R-Supervised Learning Modelling

Vivian Bwana

2022-06-06

USING ADVERTISEMENTS FOR MARKET SEGMENTATION

Defining the Question

a) Specifying the question

To identify which individuals are most likely to click on her ads.

b) Defining the Metric for success

To be able to identify who is likely to click on the ads

c) Understanding 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 help her identify which individuals are most likely to click on her ads.

d) Defining Experimental Design

1. Loading dataset into R
2. External dataset verification
3. Data understanding using Exploratory Data Analysis
4. Preparing the dataset by performing various dataset cleaning procedures
5. Perform Univariate, Bivariate and Multivariate Analysis
6. Implementing the solution
7. Challenging the solution
8. Conclusion
9. Recommendations
10. Follow up questions

e) Data Relevance

The dataset is relevant as it successfully answered our objective, we were able to identify the relevant ad target groups.

Data Understanding

Importing libraries

```
# Importing the relevant libraries to be used in this analyis
library(data.table)
library(ggplot2)
library(plyr); library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#library(ggcorrplot)
#library(moments)
```

Loading the dataset

```
# Loading our dataset
advert <- fread("http://bit.ly/IPAdvertisingData")
```

```
# checking class
class(advert)
```

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

Previewing the dataset

The first six items

```
#previewing the first 6 rows of the dataset
head(advert)
```

```
##    Daily Time Spent on Site   Age Area Income Daily Internet Usage
##                       <num> <int>       <num>                <num>
## 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
##                                   <char>         <char> <int>     <char>
## 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
##                 <POSc>         <int>
## 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 last six items

```
#previewing the last 6 rows of the dataset
tail(advert)
```

```
##    Daily Time Spent on Site   Age Area Income Daily Internet Usage
##                       <num> <int>       <num>                <num>
## 1:                    43.70    28    63126.96               173.01
## 2:                    72.97    30    71384.57               208.58
## 3:                    51.30    45    67782.17               134.42
## 4:                    51.63    51    42415.72               120.37
## 5:                    55.55    19    41920.79               187.95
## 6:                    45.01    26    29875.80               178.35
##                           Ad Topic Line          City  Male
##                                  <char>        <char> <int>
## 1:        Front-line bifurcated ability  Nicholasland     0
## 2:        Fundamental modular algorithm     Duffystad     1
## 3:      Grass-roots cohesive monitoring   New Darlene     1
## 4:         Expanded intangible solution South Jessica     1
## 5: Proactive bandwidth-monitored policy   West Steven     0
## 6:      Virtual 5thgeneration emulation   Ronniemouth     0
##                    Country           Timestamp Clicked on Ad
##                     <char>              <POSc>         <int>
## 1:                Mayotte 2016-04-04 03:57:48             1
## 2:                Lebanon 2016-02-11 21:49:00             1
```

```
## 3: Bosnia and Herzegovina 2016-04-22 02:07:01            1
## 4:               Mongolia 2016-02-01 17:24:57            1
## 5:              Guatemala 2016-03-24 02:35:54            0
## 6:                 Brazil 2016-06-03 21:43:21            1
```

Exploratory Data Analysis

Exploring the Dataset

Dimensions

```
#Checking the shape of the dataset
dim(advert)
```

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

The dataset has 1000 rows and 10 columns

Data Types

```
#Checking the datatypes of the dataset
str(advert)
```

```
## Classes 'data.table' and '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               : POSIXct, format: "2016-03-27 00:53:11" "2016-04-04 01:39:02" ...
##  $ Clicked on Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

All the data types are correct.

Data Cleaning

Editting columns names

```
# We editthe coulmn names so that they appear as one word
#this is necessaruy so as to avoid machine language readability errors
#We do this by adding dots in between each word of the column names

colnames(advert) <- c('Daily.Time.Spent.on.Site','Age','Area.Income','Daily.Internet.Usage','Ad.Topic.Li

# printing new data frame
print("New data frame : ")
```

```
## [1] "New data frame : "
```

```
print(advert)
```

```
##       Daily.Time.Spent.on.Site   Age Area.Income Daily.Internet.Usage
##                          <num> <int>       <num>                <num>
##    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
##   ---
##  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
##                                      <char>         <char> <int>
##    1:    Cloned 5thgeneration orchestration    Wrightburgh     0
##    2:    Monitored national standardization      West Jodi     1
##    3:      Organic bottom-line service-desk       Davidton     0
##    4: Triple-buffered reciprocal time-frame West Terrifurt     1
##    5:         Robust logistical utilization   South Manuel     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
##                       <char>              <POSc>         <int>
##    1:                Tunisia 2016-03-27 00:53:11             0
##    2:                 Nauru 2016-04-04 01:39:02             0
##    3:             San Marino 2016-03-13 20:35:42             0
##    4:                  Italy 2016-01-10 02:31:19             0
##    5:                Iceland 2016-06-03 03:36:18             0
##   ---
##  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 column names

```
# previewing column names
colnames(advert)
```

```
##  [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"
```

Missing Values

```
#Checking for the sum of Missing values
colSums(is.na(advert))
```

```
## 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 this dataset.

Duplicates

```
#Checking for duplicates in the dataset
advert.duplicates <- advert[duplicated(advert),]

#printing duplicated rows
advert.duplicates
```

```
## Empty data.table (0 rows and 10 cols): Daily.Time.Spent.on.Site,Age,Area.Income,Daily.Internet.Usage
```

There are no duplicated rows in the dataset

Outliers

```
#Exctracting numeric columns to analyse for outliers
num.cols <- unlist(lapply(advert, is.numeric))

#printing numeric columns
num.cols
```

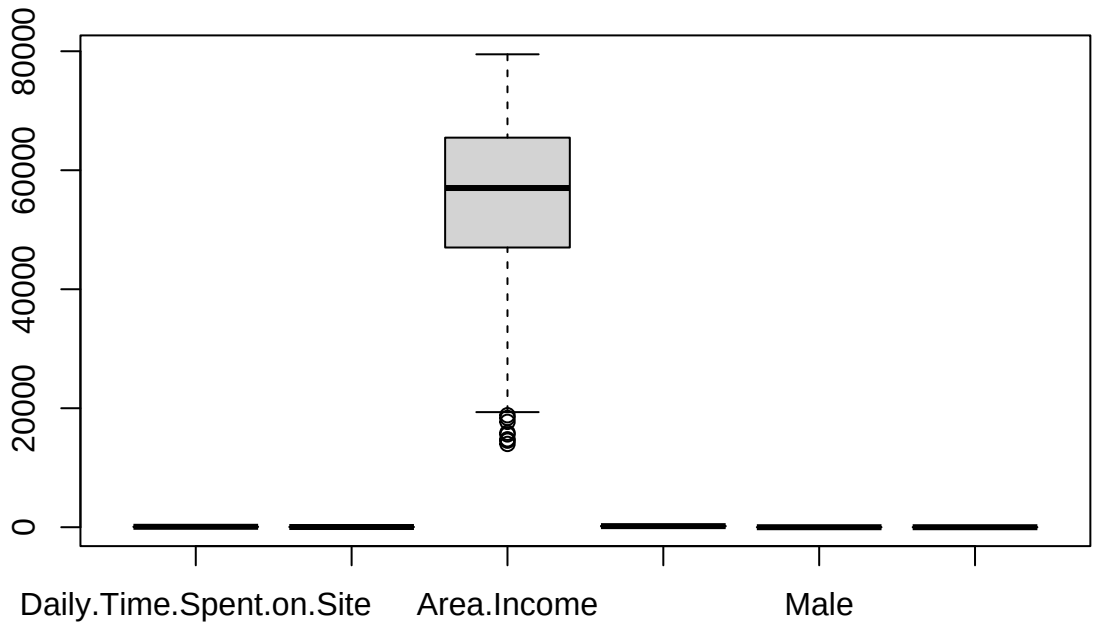
```
## Daily.Time.Spent.on.Site                      Age             Area.Income
##                     TRUE                     TRUE                    TRUE
##     Daily.Internet.Usage           Ad.Topic.Line                     City
##                     TRUE                    FALSE             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

```
#creating a dataframe with numeric columns only so as to plot a boxplot
advert.numeric <-advert[, ..num.cols]

#checking the data types, previewing
str(advert.numeric)
```

```
## Classes 'data.table' and 'data.frame':   1000 obs. of  6 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 ...
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

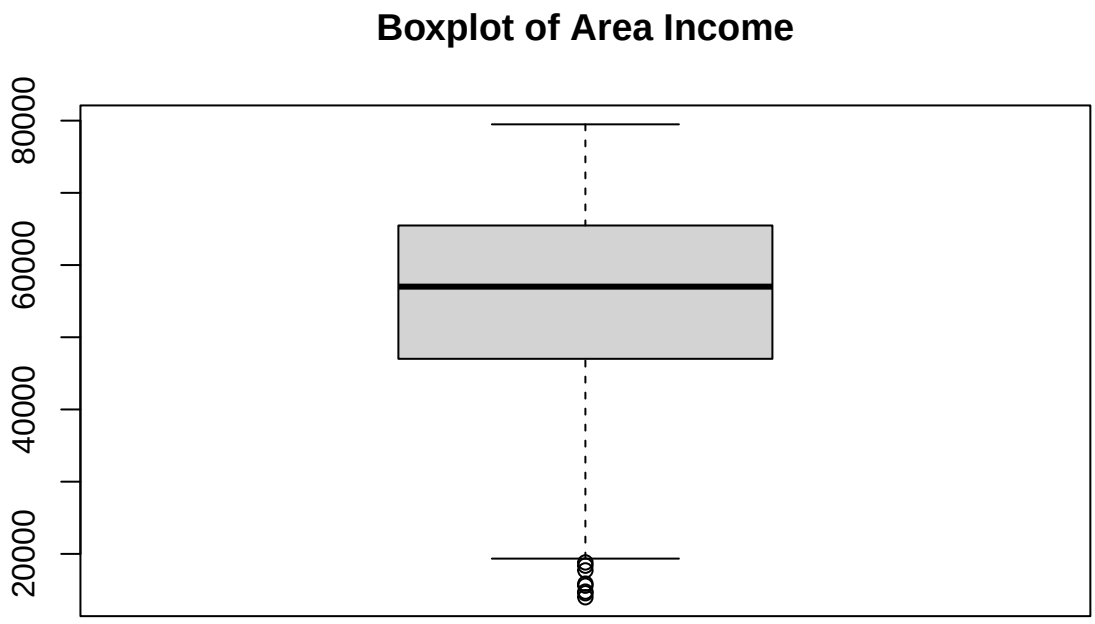
```
#Plotting a boxplot to check for outliers
boxplot(advert.numeric)
```



The only column with outliers is the in Area.income column

Removing the outliers in the Area Income column

```
#First, we plot a boxplot to show outliers in  Area.Income column
boxplot(advert$Area.Income, main = 'Boxplot of Area Income')$out
```




```
## [1] 17709.98 18819.34 15598.29 15879.10 14548.06 13996.50 14775.50 18368.57
```

```
#there are some records appearing as outliers in the lower quartile of the  Area.Income column
#These will be removed before we begin analysis
```

The outliers are seen in the lower quantile, this is the area from whic we will remove outliers.

Identifying the lower and upper quantiles

```
#Removing outliers in the lower quartile of the Area.Income

# defining the lower quantile
Q1 <- quantile(advert$Area.Income, .25)
# defining the upper quantile
Q3 <- quantile(advert$Area.Income, .75)
# calculating the IQR
IQR <- IQR(advert$Area.Income)

#Removing outliers while keeping values above 1.5*IQR of the Q1
# defining a new dataframe without outliers
no.outliers <- subset(advert, advert$Area.Income > (Q1 - 1.5*IQR)) #& advert$Area.Income < (Q3 + 1.5*IQ
dim(no.outliers)
```

```
## [1] 991  10
```

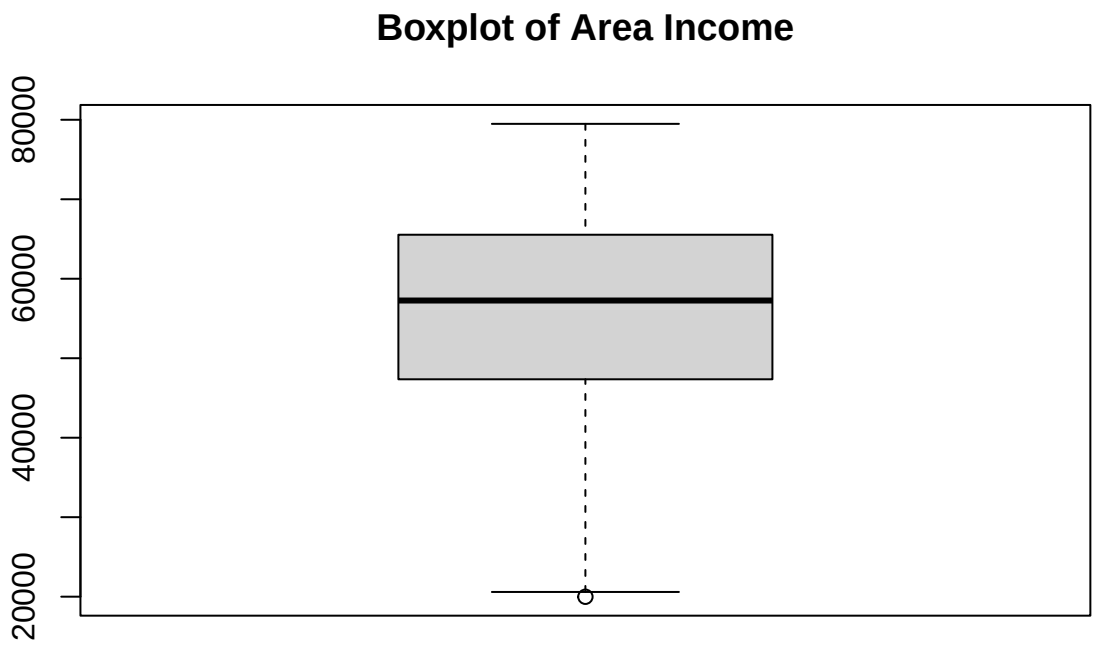
```
dim(advert)
```

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

9 rows were dropped

Plotting a boxplot for this column to confirm these changes

```
#Plotting a boxplot to check if outliers in Area.Income column have been dropped
boxplot(no.outliers$Area.Income, main = 'Boxplot of Area Income')
```



