# WEEK 13 IP

Cherono K

9/3/2020

## WEEK 13 IP

### 1. Defining the Question

In this week's project I'll be working as a Data Science Consultant to a Kenyan entrepreneur who 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.

### 2. Defining the Metric for Success

Our metrics for success would be identifying the individuals who are most likely to click on the adds by identifying their gender, age, country and city.

Our metric for success in our modelling would be building a model with an accuracy score of above 90%

### 3. Understanding the context

Every mainstream media website, news site, top blog, YouTube, and social media site uses advertising, and it's because advertising is a proven moneymaker. If you're a blogger with an audience that companies want to reach, you have the potential to make money by selling ads. Therefore, it's important that you know what type of advertising is going to sell best to your specific audience. As such, learning stats about your audience can help you to learn the basic demographics of your audience. The more you know about your audience, the better you'll be able to sell advertising to them.

### 4. Recording the Experimental Design

The following are the steps taken to implement the solution :

- Define the question, the metric for success, the context, experimental design taken.
- Read and explore the given dataset.
- Define the appropriateness of the available data to answer the given question.
- Find and deal with outliers, anomalies, and missing data within the dataset.
- Perform Univariate and Bivariate Analysis recording our observations.
- Build Supervised Learning Models i.e. KNN, Decision Trees, SVM AND Naive Bayes
- Provide a conclusion and recommendation from the analysis.

## 5. Data Relevance

Our data is very relevant to our research question. As had mentioned earlier, the more you know about your audience, the better you'll be able to sell advertising to them. The dataset provided has very relevant information about the blog's audience.

## 6. Reading the Data

```
# Importing libraries
#install.packages("moments",repos="http://cran.us.r-project.org")
library(moments)
library(ggcorrplot)
```

```
## Loading required package: ggplot2
```

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------ tidyverse 1.3.0 --

## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## v purrr   0.3.4

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

```
#Loading the Dataset
advertising_dataset <- read.csv("http://bit.ly/IPAdvertisingData")
```

## 6. Checking the Data

### a.) Previewing the Data

```
#Previewing the top of our dataset
head(advertising_dataset)
```

```
##   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 bottom of the dataset
tail(advertising_dataset)
```

```
##      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
```

```
# Checking the names of the columns
names(advertising_dataset)
```

```
##  [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"
```

```
#Displaying the structure of our dataset
str(advertising_dataset)
```

```
## '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 standardi
##  $ 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                : chr  "2016-03-27 00:53:11" "2016-04-04 01:39:02" "2016-03-13 20:35:42"
##  $ Clicked.on.Ad            : int  0 0 0 0 0 0 0 1 0 0 ...
```

```r
#Checking the dimension of our dataset
dim(advertising_dataset)
```

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

```r
#Checking the class of our data set using the class() function
class(advertising_dataset)
```

```
## [1] "data.frame"
```

```r
# Checking the datatypes for each column
columns = colnames(advertising_dataset)
for (column in seq(length(colnames(advertising_dataset)))){
    print(columns[column])
    print(class(advertising_dataset[, column]))
    cat('\n')
}
```

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

```
## [1] "Timestamp"
## [1] "character"
##
## [1] "Clicked.on.Ad"
## [1] "integer"
```

Our data frame has 1000 entries and 10 columns. The columns in our dataset are numerical, character and integer. However, the gender and clicked on Ad column are classified as integer data types but they are categorical columns.

### b.) Null Values

```
#Checking the number of missing data in our dataset
sum(is.na(advertising_dataset))
```

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

### c.) Duplicates

```
#Checking for duplicated data
duplicated <- advertising_dataset[duplicated(advertising_dataset),]
duplicated
```

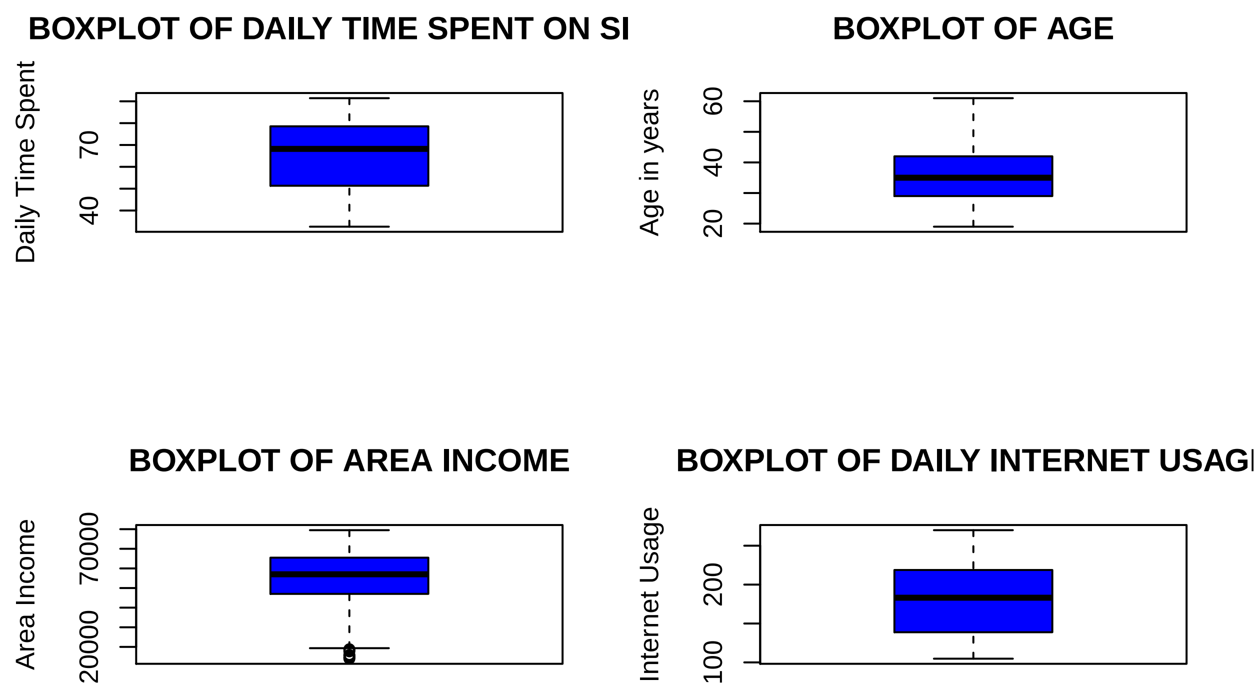
```
##  [1] Daily.Time.Spent.on.Site Age                      Area.Income
##  [4] Daily.Internet.Usage     Ad.Topic.Line            City
##  [7] Male                     Country                  Timestamp
## [10] Clicked.on.Ad
## <0 rows> (or 0-length row.names)
```

Our data frame has no null values and no dulpicated entries.

### d.) Outliers

```
# Checking for outliers on the numeric columns

#Checking for outliers in the Daily Time Spent on Site column
par(mfrow=c(2,2))
boxplot(advertising_dataset$Daily.Time.Spent.on.Site,
        main = toupper("Boxplot of Daily Time Spent on Site"),
        ylab = "Daily Time Spent",
        col = "blue")
#Checking for outliers in the Age column
boxplot(advertising_dataset$Age,
        main = toupper("Boxplot of Age"),
        ylab = "Age in years",
        col = "blue")
#Checking for outliers in the Area Income column
```

```r
boxplot(advertising_dataset$Area.Income,
        main = toupper("Boxplot of Area Income"),
        ylab = "Area Income",
        col = "blue")
#Checking for outliers in the Daily Internet Usage column
boxplot(advertising_dataset$Daily.Internet.Usage,
        main = toupper("Boxplot of Daily Internet Usage"),
        ylab = "Internet Usage",
        col = "blue")
```




We only have outliers in the Area Income column. However we'll not remove these outliers as they are actual information about the individuals who viwe the blog.

