# Final project

Melissa Kanyi

3/2/2020

## 1.1 Introduction

### 1.1.1 Defining the Question

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

### 1.1.2 The Context

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

### 1.1.3 Metrics of Success

Accuracy Score of 85% or above

### 1.1.4 Experimental Design Taken

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

### 1.1.5 Appropriateness of the Data

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

## 1.2 Installing & Loading necessary packages

```
#install.packages('tidyverse')
install.packages("dplyr")
```

```
## Installing package into '/home/melissa/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

```
library(data.table)
```

```
library(dplyr)
```

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

```
## 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
```

```
install.packages('tinytex')
```

```
## Installing package into '/home/melissa/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

```
tinytex::install_tinytex()
```

```
## Warning: Detected an existing tlmgr at /home/melissa/bin/tlmgr. It seems TeX
## Live has been installed (check tinytex::tinytex_root()). You are recommended
## to uninstall it, although TinyTeX should work well alongside another LaTeX
## distribution if a LaTeX document is compiled through tinytex::latexmk().
```

```
## TinyTeX installed to /home/melissa/.TinyTeX
```

```
## You may have to restart your system after installing TinyTeX to make sure ~/bin appears in your PATH
```

## 1.3 Loading the Data

```
library(readr)
```

```
advertising <- read_csv("~/Downloads/advertising.csv")
```

```
## Parsed with column specification:
## cols(
##   `Daily Time Spent on Site` = col_double(),
##   Age = col_double(),
##   `Area Income` = col_double(),
```

```
##   `Daily Internet Usage` = col_double(),
##   `Ad Topic Line` = col_character(),
##   City = col_character(),
##   Male = col_double(),
##   Country = col_character(),
##   Timestamp = col_datetime(format = ""),
##   `Clicked on Ad` = col_double()
## )
```

```
advertising
```

```
## # A tibble: 1,000 x 10
##    `Daily Time Spe~   Age `Area Income` `Daily Internet~ `Ad Topic Line` City
##               <dbl> <dbl>         <dbl>            <dbl> <chr>           <chr>
##  1             69.0    35        61834.             256. Cloned 5thgene~ Wrig~
##  2             80.2    31        68442.             194. Monitored nati~ West~
##  3             69.5    26        59786.             236. Organic bottom~ Davi~
##  4             74.2    29        54806.             246. Triple-buffere~ West~
##  5             68.4    35        73890.             226. Robust logisti~ Sout~
##  6             60.0    23        59762.             227. Sharable clien~ Jami~
##  7             88.9    33        53853.             208. Enhanced dedic~ Bran~
##  8             66      48        24593.             132. Reactive local~ Port~
##  9             74.5    30        68862              222. Configurable c~ West~
## 10             69.9    20        55642.             184. Mandatory homo~ Rami~
## # ... with 990 more rows, and 4 more variables: Male <dbl>, Country <chr>,
## #   Timestamp <dttm>, `Clicked on Ad` <dbl>
```

```
ads <- advertising
head(ads)
```

```
## # A tibble: 6 x 10
##   `Daily Time Spe~   Age `Area Income` `Daily Internet~ `Ad Topic Line` City
##              <dbl> <dbl>         <dbl>            <dbl> <chr>           <chr>
## 1             69.0    35        61834.             256. Cloned 5thgene~ Wrig~
## 2             80.2    31        68442.             194. Monitored nati~ West~
## 3             69.5    26        59786.             236. Organic bottom~ Davi~
## 4             74.2    29        54806.             246. Triple-buffere~ West~
## 5             68.4    35        73890.             226. Robust logisti~ Sout~
## 6             60.0    23        59762.             227. Sharable clien~ Jami~
## # ... with 4 more variables: Male <dbl>, Country <chr>, Timestamp <dttm>,
## #   `Clicked on Ad` <dbl>
```

## 1.4 Checking the Data

```
# Viewing the top of the data
head(ads)
```

```
## # A tibble: 6 x 10
##   `Daily Time Spe~   Age `Area Income` `Daily Internet~ `Ad Topic Line` City
##              <dbl> <dbl>         <dbl>            <dbl> <chr>           <chr>
```

```
## 1             69.0    35        61834.             256. Cloned 5thgene~ Wrig~
## 2             80.2    31        68442.             194. Monitored nati~ West~
## 3             69.5    26        59786.             236. Organic bottom~ Davi~
## 4             74.2    29        54806.             246. Triple-buffere~ West~
## 5             68.4    35        73890.             226. Robust logisti~ Sout~
## 6             60.0    23        59762.             227. Sharable clien~ Jami~
## # ... with 4 more variables: Male <dbl>, Country <chr>, Timestamp <dttm>,
## #   `Clicked on Ad` <dbl>
```

```
# viewing the bottom of the data
tail(ads)
```

```
## # A tibble: 6 x 10
##   `Daily Time Spe~   Age `Area Income` `Daily Internet~ `Ad Topic Line` City
##              <dbl> <dbl>         <dbl>            <dbl> <chr>           <chr>
## 1             43.7    28        63127.             173. Front-line bif~ Nich~
## 2             73.0    30        71385.             209. Fundamental mo~ Duff~
## 3             51.3    45        67782.             134. Grass-roots co~ New ~
## 4             51.6    51        42416.             120. Expanded intan~ Sout~
## 5             55.6    19        41921.             188. Proactive band~ West~
## 6             45.0    26        29876.             178. Virtual 5thgen~ Ronn~
## # ... with 4 more variables: Male <dbl>, Country <chr>, Timestamp <dttm>,
## #   `Clicked on Ad` <dbl>
```

```
# checking the number of rows and columns
dim(ads)
```

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

There are 1000 rows and 10 columns.

```
# checking the types of attributes
sapply(ads, class)
```

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

```
## $Male
## [1] "numeric"
##
## $Country
## [1] "character"
##
## $Timestamp
## [1] "POSIXct" "POSIXt"
##
## $`Clicked on Ad`
## [1] "numeric"
```

```
# checking the summary statistics of the data
summary(ads)
```

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

```
# summary information of the dataset
glimpse(ads)
```

```
## Observations: 1,000
## Variables: 10
## $ `Daily Time Spent on Site` <dbl> 68.95, 80.23, 69.47, 74.15, 68.37, 59.99...
## $ Age                        <dbl> 35, 31, 26, 29, 35, 23, 33, 48, 30, 20, ...
## $ `Area Income`              <dbl> 61833.90, 68441.85, 59785.94, 54806.18, ...
## $ `Daily Internet Usage`     <dbl> 256.09, 193.77, 236.50, 245.89, 225.58, ...
## $ `Ad Topic Line`            <chr> "Cloned 5thgeneration orchestration", "M...
## $ City                       <chr> "Wrightburgh", "West Jodi", "Davidton", ...
## $ Male                       <dbl> 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0...
## $ Country                    <chr> "Tunisia", "Nauru", "San Marino", "Italy...
## $ Timestamp                  <dttm> 2016-03-27 00:53:11, 2016-04-04 01:39:0...
## $ `Clicked on Ad`            <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0...
```

## 1.5 Data Cleaning

### Missing Values

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

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

There are no missing values in the dataset from the output above

### Duplicates

```
# checking for duplicated rows
dr <- unique(ads)
dr
```

```
## # A tibble: 1,000 x 10
##    `Daily Time Spe~   Age `Area Income` `Daily Internet~ `Ad Topic Line` City
##               <dbl> <dbl>         <dbl>            <dbl> <chr>           <chr>
##  1             69.0    35        61834.             256. Cloned 5thgene~ Wrig~
##  2             80.2    31        68442.             194. Monitored nati~ West~
##  3             69.5    26        59786.             236. Organic bottom~ Davi~
##  4             74.2    29        54806.             246. Triple-buffere~ West~
##  5             68.4    35        73890.             226. Robust logisti~ Sout~
##  6             60.0    23        59762.             227. Sharable clien~ Jami~
##  7             88.9    33        53853.             208. Enhanced dedic~ Bran~
##  8             66      48        24593.             132. Reactive local~ Port~
##  9             74.5    30        68862              222. Configurable c~ West~
## 10             69.9    20        55642.             184. Mandatory homo~ Rami~
## # ... with 990 more rows, and 4 more variables: Male <dbl>, Country <chr>,
## #   Timestamp <dttm>, `Clicked on Ad` <dbl>
```

There are no duplicated rows on our dataset

### Fixing spaces in the column names

```
# checking the column names
names(ads)
```

```
##  [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"
```