The boxplot indicates that the outliers in the lower quantile have been removed

Univariate Analysis

```
#previewing the new dataset without outliers
head(no.outliers)
```

```
##    Daily.Time.Spent.on.Site   Age Area.Income Daily.Internet.Usage
##                       <num> <int>       <num>                <num>
## 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
##                                  <char>         <char> <int>     <char>
## 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
##                 <POSc>         <int>
## 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
```

Summary The summary gives us the minimum, maximum, median, mean and quantile ranges for all the numerical and categorical columns.

```
#checking summary statistics
summary(no.outliers)
```

```
##  Daily.Time.Spent.on.Site      Age         Area.Income    Daily.Internet.Usage
##  Min.   :32.60            Min.   :19.00   Min.   :19992   Min.   :104.8
##  1st Qu.:51.34            1st Qu.:29.00   1st Qu.:47348   1st Qu.:138.6
##  Median :68.41            Median :35.00   Median :57260   Median :183.4
##  Mean   :65.06            Mean   :35.99   Mean   :55349   Mean   :180.0
##  3rd Qu.:78.59            3rd Qu.:42.00   3rd Qu.:65538   3rd Qu.:218.9
##  Max.   :91.43            Max.   :61.00   Max.   :79485   Max.   :270.0
##  Ad.Topic.Line          City                Male          Country
##  Length:991         Length:991         Min.   :0.0000   Length:991
##  Class :character   Class :character   1st Qu.:0.0000   Class :character
##  Mode  :character   Mode  :character   Median :0.0000   Mode  :character
##                                        Mean   :0.4793
##                                        3rd Qu.:1.0000
##                                        Max.   :1.0000
##    Timestamp                       Clicked.on.Ad
##  Min.   :2016-01-01 02:52:10.00   Min.   :0.0000
##  1st Qu.:2016-02-17 22:51:14.50   1st Qu.:0.0000
##  Median :2016-04-07 03:56:16.00   Median :0.0000
##  Mean   :2016-04-10 02:20:21.53   Mean   :0.4955
##  3rd Qu.:2016-05-31 01:37:57.50   3rd Qu.:1.0000
##  Max.   :2016-07-24 00:22:16.00   Max.   :1.0000
```

Extracting numerical columns from the no.outliers dataset to use for analysis

```
#Extracting a numeric subset from the no outliers dataset
no.out.num.cols <-unlist(lapply(no.outliers, is.numeric))
#Exctracting numeric columns to analyse for outliers
```

```
#num.cols <- unlist(lapply(advert, is.numeric))

#printing numeric columns
no.out.num.cols
```

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

```
#creating a dataframe with numeric columns only so as to plot a boxplot
no.outliers.numeric <-no.outliers[, ..no.out.num.cols]

#previewing
head(no.outliers.numeric)
```

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

```
#checking the data types, previewing
#str(no.outliers.numeric)
```

Measures of Central Tendency

###i) Mean

```
#means of all numeric columns in the dataset
#this has been exctracted from the datset and named no.outliers.numeric
#the variable for the column means is no.out.col.means

no.out.col.means <- colMeans(data.frame(no.outliers.numeric))

# Printing out
```

```
# ---
#
no.out.col.means
```

```
## Daily.Time.Spent.on.Site                     Age             Area.Income
##             6.505689e+01            3.598587e+01            5.534910e+04
##     Daily.Internet.Usage                    Male            Clicked.on.Ad
##             1.799846e+02            4.793138e-01            4.954591e-01
```

The average daily time spent on site was 65.05 units. The average area income was 55,349 units. The average age of respondents was 35.98 years. The average daily internet usage was 179.98 units.

###ii) Median

```
#median of all numeric columns in the dataset
#this has been exctracted from the datset and named no.outliers.numeric
#the variable for the column means is no.out.col.median
library(matrixStats)
```

```
##
## Attaching package: 'matrixStats'

## The following object is masked from 'package:dplyr':
##
##     count

## The following object is masked from 'package:plyr':
##
##     count
```

```
no.out.col.median <- colMedians(as.matrix.data.frame(no.outliers.numeric))

# Printing out
# ---
#
print(no.out.col.median)
```

```
## [1]    68.41    35.00 57260.41   183.43     0.00     0.00
```

The median of the daily time spent on site was 68.41 units. The median of the area income was 57,260.41 units. The median of the ages of the respondents was 35 years. The median of the daily internet usage was 183.43 units.

###iii) Mode

```
# We create the mode function that will perform our mode operation for us
# The mode will give us values that appeared the most number of times
# ---
# library(purrr)
FindMode <- function(no.outliers) {
   uniqv <- unique(no.outliers)
   uniqv[which.max(tabulate(match(no.outliers, uniqv)))]
```

```
}

# Calculating the mode using out getmode() function
# ---
#
#no.out.col.mode <- getmode(as.matrix(no.outliers.numeric))
no.out.col.mode <- data.frame(no.outliers)

# Printing out
# ---
#
apply(no.out.col.mode,2, FindMode)
```

```
##             Daily.Time.Spent.on.Site                                  Age
##                              "62.26"                                "31"
##                          Area.Income                  Daily.Internet.Usage
##                           "61833.90"                               "167.22"
##                        Ad.Topic.Line                                 City
## "Cloned 5thgeneration orchestration"                          "Lisamouth"
##                                 Male                              Country
##                                  "0"                    "Czech Republic"
##                            Timestamp                       Clicked.on.Ad
##                "2016-03-27 00:53:11"                                 "0"
```

Below are the observations drawn from the above analysis:

Most people spent 62.26 minutes on the sites they visited There were more males compared to females/ other gender The most common ad line was "Cloned 5thgeneration orchestration" Most people were on the site on 2016-03-27 at 00:53:11, there could have been an event that led to most people visiting the site on this day and time Most people were aged 31yesr old Most people had daily internet usage of 167.22 units Most people were from Lismouth city and also from the country of Czech Republic Most people did not Click on the ads

Measures of Dispersion

We will use the numeric data-frame while calculating measures of dispersion

i)Minimum

```
# Finding the minimum values of the numerica columns
sapply(no.outliers.numeric, min)
```

```
## Daily.Time.Spent.on.Site                      Age              Area.Income
##                    32.60                    19.00                 19991.72
##     Daily.Internet.Usage                     Male             Clicked.on.Ad
##                   104.78                     0.00                     0.00
```

The minimum of the daily time spent on site was 32.60 units. The minimum of the area income was 19,991.72 units. The minimum of the ages of the respondents was 19 years. The minimum of the daily internet usage was 104.78 units.

ii)Maximum

```
# Finding the maximum values of the numerical columns
sapply(no.outliers.numeric, max)
```

```
## Daily.Time.Spent.on.Site                     Age             Area.Income
##                    91.43                   61.00                79484.80
##     Daily.Internet.Usage                    Male            Clicked.on.Ad
##                   269.96                    1.00                    1.00
```

The maximum of the daily time spent on site was 91.43 units. The maximum of the area income was 79,484.80 units. The maximum of the ages of the respondents was 61 years. The maximum of the daily internet usage was 269.96 units. The maximum value of whether male or not is 1. The maximum value of whether clicked on advert or not is 1. ###iii) Variance

```
# Finding the variance of all the variables
# area <-sd(no.outliers.numeric$Area.Income)
sapply(no.outliers.numeric, var)
```

```
## Daily.Time.Spent.on.Site                     Age             Area.Income
##             2.528258e+02            7.752303e+01             1.680004e+08
##     Daily.Internet.Usage                    Male            Clicked.on.Ad
##             1.940743e+03            2.498242e-01             2.502319e-01
```

The variance of the daily time spent on site was 252.82. The variance of the area income was 168,000,385. The variance of the ages of the respondents was 77.52. The variance of the daily internet usage was 1940.74. The variance of male column is 0.2498 . The variance of whether ad was clicked or not 0.2502 .

###iv) Standard Deviation

```
# Finding the standard deviation for all numeric variables
sapply(no.outliers.numeric, sd)
```

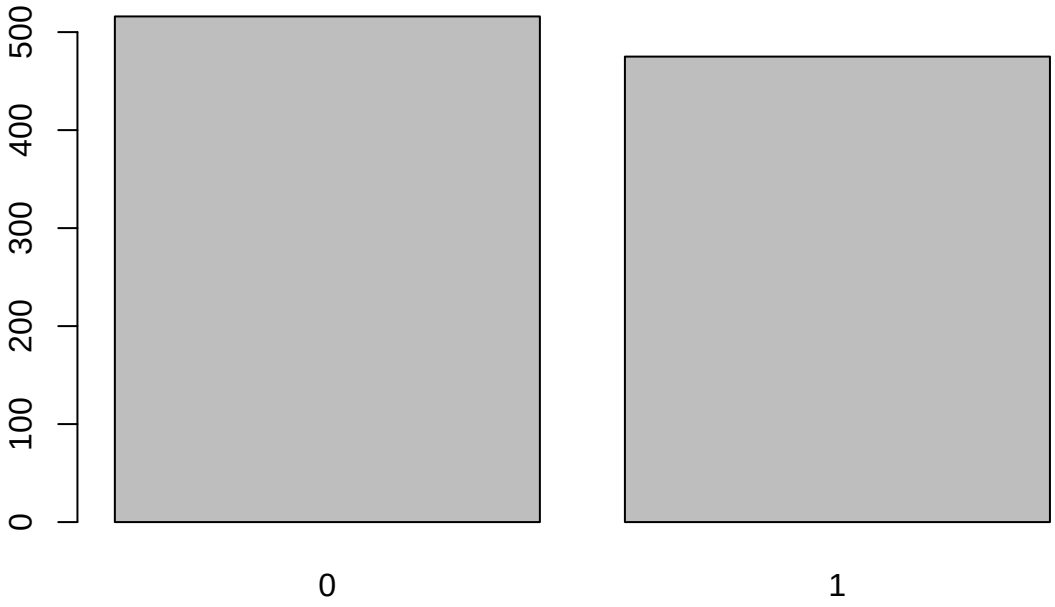
```
## Daily.Time.Spent.on.Site                     Age             Area.Income
##             1.590050e+01            8.804716e+00             1.296150e+04
##     Daily.Internet.Usage                    Male            Clicked.on.Ad
##             4.405386e+01            4.998241e-01             5.002318e-01
```

The standard deviation of the daily time spent on site was 15.90. The standard deviation of the area income was 12,961.5. The standard deviation of the ages of the respondents was 8.80. The standard deviation of the daily internet usage was 44.05.

Univariate Graphicals

Males

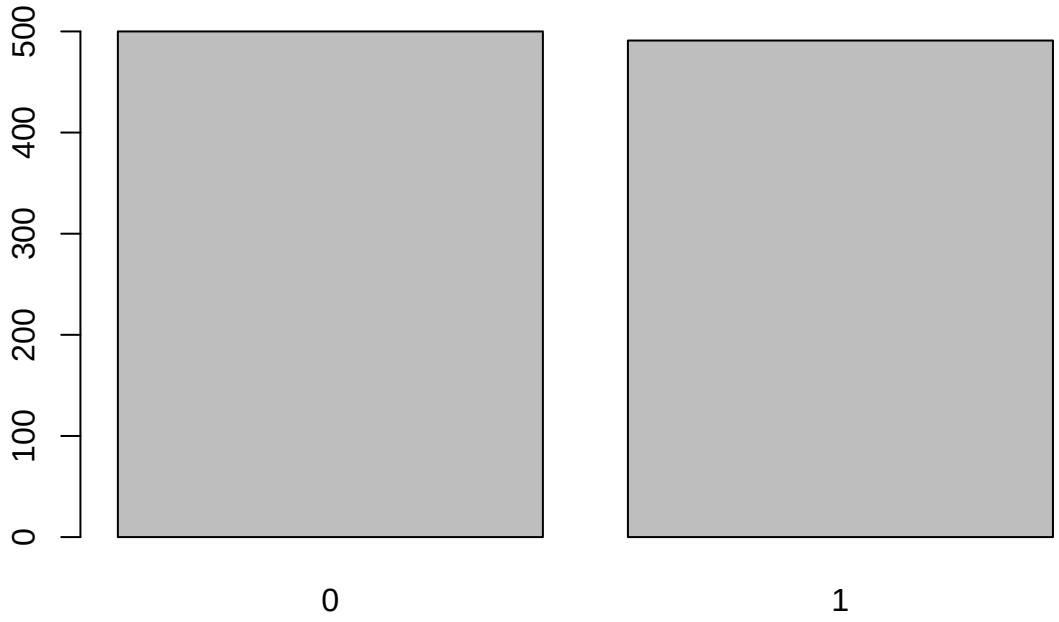
```
# Plotting a bar-graph to see the frequency of the categorical variables
# The table() function computes the frequency distribution of the categorical variables

# for the male column
barplot(table(no.outliers.numeric$Male))
```



The respondents who were not males were more than those who were males

Clicked on ad

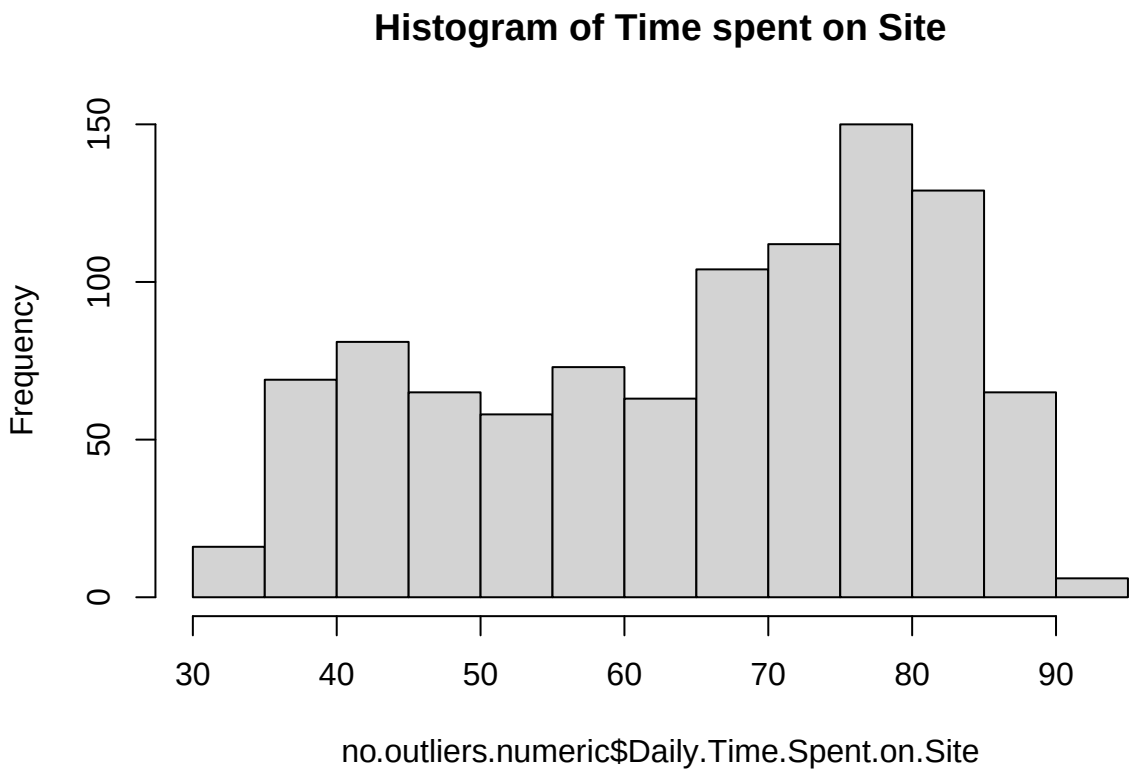
```
# for the male column
barplot(table(no.outliers.numeric$Clicked.on.Ad))
```