## 7. Tidying the Dataset

```r
#Changing the  name of the Male column to Gender
colnames(advertising_dataset)[colnames(advertising_dataset) == 'Male'] = 'Gender'
colnames(advertising_dataset)
```

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

```r
#Separating the date from the time stamp column
Dates <- format(as.POSIXct(strptime(advertising_dataset$Timestamp,"%Y-%m-%d %H:%M:%S",tz="")) ,format =
head(Dates)
```

```
## [1] "2016-03-27" "2016-04-04" "2016-03-13" "2016-01-10" "2016-06-03"
## [6] "2016-05-19"
```

```r
#Separating the time from the Timestamp column
Time <- format(as.POSIXct(strptime(advertising_dataset$Timestamp,"%Y-%m-%d %H:%M:%S",tz="")) ,format =
head(Time)
```

```
## [1] "00:53:11" "01:39:02" "20:35:42" "02:31:19" "03:36:18" "14:30:17"
```

```r
#Creating new columns for date and time
advertising_dataset$Dates <- Dates
advertising_dataset$Time <- Time
head(advertising_dataset)
```

```
##   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 Gender    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      Dates     Time
## 1 2016-03-27 00:53:11             0 2016-03-27 00:53:11
## 2 2016-04-04 01:39:02             0 2016-04-04 01:39:02
## 3 2016-03-13 20:35:42             0 2016-03-13 20:35:42
## 4 2016-01-10 02:31:19             0 2016-01-10 02:31:19
## 5 2016-06-03 03:36:18             0 2016-06-03 03:36:18
## 6 2016-05-19 14:30:17             0 2016-05-19 14:30:17
```

```r
#Separating the Year, month and day into different column.
advertising_dataset <- separate(advertising_dataset, "Dates", c("Year", "Month", "Day"), sep = "-")
#Separating the houe, minutes and seconds into different columns
advertising_dataset <- separate(advertising_dataset, "Time", c("Hour", "Minutes", "Seconds"), sep = ":"
```

```r
#Changing the data type for gender and clicked on ad from integer to factor
cols <- c('Gender' ,'Clicked.on.Ad','Year','Month','Day','Hour','Minutes','Seconds', 'City','Country')
advertising_dataset[,cols] <- lapply(advertising_dataset[,cols] , factor)
# Checking the datatypes for each column
columns = colnames(advertising_dataset)
for (column in seq(length(colnames(advertising_dataset)))){
```

```r
    print(columns[column])
    print(class(advertising_dataset[, column]))
    cat('\n')
}
```

```
## [1] "Daily.Time.Spent.on.Site"
## [1] "numeric"
##
## [1] "Age"
## [1] "integer"
##
## [1] "Area.Income"
## [1] "numeric"
##
## [1] "Daily.Internet.Usage"
## [1] "numeric"
##
## [1] "Ad.Topic.Line"
## [1] "character"
##
## [1] "City"
## [1] "factor"
##
## [1] "Gender"
## [1] "factor"
##
## [1] "Country"
## [1] "factor"
##
## [1] "Timestamp"
## [1] "character"
##
## [1] "Clicked.on.Ad"
## [1] "factor"
##
## [1] "Year"
## [1] "factor"
##
## [1] "Month"
## [1] "factor"
##
## [1] "Day"
## [1] "factor"
##
## [1] "Hour"
## [1] "factor"
##
## [1] "Minutes"
## [1] "factor"
##
## [1] "Seconds"
## [1] "factor"
```

```r
library(tidyr)
#Dropping the Ad Topic line column and the Timestamp column.
advertising_dataset = advertising_dataset[,!(names(advertising_dataset) %in% c("Ad.Topic.Line","Timestam
colnames(advertising_dataset)
```

```
##  [1] "Daily.Time.Spent.on.Site" "Age"
##  [3] "Area.Income"              "Daily.Internet.Usage"
##  [5] "City"                     "Gender"
##  [7] "Country"                  "Clicked.on.Ad"
##  [9] "Year"                     "Month"
## [11] "Day"                      "Hour"
## [13] "Minutes"                  "Seconds"
```

## 8. Exploratory Data Analysis

### a.) Univariate Analysis

```r
# Summary statistics of the data
(summary(advertising_dataset))
```

#### i.) Measures of Central Tendency

```
##  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
##
##               City      Gender            Country     Clicked.on.Ad   Year
##  Lisamouth      :  3   0:519   Czech Republic:  9   0:500         2016:1000
##  Williamsport   :  3   1:481   France        :  9   1:500
##  Benjaminchester:  2           Afghanistan   :  8
##  East John      :  2           Australia     :  8
##  East Timothy   :  2           Cyprus        :  8
##  Johnstad       :  2           Greece        :  8
##  (Other)        :986           (Other)       :950
##  Month         Day            Hour          Minutes         Seconds
##  01:147   03     : 46   07     : 54   02     : 26   22     : 28
##  02:160   17     : 42   20     : 50   07     : 24   10     : 27
##  03:156   15     : 41   09     : 49   13     : 24   35     : 27
##  04:147   10     : 37   21     : 48   10     : 22   37     : 27
##  05:147   04     : 36   00     : 45   21     : 21   38     : 24
##  06:142   26     : 36   05     : 44   33     : 21   15     : 23
##  07:101   (Other):762   (Other):710   (Other):862   (Other):844
```

The above output gives us the minimum, maximum, median. mean, 1st and 3rd quantile of each of our numerical columns.

```r
#Getting the mode
#Creating a function to calculate the mode
getmode <- function(v){
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v,uniqv)))]
}
#Getting the mode of each continous variable
getmode(advertising_dataset$Daily.Time.Spent.on.Site)
```

```
## [1] 62.26
```

```r
getmode(advertising_dataset$Age)
```

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

```r
getmode(advertising_dataset$Area.Income)
```

```
## [1] 61833.9
```

```r
getmode(advertising_dataset$Daily.Internet.Usage)
```

```
## [1] 167.22
```

The above output is the mode of each numerical column.

Most individuals in our data frame spend 62.26 minutes on the site, they are aged 31, they have an area income of $61,833.9 and their daily internet usage is 167.22

```r
#Finding the Range of numerical variables
range(advertising_dataset$Daily.Time.Spent.on.Site, na.rm=TRUE)
```

### ii.) Measures of Dispersion

```
## [1] 32.60 91.43
```

```r
max(advertising_dataset$Daily.Time.Spent.on.Site, na.rm=TRUE) - min(advertising_dataset$Daily.Time.Spent
```

```
## [1] 58.83
```

```r
range(advertising_dataset$Age, na.rm=TRUE)
```

```
## [1] 19 61
```

```r
max(advertising_dataset$Age, na.rm=TRUE) - min(advertising_dataset$Age, na.rm=TRUE)
```

```
## [1] 42
```

```r
range(advertising_dataset$Area.Income, na.rm=TRUE)
```

```
## [1] 13996.5 79484.8
```

```r
max(advertising_dataset$Area.Income, na.rm=TRUE) - min(advertising_dataset$Area.Income, na.rm=TRUE)
```

```
## [1] 65488.3
```

```r
range(advertising_dataset$Daily.Internet.Usage, na.rm=TRUE)
```

```
## [1] 104.78 269.96
```

```r
max(advertising_dataset$Daily.Internet.Usage, na.rm=TRUE) - min(advertising_dataset$Daily.Internet.Usag
```

```
## [1] 165.18
```

The range of each numerical value is as listed below:

- Daily Time Spent on Site is 58.83 minutes with a maximum of 91.43 minutes and 32.60 minutes
- Age is 41 years with the maximum being 61 years and minimum being 19 years.
- Area Income is 65,488.3 with a maximum of 79,484.8 and minimum of 13,996.5.
- Daily Internet usage is 165.18 with a maximum of 269.96 and minimum of 104.78.

```r
#Finding the Interquartile Range
quantile(advertising_dataset$Daily.Time.Spent.on.Site, na.rm=TRUE)
```

```
##      0%     25%     50%     75%    100% 
## 32.6000 51.3600 68.2150 78.5475 91.4300
```

```r
quantile(advertising_dataset$Age, na.rm=TRUE)
```

```
##   0%  25%  50%  75% 100% 
##   19   29   35   42   61
```

```r
quantile(advertising_dataset$Area.Income, na.rm=TRUE)
```

```
##       0%      25%      50%      75%     100% 
## 13996.50 47031.80 57012.30 65470.64 79484.80
```

```r
quantile(advertising_dataset$Daily.Internet.Usage, na.rm=TRUE)
```

```
##       0%      25%      50%      75%     100% 
## 104.7800 138.8300 183.1300 218.7925 269.9600
```