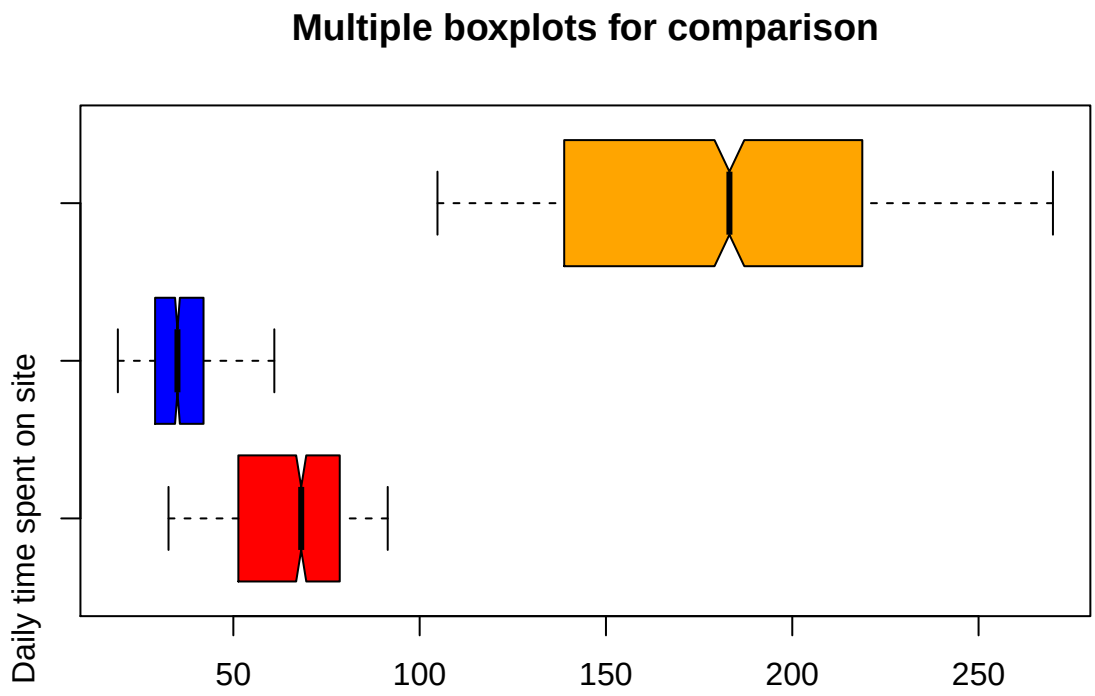
```
# Replacing spaces in the column names with an underscore
names(ads) <- gsub(" ", "_", names(ads))
```

```
# confirming the column names have changed
names(ads)
```

```
##  [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"
```

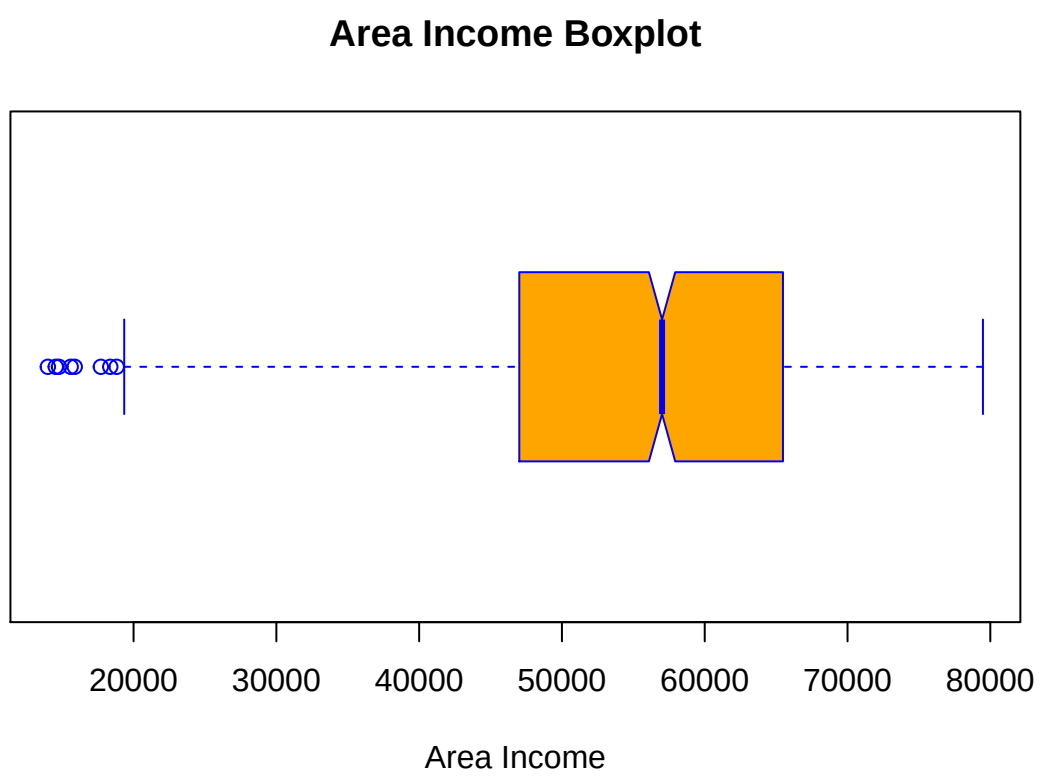
## Outliers

```
# Using boxplot to check for observations far away from other points
# Using all the three double type columns: while specifying each
Daily_Time_Spent_on_Site <- ads$Daily_Time_Spent_on_Site
Age <- ads$Age
Daily_Internet_Usage <- ads$Daily_Internet_Usage

boxplot(Daily_Time_Spent_on_Site,Age,Daily_Internet_Usage,
        main = "Multiple boxplots for comparison",
        at = c(1,2,3),
        names = c("Daily time spent on site","Age","Daily Internet Usage"),
        col = c("red","blue","orange"),
        border = "black",
        horizontal = TRUE,
        notch = TRUE
        )
```




There are no outliers in the three features plotted

```
# Boxplot for the Area Income
# labeling the title
# labeling the x axis
# Specifying color options

boxplot(ads$Area_Income,
        main = "Area Income Boxplot",
        xlab = "Area Income",
        col = "orange",
        border = "blue",
        horizontal = TRUE,
        notch = TRUE
        )
```




There are a few outliers on the first quartile of the Area income boxplot.

## 1.6 Visualizations

### Univariate Analysis

```
# getting the age range on the target group
age_range <- range(ads$Age)
age_range
```

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

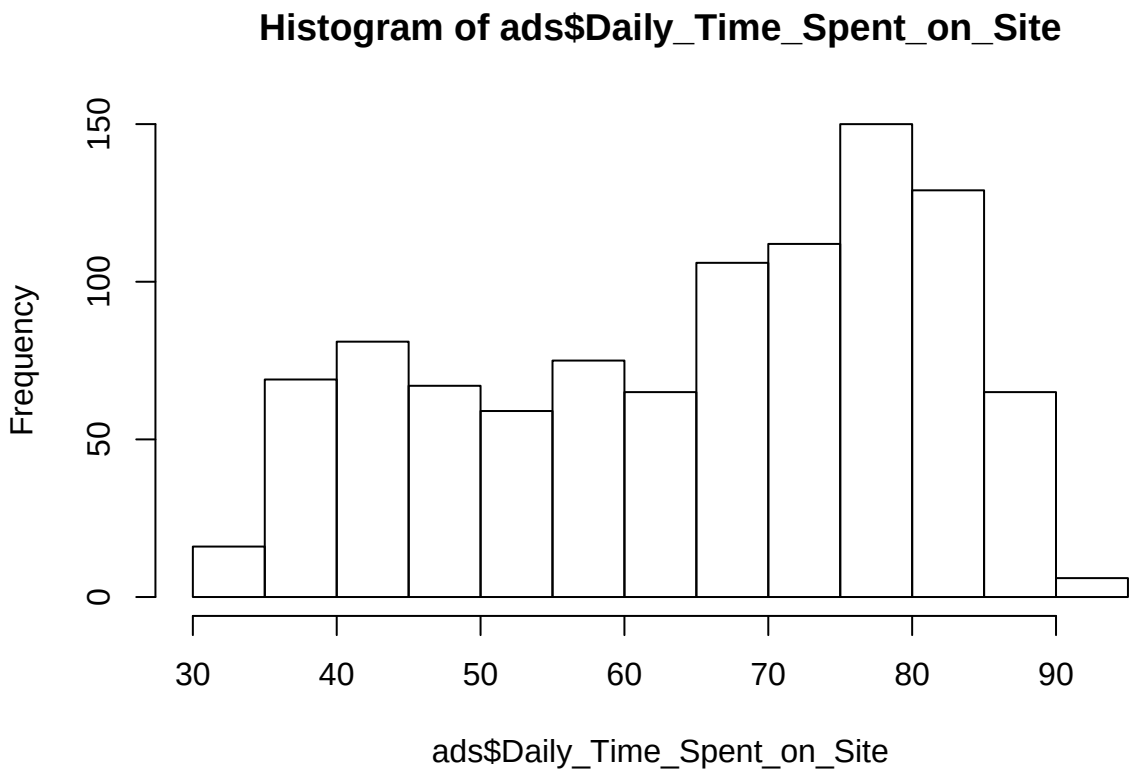
- The age ranges from 19-61

```
# getting the range of the area income
income_range <- range(ads$Area_Income)
income_range
```