The number of respondents who clicked and those who did not click on adverts were almost similar

Time spent on site

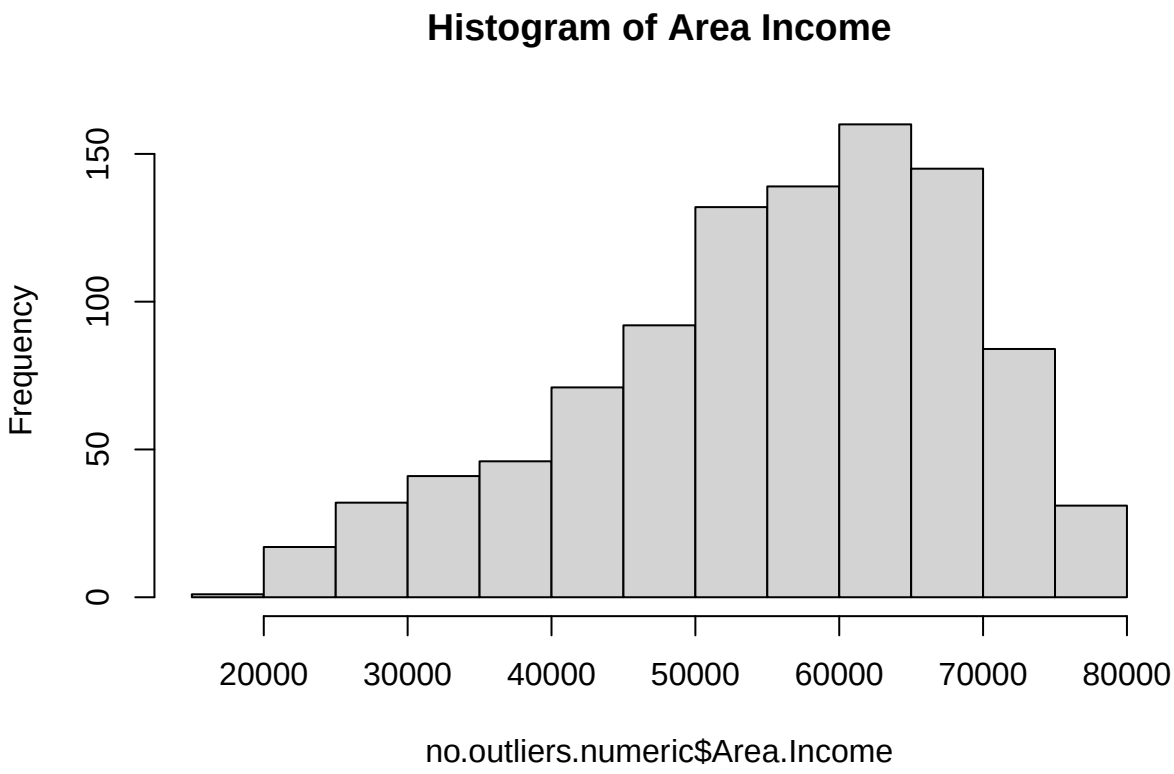
```
# Plotting histograms to show the distribution of the numerical variables

# Histogram of time spent on site
hist(no.outliers.numeric$Daily.Time.Spent.on.Site, main = "Histogram of Time spent on Site")
```



The time spent on sight is not skewed, meaning the data points tend to be evenly distributed

Area Income

```
# Histogram of area income
hist(no.outliers.numeric$ Area.Income, main = "Histogram of Area Income")
```



The area income is left skewed, meaning the data points extend to the left of the distribution

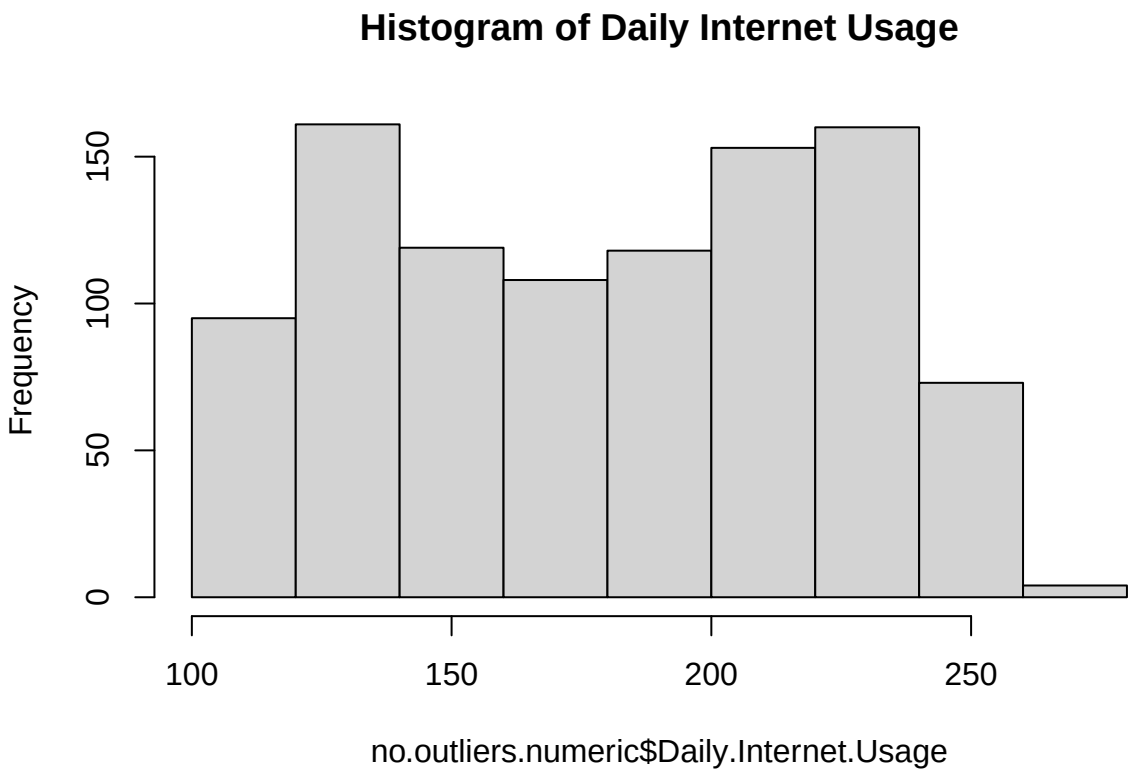
Age

```
# Histogram of Age
hist(no.outliers.numeric$Age, main = "Histogram of Age")
```

The age variable is right skewed, meaning the data points extend to the right of the data points distribution

Daily internet usage

```
# Histogram of daily internet usage
hist(no.outliers.numeric$Daily.Internet.Usage, main = "Histogram of Daily Internet Usage")
```



Daily internet usage is not skewed, meaning the data points tend to be normally distributed

Countries

```
# Checking the number of countries
# Checking the unique entries
countries <-unique(no.outliers$Country)

# printing the number of unique countries
# we will use the length function to do a unique value count
length(countries)
```

```
## [1] 237
```

There are 237 countries in the data-set

Cities

```
# Checking the number of cities
# Checking the unique entries
cities <-unique(no.outliers$City)

# printing the number of unique cities
# we will use the length function to do a unique value count
length(cities)
```

```
## [1] 960
```

There are 960 cities in the dataset

Bivariate Analysis

Previewing the top 6 rows

```
# previewing
head(no.outliers)
```

```
##    Daily.Time.Spent.on.Site   Age Area.Income Daily.Internet.Usage
##                       <num> <int>       <num>                <num>
## 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
##                                   <char>         <char> <int>     <char>
## 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
##                  <POSc>         <int>
## 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
```

i) Covariance

```
# finding the covariance of the target variable variables
# we assign different variables for the specific columns

# Assigning Daily.Time.Spent.on.Site column to variable time.site
time.site <- no.outliers$Daily.Time.Spent.on.Site

# Assigning Age column to variable age
age <-no.outliers$Age

# Assigning Area.income column to variable area.income
area.income <-no.outliers$Area.Income

# Assigning Daily.Internet.Usage column to variable daily.internet
daily.internet <-no.outliers$Daily.Internet.Usage
```

```
# Assigning Male column to variable male
#male <-no.outliers$Male

# Assigning clicked on ads column to variable clicks.target
#clicks.target <-no.outliers$Clicked.on.Ad
```

```
# Finding co-variances of the numerical variables

# covariance of age and time spent on site
cov(time.site,age )
```

```
## [1] -46.59899
```

There is a negative linear relationship between the variables

```
# covariance of age and time spent on site
cov(time.site,area.income )
```

```
## [1] 64600.67
```

There is a strong positive linear relationship between the time spent on the site and the area income

```
# covariance of age and time spent on site
cov(time.site,daily.internet)
```

```
## [1] 364.2711
```

There is a positive linear relationship between the time spent on the site and the daily internet usage

```
# covariance of age and time spent on site
cov(age,area.income )
```

```
## [1] -20744.22
```

There is a strong negative linear relationship between the age and area income variables

```
# covariance of age and time spent on site
cov(age,daily.internet )
```

```
## [1] -142.7226
```

There is a negative linear relationship between the age and daily internet usage variables

```
#covariance
cov(area.income,daily.internet )
```

```
## [1] 201115
```

There is a strong positive linear relationship between the daily internet usage and area income variables

ii) Correlation

We will use the numeric dataframe

```
# correlation matrix
ad_cor <- cor(no.outliers.numeric, use="pairwise.complete.obs",method = "pearson")
round(ad_cor, 2)
```

```
##                           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        0.35
## Male                                        -0.02 -0.02        0.01
## Clicked.on.Ad                               -0.75  0.49       -0.47
##                           Daily.Internet.Usage  Male Clicked.on.Ad
## Daily.Time.Spent.on.Site                 0.52 -0.02         -0.75
## Age                                     -0.37 -0.02          0.49
## Area.Income                              0.35  0.01         -0.47
## Daily.Internet.Usage                     1.00  0.03         -0.79
## Male                                     0.03  1.00         -0.04
## Clicked.on.Ad                           -0.79 -0.04          1.00
```

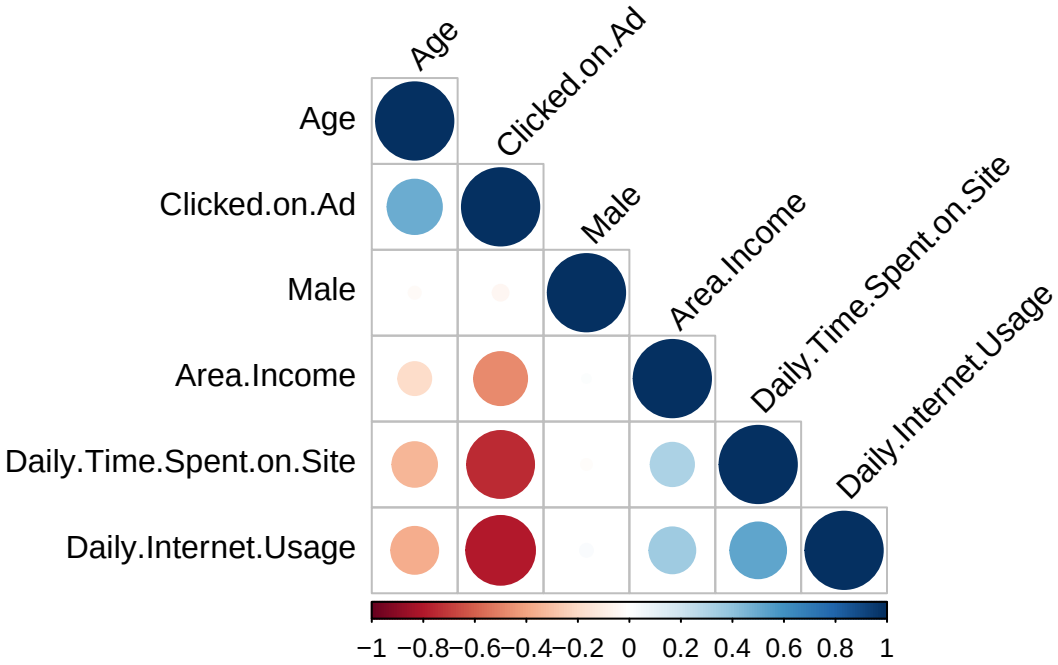
```
# gives correlation co-efficients in pairs and rounding them off to decimal places

# When the correlation the coefficient value is next to 1 it shows a positive linear relationship,
# when next to -1, it indicates that the variables are negatively linearly related
# When close to zero, it would indicate a weak linear relationship between the variables.
```

Correlation matrix

```
# Visualizing the correlation matrix
library(corrplot)
```