The above output gives the quantiles of each of the numeric columns.

```r
#Finding the Standard Deviation
sd(advertising_dataset$Daily.Time.Spent.on.Site, na.rm=TRUE)
```

```
## [1] 15.85361
```

```r
sd(advertising_dataset$Age, na.rm=TRUE)
```

```
## [1] 8.785562
```

```r
sd(advertising_dataset$Area.Income, na.rm=TRUE)
```

```
## [1] 13414.63
```

```r
sd(advertising_dataset$Daily.Internet.Usage, na.rm=TRUE)
```

```
## [1] 43.90234
```

```r
#Finding the Variance
var(advertising_dataset$Daily.Time.Spent.on.Site, na.rm=TRUE)
```

```
## [1] 251.3371
```

```r
var(advertising_dataset$Age, na.rm=TRUE)
```

```
## [1] 77.18611
```

```r
var(advertising_dataset$Area.Income, na.rm=TRUE)
```

```
## [1] 179952406
```

```r
var(advertising_dataset$Daily.Internet.Usage, na.rm=TRUE)
```

```
## [1] 1927.415
```

```r
#Finding the Skewness
skewness(advertising_dataset$Daily.Time.Spent.on.Site, na.rm=TRUE)
```

```
## [1] -0.3712026
```

```r
skewness(advertising_dataset$Age, na.rm=TRUE)
```

```
## [1] 0.4784227
```

```r
skewness(advertising_dataset$Area.Income, na.rm=TRUE)
```

```
## [1] -0.6493967
```

```r
skewness(advertising_dataset$Daily.Internet.Usage, na.rm=TRUE)
```

```
## [1] -0.03348703
```

```r
#Finding the Kurtosis
kurtosis(advertising_dataset$Daily.Time.Spent.on.Site, na.rm=TRUE)
```

```
## [1] 1.903942
```

```r
kurtosis(advertising_dataset$Age, na.rm=TRUE)
```

```
## [1] 2.595482
```

```r
kurtosis(advertising_dataset$Area.Income, na.rm=TRUE)
```

```
## [1] 2.894694
```

```r
kurtosis(advertising_dataset$Daily.Internet.Usage, na.rm=TRUE)
```

```
## [1] 1.727701
```

- The standard deviation, variance, skewness and kurtosis of each numeric variable is as listed in the above outputs.
- Only Age is positively skewed implying the mean age is greater than the mode. The other variables are negatively skewed.
- The kurtosis of each of the numeric variable is greater than zero. This implies that our dataframe has outliers as we had seen in the boxplots plotted. The area income has high kurtosis and as we had seen, it had outliers present.

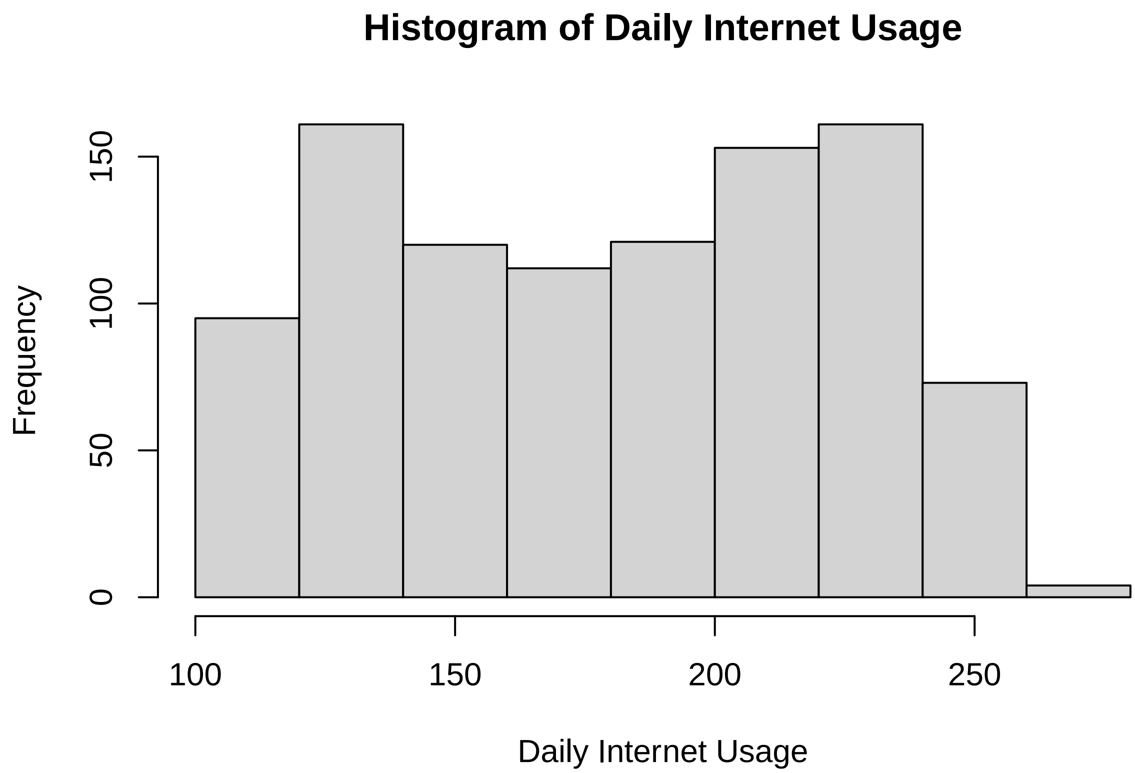
### iii.) Histograms

We'll plot histograms of each of the numerical data types to see their distribution.

```r
#Plotting a histogram for age
hist(advertising_dataset$Age,
     main = "Histogram of Age",
     xlab = "Age in Years")
```

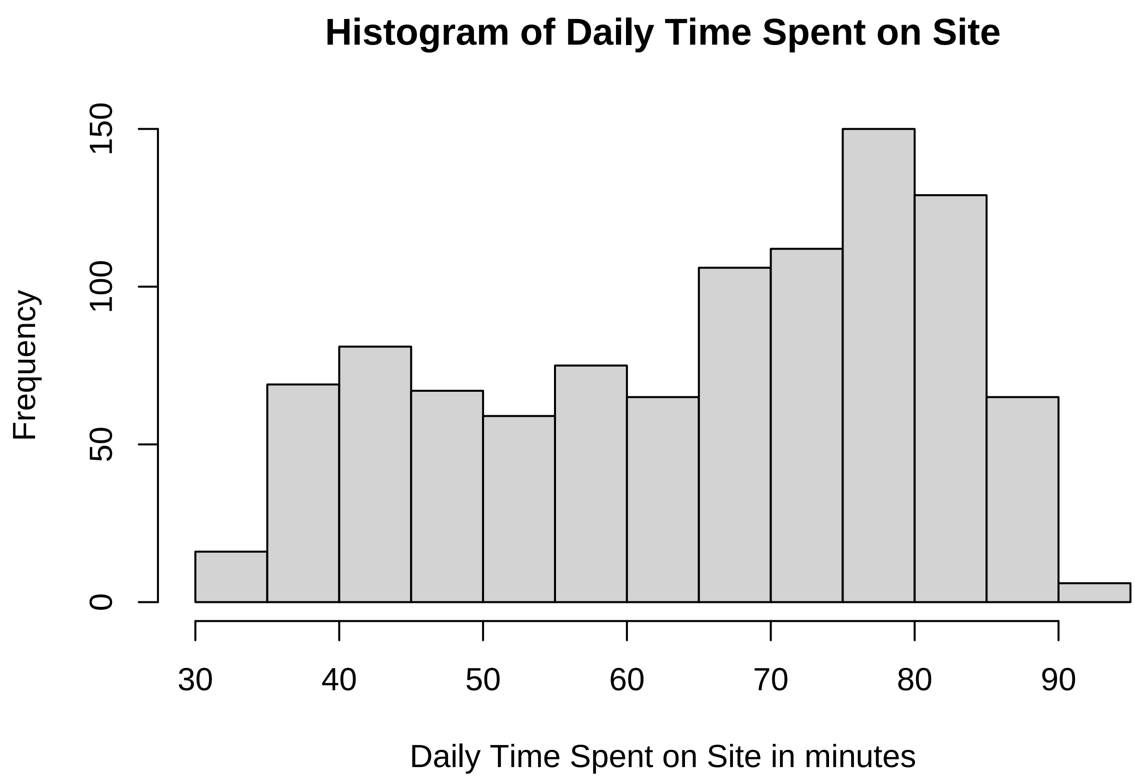

Many of the individuals in our data frame are between the ages of 25 and 40. This shows that majority of the audience of the blog is between this age bracket with very few being below 25 years and above 50 years old.

```
#Plotting a histogram for Daily Internet Usage
hist(advertising_dataset$Daily.Internet.Usage,
     main = "Histogram of Daily Internet Usage",
     xlab = "Daily Internet Usage")
```



