IP WEEK 12

IYLINE CHUMO

26/08/2021

```
library(tinytex)
```

Defining The Question

To identify which individuals are most likely to click on an online cryptography course advert.

Metric of Success

Our project will be considered successful if we are able to effectively perform EDA to determine the individuals who are most likely to click the ads.

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 our services as Data Scientists Consultants identify which individuals are most likely to click on her ads.

Experimental Design

- Loading the dataset
- Performing data cleaning
- Exploratory Data Analysis
- Conclusion and recommendation

Loading the Dataset

```
data <- read.csv('http://bit.ly/IPAdvertisingData')

head(data)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    68.95  35    61833.90               256.09
## 2                    80.23  31    68441.85               193.77
## 3                    69.47  26    59785.94               236.50
## 4                    74.15  29    54806.18               245.89
## 5                    68.37  35    73889.99               225.58
## 6                    59.99  23    59761.56               226.74
##                          Ad.Topic.Line           City Male    Country
## 1    Cloned 5thgeneration orchestration    Wrightburgh    0    Tunisia
## 2    Monitored national standardization      West Jodi    1      Nauru
```

```
## 3      Organic bottom-line service-desk       Davidton    0 San Marino
## 4 Triple-buffered reciprocal time-frame West Terrifurt    1      Italy
## 5         Robust logistical utilization   South Manuel    0    Iceland
## 6       Sharable client-driven software      Jamieberg    1     Norway
##             Timestamp Clicked.on.Ad
## 1 2016-03-27 00:53:11             0
## 2 2016-04-04 01:39:02             0
## 3 2016-03-13 20:35:42             0
## 4 2016-01-10 02:31:19             0
## 5 2016-06-03 03:36:18             0
## 6 2016-05-19 14:30:17             0

tail(data)

##      Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 995                     43.70  28    63126.96               173.01
## 996                     72.97  30    71384.57               208.58
## 997                     51.30  45    67782.17               134.42
## 998                     51.63  51    42415.72               120.37
## 999                     55.55  19    41920.79               187.95
## 1000                    45.01  26    29875.80               178.35
##                           Ad.Topic.Line          City Male
## 995         Front-line bifurcated ability  Nicholasland    0
## 996         Fundamental modular algorithm     Duffystad    1
## 997       Grass-roots cohesive monitoring   New Darlene    1
## 998          Expanded intangible solution South Jessica    1
## 999  Proactive bandwidth-monitored policy   West Steven    0
## 1000      Virtual 5thgeneration emulation   Ronniemouth    0
##                      Country           Timestamp Clicked.on.Ad
## 995                  Mayotte 2016-04-04 03:57:48             1
## 996                  Lebanon 2016-02-11 21:49:00             1
## 997  Bosnia and Herzegovina 2016-04-22 02:07:01             1
## 998                 Mongolia 2016-02-01 17:24:57             1
## 999                Guatemala 2016-03-24 02:35:54             0
## 1000                  Brazil 2016-06-03 21:43:21             1
```

Cleaning Data

Finding the total missing values in our dataset.

```
colSums(is.na(data))

## 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 our dataset

##Checking for duplicates across our rows.

```
data[duplicated(data),]
```

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

there are no duplicated values in our dataset

Exploring the dataset

#Checking the descriptive statistics of our dataset

```
summary(data)
```

```
##  Daily.Time.Spent.on.Site      Age         Area.Income
Daily.Internet.Usage
##  Min.   :32.60            Min.   :19.00   Min.   :13996   Min.   :104.8
##  1st Qu.:51.36            1st Qu.:29.00   1st Qu.:47032   1st Qu.:138.8
##  Median :68.22            Median :35.00   Median :57012   Median :183.1
##  Mean   :65.00            Mean   :36.01   Mean   :55000   Mean   :180.0
##  3rd Qu.:78.55            3rd Qu.:42.00   3rd Qu.:65471   3rd Qu.:218.8
##  Max.   :91.43            Max.   :61.00   Max.   :79485   Max.   :270.0
##  Ad.Topic.Line          City                Male         Country
##  Length:1000        Length:1000        Min.   :0.000   Length:1000
##  Class :character   Class :character   1st Qu.:0.000   Class :character
##  Mode  :character   Mode  :character   Median :0.000   Mode  :character
##                                        Mean   :0.481
##                                        3rd Qu.:1.000
##                                        Max.   :1.000
##   Timestamp         Clicked.on.Ad
##  Length:1000        Min.   :0.0
##  Class :character   1st Qu.:0.0
##  Mode  :character   Median :0.5
##                     Mean   :0.5
##                     3rd Qu.:1.0
##                     Max.   :1.0
```

##Checking the structure of our dataframe

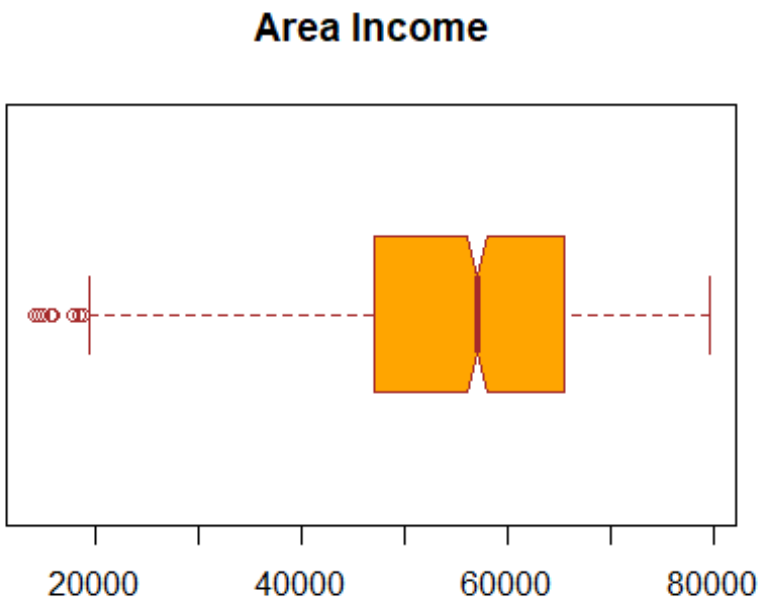
```
str(data)
```