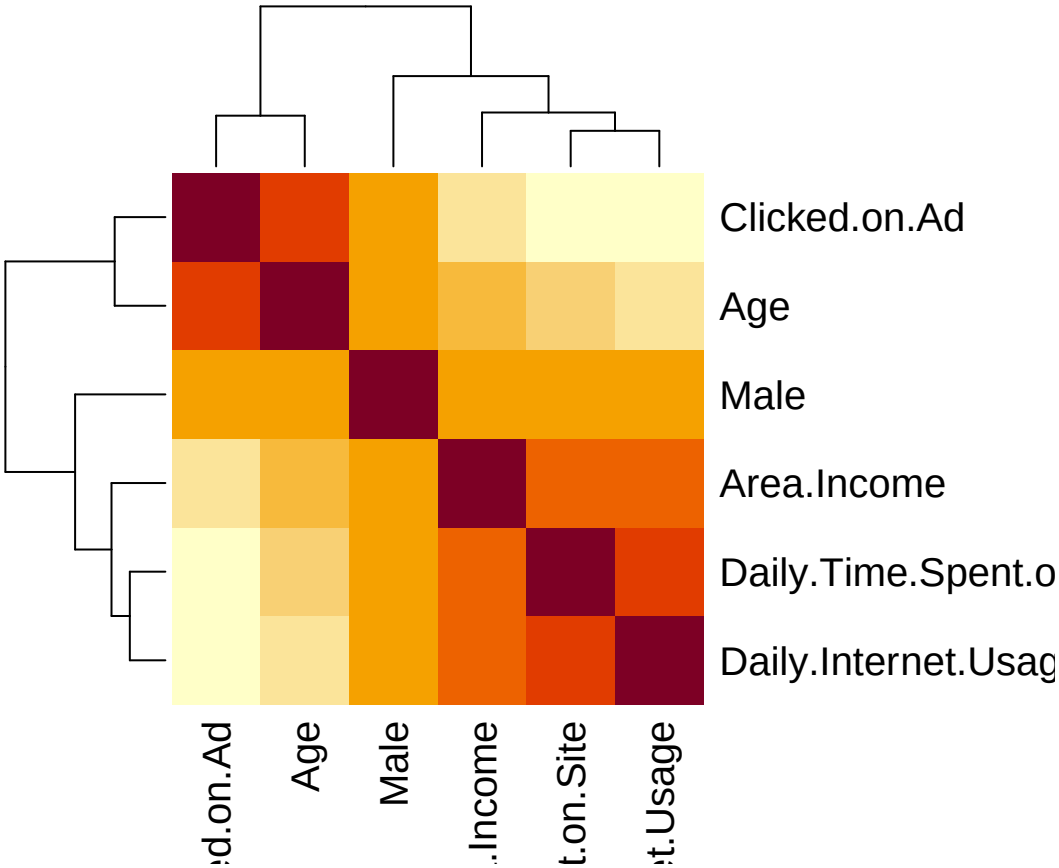
```
## corrplot 0.92 loaded
```

```
corrplot(ad_cor, type = "lower", order = "hclust",
         tl.col = "black", tl.srt = 45)
```



Correlation heatmap

```
# Plotting a correlation Heatmap
# Get some colors
#col<- colorRampPalette(c("blue", "white", "red"))(20)
heatmap(x = ad_cor, symm = TRUE)
```



Graphical Representations

```
# Plotting bivariate bar graphs and scatter plots
# we will use the variables we assigned earlier
#time.site
#age
#area.income
#daily.internet
#male
#clicks.target
```

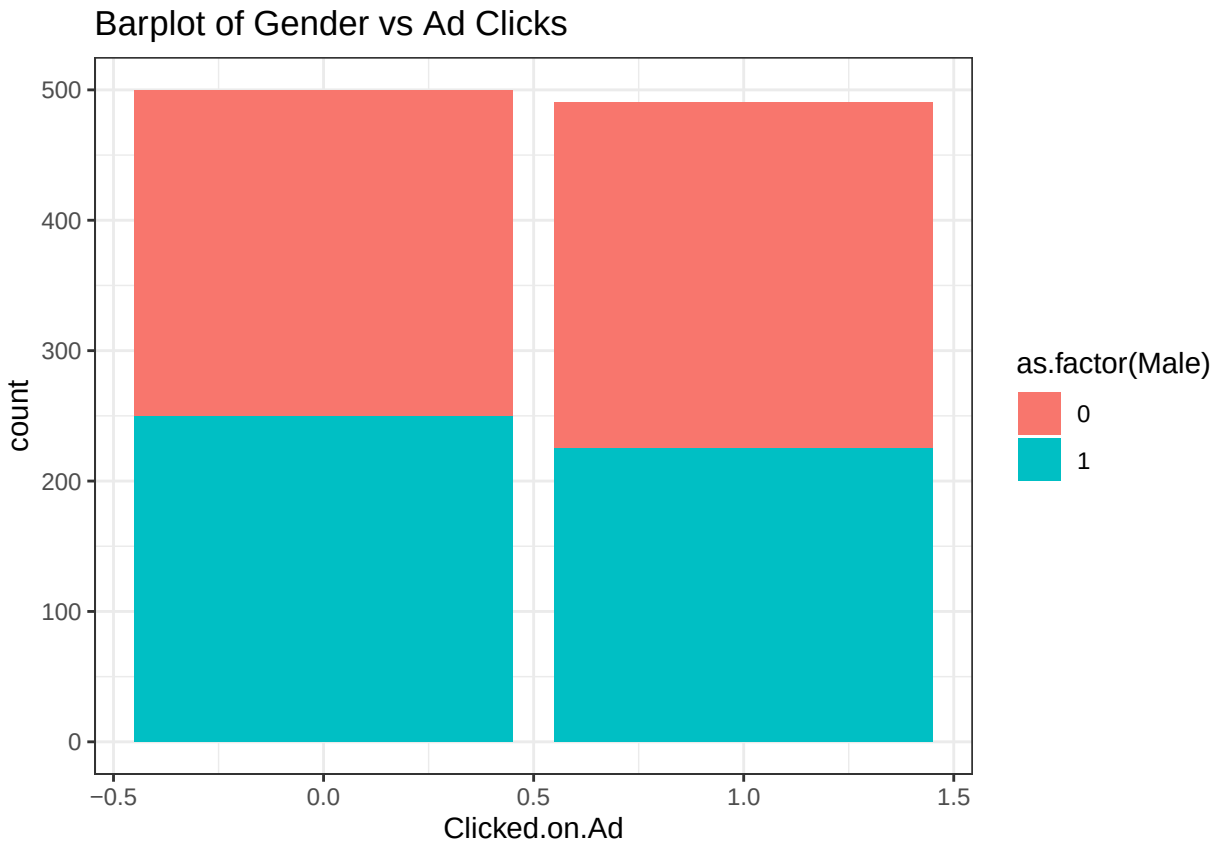
Categorical vs. Categorical

```
# printing column names
colnames(no.outliers)
```

```
##  [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"
```

Stacked bar graphs

```
# we will use stacked bargraphs to show the distribution of
# ad clicks among different genders
# we will have the distribution of the ad clicks on the x axis and
# the male column as fill

ggplot(data= no.outliers)+geom_bar(aes(x=Clicked.on.Ad, fill=as.factor(Male)))+
  ggtitle(label="Barplot of Gender vs Ad Clicks")+
  theme_bw() # picks a color theme
```

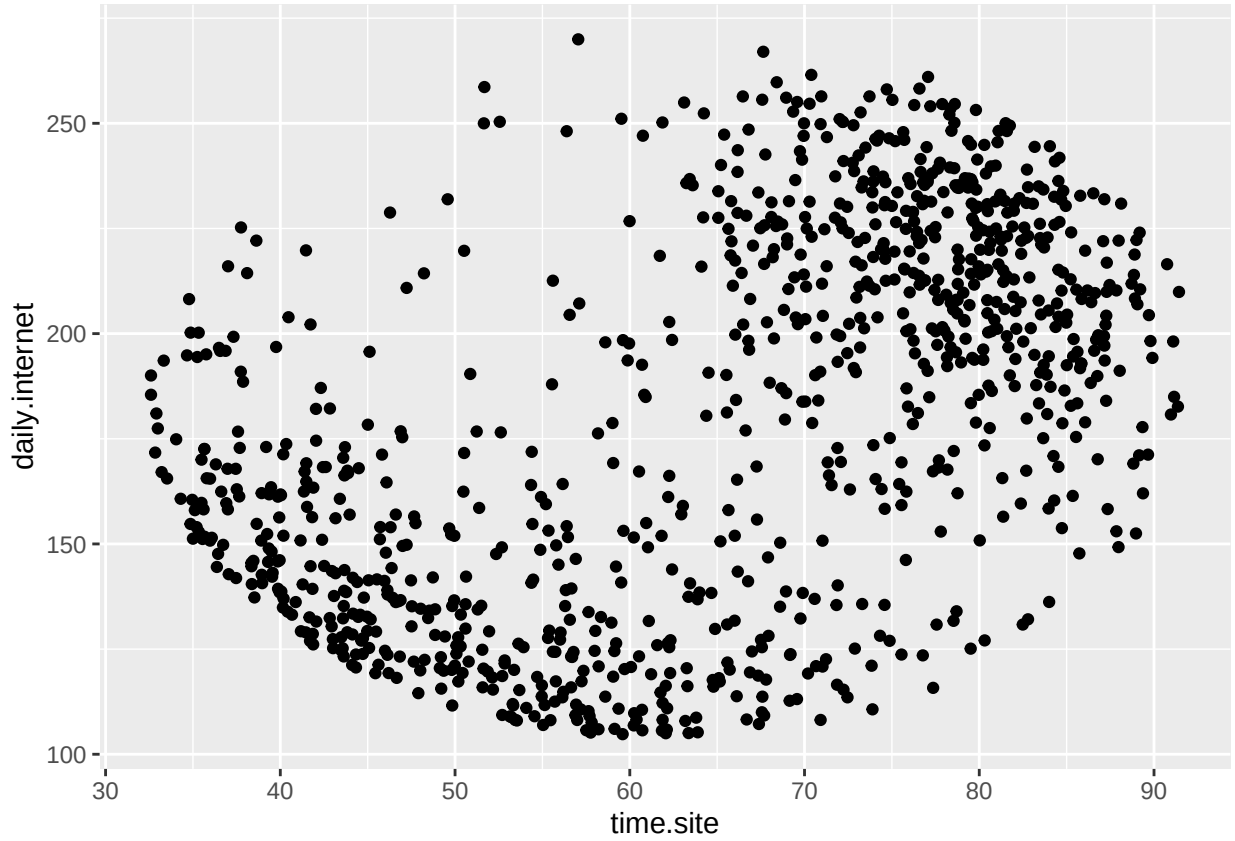


Most of the people who clicked on the ads were not males The number of males and other gender that did not click on the ads were equal

Numerical vs Numerical

Time spent on site versus daily internet usage

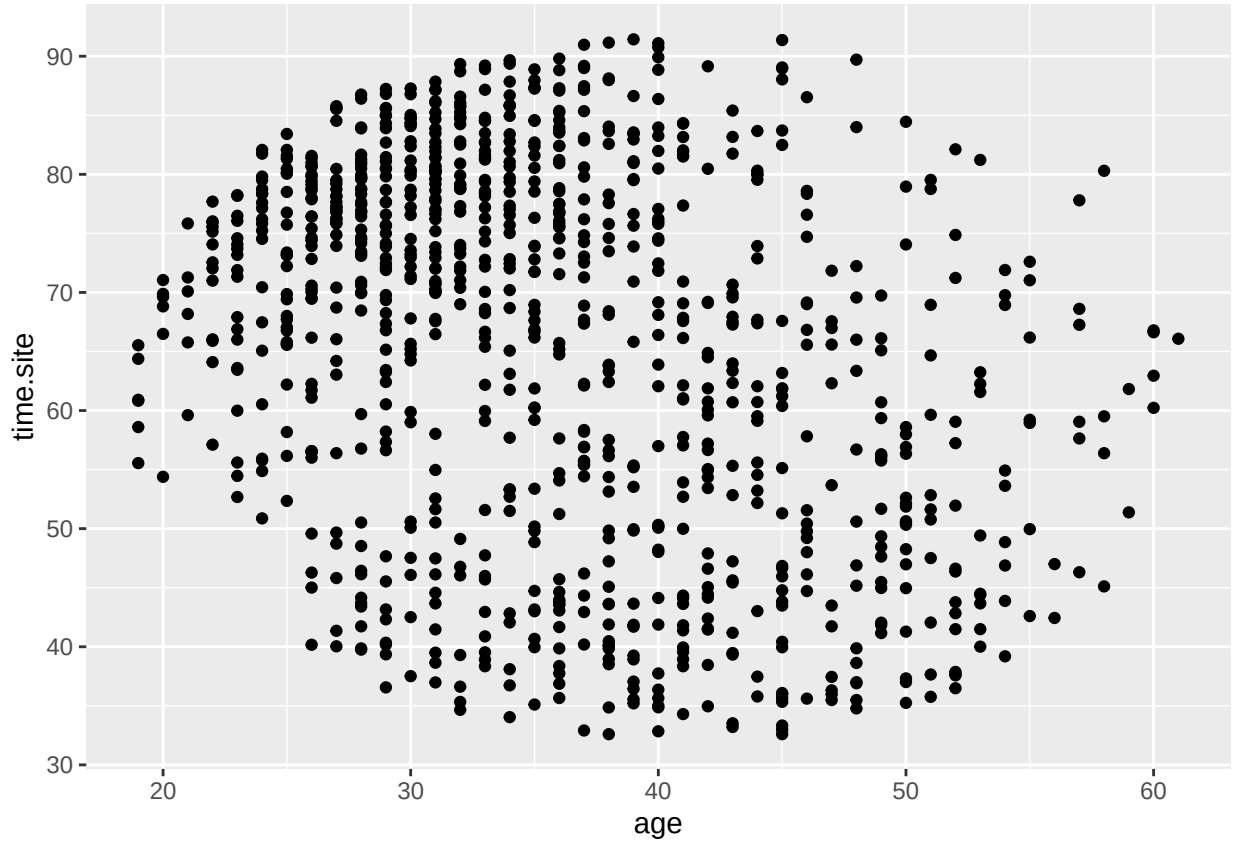
```
# plotting a scatter plot to show the distribution of
# time spent on site versus the daily internet usage

ggplot(no.outliers,
       aes(x = time.site,
           y = daily.internet)) +
  geom_point()
```



Most people that spent 80-90 minutes on the sites used more internet, between 150-300 units Most people that spent less than 60 minutes on the sites used less internet, approximately below 200 units

