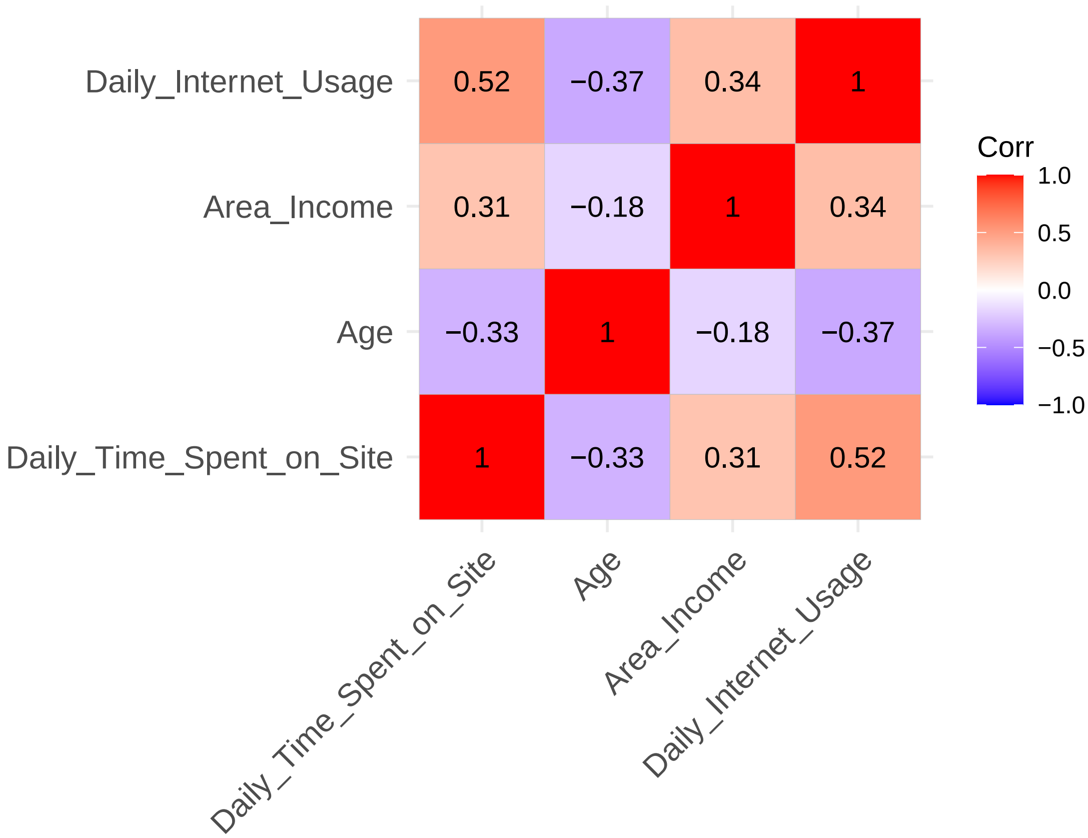



```
# scatterplot of Time on site vs Internet usage
plot((ad$Daily_Time_Spent_on_Site), (ad$Daily_Internet_Usage),
     main = "A scatterplot of Time Spent on site against Daily Internet Usage",
     xlab = 'Time Spent on Site',
     ylab = 'Daily Internet Usage')
```



Heatmap

```
# Heat map
# Checking the relationship between the variables
# Using Numeric variables only
numeric_tbl <- ad %>%
  select_if(is.numeric) %>%
  select(Daily_Time_Spent_on_Site, Age, Area_Income,Daily_Internet_Usage)
# Calculate the correlations
corr <- cor(numeric_tbl, use = "complete.obs")
ggcorrplot(round(corr, 2),
           type = "full", lab = T)
```