```
## 'data.frame':    1000 obs. of  10 variables:
##  $ Daily.Time.Spent.on.Site: num  69 80.2 69.5 74.2 68.4 ...
##  $ Age                     : int  35 31 26 29 35 23 33 48 30 20 ...
##  $ Area.Income             : num  61834 68442 59786 54806 73890 ...
##  $ Daily.Internet.Usage    : num  256 194 236 246 226 ...
```

```
##  $ Ad.Topic.Line           : chr  "Cloned 5thgeneration orchestration"
"Monitored national standardization" "Organic bottom-line service-desk"
"Triple-buffered reciprocal time-frame" ...
##  $ City                    : chr  "Wrightburgh" "West Jodi" "Davidton"
"West Terrifurt" ...
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Country                 : chr  "Tunisia" "Nauru" "San Marino" "Italy"
...
##  $ Timestamp               : chr  "2016-03-27 00:53:11" "2016-04-04
01:39:02" "2016-03-13 20:35:42" "2016-01-10 02:31:19" ...
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
```

##checking for outliers plotting the boxplots to to check the data distribution in the numeric columns

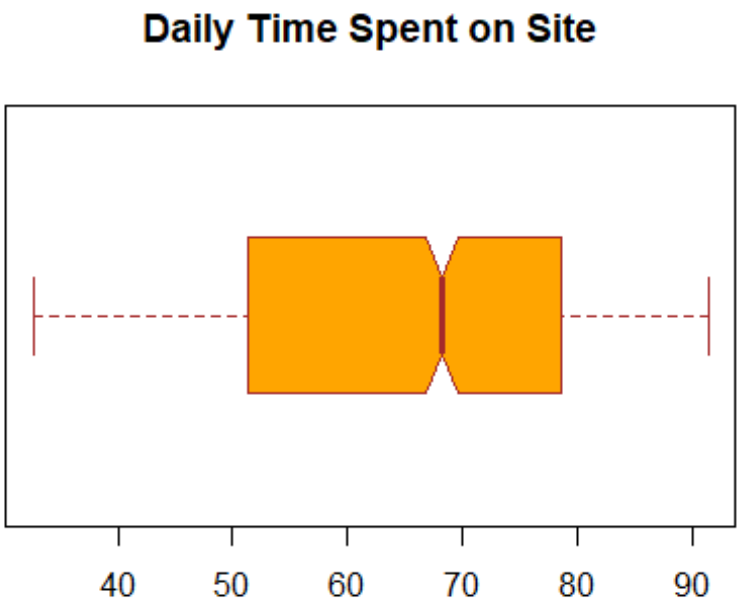
```
boxplot(data$Area.Income,
        main ="Area Income",
        col = "orange",
        border  = 'brown',
        horizontal = TRUE,
        notch = TRUE)
```



There are a few outliers in the area.income column

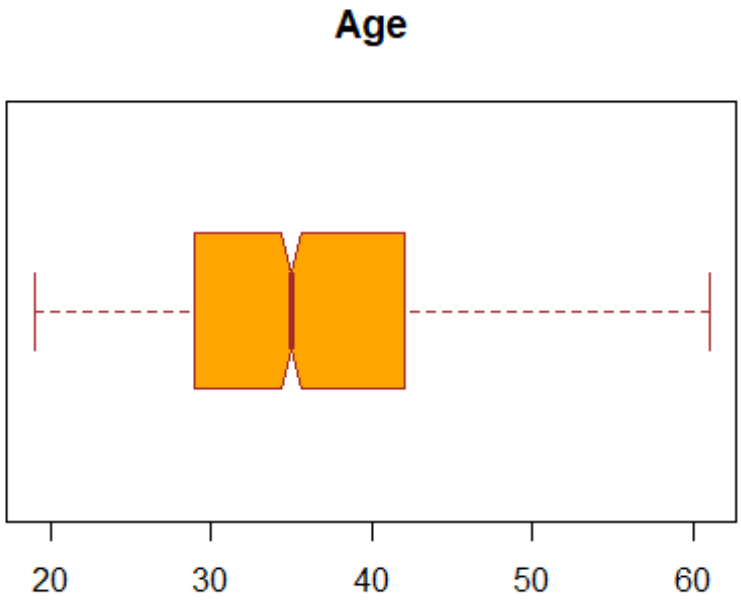
```
boxplot(data$Daily.Time.Spent.on.Site,
        main ="Daily Time Spent on Site",
        col = "orange",
        border  = 'brown',
```