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

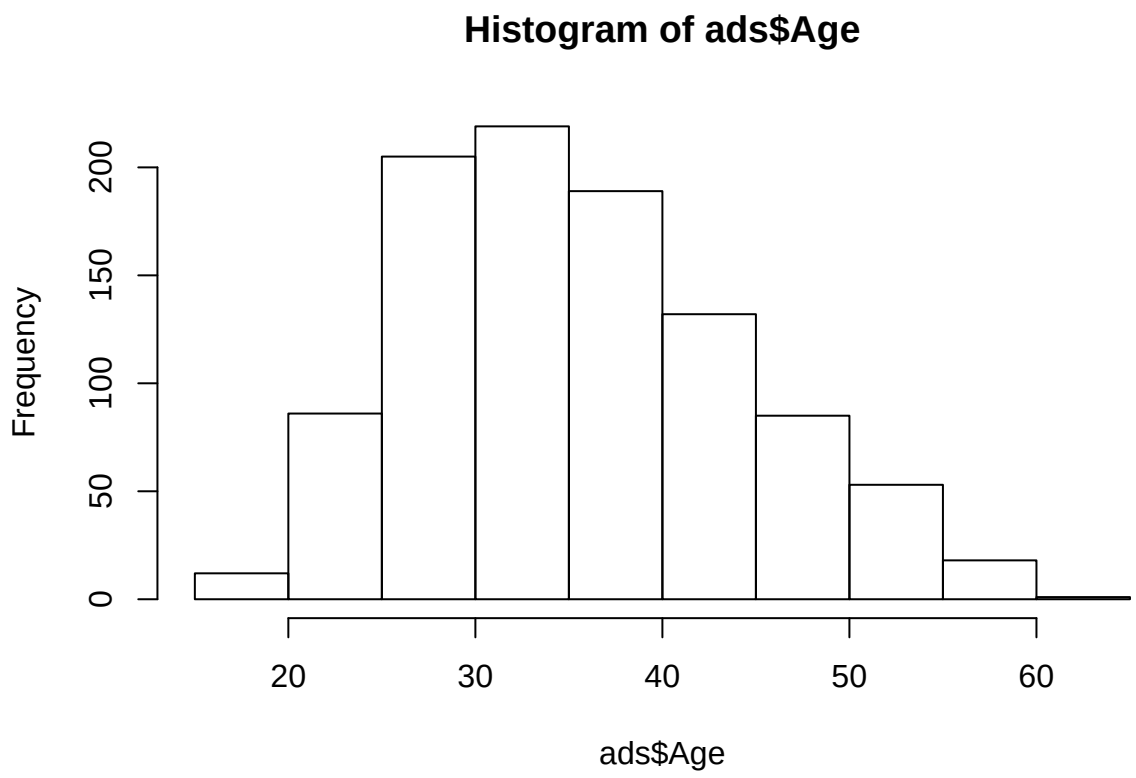
- The area income ranges from 13996-79484

```r
hist(ads$Daily_Time_Spent_on_Site)
```

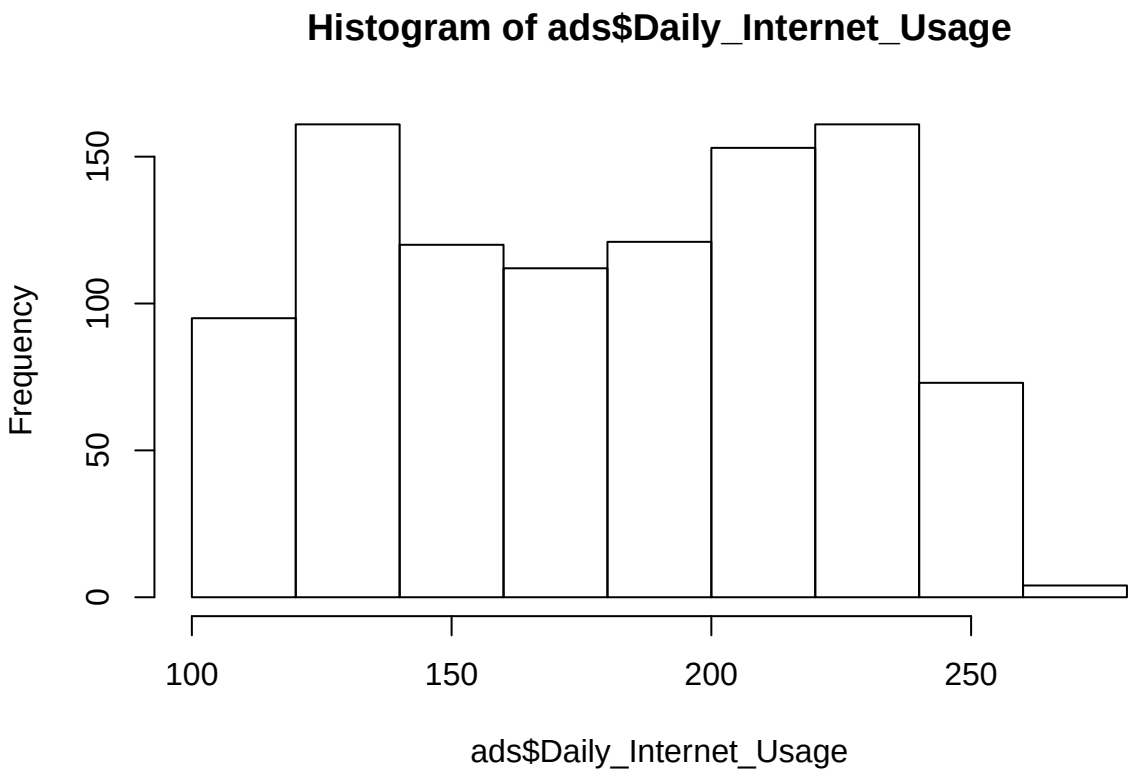



- Histogram is skewed to the left showing that most people spent their 65-85 of their time on the site .

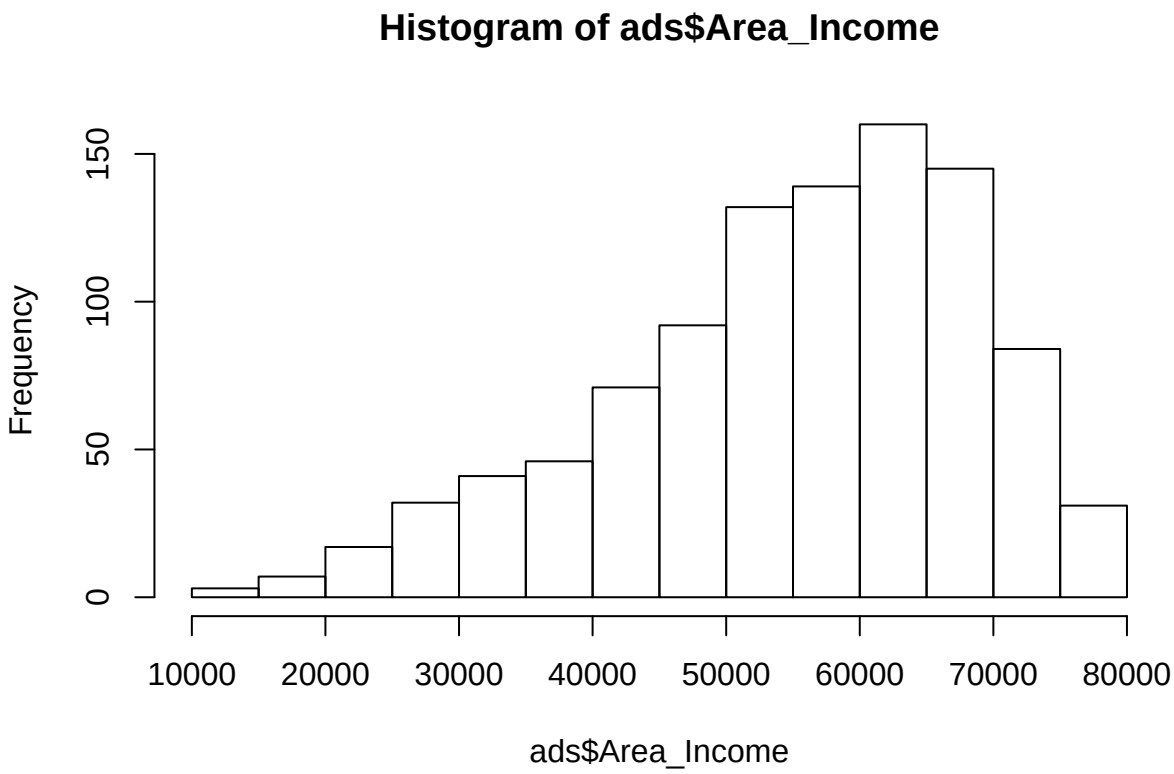
```r
hist(ads$Age)
```




```
hist(ads$Daily_Internet_Usage)
```