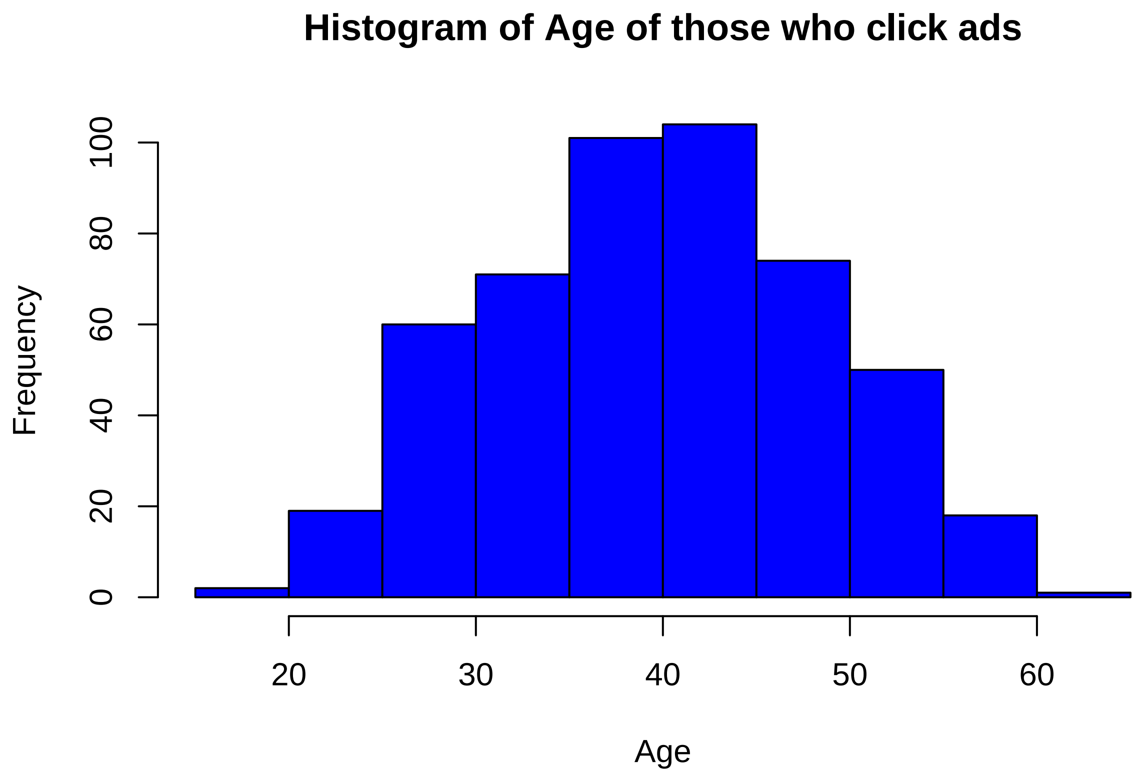


Analysis of those who clicked on ads:

```
# Analysis of people who click on the ads
ad_click <- ad[which(ad$Clicked_on_Ad == 1),] # Creating a new dataset that only has those who clicked
```

- Most popular age group of people clicking on ads:

```
# Most popular age group of people clicking on ads
hist((ad_click$Age),
     main = "Histogram of Age of those who click ads",
     xlab = 'Age',
     ylab = 'Frequency',
     col = "blue")
```



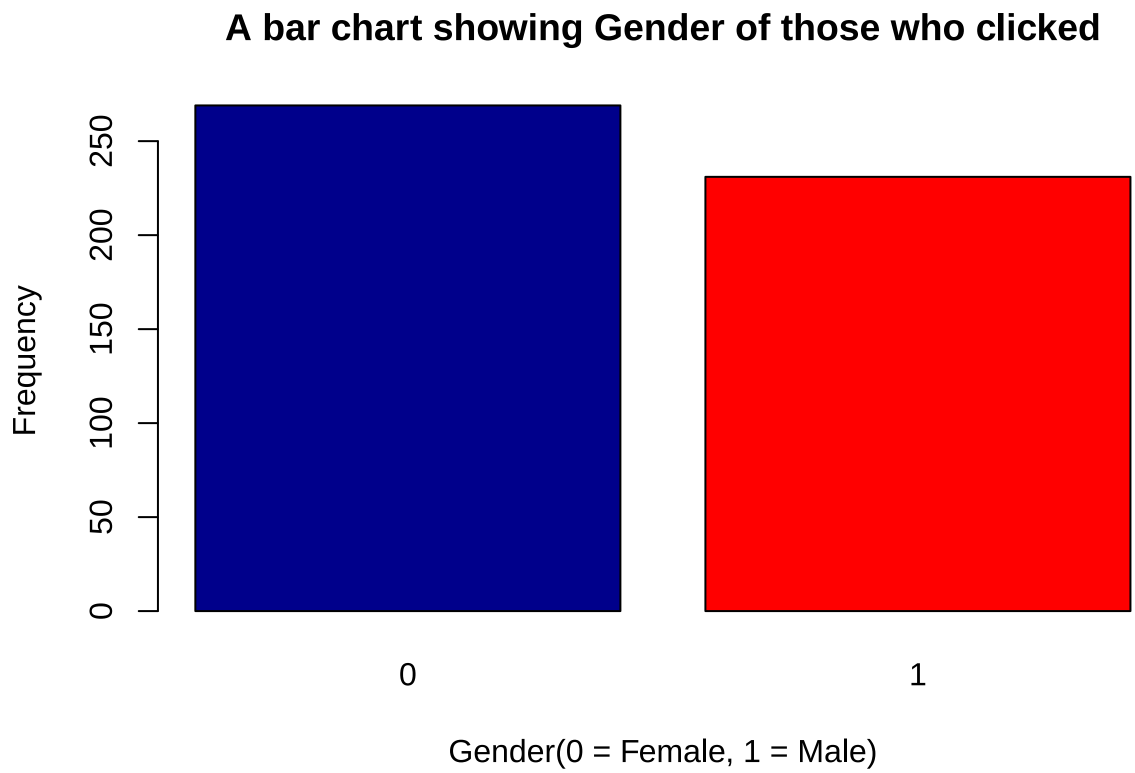
- 40 - 45 year olds click on the most ads.