Age versus Time spent on the site

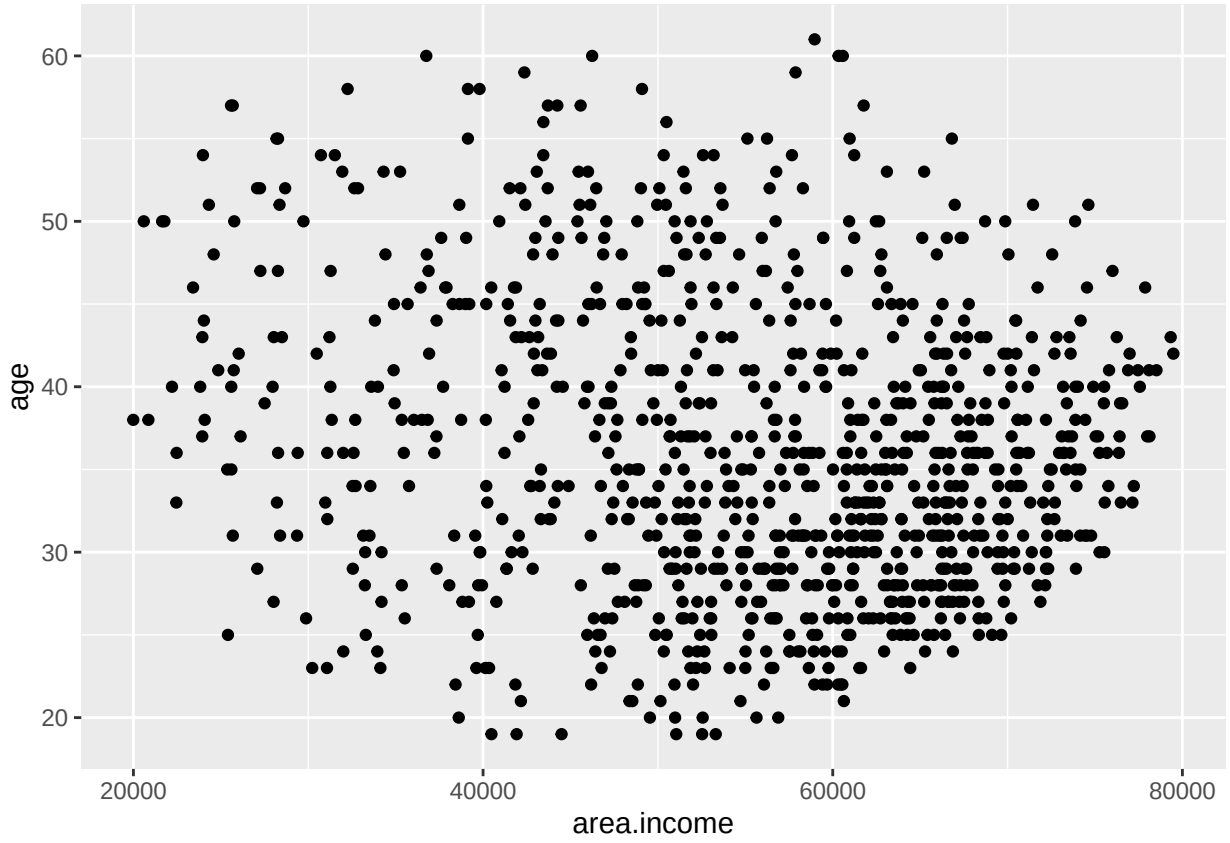
```
# plotting a scatter plot showing age versus
#time spent on site
#plot(age, time.site, xlab="Time on site", ylab="Age of Respondent")

ggplot(no.outliers,
       aes(x = age,
           y = time.site)) +
  geom_point()
```



There is no observable strong relationship between the age of the people and the time spent on the site. However, most people below 40 years are seen to spend between 70 to 90 minutes on the sites.

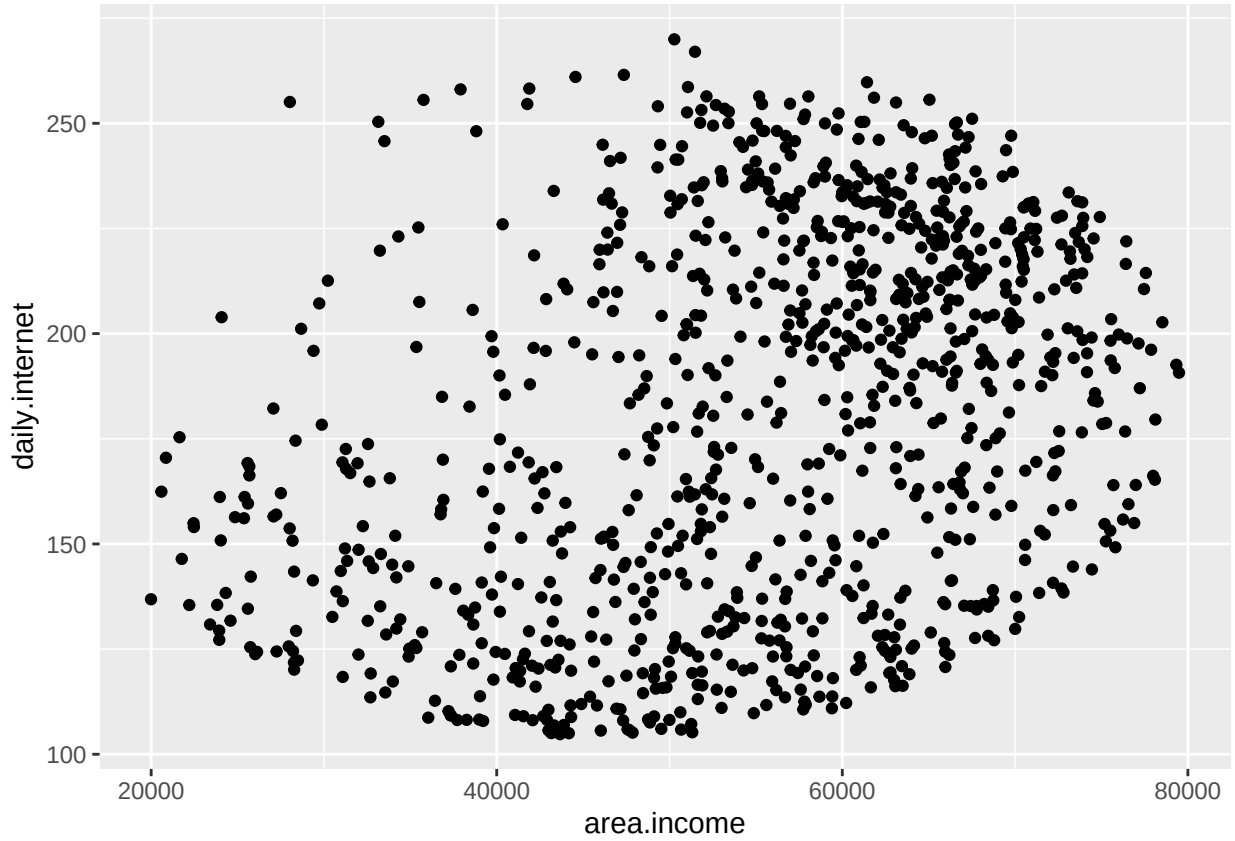
Area income versus Age

```
# plotting a scatter plot to show the correlation of
# area income versus age
#plot(age, area.income, xlab="Age", ylab="Area.Income")

ggplot(no.outliers,
       aes(x = area.income,
           y = age)) +
  geom_point()
```



Area income versus Age

```
# plotting a scatter plot to show the relationship
# between the area income versus the daily internet usage
#plot(area.income, daily.internet, xlab="Daily Internet", ylab="Area Income")

ggplot(no.outliers,
       aes(x = area.income,
           y = daily.internet)) +
  geom_point()
```



There is no significant relationship between the area income and the time spent on the internet. However, areas above 50,000 units were seen to have a wide range of daily internet usage, from as low as around 100 to as high as around 275 units per day. 1.00

Categorical vs. Numerical

Plotted using grouped kernel density plots

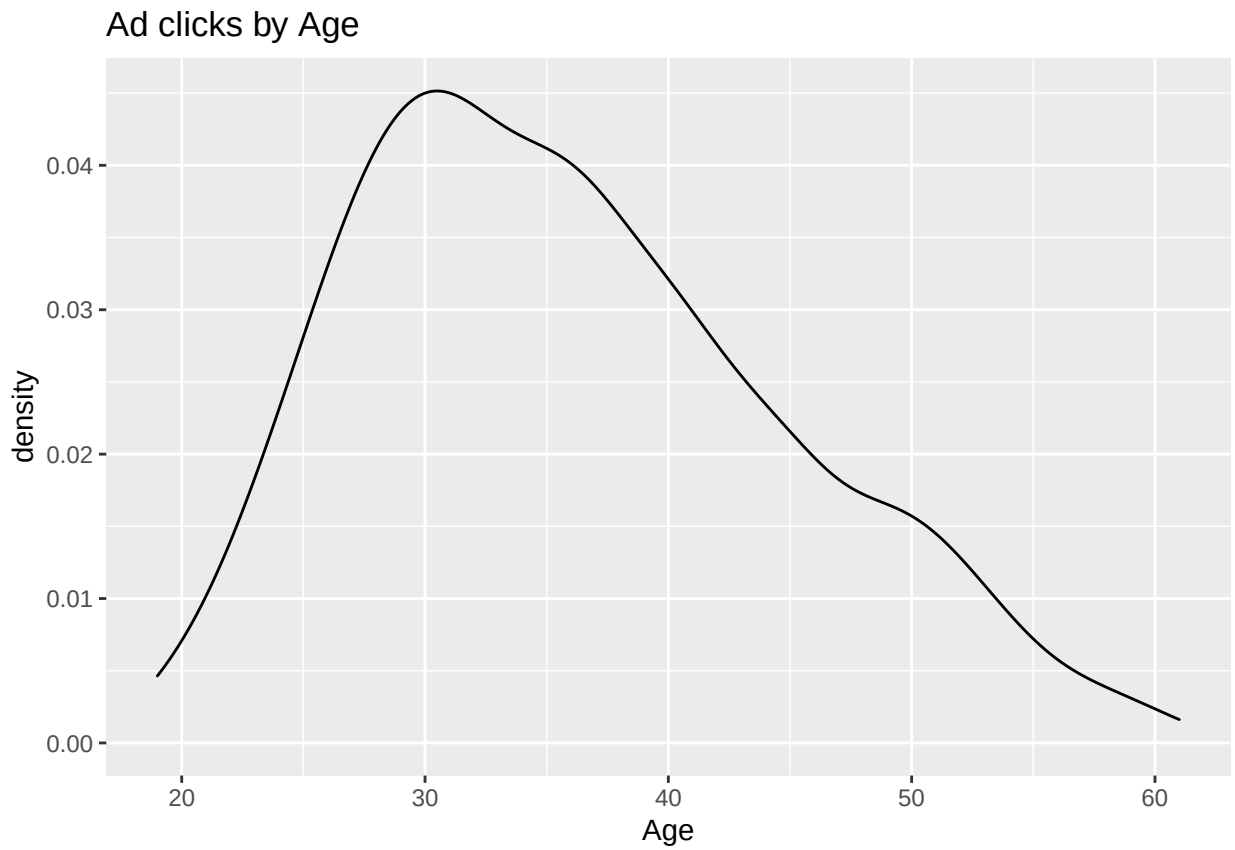
```
# checking column names
colnames(no.outliers)
```

```
##  [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"
```

Ad Clicks vs. Age

```
# plotting a grouped kernel density plot
# to show the distribution of ad clicks in
# different age groups
ggplot(no.outliers,
       aes(x = Age,
           fill = Clicked.on.Ad)) +
```

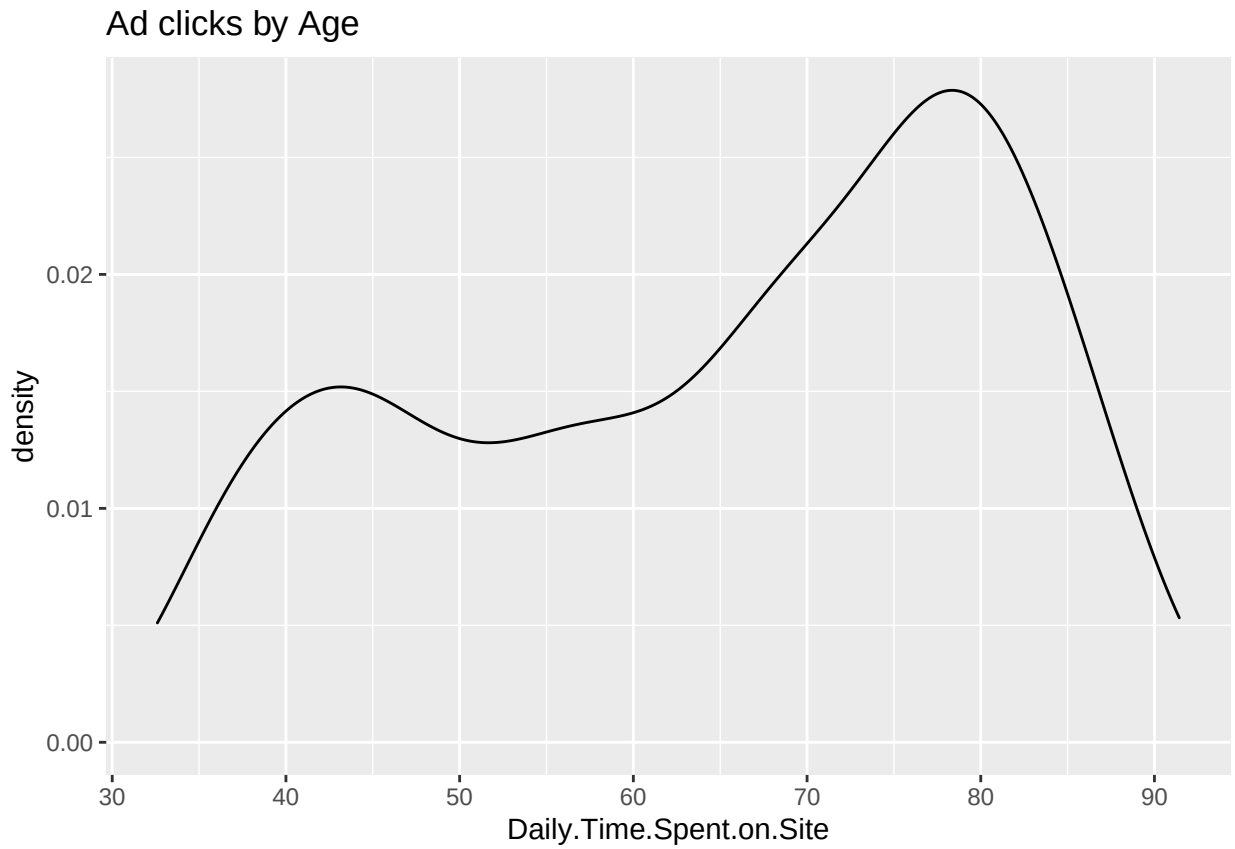
```
  geom_density(alpha = 0.8) +
  labs(title = "Ad clicks by Age")
```



Most people that clicked on the ads were 40years and above, with the peak being at around age 30.