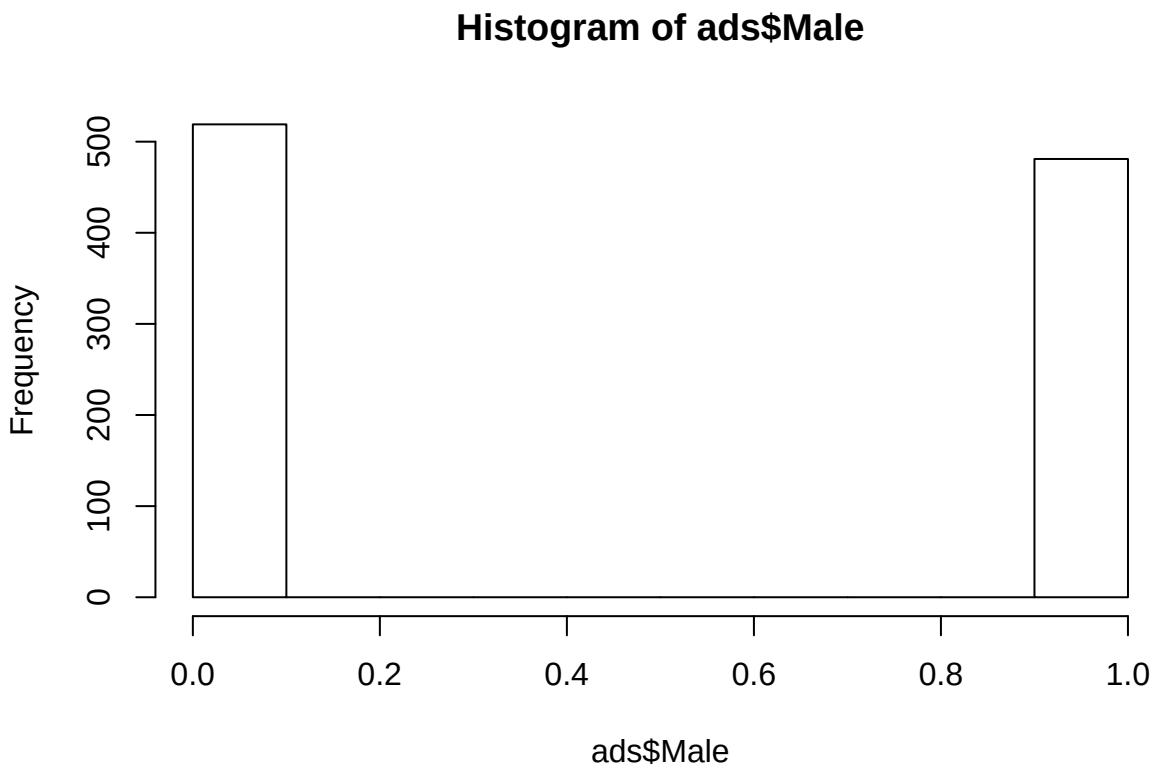



```
hist(ads$Area_Income)
```




- Most of the people earned between 5,000 and 7,000

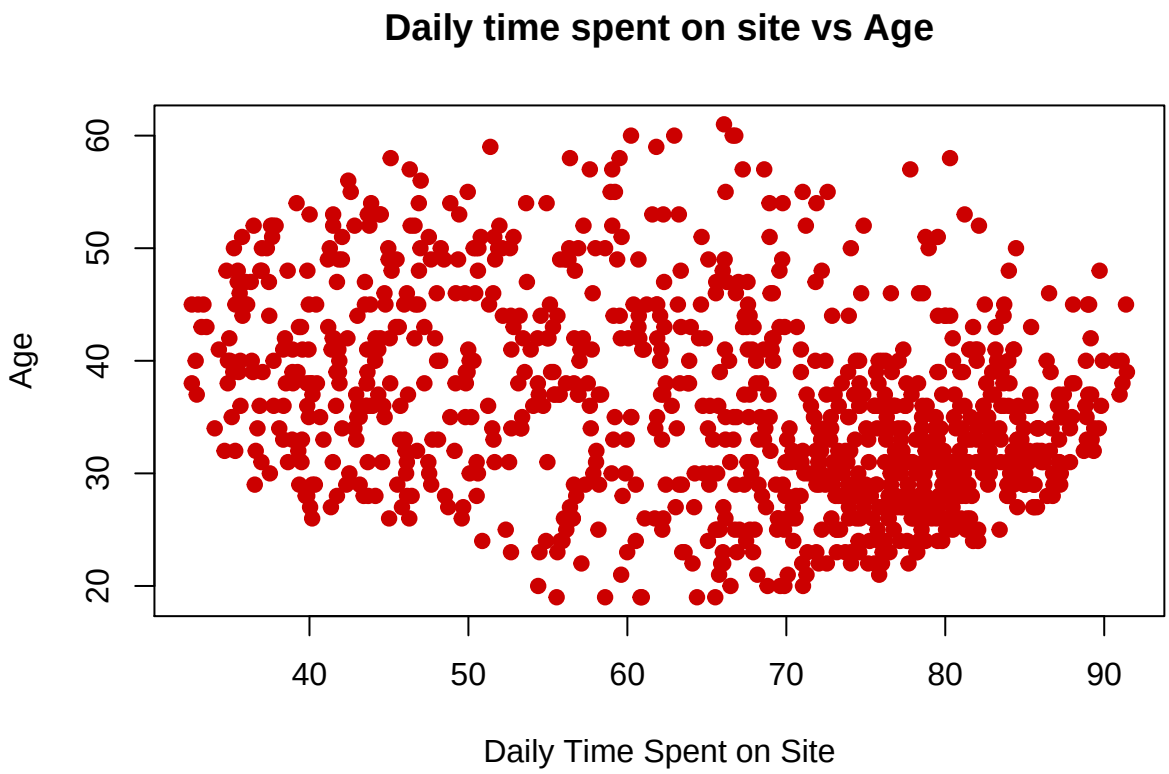
```
hist(ads$Male)
```




- There are more females than males.

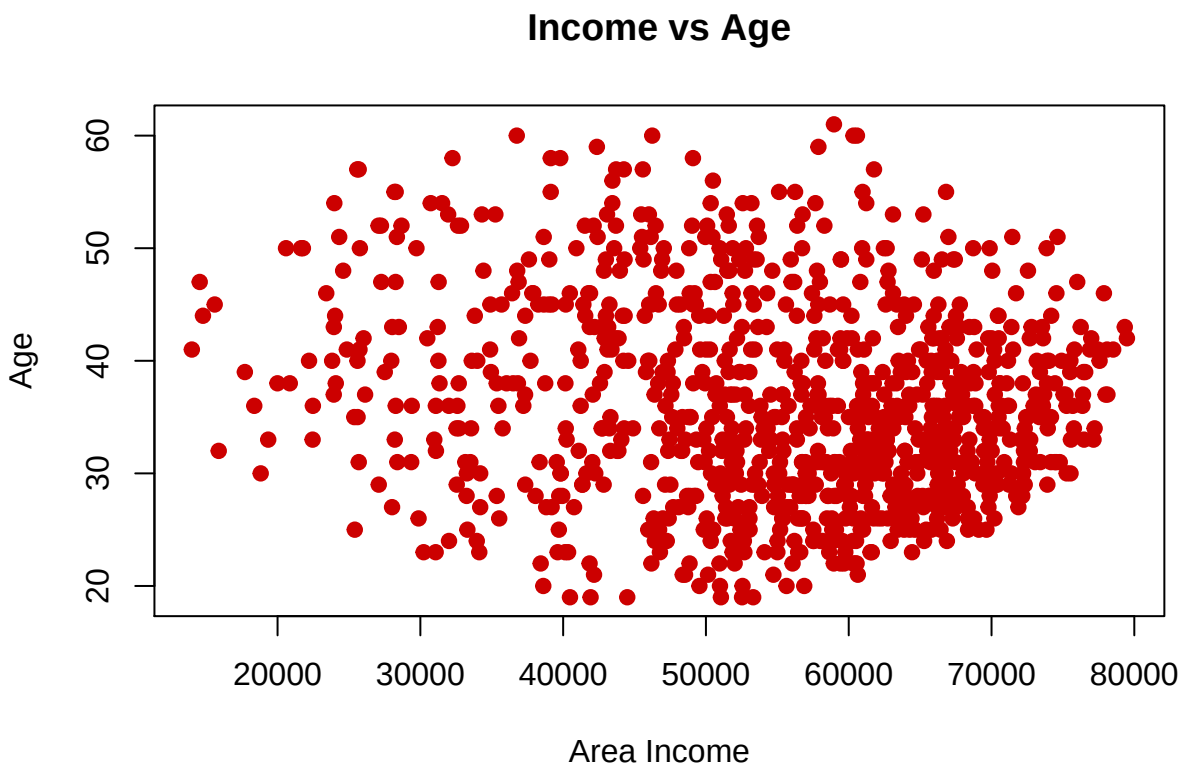
### Bivariate Analysis Visualizations

```
plot(ads$Daily_Time_Spent_on_Site, ads$Age,
     col = "#cc0000",
     pch = 19,
     main = "Daily time spent on site vs Age",
     xlab = "Daily Time Spent on Site",
     ylab = "Age"
     )
```