Plotting to visualize the gender distribution:

```
gender2 <- (ad_click$Male)
gender2.frequency <- table(gender2)
gender2.frequency
```

```
## gender2
##   0   1
## 269 231
```

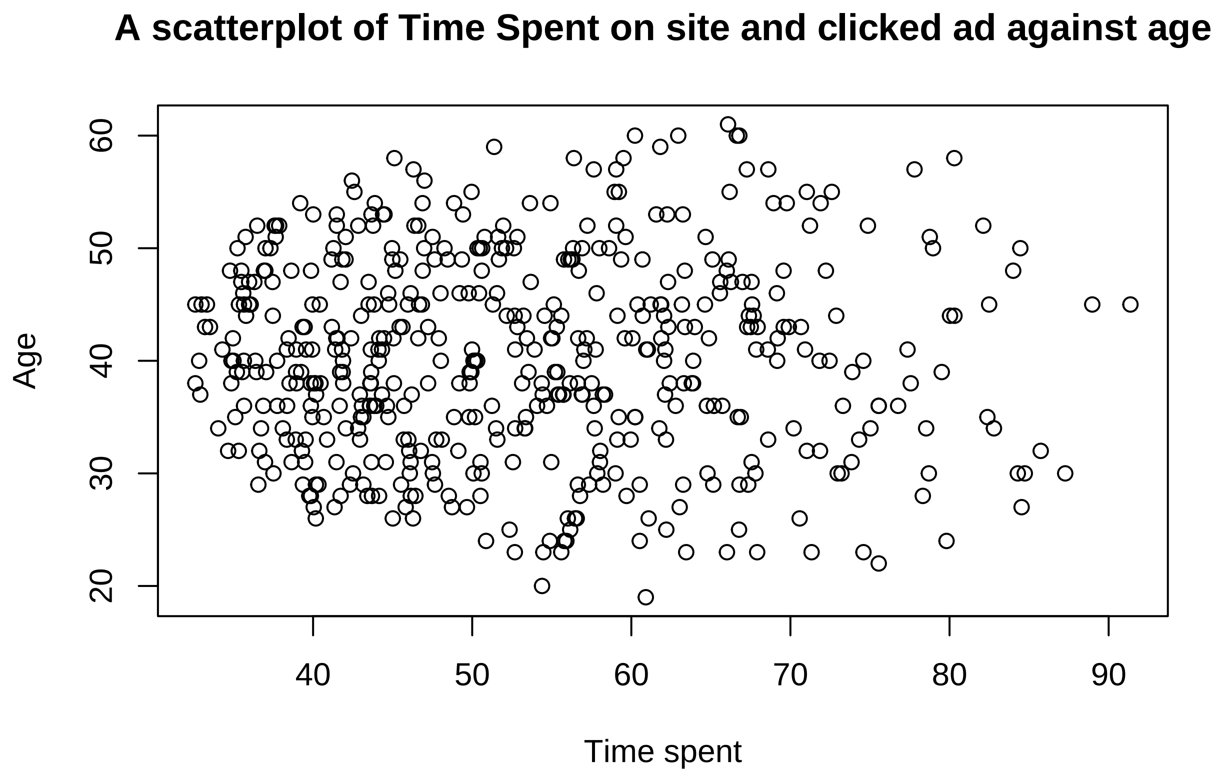
```
# plotting to visualize the gender distribution
barplot(gender2.frequency,
  main="A bar chart showing Gender of those who clicked",
  xlab="Gender(0 = Female, 1 = Male)",
  ylab = "Frequency",
  col=c("darkblue","red"),
  )
```



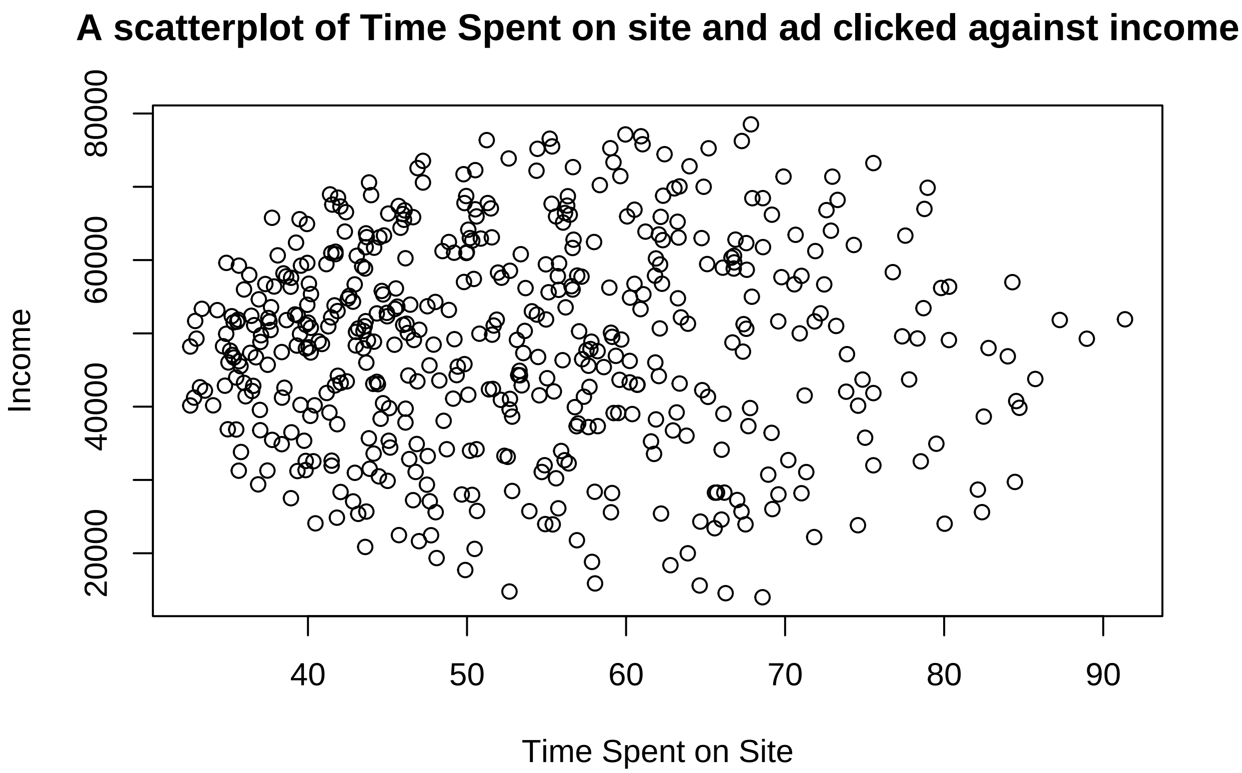
- Females clicked more ads than males.