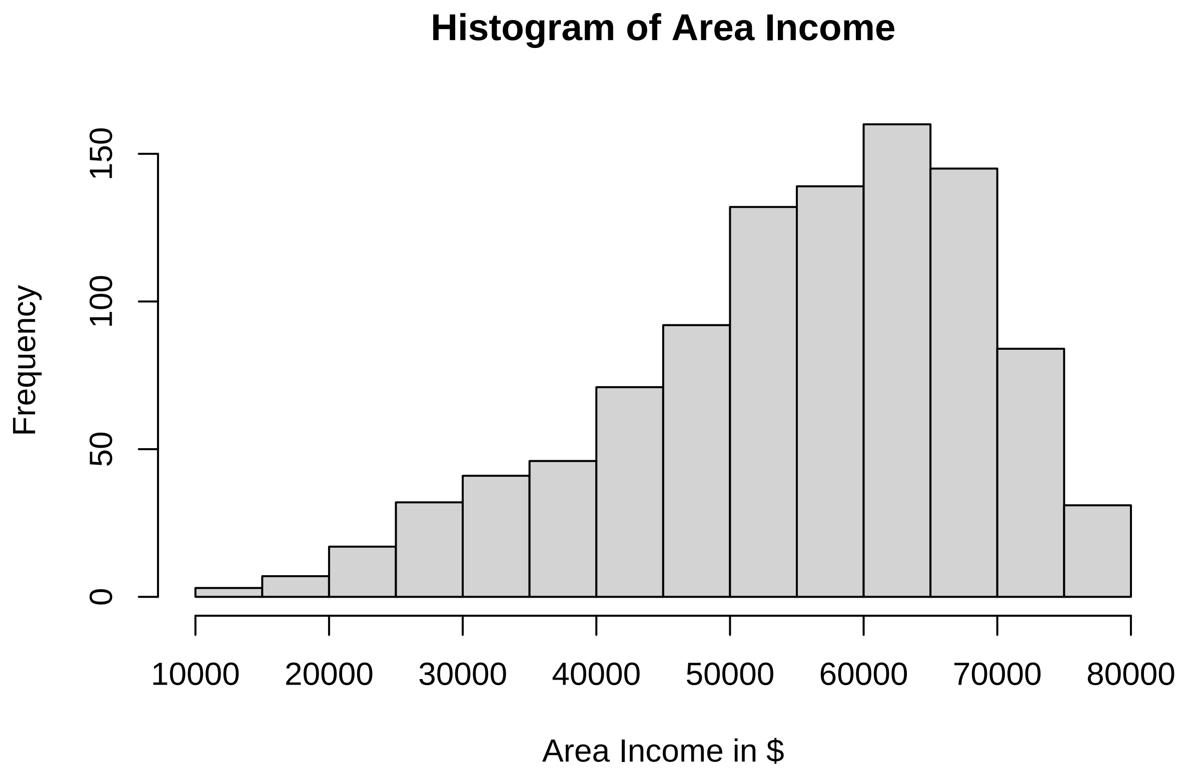

Majority of the audience's Daily Internet usage is between 100-140 and 200-240.

```
#Plotting a histogram for Daily Time Spent on Site
hist(advertising_dataset$Daily.Time.Spent.on.Site,
     main = "Histogram of Daily Time Spent on Site",
     xlab = "Daily Time Spent on Site in minutes")
```




Majority of the audience spend 75-80 minutes daily on the site with very few people spending less than 40 minutes and more than 85 minutes on the site.

```
#Plotting a histogram for Area Income
hist(advertising_dataset$Area.Income,
     main = "Histogram of Area Income",
     xlab = "Area Income in $")
```




The area income for majority of the audience in our data frame have a relatively high area income of 60,500 with very few having a lower income of between 10,000 and 40,000.

#### b.) Bivariate Analysis

```r
daily_time <- advertising_dataset$Daily.Time.Spent.on.Site
age <- advertising_dataset$Age
area <- advertising_dataset$Area.Income
daily_usage <- advertising_dataset$Daily.Internet.Usage
cov(daily_time, age)
```

#### i.) Covariance

```
## [1] -46.17415
```

```r
cov(daily_time, area)
```

```
## [1] 66130.81
```

```r
cov(daily_time, daily_usage)
```

```
## [1] 360.9919
```

```r
cov(age, area)
```

```
## [1] -21520.93
```

```r
cov(age, daily_usage)
```

```
## [1] -141.6348
```

```r
cov(area, daily_usage)
```

```
## [1] 198762.5
```

From the above, we can tell that Daily time spent on site and Age, Age and Area Income, Age and Daily Internet Usage have a negative linear relationship with each other.

On the other hand, Daily time spent on site and area, Daily time spent on site and Daily Internet Usage and Area Income and Daily Internet Usage have a positive linear relationship with each other.

**ii.) Correlation** The correlation of each numeric variable will help in understanding the association of these random variables.

```r
correlation <- round(cor(select_if(advertising_dataset, is.numeric)), 2)
head(correlation)
```

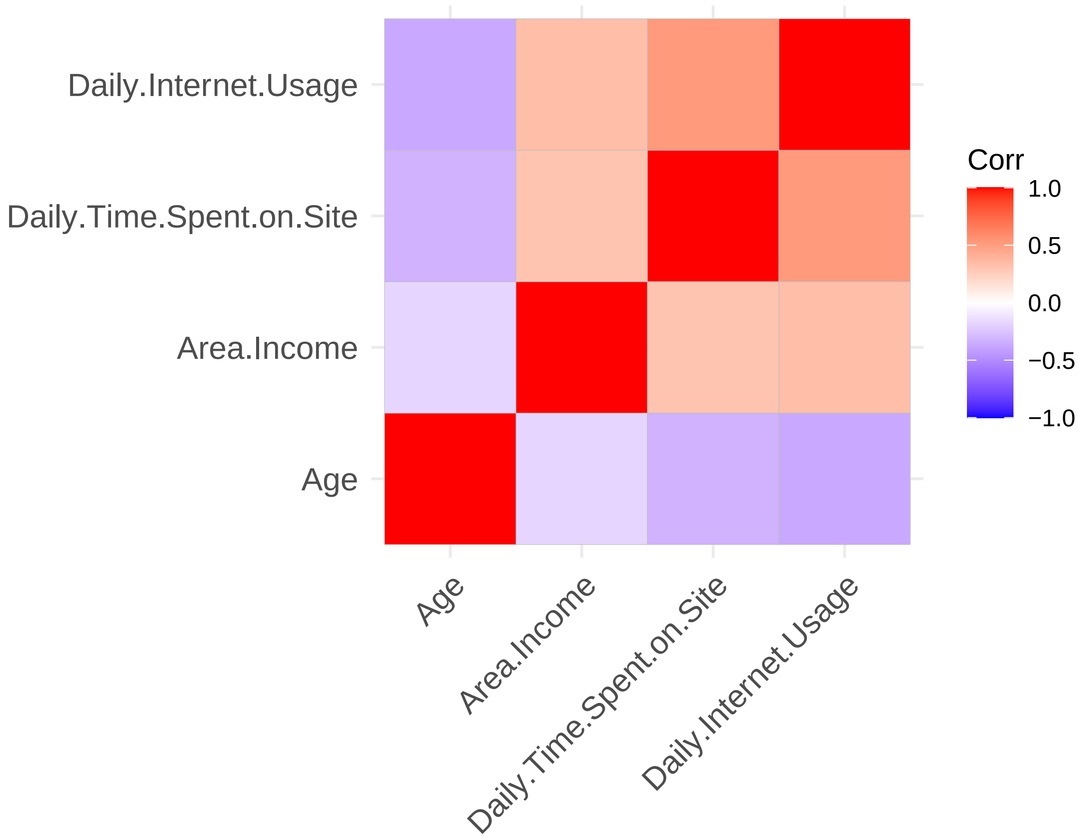
```
##                          Daily.Time.Spent.on.Site   Age Area.Income
## Daily.Time.Spent.on.Site                     1.00 -0.33        0.31
## Age                                         -0.33  1.00       -0.18
## Area.Income                                  0.31 -0.18        1.00
## Daily.Internet.Usage                         0.52 -0.37        0.34
##                          Daily.Internet.Usage
## Daily.Time.Spent.on.Site                 0.52
## Age                                     -0.37
## Area.Income                              0.34
## Daily.Internet.Usage                     1.00
```

The above output gives us the correlation of each variable. We can deduce that:

- The daily time spent on site is positively linearly related to Area Income and Daily Internet Usage and negatively correlated to age.
- Age is negatively linearly correlated to all other variables.
- Area Income is positively correlated to Daily Time spent on site, as mentioned earlier, and Daily Internet Usage.