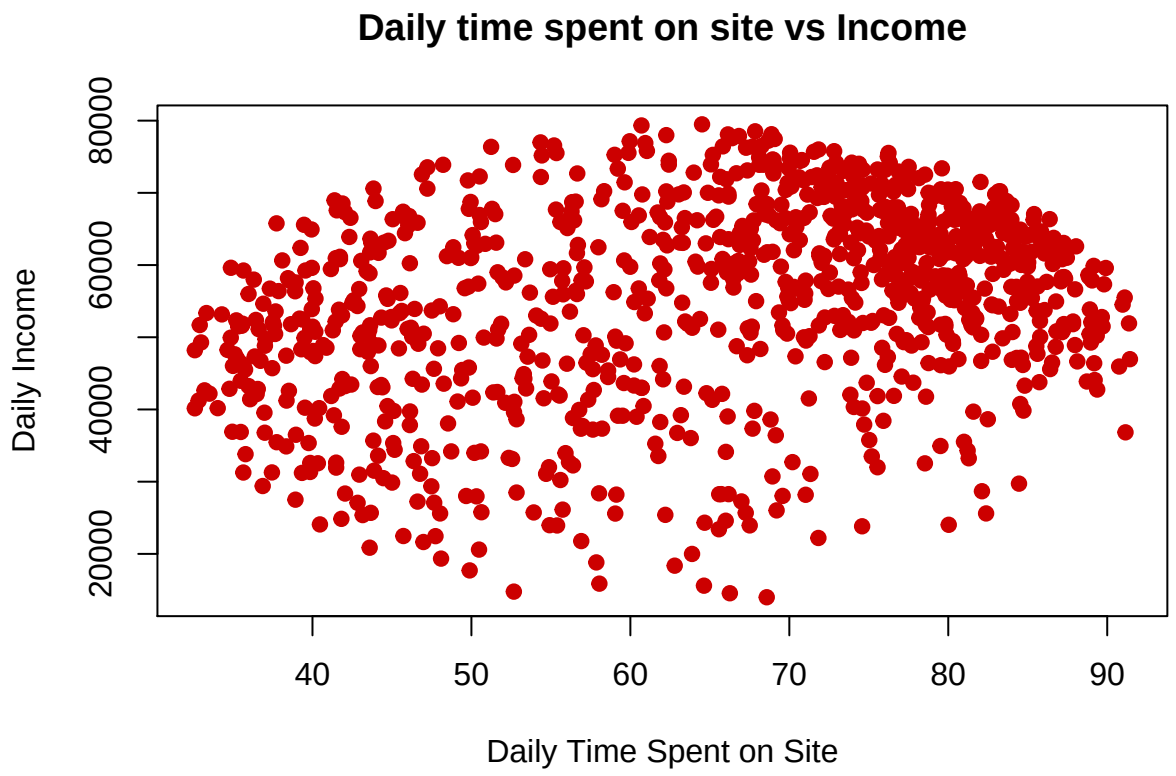

- This slightly shows that people below the age of 40 spend more time on the site.

```
plot(ads$Area_Income, ads$Age,
     col = "#cc0000",
     pch = 19,
     main = "Income vs Age",
     xlab = "Area Income",
     ylab = "Age"
     )
```

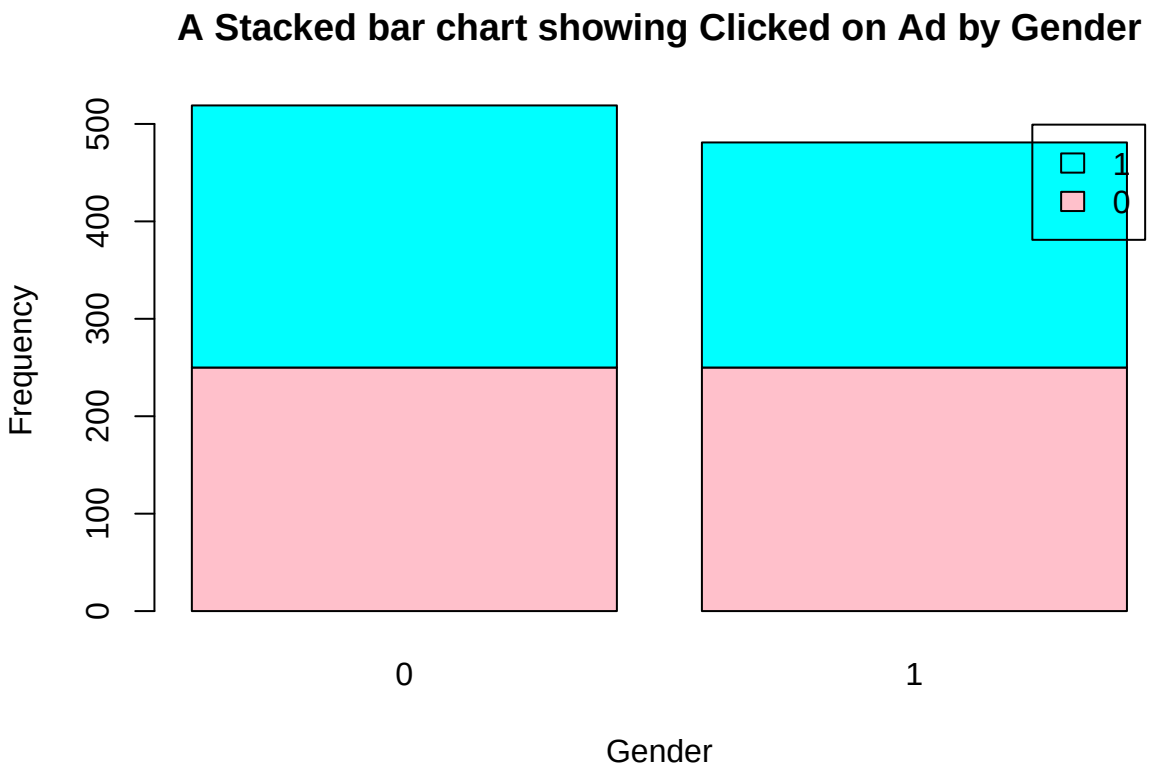



- This just shows that regardless of age, most of the people earn more than 50000.

```
plot(ads$Daily_Time_Spent_on_Site, ads$Area_Income,
     col = "#cc0000",
     pch = 19,
     main = "Daily time spent on site vs Income",
     xlab = "Daily Time Spent on Site",
     ylab = "Daily Income"
)
```