Scatterplots of those who clicked:

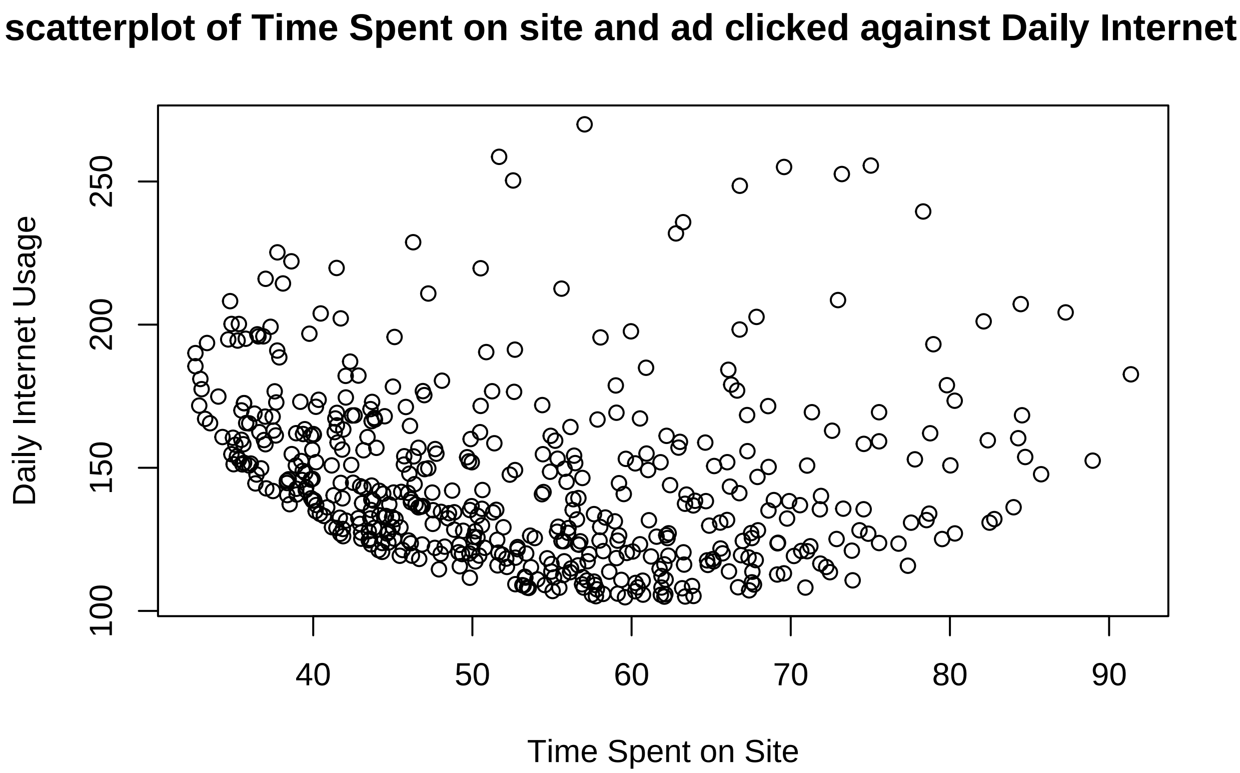
```
# scatterplot
plot((ad_click$Daily_Time_Spent_on_Site), (ad_click$Age),
     main = "A scatterplot of Time Spent on site and clicked ad against age",
     xlab = 'Time spent',
     ylab = 'Age')
```




```r
# scatterplot of Time on site vs income
plot((ad_click$Daily_Time_Spent_on_Site), (ad_click$Area_Income),
     main = "A scatterplot of Time Spent on site and ad clicked against income",
     xlab = 'Time Spent on Site',
     ylab = 'Income')
```