Ad Clicks vs. Daily Time spent on Site

```
# plotting
#plot(time.site, daily.internet, xlab="Time on Site", ylab="Daily Internet Usage")

ggplot(no.outliers,
       aes(x = Daily.Time.Spent.on.Site,
           fill = Clicked.on.Ad)) +
  geom_density(alpha = 0.4) +
  labs(title = "Ad clicks by Age")
```



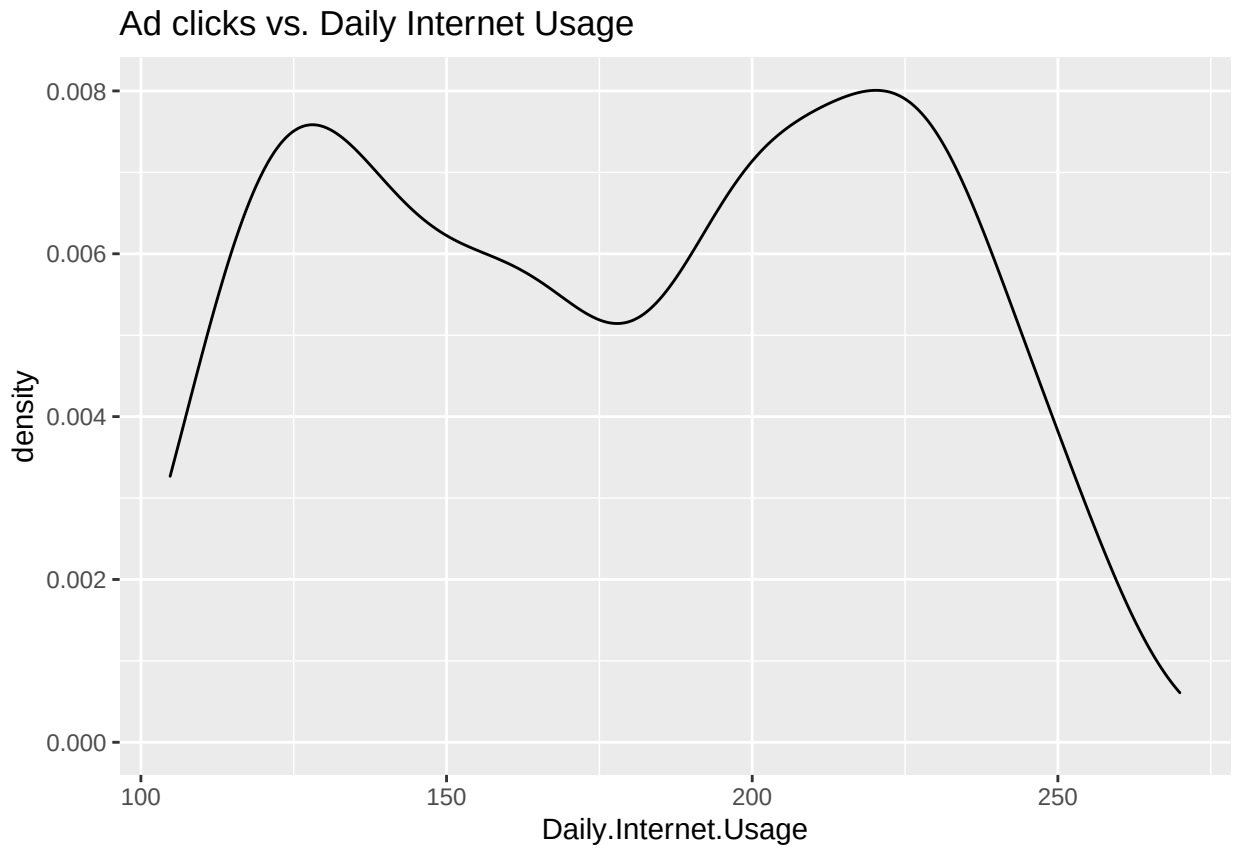
Most people that clicked on the ads spent between 70-80 minutes on the site, with a low of between 50-55 minutes. There is a steep drop after around 80 minutes on the site.

```
colnames(no.outliers)
```

```
##  [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"
```

Ad clicks vs. Daily Internet Usage

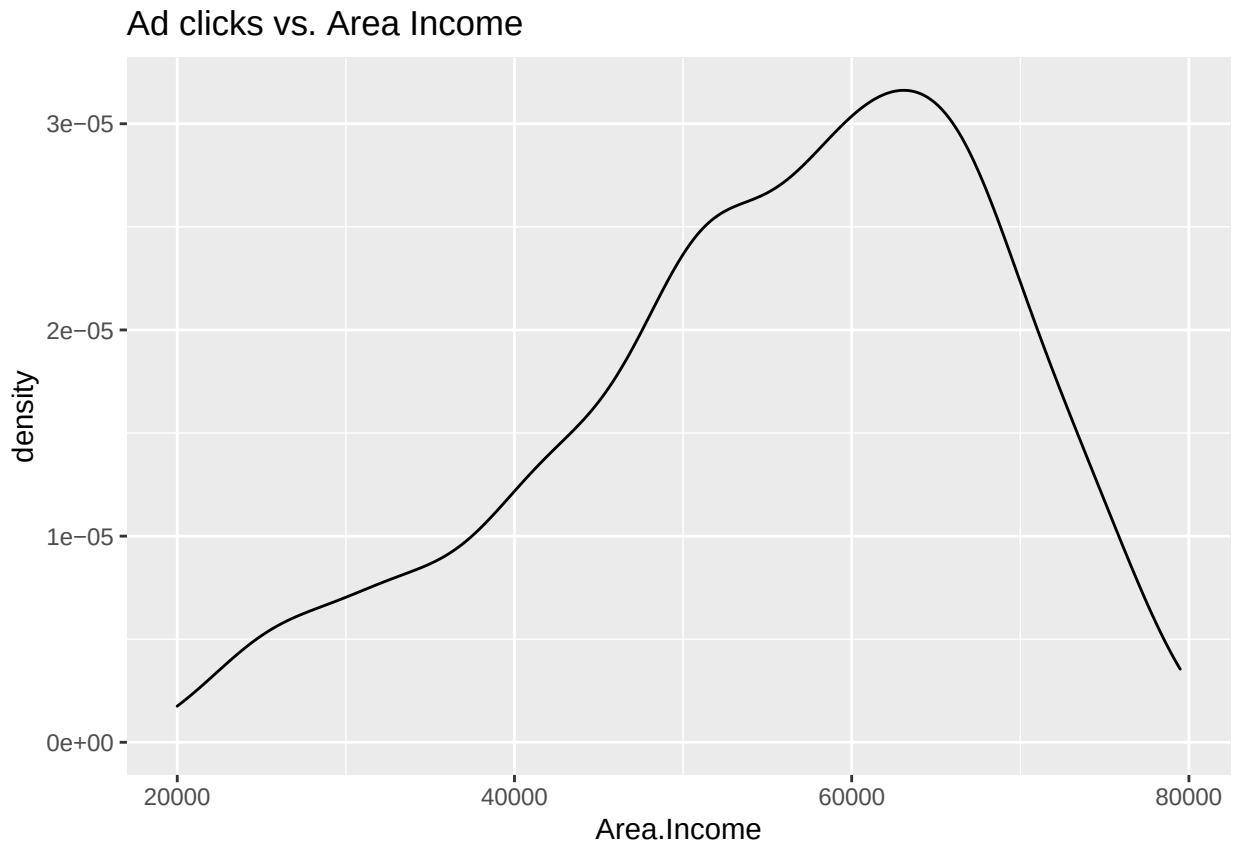
```
# plotting
#plot(clicks.target, daily.internet, xlab="Daily Internet", ylab="Clicks on Ad")

ggplot(no.outliers,
       aes(x = Daily.Internet.Usage,
           fill = Clicked.on.Ad)) +
  geom_density(alpha = 0.4) +
  labs(title = "Ad clicks vs. Daily Internet Usage")
```



There are two peaks observed for this distribution. Most people that clicked on the ads used around 125 and 225 units of internet daily. At, 175 units, there is a drop in the clicks. Past 225 units of daily internet usage, the number of clicks reduces very steadily.

Ad clicks vs. Area Income

```
# plotting
# plot(clicks.target, age, xlab="Clicks on ad", ylab="Age")
ggplot(no.outliers,
       aes(x = Area.Income,
           fill = Clicked.on.Ad)) +
  geom_density(alpha = 0.4) +
  labs(title = "Ad clicks vs. Area Income")
```




```
# there is no relationship between time spent on site and age
```

Most ads were clicked by people who's area income was less than 50,000 units, with a rise at 60,000units. There is seen a steep drop on the clicks fro people with area income of over 60,000 units.

Multivariate Analysis

We will print the correlation matrix initially computed to recall the variable correlations

```
# printing our correlation matrix
round(ad_cor, 2)
```

```
##                           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        0.35
## Male                                        -0.02 -0.02        0.01
## Clicked.on.Ad                               -0.75  0.49       -0.47
##                           Daily.Internet.Usage  Male Clicked.on.Ad
## Daily.Time.Spent.on.Site                 0.52 -0.02         -0.75
## Age                                     -0.37 -0.02          0.49
## Area.Income                              0.35  0.01         -0.47
```

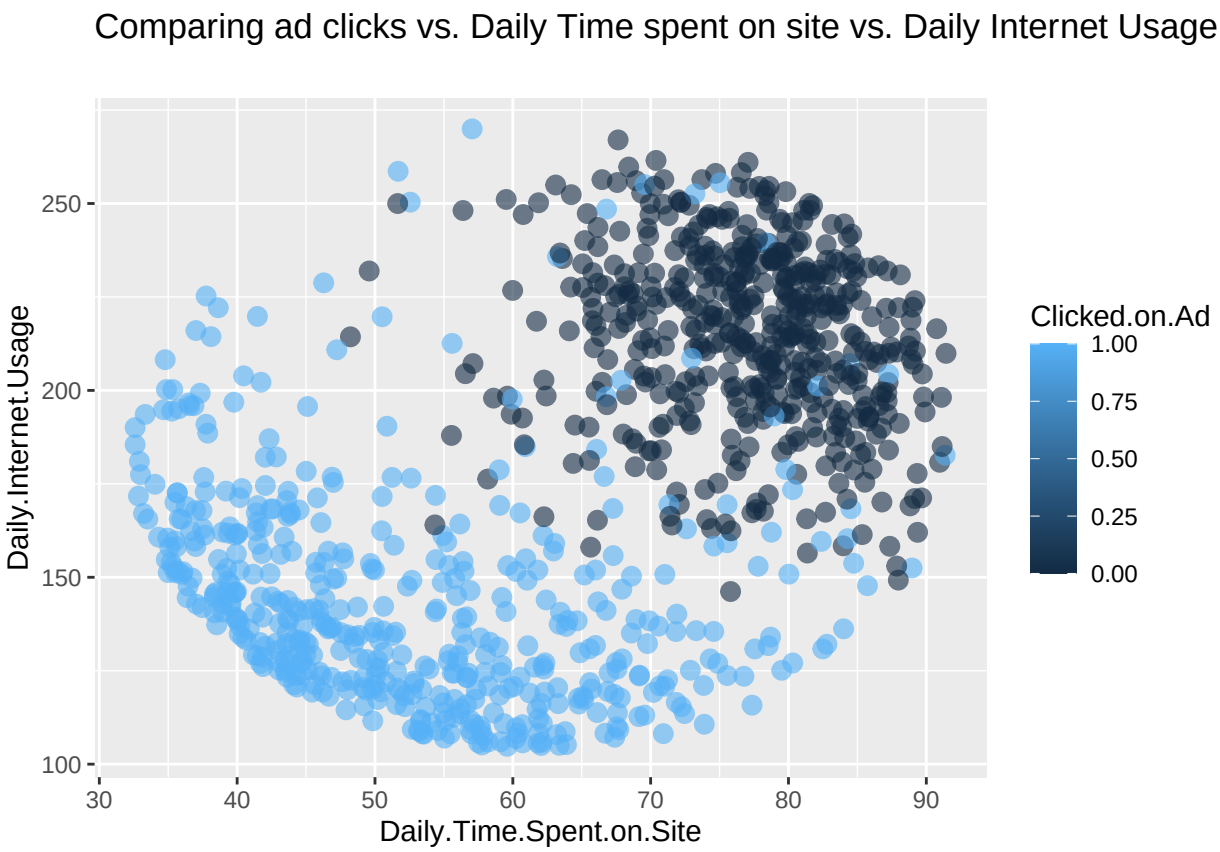
```
## Daily.Internet.Usage                    1.00  0.03         -0.79
## Male                                    0.03  1.00         -0.04
## Clicked.on.Ad                          -0.79 -0.04          1.00
```

Ad clicks have moderate correlations with; daily internet usage, daily time spent on site, age and area income.

We will therefore do multivariate scatter plots for the same.