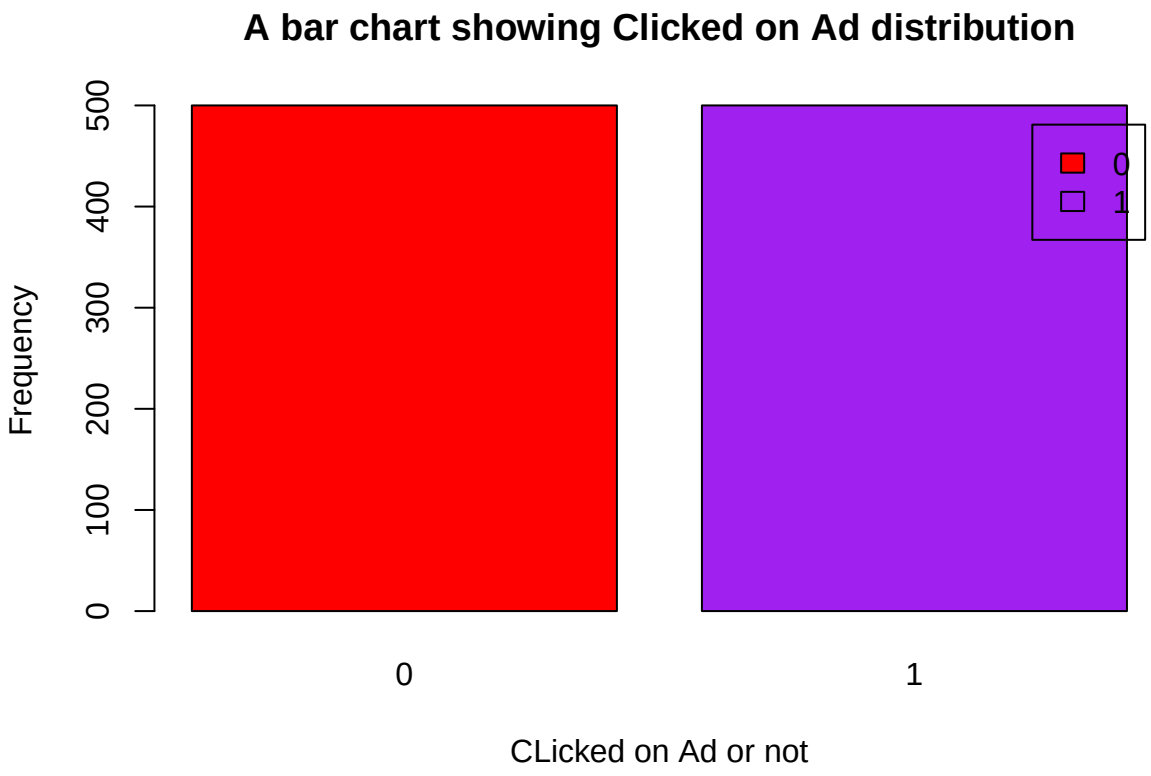



```
# stacked bar chart
# giving a title to the chart
# labeling the x and y axis
# Setting the color options
# Creating a legend for easier reference

count <- table(ads$Clicked_on_Ad, ads$Male)
barplot(count,
        main = "A Stacked bar chart showing Clicked on Ad by Gender",
        xlab = "Gender",
        ylab = "Frequency",
        col = c("pink","cyan"),
        legend = rownames(count)
        )
```




- There are slightly more females than males in the dataset. - More females clicked on Ad compared to males.

```
count <- table(ads$Clicked_on_Ad)
barplot(count,
        main = "A bar chart showing Clicked on Ad distribution",
        xlab = "CLicked on Ad or not",
        ylab = "Frequency",
        col = c("red","purple"),
        legend = rownames(count))
```




- The data is balanced since the number of people who clicked on Ad were equal to those who did not.

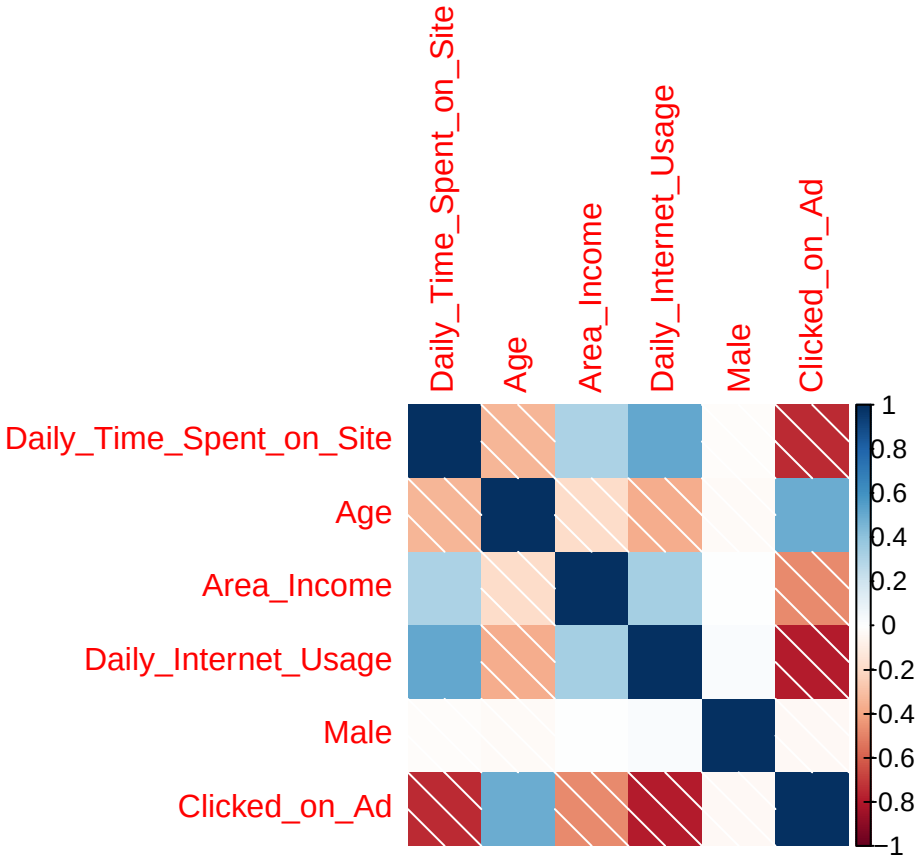
```
#installing the necessary packages and the libraries.
install.packages("corrplot")
```

```
## Installing package into '/home/melissa/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
#`getting the numeric values of our dataaset
num_data = ads[, sapply(ads, is.numeric)]

#plotting the numeric values.
corrplot(cor(num_data), method = 'shade')
```




- We can see that the daily time spent on the site and the daily internet usage have a slightly high correlation. - Whether you're male or not has no correlation with all the other varirables.

```
cor(num_data, use = "complete.obs", method = "pearson")
```

```
##                          Daily_Time_Spent_on_Site         Age  Area_Income
## Daily_Time_Spent_on_Site              1.00000000 -0.33151334  0.310954413
## Age                                  -0.33151334  1.00000000 -0.182604955
## Area_Income                           0.31095441 -0.18260496  1.000000000
## Daily_Internet_Usage                  0.51865848 -0.36720856  0.337495533
## Male                                 -0.01895085 -0.02104406  0.001322359
## Clicked_on_Ad                        -0.74811656  0.49253127 -0.476254628
##                          Daily_Internet_Usage         Male Clicked_on_Ad
## Daily_Time_Spent_on_Site           0.51865848 -0.018950855   -0.74811656
## Age                               -0.36720856 -0.021044064    0.49253127
## Area_Income                        0.33749553  0.001322359   -0.47625463
## Daily_Internet_Usage               1.00000000  0.028012326   -0.78653918
## Male                               0.02801233  1.000000000   -0.03802747
## Clicked_on_Ad                     -0.78653918 -0.038027466    1.00000000
```

# Modeling

```
# loading the rpart library for modeling
library(rpart)
```

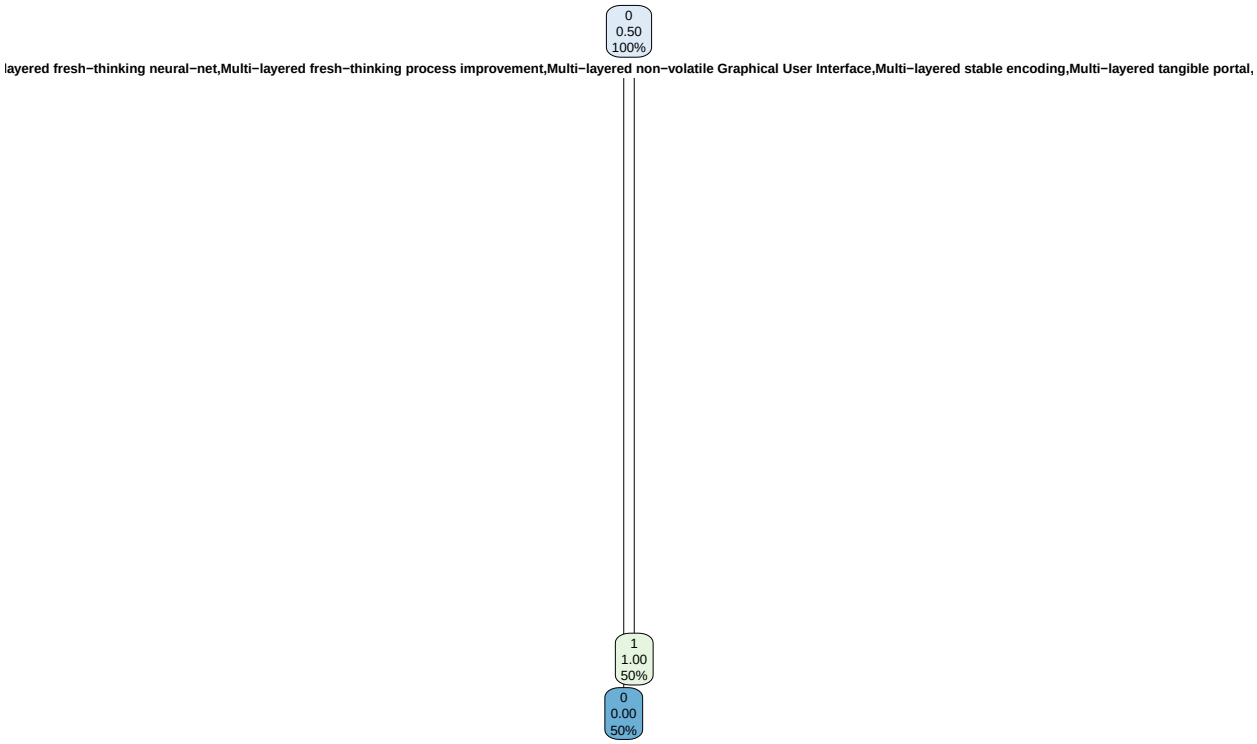
```
# Using decision tree
# Fitting the model
# Specifying the target and predictor variables

m <- rpart(Clicked_on_Ad ~ . ,
    data = ads,
    method = "class")
```

```
install.packages('rpart.plot')
```

```
## Installing package into '/home/melissa/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

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