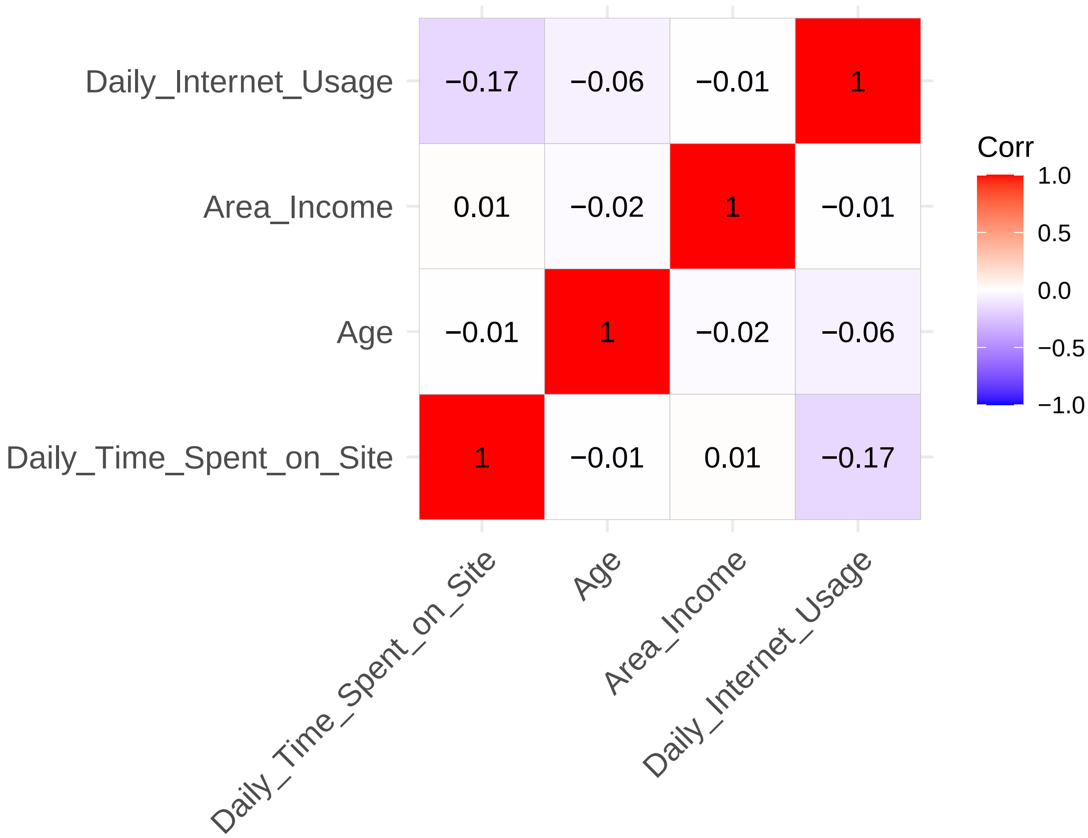

```r
# scatterplot of Time on site vs Internet usage
plot((ad_click$Daily_Time_Spent_on_Site), (ad_click$Daily_Internet_Usage),
     main = "A scatterplot of Time Spent on site and ad clicked against Daily Internet Usage",
     xlab = 'Time Spent on Site',
     ylab = 'Daily Internet Usage')
```




```r
# Heat map
# Checking the relationship between the variables

# Using Numeric variables only
numeric_tbl <- ad_click %>%
  select_if(is.numeric) %>%
  select(Daily_Time_Spent_on_Site, Age, Area_Income,Daily_Internet_Usage)

# Calculate the correlations
corr <- cor(numeric_tbl, use = "complete.obs")
ggcorrplot(round(corr, 2),
           type = "full", lab = T)
```



- There is low correlation between the numerical variables.
- The country with the most ad clicks:

```
mode(ad_click$Country)
```

```
## [1] "Australia"
```

- The income that clicks most:

```
mode(ad_click$Area_Income)
```

```
## [1] 24593.33
```