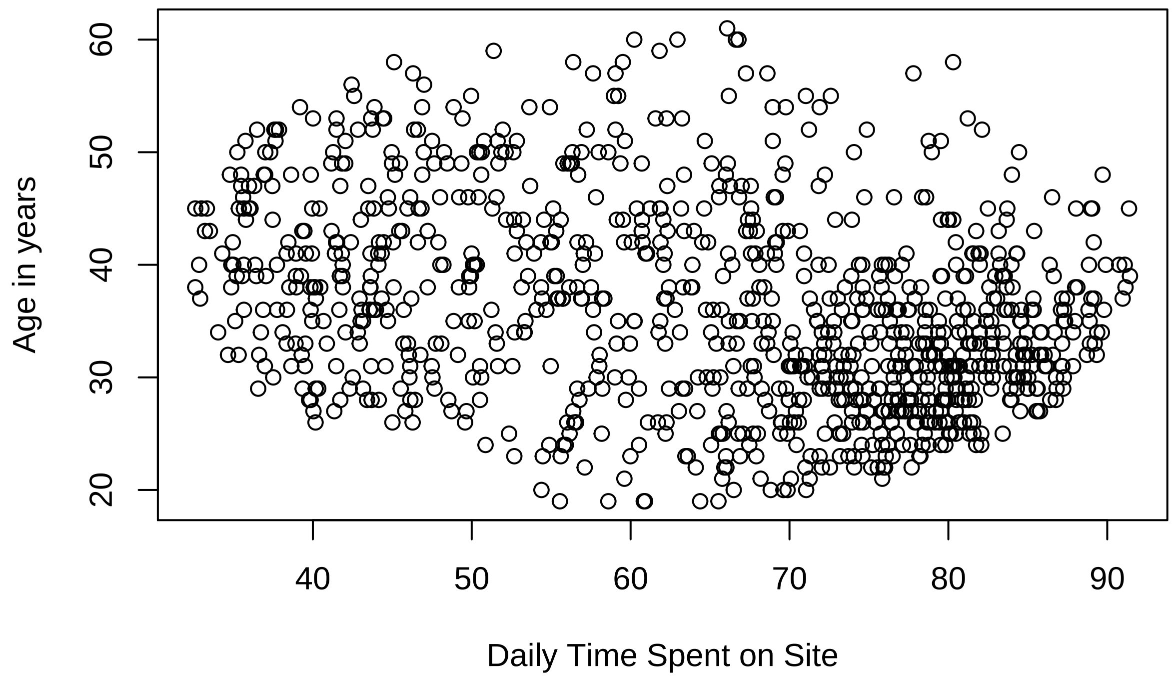
We'll plot a heatmap to visualize the correlation matrix of our numeric variables.

```r
corr = round(cor(select_if(advertising_dataset, is.numeric)), 2)
ggcorrplot(corr, hc.order = T)
```



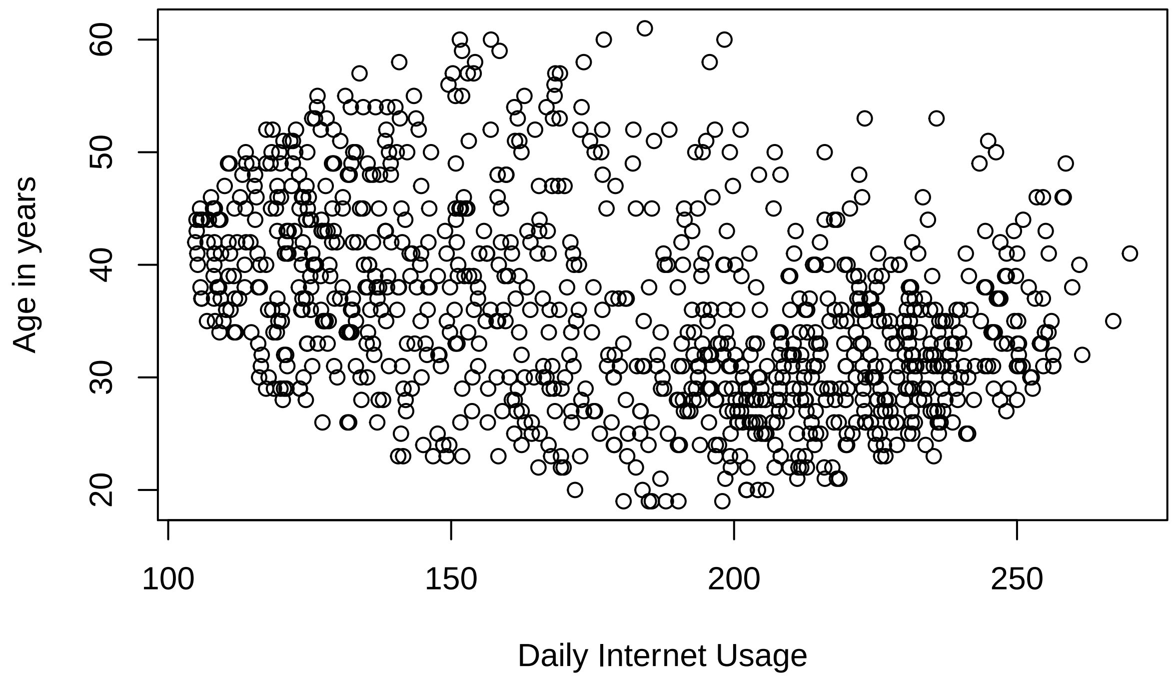

**iii.) Scatter Plot** We'll plot scatter plots of the numerical variables to understand how they are related.

```
# Plotting a scatter plot of Daily time spent on site and age
plot(daily_time, age, xlab="Daily Time Spent on Site", ylab="Age in years")
```



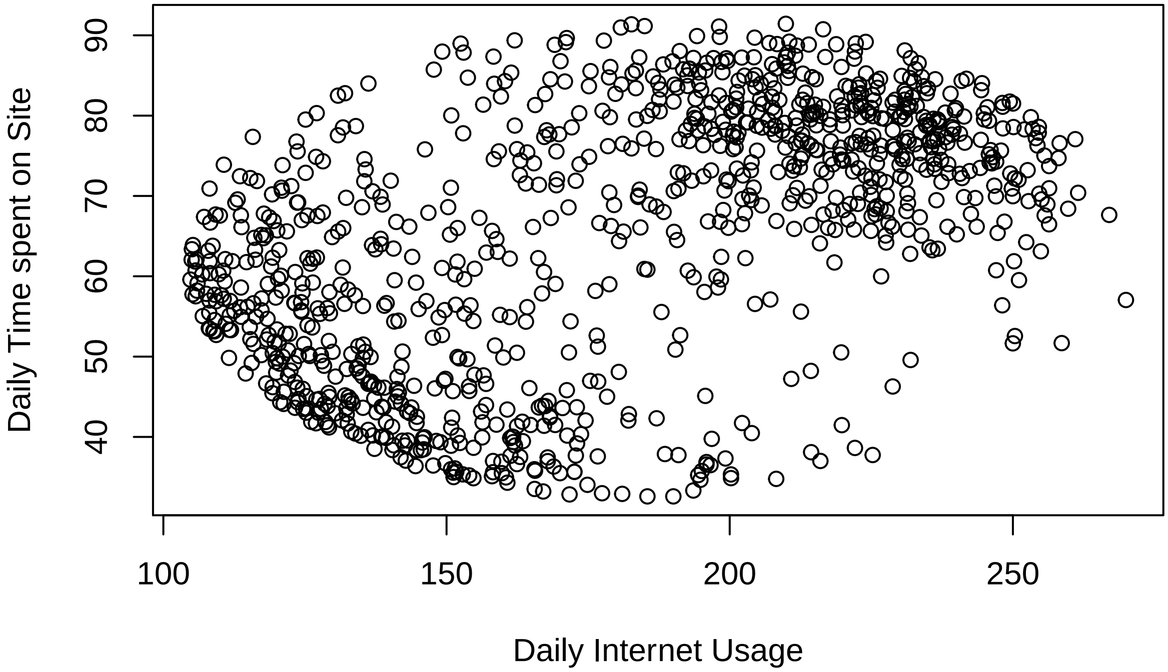

Younger people tend to spend more time on the site than older people as shown.

```
# Plotting a scatter plot of Daily Internet Usage and age
plot(daily_usage, age, xlab="Daily Internet Usage", ylab="Age in years")
```




Younger people seem to have a relatively higher Daily Internet Usage than Older people.

```
# Plotting a scatter plot of Daily Internet usage and Daily time spent on site
plot(daily_usage, daily_time, xlab="Daily Internet Usage", ylab="Daily Time spent on Site")
```