Comparing ad clicks vs. Daily Time spent on site vs. Daily Internet Usage

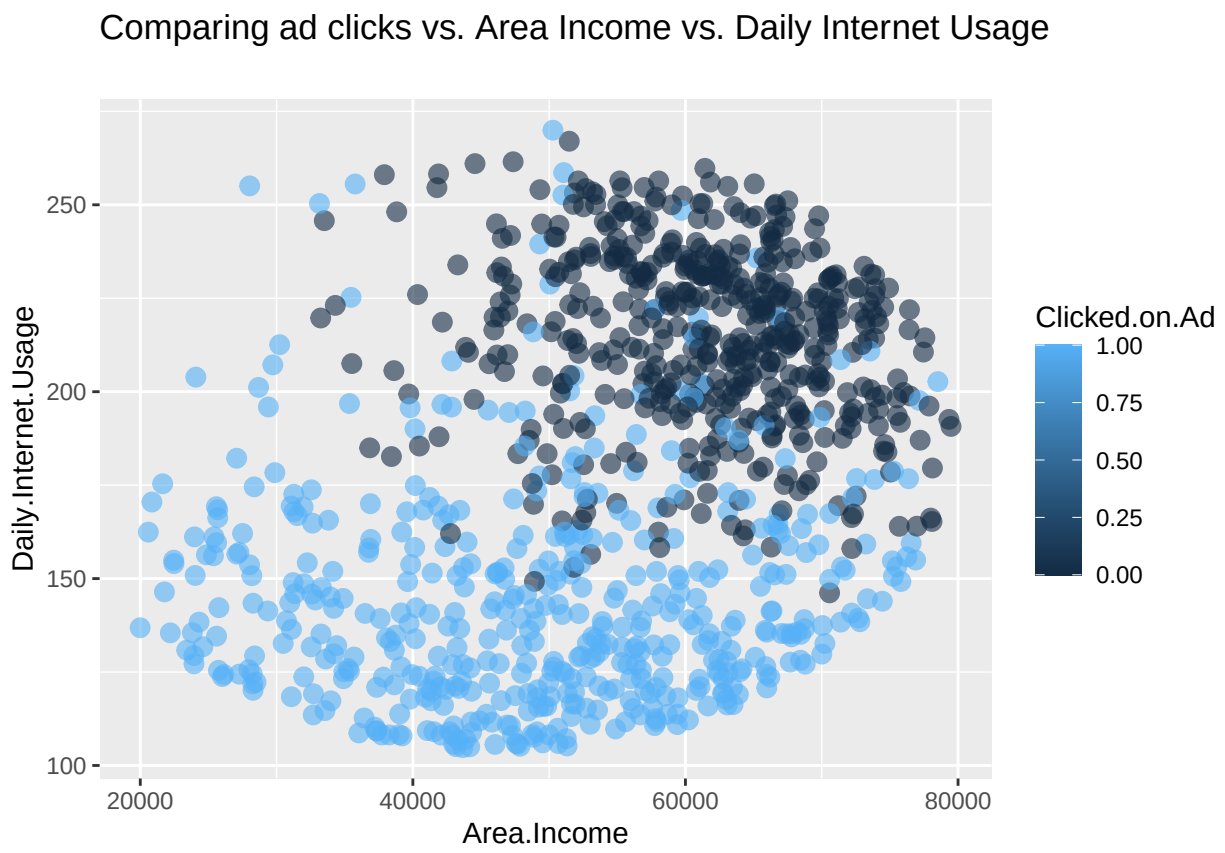
```
# plotting a scatter plot to compare the three variables above
library("ggplot2")

ggplot(no.outliers,
       aes(x = Daily.Time.Spent.on.Site,
           y = Daily.Internet.Usage,
           color = Clicked.on.Ad)) +
       geom_point(size = 3,
                  alpha = .6) +
        labs(title = "Comparing ad clicks vs. Daily Time spent on site vs. Daily Internet Usage
")
```



Most people that spent over 65 minutes on the internet and used over 150 units of internet daily clicked on the ads.

Comparing ad clicks vs. Area Income vs. Daily Internet Usage

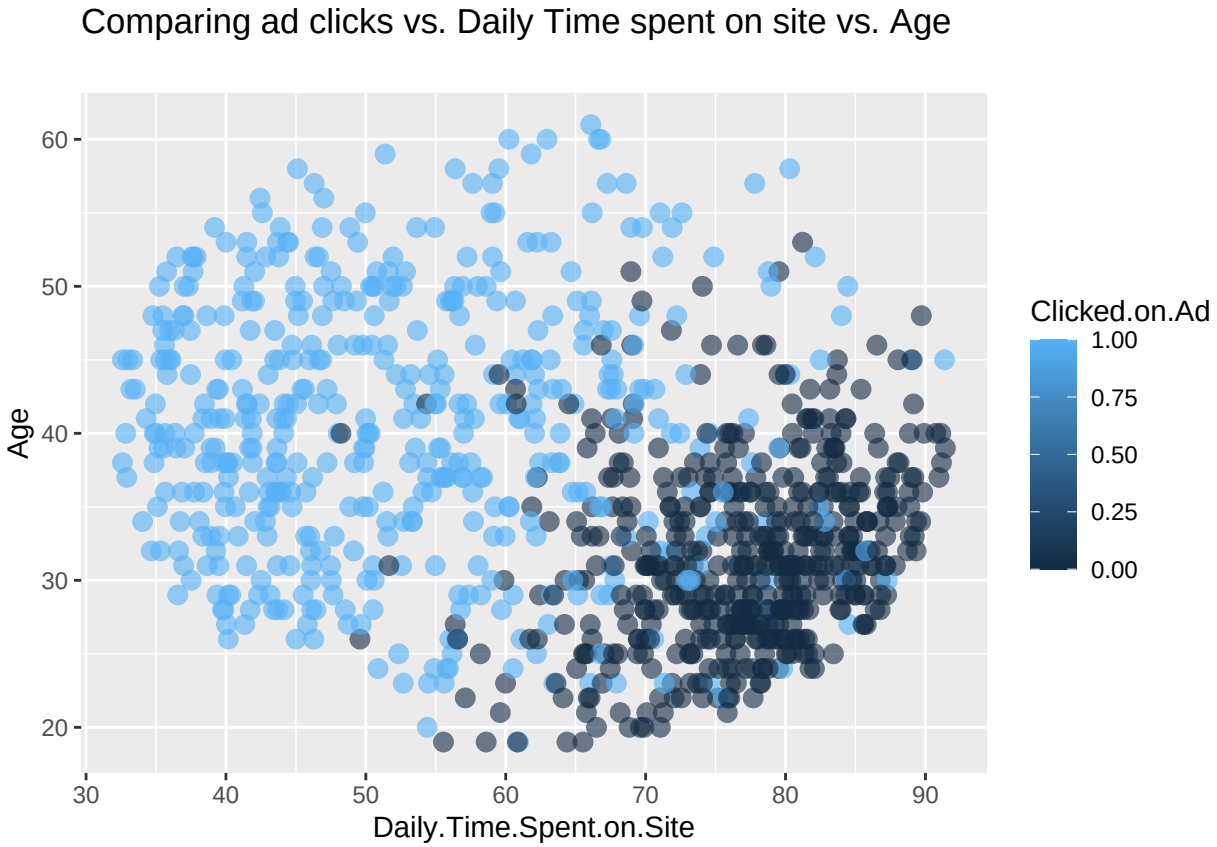
```
# plotting a scatter plot to compare the three variables above
ggplot(no.outliers,
       aes(x =Area.Income,
           y = Daily.Internet.Usage,
           color = Clicked.on.Ad)) +
       geom_point(size = 3,
                  alpha = .6) +
        labs(title = "Comparing ad clicks vs. Area Income vs. Daily Internet Usage
")
```



People with area income of above 50,000 units and spent over 150 units on daily internet usage clicked on the ads

Comparing ad clicks vs. Daily Time spent on site vs. Age

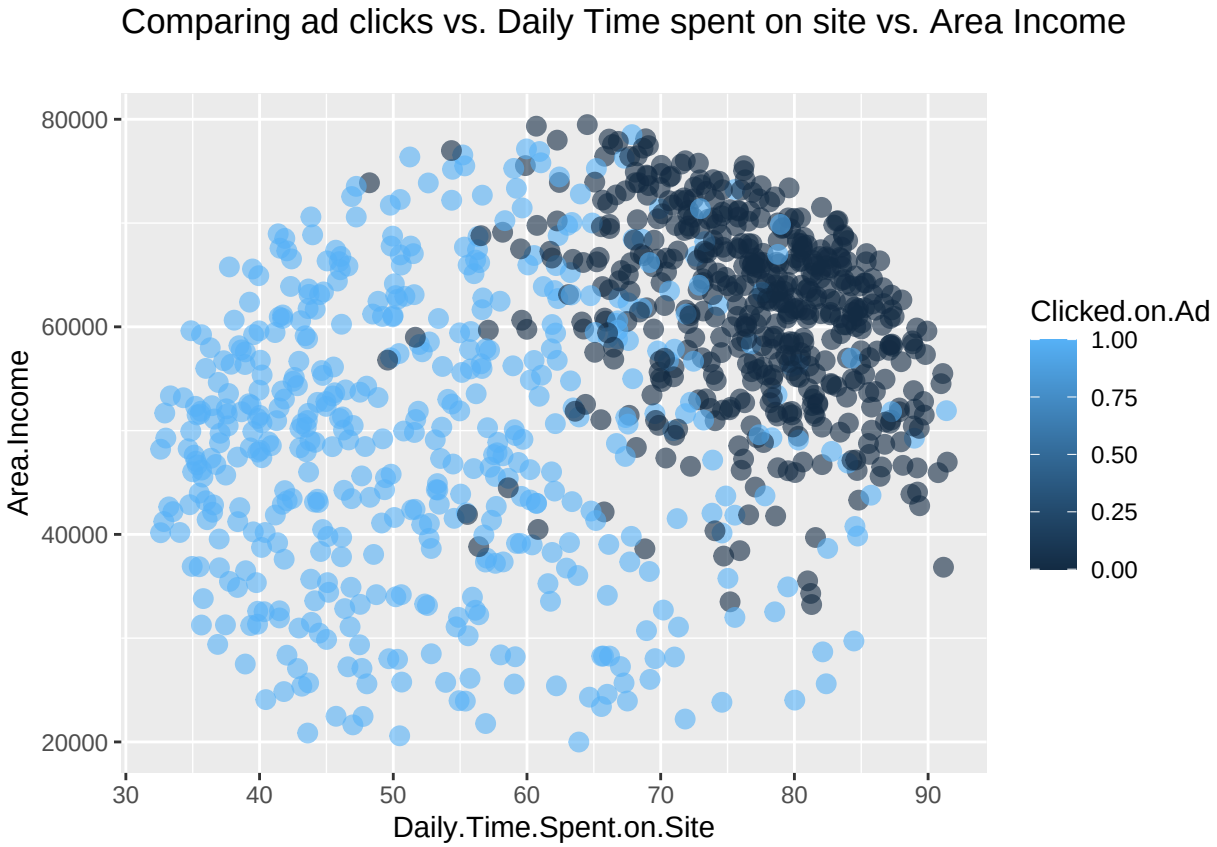
```
# plotting a scatter plot to compare the three variables above
library("ggplot2")

ggplot(no.outliers,
       aes(x = Daily.Time.Spent.on.Site,
           y = Age,
           color = Clicked.on.Ad)) +
       geom_point(size = 3,
                  alpha = .6) +
        labs(title = "Comparing ad clicks vs. Daily Time spent on site vs. Age
")
```



People below 40 years old that spent over 60minutes on the site clicked on the ads.

Comparing ad clicks vs. Area Income vs. Daily Time spent on Site

```
# plotting a scatter plot to compare the three variables above
library("ggplot2")

ggplot(no.outliers,
       aes(x = Daily.Time.Spent.on.Site,
           y = Area.Income,
           color = Clicked.on.Ad)) +
       geom_point(size = 3,
                  alpha = 0.6) +
        labs(title = "Comparing ad clicks vs. Daily Time spent on site vs. Area Income
")
```



People that spent over 60 minutes on the site and with area income of over 40,000 clicked on the ads

Modelling

We will use the Decision Tree Algorithm This is because this is a classification problem We are to classify where an ad will be clicked or not Labels : 0 = ad not getting clicked, 1 = ad getting clicked

```
# we will use the numeric dataset
# we will create a copy of it to use for modellling
# creating a copy
ad <- data.frame(no.outliers.numeric)

# previewing our dataset
head(ad)
```

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

```
## 3             0
## 4             0
## 5             0
## 6             0
```

Deleting the numerical column, Male

```
# deleting numerical column Male as it cannot be used for modelling
ad1 = select(ad, -Male)
```

Preview after drop

```
# checking columns
colnames(ad1)
```

```
## [1] "Daily.Time.Spent.on.Site" "Age"
## [3] "Area.Income"              "Daily.Internet.Usage"
## [5] "Clicked.on.Ad"
```

Multiple Linear Regression Model

The model

```
# Applying linear regression model function
multiple_lm <- lm(Clicked.on.Ad~ ., ad1)
```

Model Summary

```
# generating the model summary
summary(multiple_lm)
```

```
##
## Call:
## lm(formula = Clicked.on.Ad ~ ., data = ad1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.64701 -0.11592 -0.03121  0.05093  1.02847
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               2.279e+00  5.754e-02   39.61   <2e-16 ***
## Daily.Time.Spent.on.Site -1.275e-02  5.068e-04  -25.17   <2e-16 ***
## Age                       9.001e-03  8.309e-04   10.83   <2e-16 ***
## Area.Income              -5.745e-06  5.593e-07  -10.27   <2e-16 ***
## Daily.Internet.Usage     -5.334e-03  1.878e-04  -28.40   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2105 on 986 degrees of freedom
## Multiple R-squared:  0.8236, Adjusted R-squared:  0.8229
## F-statistic:  1151 on 4 and 986 DF,  p-value: < 2.2e-16
```

The anova table

```
#Generating the anova table

anova(multiple_lm)
```

```
## Analysis of Variance Table
##
## Response: Clicked.on.Ad
##                            Df  Sum Sq Mean Sq F value    Pr(>F)
## Daily.Time.Spent.on.Site    1 139.021 139.021 3137.13 < 2.2e-16 ***
## Age                         1  16.579  16.579  374.13 < 2.2e-16 ***
## Area.Income                 1  12.694  12.694  286.45 < 2.2e-16 ***
## Daily.Internet.Usage        1  35.741  35.741  806.52 < 2.2e-16 ***
## Residuals                 986  43.694   0.044
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Predicting

```
# Performing our prediction
pred_click <- predict(multiple_lm, ad1)

#printing a sample of the predictions
head(pred_click, 10)
```

```
##           1           2           3           4           5           6
## -0.006304017  0.108294719  0.022321201 -0.031846296  0.094579197  0.168435093
##           7           8           9          10
##  0.021577864  1.025523486  0.021604501  0.267908310
```

Model error

```
# using the train object as input to predict error
pred4 <- predict(multiple_lm, ad1)

error <- pred4 - ad1$Clicked.on.Ad

rmse_xval <- sqrt(mean(error^2)) ## xval RMSE

rmse_xval
```

```
## [1] 0.2099788
```

The model without tuning has 20% error.This is not so high hence model can be considered optimal.

Decision Tree

Importing Decision Tree Libraries

```
library(plyr); library(dplyr)
library(readr)
library(dplyr)
library(caret)
```

```
## Loading required package: lattice
```

```
library(rpart)
library(rpart.plot)
library(mlbench)
library(caTools)
library(party)
```

```
## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

##
## Attaching package: 'modeltools'

## The following object is masked from 'package:plyr':
##
##     empty

## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:data.table':
##
##     yearmon, yearqtr