- Ad title that garners most clicks:

```
## [1] "Reactive local challenge"
```

- All the data profiling statistics will be organized into the report below

```
create_report(ad)
```

```
##
##
## processing file: report.rmd

##   |                                                                      |
##    inline R code fragments
##
##   |                                                                      |...
## label: global_options (with options)
## List of 1
##  $ include: logi FALSE
##
##   |                                                                      |.....
##   ordinary text without R code
##
##   |                                                                      |.......
## label: introduce
##   |                                                                      |........
##   ordinary text without R code
##
##   |                                                                      |..........
## label: plot_intro

##   |                                                                      |............
##   ordinary text without R code
##
##   |                                                                      |.............
## label: data_structure
##   |                                                                      |...............
##   ordinary text without R code
##
##   |                                                                      |.................
## label: missing_profile

##   |                                                                      |..................
##   ordinary text without R code
##
##   |                                                                      |..................
## label: univariate_distribution_header
##   |                                                                      |..................
##   ordinary text without R code
##
##   |                                                                      |..................
## label: plot_histogram

##   |                                                                      |..................
##   ordinary text without R code
##
##   |                                                                      |..................
## label: plot_density
##   |                                                                      |..................
##   ordinary text without R code
##
```

```
##   |                                                                  |.................
## label: plot_frequency_bar

##   |                                                                  |.................
##   ordinary text without R code
##
##   |                                                                  |.................
## label: plot_response_bar
##   |                                                                  |.................
##   ordinary text without R code
##
##   |                                                                  |.................
## label: plot_with_bar
##   |                                                                  |.................
##   ordinary text without R code
##
##   |                                                                  |.................
## label: plot_normal_qq

##   |                                                                  |.................
##   ordinary text without R code
##
##   |                                                                  |.................
## label: plot_response_qq
##   |                                                                  |.................
##   ordinary text without R code
##
##   |                                                                  |.................
## label: plot_by_qq
##   |                                                                  |.................
##   ordinary text without R code
##
##   |                                                                  |.................
## label: correlation_analysis

##   |                                                                  |.................
##   ordinary text without R code
##
##   |                                                                  |.................
## label: principal_component_analysis

##   |                                                                  |.................
##   ordinary text without R code
##
##   |                                                                  |.................
## label: bivariate_distribution_header
##   |                                                                  |.................
##   ordinary text without R code
##
##   |                                                                  |.................
## label: plot_response_boxplot
##   |                                                                  |.................
##   ordinary text without R code
```

```
##
##   |                                                                            |.................
## label: plot_by_boxplot
##   |                                                                            |.................
##   ordinary text without R code
##
##   |                                                                            |.................
## label: plot_response_scatterplot
##   |                                                                            |.................
##   ordinary text without R code
##
##   |                                                                            |.................
## label: plot_by_scatterplot

## output file: C:/Users/user/Documents/Geoffrey Chege Moringa IP W12/report.knit.md

## "C:/Program Files/RStudio/bin/quarto/bin/pandoc" +RTS -K512m -RTS "C:/Users/user/Documents/Geoffrey C

##
## Output created: report.html
```

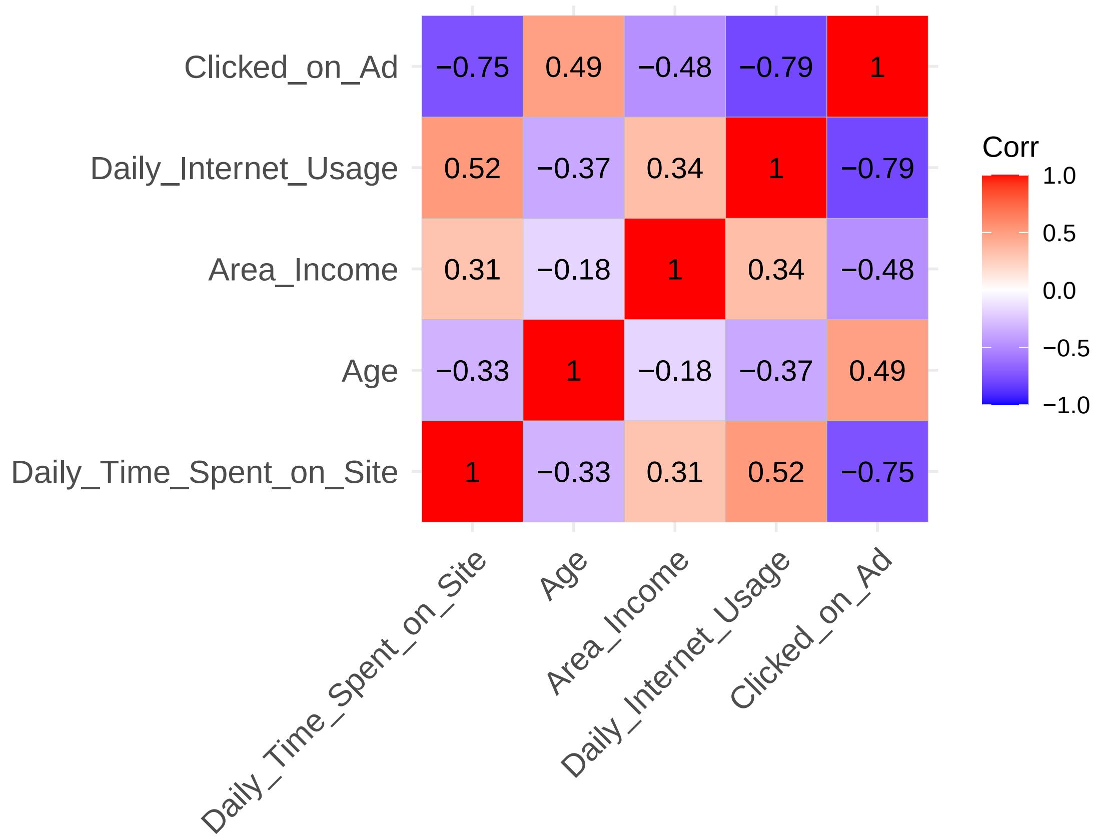
- A link to the report: "https://github.com/Geoffrey-Chege/Supervised-and-Unsupervised-Learning/blob/main/Ad%20Clicks/report.html"

6. Modelling

```
# Heat map
# Checking the relationship between the variables

# Using Numeric variables only
numeric_tbl2 <- ad %>%
  select_if(is.numeric) %>%
  select(Daily_Time_Spent_on_Site, Age, Area_Income,Daily_Internet_Usage, Clicked_on_Ad)

# Calculate the correlations
corr <- cor(numeric_tbl2, use = "complete.obs")
ggcorrplot(round(corr, 2),
           type = "full", lab = T)
```