```
# Plotting the decision tree model
rpart.plot(m)
```




```
# Making predictions
# Printing the confusion matrix

p <- predict(m, ads, type ="class")
table(p, ads$Clicked_on_Ad)
```

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

```
# Printing the Accuracy

mean(ads$Clicked_on_Ad == p)
```

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

- The model accuracy is 100%
- This is a good model to make predictions.
- We will evaluate this model or challenge it using another model.

# Challenging the Solution

## Random Forest Classifier

```
install.packages("lattice")
```

```
## Installing package into '/home/melissa/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

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

```
## Installing package into '/home/melissa/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

```
library(caret)
```

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

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

```
# Training the model
# Setting seed for randomness

set.seed(12)
model <- train(Clicked_on_Ad ~. ,
               data = ads,
               method = "ranger")
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

```
## Growing trees.. Progress: 93%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 82%. Estimated remaining time: 6 seconds.
## Growing trees.. Progress: 86%. Estimated remaining time: 4 seconds.
## Growing trees.. Progress: 81%. Estimated remaining time: 7 seconds.
## Growing trees.. Progress: 89%. Estimated remaining time: 3 seconds.
## Growing trees.. Progress: 83%. Estimated remaining time: 6 seconds.
## Growing trees.. Progress: 88%. Estimated remaining time: 4 seconds.
## Growing trees.. Progress: 73%. Estimated remaining time: 11 seconds.
## Growing trees.. Progress: 92%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 86%. Estimated remaining time: 5 seconds.
## Growing trees.. Progress: 89%. Estimated remaining time: 3 seconds.
## Growing trees.. Progress: 77%. Estimated remaining time: 9 seconds.
## Growing trees.. Progress: 89%. Estimated remaining time: 3 seconds.
## Growing trees.. Progress: 98%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 83%. Estimated remaining time: 6 seconds.
## Growing trees.. Progress: 93%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 82%. Estimated remaining time: 6 seconds.
## Growing trees.. Progress: 97%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 88%. Estimated remaining time: 4 seconds.
## Growing trees.. Progress: 99%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 89%. Estimated remaining time: 3 seconds.
## Growing trees.. Progress: 86%. Estimated remaining time: 4 seconds.
## Growing trees.. Progress: 82%. Estimated remaining time: 6 seconds.
## Growing trees.. Progress: 94%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 91%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 79%. Estimated remaining time: 8 seconds.
## Growing trees.. Progress: 93%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 99%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 84%. Estimated remaining time: 5 seconds.
## Growing trees.. Progress: 93%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 80%. Estimated remaining time: 7 seconds.
## Growing trees.. Progress: 90%. Estimated remaining time: 3 seconds.
## Growing trees.. Progress: 100%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 97%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 89%. Estimated remaining time: 3 seconds.
## Growing trees.. Progress: 100%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 98%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 94%. Estimated remaining time: 1 seconds.
## Growing trees.. Progress: 95%. Estimated remaining time: 1 seconds.
## Growing trees.. Progress: 88%. Estimated remaining time: 4 seconds.
```

```
# Printing the model
model
```

```
## Random Forest
##
## 1000 samples
##    9 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 1000, 1000, 1000, 1000, 1000, 1000, ...
## Resampling results across tuning parameters:
##
```

```
##   mtry  splitrule   RMSE       Rsquared   MAE
##      2  variance    0.4976364  0.5401973  0.49751040
##      2  extratrees  0.4987522  0.4177158  0.49863035
##     66  variance    0.1969787  0.8583809  0.12509945
##     66  extratrees  0.2083152  0.8567880  0.15114435
##   2209  variance    0.1917676  0.8525418  0.05825788
##   2209  extratrees  0.1705530  0.8830922  0.05948088
##
## Tuning parameter 'min.node.size' was held constant at a value of 5
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were mtry = 2209, splitrule =
##  extratrees and min.node.size = 5.
```

## Decision Trees

```
install.packages("lattice")
```

```
## Installing package into '/home/melissa/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)
```

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

```
## Installing package into '/home/melissa/R/x86_64-pc-linux-gnu-library/3.4'
## (as 'lib' is unspecified)

## Warning in download.file(url, destfile, method, mode = "wb", ...): downloaded
## length 147456 != reported length 3031461

## Warning in download.file(url, destfile, method, mode = "wb", ...): URL 'https://
## cloud.r-project.org/src/contrib/ggplot2_3.3.0.tar.gz': status was 'Failure when
## receiving data from the peer'

## Error in download.file(url, destfile, method, mode = "wb", ...) :
##   download from 'https://cloud.r-project.org/src/contrib/ggplot2_3.3.0.tar.gz' failed

## Warning in download.packages(pkgs, destdir = tmpd, available = available, :
## download of package 'ggplot2' failed
```

```
library(caret)
```

```
## Example
# ---
# Decision Tree Code Example 1.2
# ---
#
set.seed(12)
model <- train(Clicked_on_Ad ~ .,
               data = ads,
               method = "ranger")
```

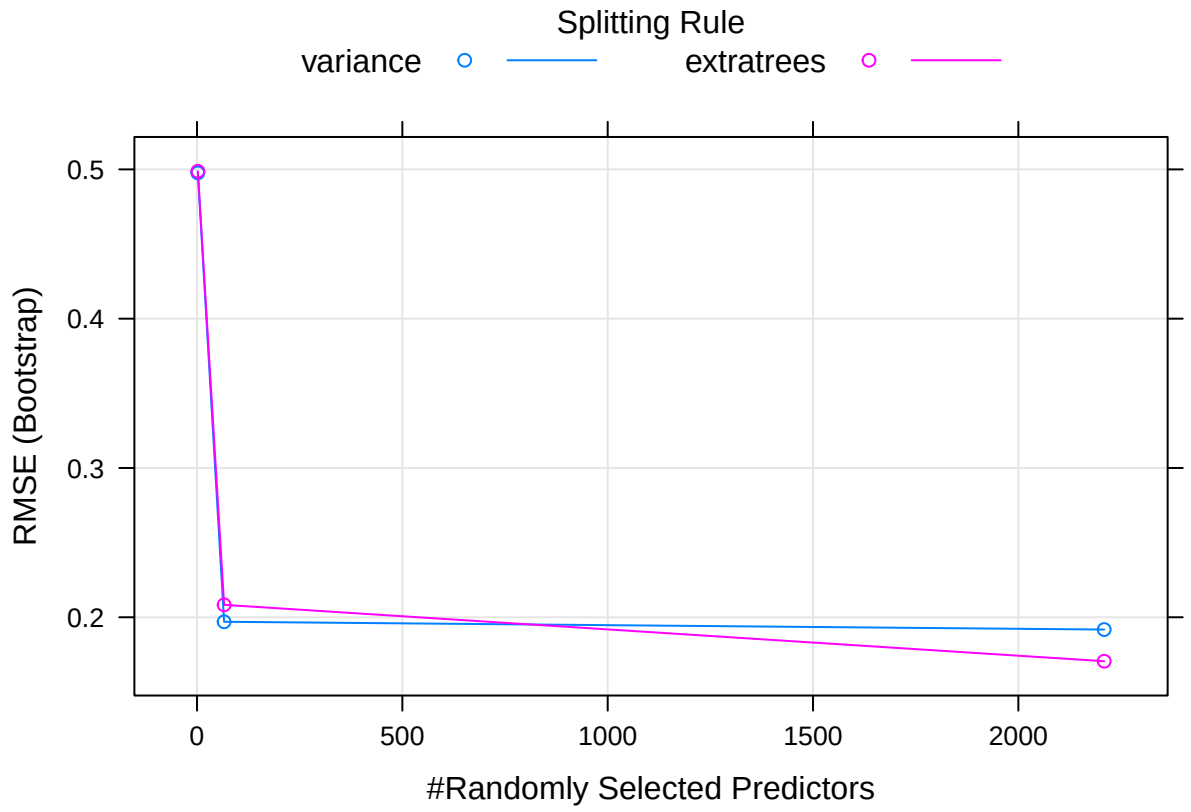
```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