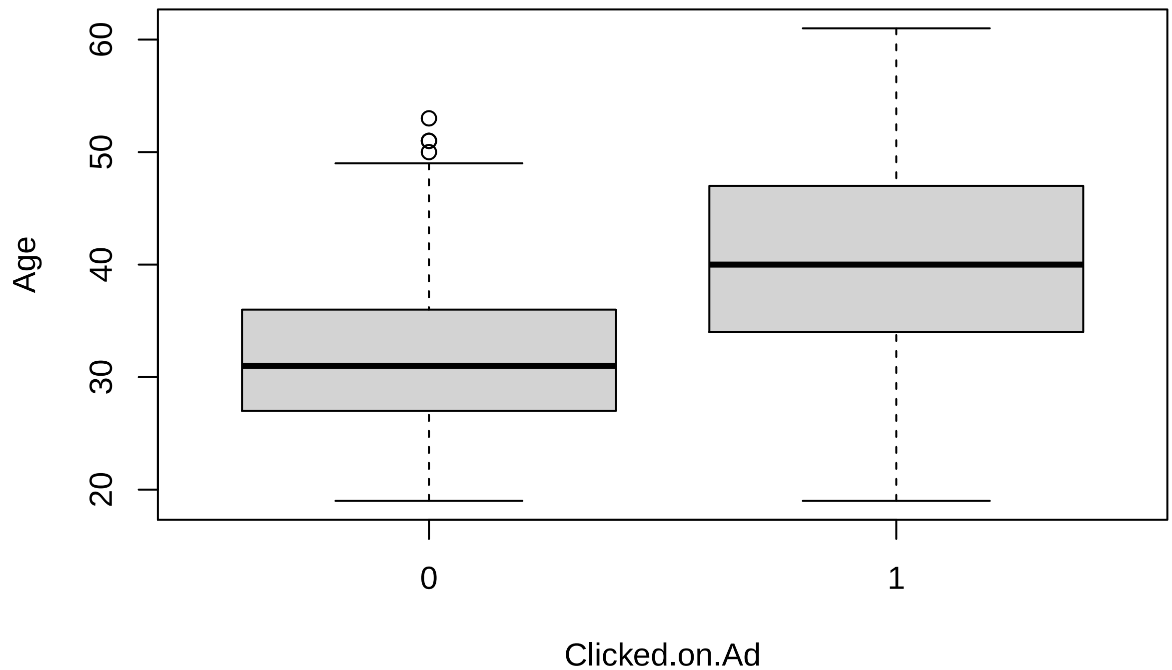



Majority of the individuals with a high Daily internet usage spend more time on the site while those with a low daily internet usage spend less time on the site.

### iv.) Boxplots

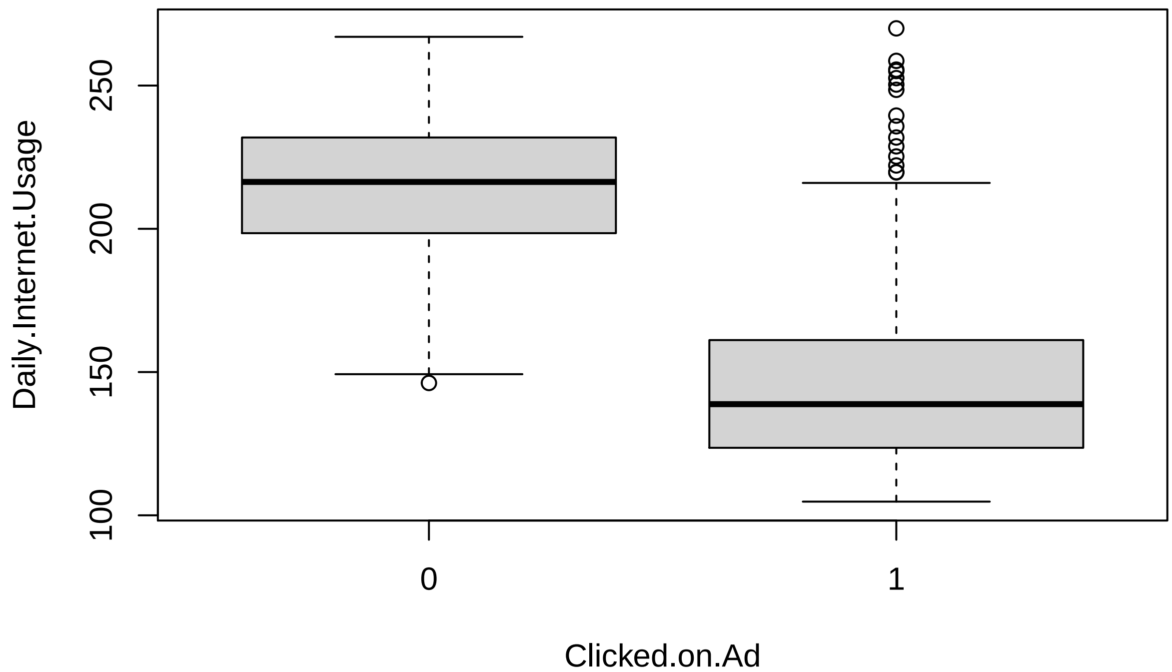
We'll plot boxplots to visualize Numerical and Factor data type.

```r
#Plotting boxplots for age and clicked on ad
plot(Age ~ Clicked.on.Ad , data = advertising_dataset)
```



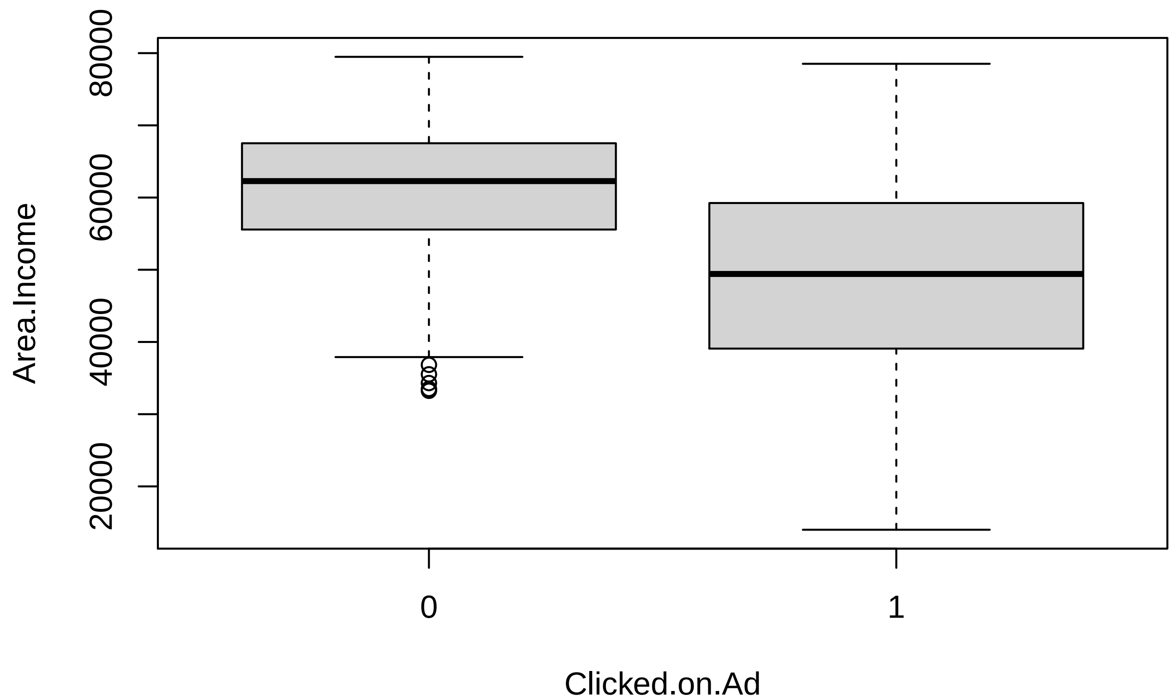

Individuals aged between 27 and 36 are the majority of the audience who did not click on the Ad while those who clicked on the ad were aged above 36. Therefore older people click on ads more than younger people do.

```
#Plotting boxplots for Daily Internet Usage and clicked on ad
plot(Daily.Internet.Usage ~ Clicked.on.Ad , data = advertising_dataset)
```