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich
```

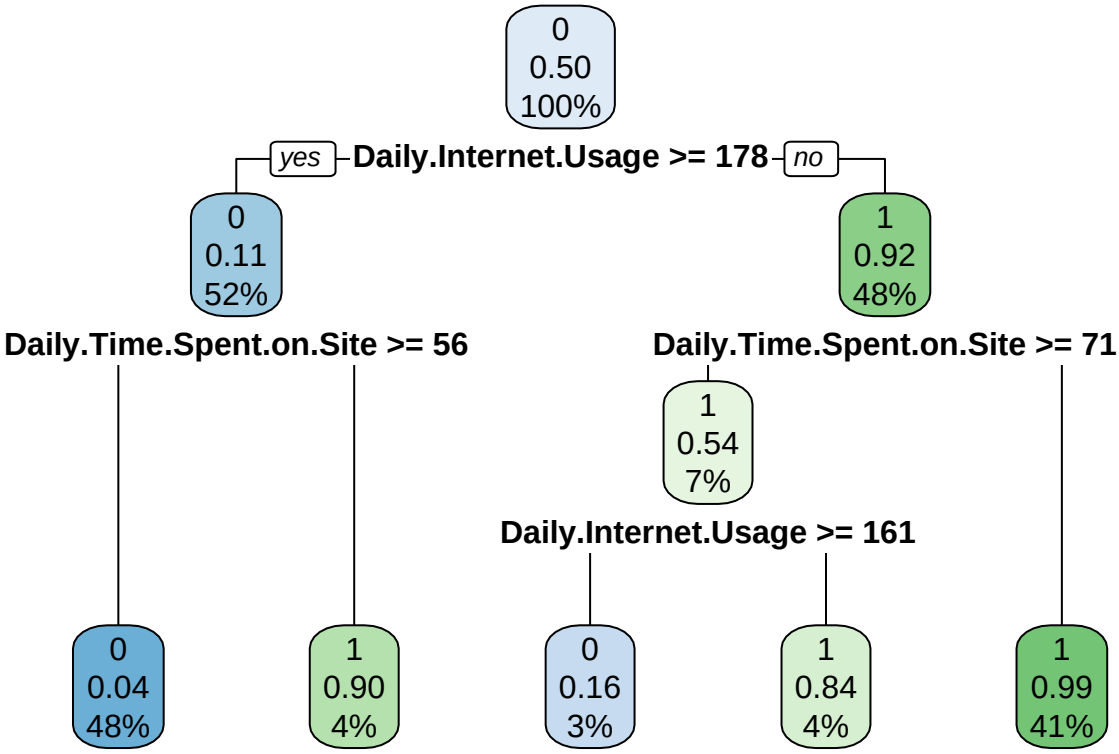
```
library(magrittr)
```

Decision Tree model Splitting the dataset

```
#data splicing
set.seed(12345)
train <- sample(1:nrow(ad1),size = ceiling(0.80*nrow(ad1)),replace = FALSE)
# training set
ad1_train <- ad1[train,]
# test set
ad1_test <- ad1[-train,]
```

Plotting a decision tree

```
# Plotting a decision tree
set.seed(35)

# creating the model
model_dt <- rpart(Clicked.on.Ad ~ ., data = ad1,
           method = "class")

#printing the model
rpart.plot(model_dt)
```



Making ad click predictions

```
# making predictions
advert_pred <- predict(model_dt, ad1, type = "class")
```

```
# creating a table function that
table(advert_pred, ad1$Clicked.on.Ad)
```

```
##
## advert_pred   0   1
##           0 485  25
##           1  15 466
```

Naive Bayes

Importing libraries

```
#install.packages('tidyverse')
library(tidyverse)
```

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

```
## v tibble  3.1.7     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()     masks plyr::arrange()
## x dplyr::between()     masks data.table::between()
## x stringr::boundary()  masks strucchange::boundary()
## x purrr::compact()     masks plyr::compact()
## x matrixStats::count() masks dplyr::count(), plyr::count()
## x tidyr::extract()     masks magrittr::extract()
## x dplyr::failwith()    masks plyr::failwith()
## x dplyr::filter()      masks stats::filter()
## x dplyr::first()       masks data.table::first()
## x dplyr::id()          masks plyr::id()
## x dplyr::lag()         masks stats::lag()
## x dplyr::last()        masks data.table::last()
## x purrr::lift()        masks caret::lift()
## x dplyr::mutate()      masks plyr::mutate()
## x dplyr::rename()      masks plyr::rename()
## x purrr::set_names()   masks magrittr::set_names()
## x dplyr::summarise()   masks plyr::summarise()
## x dplyr::summarize()   masks plyr::summarize()
## x purrr::transpose()   masks data.table::transpose()
```

```
#install.packages('ggplot2')
library(ggplot2)

#install.packages('caret')
library(caret)

#install.packages('caretEnsemble')
library(caretEnsemble)
```

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

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

```
#install.packages('psych')
library(psych)
```

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

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

```
#install.packages('Amelia')
library(Amelia)
```

```
## Loading required package: Rcpp

## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.0, built: 2021-05-26)
## ## Copyright (C) 2005-2022 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
#install.packages('mice')
library(mice)
```

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

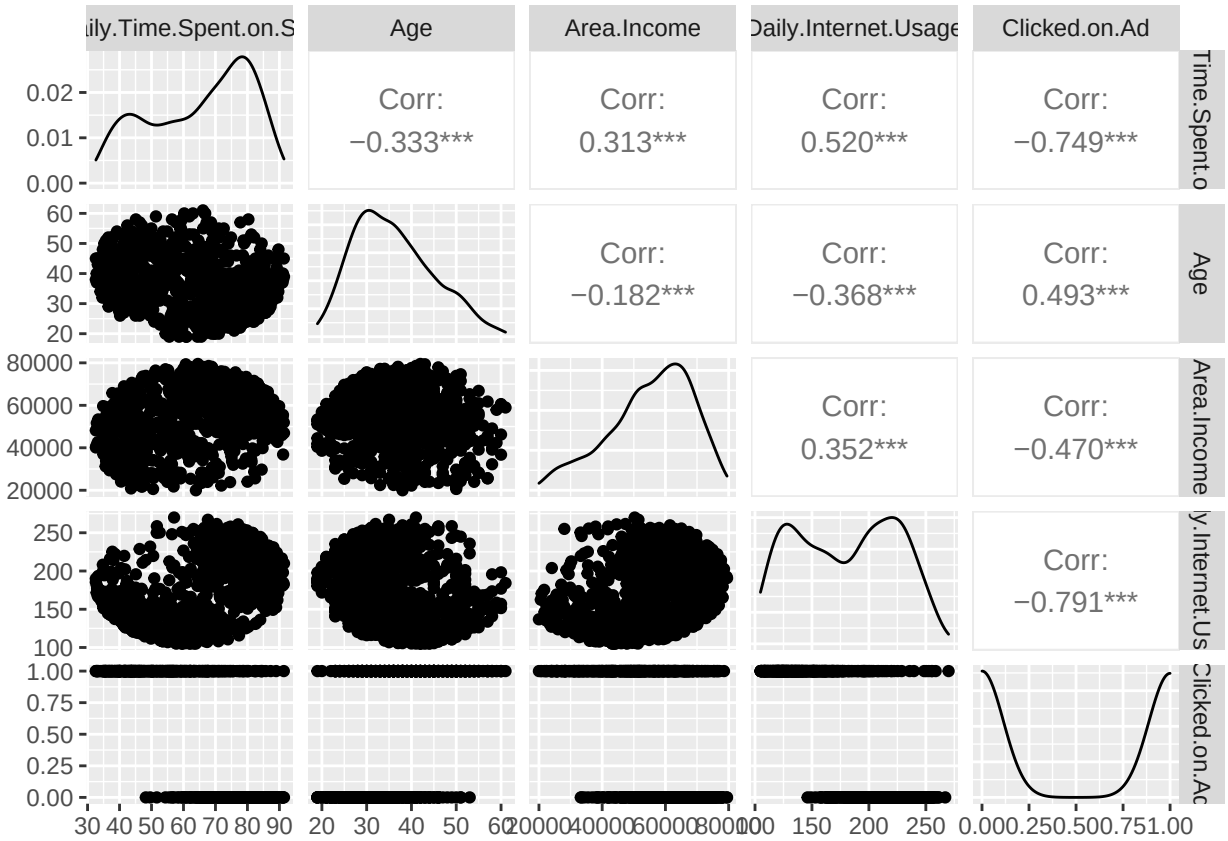
```
#install.packages('GGally')
library(GGally)
```

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

```r
#install.packages('rpart')
library(rpart)

#install.packages('randomForest')
library(randomForest)
```

```
## randomForest 4.7-1.1

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

##
## Attaching package: 'randomForest'

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

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

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

```r
# plotting a ggplot parwise summary
ggpairs(ad1)
```



Splitting the dataset

```
# Splitting data into training and test data sets

ad_train <- createDataPartition(y = ad1$Clicked.on.Ad,p = 0.75,list = FALSE)
training <- ad1[ad_train,]
testing <- ad1[-ad_train,]
```

Checking dimensions

```
# Checking percentage dimensions of the split

prop.table(table(ad1$Clicked.on.Ad)) * 100
```

```
##
##        0        1
## 50.45409 49.54591
```

```
prop.table(table(training$Clicked.on.Ad)) * 100
```

```
##
##        0        1
## 49.05914 50.94086
```

```
prop.table(table(testing$Clicked.on.Ad)) * 100
```

```
##
##        0        1
## 54.65587 45.34413
```

Previewing column names

```
# column names
print(colnames(ad1))
```

```
## [1] "Daily.Time.Spent.on.Site" "Age"
## [3] "Area.Income"              "Daily.Internet.Usage"
## [5] "Clicked.on.Ad"
```

```
#the first 6 records
head(ad1)
```

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

Splitting the dataset into train and test sets

```
# Comparing the outcome of the training and testing phase
# Creating objects x which holds the predictor variables and y which holds the response variables

# scaling our x variable
x = scale(training[,-5])
y = as.factor(training$Clicked.on.Ad)
```

Building the naive bayes model

```
# loading necessary libraries
library(e1071)
library(klaR)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

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

```
# building our model
model = train(x,y,'nb',trControl=trainControl(method='cv',number=10))
```

Model evaluation

```
# Predicting our testing set
#
library(caret)

#Predict <- predict(model,newdata = as.factor(testing))
Predict <- predict(model,newdata = testing)
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 2

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 3

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 4

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 5

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 6

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 7

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 8

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 9

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 10

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 11

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 12

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 13
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 14

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 16

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 17

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 18

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 19

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 20

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 21

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 22

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 23

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 24

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 25

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 26

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 27

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 28

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 29

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 30
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 31

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 32

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 33

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 34

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 35

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 36

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 37

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 38

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 39

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 40

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 41

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 42

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 47
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 48

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 49

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 50

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 51

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 52

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 53

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 54

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 55

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 56

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 57

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 58

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 59

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 60

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 61

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 62

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 63

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 64
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 65

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 66

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 67

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 68

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 69

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 70

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 71

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 72

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 73

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 74

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 75

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 76

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 77

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 78

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 79

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 80

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 81
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 82

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 83

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 84

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 85

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 86

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 87

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 88

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 89

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 90

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 91

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 92

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 93

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 94

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 95

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 96

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 97

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 98
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 99

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 100

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 101

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 102

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 103

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 104

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 105

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 106

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 107

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 108

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 109

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 110

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 111

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 112

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 113

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 114

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 115
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 116

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 117

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 118

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 119

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 120

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 121

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 122

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 123

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 124

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 125

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 126

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 127

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 128

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 129

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 130

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 131

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 132
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 133

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 134

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 135

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 136

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 137

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 138

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 139

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 140

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 141

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 142

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 143

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 144

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 145

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 146

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 147

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 148

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 149
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 150

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 151

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 152

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 153

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 154

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 155

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 156

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 157

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 158

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 159

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 160

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 161

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 162

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 163

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 164

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 165

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 166
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 167

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 168

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 169

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 170

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 171

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 172

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 173

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 174

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 175

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 176

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 177

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 178

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 179

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 180

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 181

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 182

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 183
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 184

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 185

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 186

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 187

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 188

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 189

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 190

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 191

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 192

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 193

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 194

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 195

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 196

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 197

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 198

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 199

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 200
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 201

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 202

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 203

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 204

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 205

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 206

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 207

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 208

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 209

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 210

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 211

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 212

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 213

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 214

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 215

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 216

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 217
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 218

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 219

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 220

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 221

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 222

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 223

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 224

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 225

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 226

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 227

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 228

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 229

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 230

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 231

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 232

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 233

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 234
```

```
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 235

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 236

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 237

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 238

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 239

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 240

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 241

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 242

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 243

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 244

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 245

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 246

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with
## observation 247
```

```
# Getting the confusion matrix to see accuracy value and other parameter values
co.matrix <-confusionMatrix(data = Predict, factor(testing$Clicked.on.Ad ))

# displaying the results
co.matrix
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   0   1
##          0 135 112
```

```
##          1   0   0
##
##                Accuracy : 0.5466
##                  95% CI : (0.4822, 0.6098)
##     No Information Rate : 0.5466
##     P-Value [Acc > NIR] : 0.5263
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.5466
##          Neg Pred Value :    NaN
##              Prevalence : 0.5466
##          Detection Rate : 0.5466
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

#Conclusion

All the models built had optimally good performances. The Multiple Linear Regression Model had an error of only 20%, Decision Tree Model had a mean squared error of 17%, Naive Bayes Model had an accuracy of 95%. Gender has the least influence on whether the ad is being clicked on or not. Age has a moderately high positive influence on an ad being clicked on, with a mean of about 35 years old. People with area income of above 50,000 units and spent over 150 units on daily internet usage clicked on the ads Area Income has a moderately high negative influence on an ad being clicked on. However since this data is skewed to the right, this could have an influence on this analysis. Daily internet usage and Daily time spent on the site has high negative correlations, this means that when these measurements increase, the chances of an ad being clicked go down. People that spent over 60 minutes on the site and with area income of over 40,000 clicked on the ads

Recommendation

The entrepreneur is advised to custom the advert to target this age group of about 35years old. This data is however skewed and hence could be causing this observation. The adverts should be set to pop at up in between the 55th minute and the 65th minutes. Adverts should target people with area income of above 50,000 units and spent over 150 units on daily internet usage. The adverts should target people who spent more time on the sites and used more daily internet. A more balanced data-set could lead to better results. We recommend the application of the Naive Bayes Model had a high accuracy of 95%.