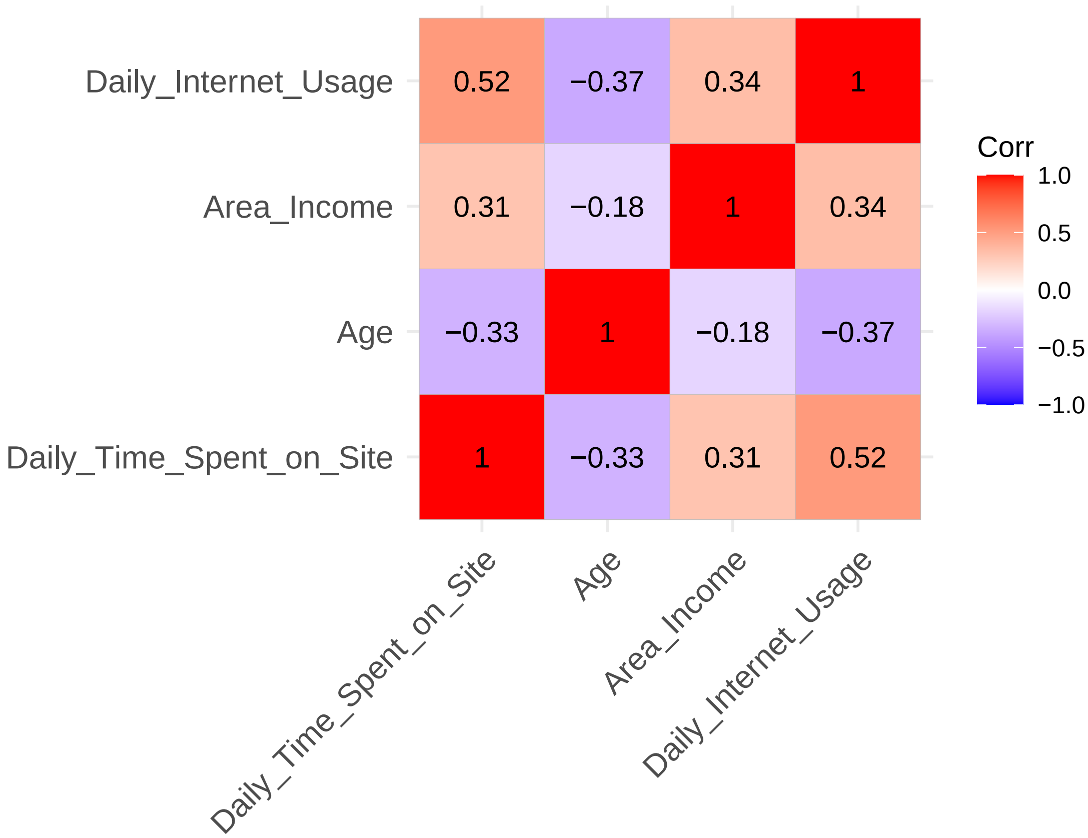
1. Daily_Time_Spent_on_Site and Clicked_on_Ad variables are strongly inversely related with a correlation of -0.75.
2. Daily_Internet_Usage and Clicked_on_Ad are strongly variable are strongly inversely related with a correlation of - 0.79.
3. Daily_Time_Spent_on_Site and Daily_Internet_Usage variables are positively related with 0.52. correlation.
4. Age and Daily_Internet_Usage variables are positively related with 0.49 correlation.

Clicked_on_Ad is the target variable so I will get correlation without it included.

```
# Heat map
# Checking the relationship between the variables

# Using Numeric variables only
numeric_tbl3 <- ad %>%
  select_if(is.numeric) %>%
  select(Daily_Time_Spent_on_Site, Age, Area_Income,Daily_Internet_Usage)

# Calculating the correlations
corr <- cor(numeric_tbl3, use = "complete.obs")
ggcorrplot(round(corr, 2),
           type = "full", lab = T)
```



- There are no highly correlated numeric independent variables, so I will use them all in analysis.

Normalizing the independent variables to ensure all the data is on the same scale

```
# Normalizing the dataset
normalize <- function(x){
  return ((x-min(x)) / (max(x)-min(x)))
}
ad$Daily_Time_Spent_on_Site <- normalize(ad$Daily_Time_Spent_on_Site)
ad$Age <- normalize(ad$Age)
ad$Area_Income <- normalize(ad$Area_Income)
ad$Male <- normalize(ad$Male)

#previewing normalized dataset
head(ad)
```

```
##   Daily_Time_Spent_on_Site       Age Area_Income Daily_Internet_Usage
## 1                0.6178820 0.3809524   0.7304725               256.09
## 2                0.8096209 0.2857143   0.8313752               193.77
## 3                0.6267211 0.1666667   0.6992003               236.50
## 4                0.7062723 0.2380952   0.6231599               245.89
## 5                0.6080231 0.3809524   0.9145678               225.58
## 6                0.4655788 0.0952381   0.6988280               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
```

- The dataset is on the same scale.