```
## Growing trees.. Progress: 99%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 99%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 97%. Estimated remaining time: 1 seconds.
## Growing trees.. Progress: 100%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 100%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 93%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 100%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 100%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 88%. Estimated remaining time: 4 seconds.
```

```
model
```

```
## Random Forest
##
## 1000 samples
##    9 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 1000, 1000, 1000, 1000, 1000, 1000, ...
## Resampling results across tuning parameters:
##
##   mtry  splitrule   RMSE       Rsquared   MAE
##      2  variance    0.4976364  0.5401973  0.49751040
##      2  extratrees  0.4987522  0.4177158  0.49863035
##     66  variance    0.1969787  0.8583809  0.12509945
##     66  extratrees  0.2083152  0.8567880  0.15114435
##   2209  variance    0.1917676  0.8525418  0.05825788
##   2209  extratrees  0.1705530  0.8830922  0.05948088
##
## Tuning parameter 'min.node.size' was held constant at a value of 5
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were mtry = 2209, splitrule =
##  extratrees and min.node.size = 5.
```

```
plot(model)
```




```
# Training the model
model <- train(Clicked_on_Ad ~ .,
               data = ads,
               method = "ranger",
               tuneLength = 5)
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.

## Growing trees.. Progress: 99%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 98%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 97%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 98%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 100%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 98%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 100%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 100%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 89%. Estimated remaining time: 3 seconds.
```

```
set.seed(42)
myGrid <- expand.grid(mtry = c(5, 10, 20, 40, 60),
                      splitrule = c("gini", "extratrees"),
                      min.node.size = 10)
model <- train(Clicked_on_Ad ~ .,
```

```
               data = ads,
               method = "ranger",
               tuneGrid = myGrid,
               trControl = trainControl(method = "cv",
                                        number = 5,
                                        verboseIter = FALSE))
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

```
## Warning: model fit failed for Fold1: mtry= 5, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.
```

```
## Warning: model fit failed for Fold1: mtry=10, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.
```

```
## Warning: model fit failed for Fold1: mtry=20, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.
```

```
## Warning: model fit failed for Fold1: mtry=40, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.
```

```
## Warning: model fit failed for Fold1: mtry=60, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.
```

```
## Warning: model fit failed for Fold2: mtry= 5, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.
```

```
## Warning: model fit failed for Fold2: mtry=10, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.
```

```
## Warning: model fit failed for Fold2: mtry=20, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.
```

```
## Warning: model fit failed for Fold2: mtry=40, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.
```

```
## Warning: model fit failed for Fold2: mtry=60, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.
```

```
## Warning: model fit failed for Fold3: mtry= 5, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.
```

```
## Warning: model fit failed for Fold3: mtry=10, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.
```

```
## Warning: model fit failed for Fold3: mtry=20, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.
```

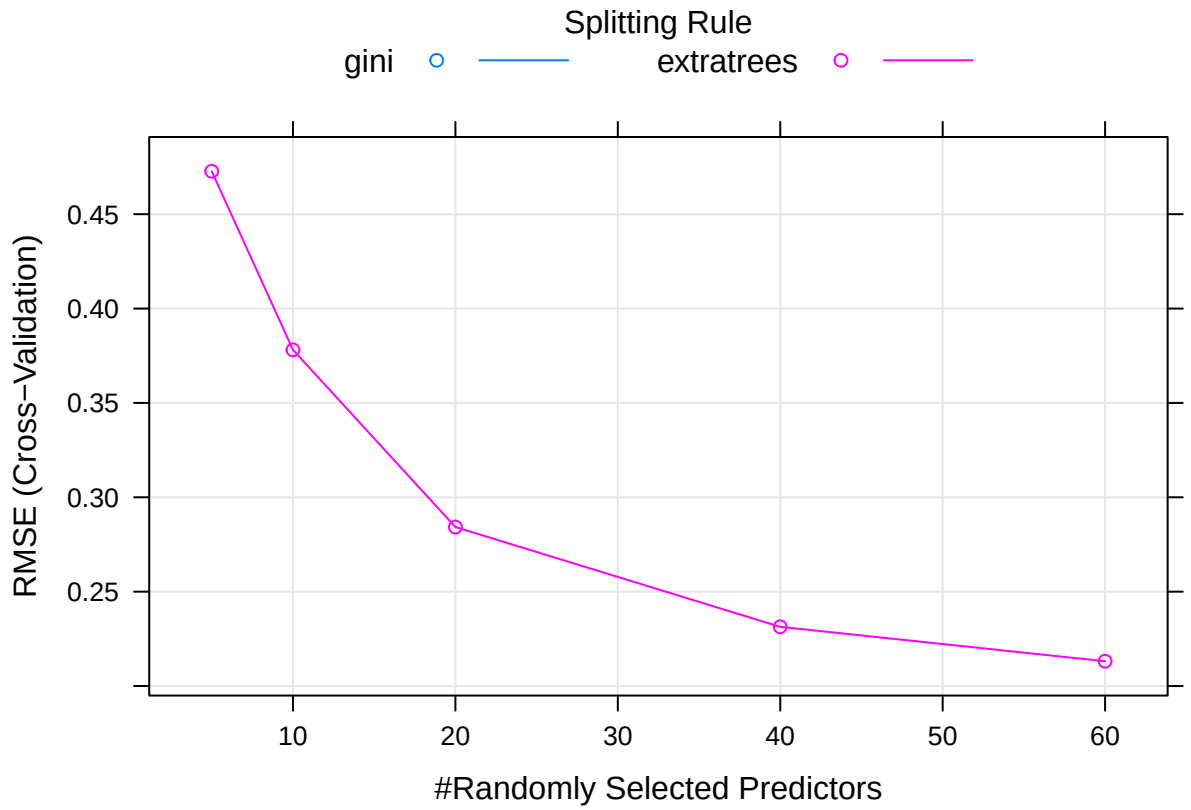
```
## Warning: model fit failed for Fold3: mtry=40, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.

## Warning: model fit failed for Fold3: mtry=60, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.

## Warning: model fit failed for Fold4: mtry= 5, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.

## Warning: model fit failed for Fold4: mtry=10, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.

## Warning: model fit failed for Fold4: mtry=20, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.

## Warning: model fit failed for Fold4: mtry=40, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.

## Warning: model fit failed for Fold4: mtry=60, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.

## Warning: model fit failed for Fold5: mtry= 5, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.

## Warning: model fit failed for Fold5: mtry=10, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.

## Warning: model fit failed for Fold5: mtry=20, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.

## Warning: model fit failed for Fold5: mtry=40, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.

## Warning: model fit failed for Fold5: mtry=60, splitrule=gini, min.node.size=10 Error in ranger::rang
##   Error: Gini splitrule applicable to classification data only.

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.

## Warning in train.default(x, y, weights = w, ...): missing values found in
## aggregated results
```

```
# Printing the model
model
```

```
## Random Forest
##
## 1000 samples
##    9 predictor
##
```

```
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 800, 800, 800, 800, 800
## Resampling results across tuning parameters:
##
##   mtry  splitrule   RMSE       Rsquared   MAE
##    5    gini              NaN        NaN        NaN
##    5    extratrees  0.4727430  0.7089443  0.4724141
##   10    gini              NaN        NaN        NaN
##   10    extratrees  0.3781414  0.7723433  0.3716844
##   20    gini              NaN        NaN        NaN
##   20    extratrees  0.2842453  0.8177107  0.2605211
##   40    gini              NaN        NaN        NaN
##   40    extratrees  0.2313547  0.8426705  0.1872970
##   60    gini              NaN        NaN        NaN
##   60    extratrees  0.2131213  0.8533135  0.1587107
##
## Tuning parameter 'min.node.size' was held constant at a value of 10
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were mtry = 60, splitrule = extratrees
##  and min.node.size = 10.
```

```
# Plotting the model
plot(model)
```




# Conclusion Random Forest Classifier is the best model among the the two with an accuracy of 100% It

is also better because it is a bagging method and uses many trees compared to decision tree classifier which uses only one tree.