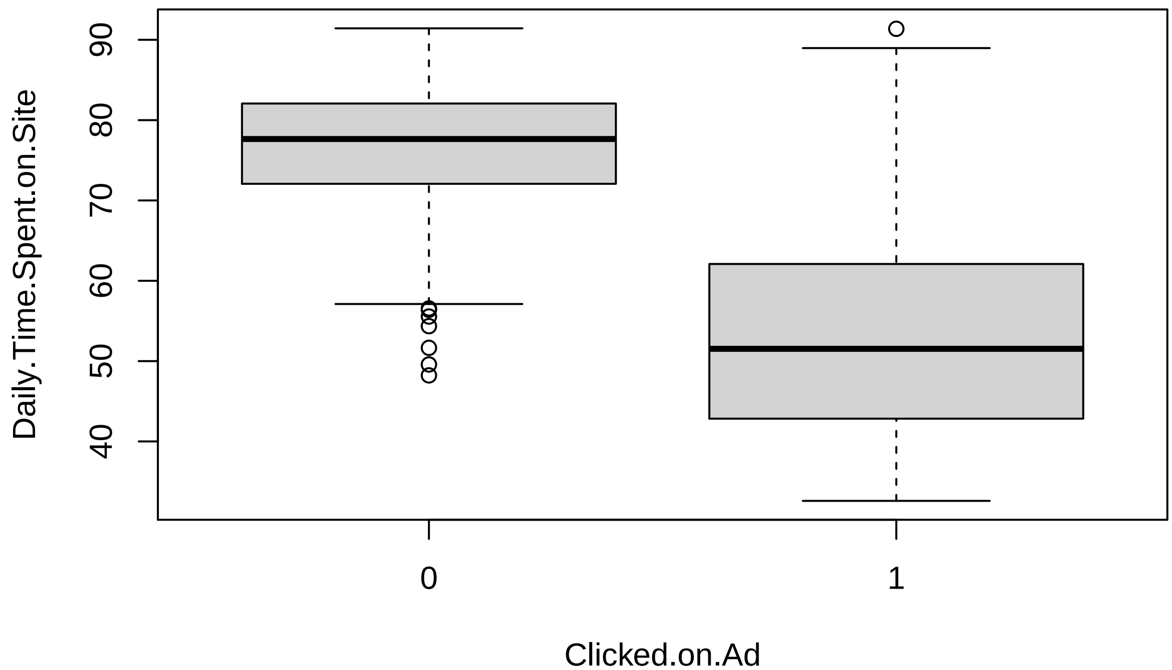

Individuals with a lower Daily Internet Usage clicked on the ad more than those with a higher internet usage.

```
#Plotting boxplots for Area Income and clicked on ad
plot(Area.Income ~ Clicked.on.Ad , data = advertising_dataset)
```



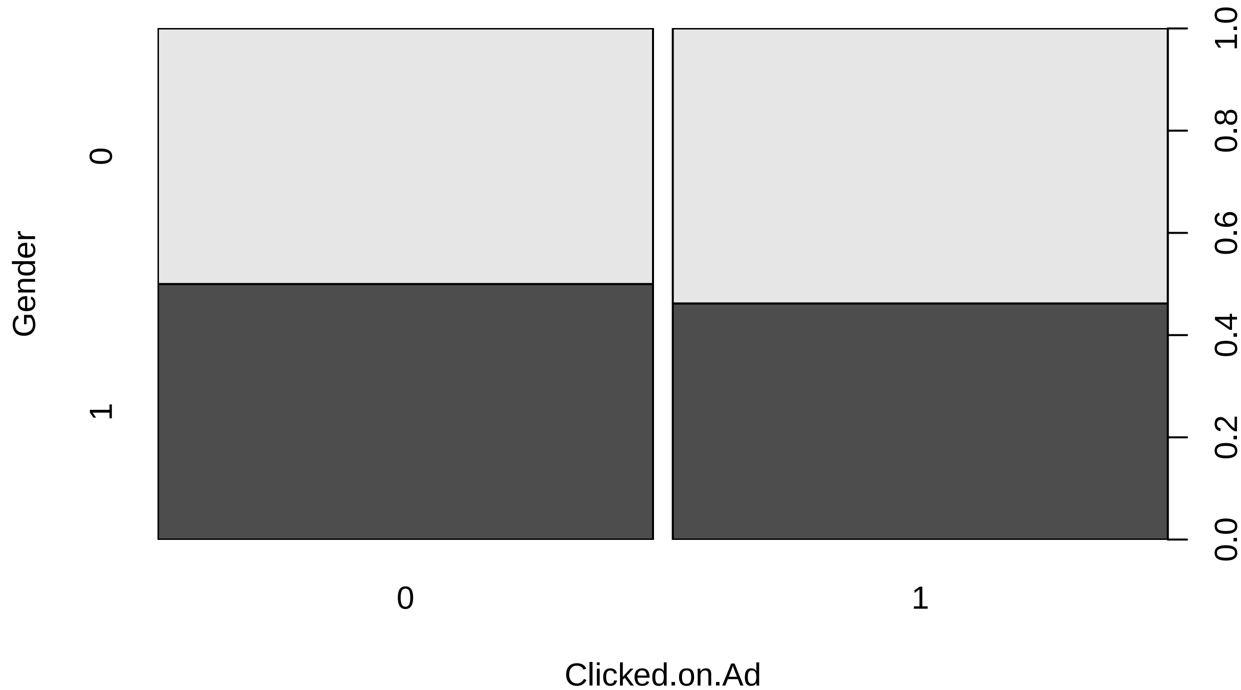

Individuals with a lower Area Income clicked on the Ad more than those who have a higher Area Income.

```r
#Plotting boxplots for Daily Time Spent on Site and clicked on ad
plot(Daily.Time.Spent.on.Site ~ Clicked.on.Ad , data = advertising_dataset)
```




Individuals who spend less time on the site click on the ad more than those who spent more time on the site.

```
#Plotting boxplots for Gender and clicked on ad
plot(Gender ~ Clicked.on.Ad , data = advertising_dataset)
```




Females clicked on the ads more than the male gender did.

## 9. Implementing the Solution

We'll now implement our solution by building supervised learning models to help identify which individuals are most likely to click on the ads in the blog.

Here we'll build models using the K-Nearest Neighbours, Decision Trees, Support Vector Machine and Naive Bayes algorithms.

### a.) K-Nearest Neighbours

We'll first normalize our data

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

```r
#Selecting the columns we'll use for modelling.
cols = c('Daily.Time.Spent.on.Site', 'Age', 'Area.Income', 'Daily.Internet.Usage', 'Gender', 'Clicked.o
ad = select(advertising_dataset, 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.
```

```r
head(ad)
```

```
##   Daily.Time.Spent.on.Site       Age Area.Income Daily.Internet.Usage Gender
## 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 Year Month Day
## 1             0 2016    03  27
## 2             0 2016    04  04
## 3             0 2016    03  13
## 4             0 2016    01  10
## 5             0 2016    06  03
## 6             0 2016    05  19
```

```r
# 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]
```

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

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

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

```
##        model
```

```
## test_sp  0  1
##       0 94  5
##       1  5 96
```

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

```
## [1] 95
```

We get an accurcay of 95% which is a really good score. The confusion matrix also shows that only 10 data points were misclassified implying that our model is a good model.

**Challenging our model.** We'll challenge our KNN model by using another value of k. We'll use k of 3 to see if our accuracy scores improve.

```r
#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] 96
```

Our accuracy score improves and we end up with a better accuarcy of 96%.

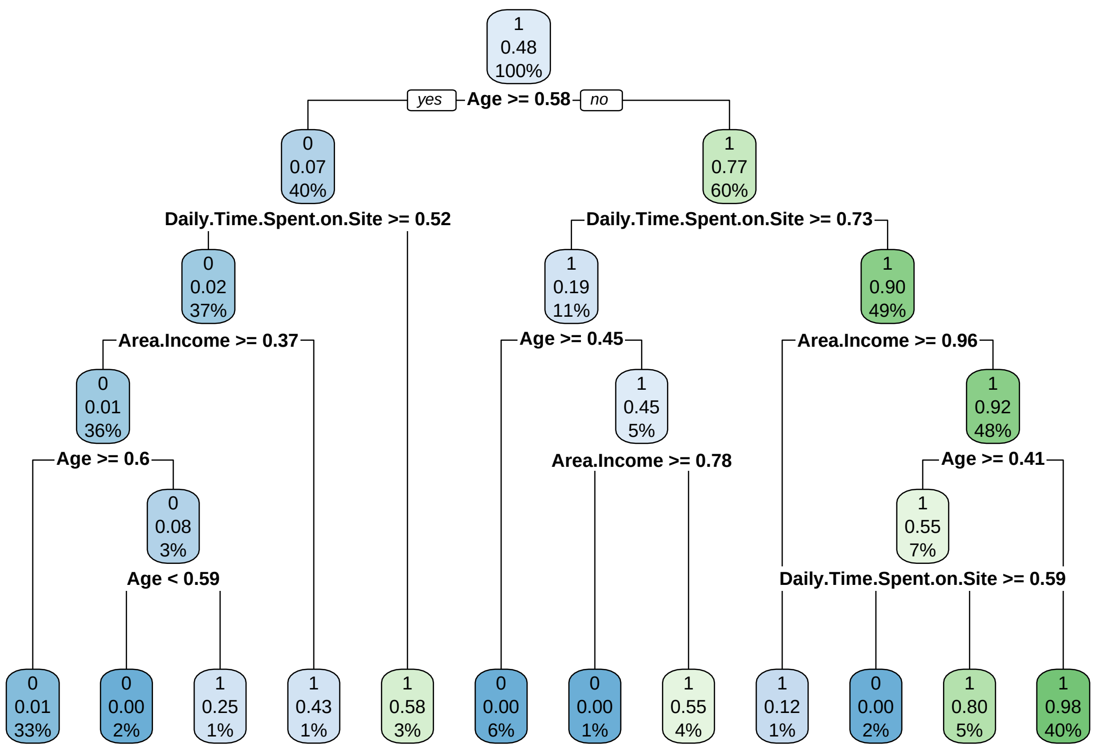
### b.) Decision Trees

```r
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

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

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

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