Splitting Data into Training and Testing Sets

```r
# splitting the data into training and testing sets
# I will split it 70:30
intrain <- createDataPartition(y = ad$Clicked_on_Ad, p = 0.7, list = FALSE)
training <- ad[intrain,]
testing <- ad[-intrain,]
```

```r
# checking the dimensions of our training and testing sets
dim(training)
```

```
## [1] 700  10
```

```r
dim(testing)
```

```
## [1] 300  10
```

- 700 of data will be used for training while 300 will be for testing.

```r
# checking the dimensions of our split
prop.table(table(ad$Clicked_on_Ad)) * 100
```

```
##
##  0  1
## 50 50
```

```r
prop.table(table(training$Clicked_on_Ad)) * 100
```

```
##
##  0  1
## 50 50
```

```r
prop.table(table(testing$Clicked_on_Ad)) * 100
```

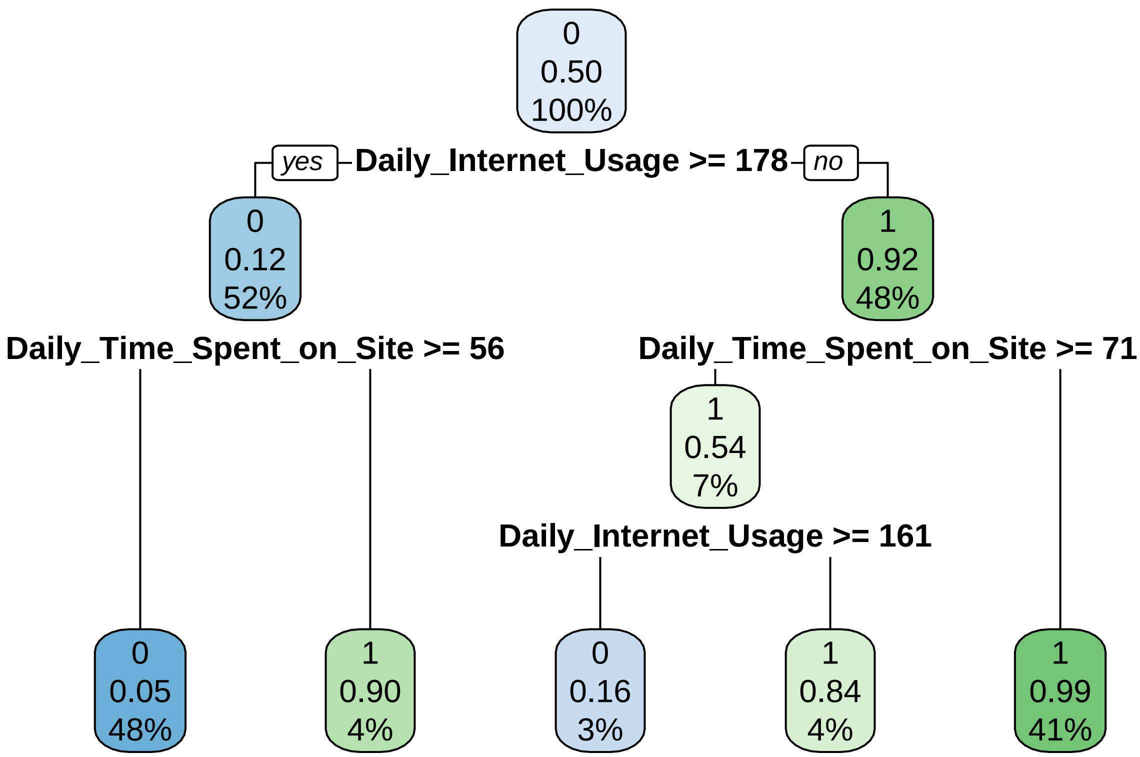
```
##
##  0  1
## 50 50
```

- The target data is equal in the data,training set and test set.

Decision Tree Classifier

```r
# Specifying target and predictor variables
m <- rpart(Clicked_on_Ad ~ . ,
         data = numeric_tbl2,
         method = "class")
```

```r
# Plotting model
rpart.plot(m)
```




```r
# Making predictions
p <- predict(m, numeric_tbl2, type ="class")

# Printing the confusion matrix
table(p, numeric_tbl2$Clicked_on_Ad)
```

```
##
## p     0   1
##   0 485  28
##   1  15 472
```

- The model correctly classified 485 did not clicks as '0' and 472 clicks as '1' . However, it also incorrectly classified 28 did not clicks as '1'(clicked) and 15 clicks as '0'(did not click).

```
# Printing the Accuracy
(mean(numeric_tbl2$Clicked_on_Ad == p))*100
```

```
## [1] 95.7
```

- The model has an accuracy of 95.7%
- This is a good model for making predictions

7. Conclusions

- Decision Tree gives an accuracy of 95.7%
- The females have the majority site visits but they don't click on the ad.
- The minimum age of the participant was 19 years old while the oldest was 60 years old.
- The minimum daily time spent on the site was 32 minutes while the maximum time spent was 91 minutes.
- The youth have most site visits as compared to the teenagers and older people.

8. Recommendations

- Appropriate content targeting different age groups should be uploaded when it comes to the ads. This will lead to an increase in the number of clicks on ads.
- There should be more locally targeted ads, seeing as the key word 'local' prompted more clicks.