```r
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)
```




```r
#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
```

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

```
## [1] 92.1
```

The decision trees model gives us an accuracy of 91.7%. This is a good model. The confusion matrix also shows that there were only a few misclassifications.

**Challenging our model.** We'll challenge our Decision Tree by building a Random Forest model.

```r
#Loading  the randomForest package.
require(randomForest)
```

```
## Loading required package: randomForest
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

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

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
set.seed(101)
#Extracting the training set
ad_train <- ad[1:800,]
#Building the classification tree using the random forest library
tree <- randomForest(Clicked.on.Ad~.,data=ad_train, parms = list(loss = penalty.matrix), method = "clas
tree
```

```
##
## Call:
##  randomForest(formula = Clicked.on.Ad ~ ., data = ad_train, parms = list(loss = penalty.matrix),
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 4.5%
## Confusion matrix:
##     0   1 class.error
## 0 396  16  0.03883495
## 1  20 368  0.05154639
```

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

```
##
## p     0   1
##   0 497   7
##   1   3 493
```

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

```
## [1] 99
```

Our Random Forest model gives us an accuracy score of 99%. This is such a good model, however this model is very prone to overfitting as it has a very high score hence it is highly unrecommended.

### c.) Support Vector Machine

```
# 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]
```

```
set.seed(100)
# Train the model using support vector machine
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
svmLinear = train(Clicked.on.Ad ~ ., data=ad_train, method = "svmLinear",
trControl=trctrl,
preProcess = c("center", "scale"),
tuneLength = 10)
#Checking the result of our train() model as shown below
svmLinear
```

```
## Support Vector Machines with Linear Kernel
##
## 800 samples
##   5 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (5), scaled (5)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 720, 720, 719, 720, 721, 721, ...
```

```
## Resampling results:
##
##   Accuracy   Kappa
##   0.9550028  0.9098059
##
## Tuning parameter 'C' was held constant at a value of 1
```

```r
#Predicting using the predict() method
test_pred <- predict(svmLinear, newdata = ad_test)
test_pred
```

```
##   [1] 1 1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 1 1 1 1 1 1 1 0 0 1
##  [38] 1 1 1 1 1 0 0 0 1 1 0 0 1 0 1 1 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0
##  [75] 0 1 1 0 0 0 1 0 0 1 0 1 1 1 0 1 0 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 1 0 1 0 1
## [112] 1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 0 0 0 1 0 1 1 1 0 0 1 1 1 1 1 0 1 1 1 0 0 1
## [149] 0 0 1 1 1 1 0 1 1 0 0 0 1 0 0 0 0 1 1 0 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 0 0
## [186] 1 0 1 0 0 1 1 1 0 1 0 1 1 1 1
## Levels: 0 1
```

```r
#Evaluating our model using a confusion matrix and accuarcy score
confusionMatrix(table(test_pred, ad_test$Clicked.on.Ad))
```

```
## Confusion Matrix and Statistics
##
##
## test_pred   0   1
##         0  86   7
##         1   2 105
##
##                Accuracy : 0.955
##                  95% CI : (0.9163, 0.9792)
##     No Information Rate : 0.56
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9092
##
##  Mcnemar's Test P-Value : 0.1824
##
##             Sensitivity : 0.9773
##             Specificity : 0.9375
##          Pos Pred Value : 0.9247
##          Neg Pred Value : 0.9813
##              Prevalence : 0.4400
##          Detection Rate : 0.4300
##    Detection Prevalence : 0.4650
##       Balanced Accuracy : 0.9574
##
##        'Positive' Class : 0
##
```

Our SVM model gives us an accuarcy of 95.5%. This accuarcay score implies that this a very good model and it's confuzion matrix proves the same as it classified only 9 incorrectly.

### d.) Naive Bayes

```
#Loading the required libraries
library(tidyverse)
library(caret)
library(caretEnsemble)
```

```
##
## Attaching package: 'caretEnsemble'

## The following object is masked from 'package:ggplot2':
##
##     autoplot
```

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:randomForest':
##
##     outlier

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(rpart)
library(randomForest)
#Ensuring the target variable is a factor i.e. categorical variable
ad$Clicked.on.Ad <- factor(ad$Clicked.on.Ad, levels = c(0,1), labels = c("False", "True"))
```

```
# Splitting data into training and test data sets
library(caret)
indxTrain <- createDataPartition(ad$Clicked.on.Ad, p=0.7)$Resample1
training <- ad[indxTrain,]
testing <- ad[-indxTrain,]
x = training[,-9]
y = training$Clicked.on.Ad
```

```
#Loading the inbuilt e1071 package that holds the Naive Bayes function
library(e1071)
```

```
##
## Attaching package: 'e1071'

## The following objects are masked from 'package:moments':
##
##     kurtosis, moment, skewness
```

```r
#Now building our model
NBclassfier=naiveBayes(Clicked.on.Ad~., data=training)
print(NBclassfier)
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
## False  True
##   0.5   0.5
##
## Conditional probabilities:
##        Daily.Time.Spent.on.Site
## Y             [,1]      [,2]
##   False 0.7483160 0.1281597
##   True  0.3458012 0.2195476
##
##        Age
## Y             [,1]      [,2]
##   False 0.6659705 0.1469108
##   True  0.2453974 0.1727148
##
##        Area.Income
## Y             [,1]      [,2]
##   False 0.7252849 0.1412367
##   True  0.5253113 0.2156702
##
##        Daily.Internet.Usage
## Y             [,1]      [,2]
##   False 0.6659705 0.1469108
##   True  0.2453974 0.1727148
##
##        Gender
## Y               0         1
##   False 0.5028571 0.4971429
##   True  0.5200000 0.4800000
```

```r
#Predicting
pre <- predict(NBclassfier, testing, type = "raw") %>%
  as.data.frame() %>%
  mutate(prediction = if_else(0 < 1, 0, 1)) %>%
  pull(prediction)
```

```r
a <- table(pre, testing$Clicked.on.Ad)
a
```

```
##
## pre False True
##   0   150  150
```

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

```
## [1] 50
```

The accuracy score of our Naive Bayes model is very poor i.e. 50% This model is highly unrecommended for the problem at hand.

## 11. Conclusions and Recommendations

From this analysis, we can conclude the following:

Majority of the audience of the blog are:

- Aged between 25 and 40 years
- Females
- Spend 75-80 minutes on the blog
- Have a high area income
- Have a moderate Daily internet usage.

Majority of the individuals who clicked on the ad have the following attributes:

- They are older i.e. ages above 36
- They have a low daily internet usage
- They have a lower area income
- They spend less time on the site.
- They are female.

Younger people(below 36), people with a high daily internet usage, high area income and spend more time on the site are least likely to click on the ad.

On modelling, the best performing model is the Random Forest. However this model is not recommended as it is very prone to overfitting. The most recommended model is the KNN with an accuracy score of 96%. The SVM and the Decision Tress also do well. The Naive Bayes modle is the least recommended model as it performs poorly with an accuarcy score of 50%, this model is highly unrecmmended.

Based on this analysis I would recommend to the Kenyan entrepreneur to create ads that are more accommodating i.e. the ads should be able to influence all types of gender, both young and old people. I would also recommedn the use of the KNN model when predicting the individuals who would click on her ads.