```
        horizontal = TRUE,
        notch = TRUE)
```



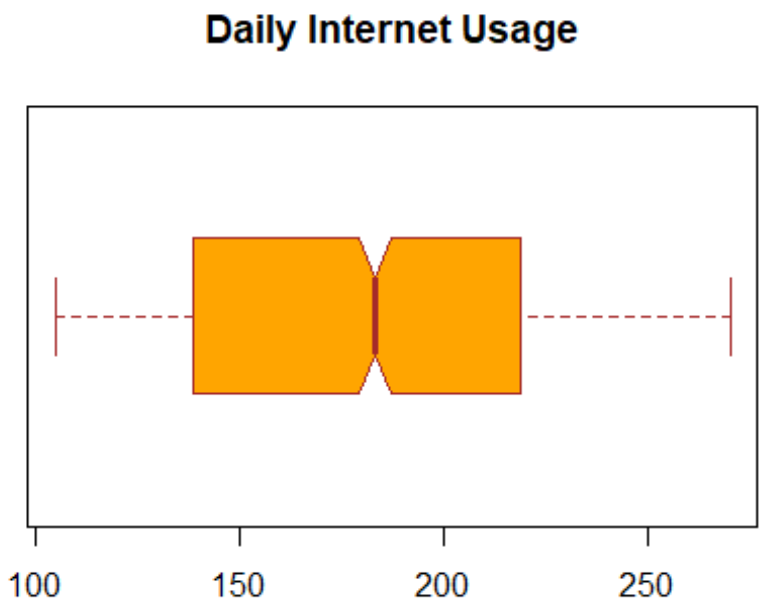
There are no outliers in time spent on site column.

```
boxplot(data$Age,
        main ="Age",
        col = "orange",
        border  = 'brown',
        horizontal = TRUE,
        notch = TRUE)
```



There are no outliers in the age column

```
boxplot(data$Daily.Internet.Usage,
        main ="Daily Internet Usage",
        col = "orange",
        border  = 'brown',
        horizontal = TRUE,
        notch = TRUE)
```




```
#There are no outliers in the daily internet usage column
```

Exploratory Data Analysis

Univariate Analysis

Measures of Central Tendency

#Finding the mean of our numeric columns

```
colMeans(data[sapply(data,is.numeric)])
```

```
## Daily.Time.Spent.on.Site                      Age             Area.Income
##                  65.0002                  36.0090              55000.0001
##     Daily.Internet.Usage                     Male           Clicked.on.Ad
##                 180.0001                   0.4810                   0.5000
```

#Finding the median of our numeric columns

```
ad_time_median <- median(data$Daily.Time.Spent.on.Site)
print(ad_time_median)
```

```
## [1] 68.215
```

```
ad_age_median <- median(data$Age)
ad_age_median
```

```
## [1] 35
```

```
ad_income_median <- median(data$Area.Income)
ad_income_median
```

```
## [1] 57012.3
```

```
ad_internet_usage_median <- median(data$Daily.Internet.Usage)
ad_internet_usage_median
```

```
## [1] 183.13
```

Finding the mode of our numeric columns. creating the mode function

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

getmode(data$Age)
```

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

```
getmode(data$Daily.Time.Spent.on.Site)
```

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

```
getmode(data$Area.Income)
```

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

```
getmode(data$Daily.Internet.Usage)
```

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

```
getmode(data$City)
```

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

```
getmode(data$Ad.Topic.Line)
```

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

```
getmode(data$Male)
```

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

```
getmode(data$Country)
```

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

```
getmode(data$Timestamp)
```

```
## [1] "2016-03-27 00:53:11"
```

finding the minimum values in the numeric columns

```
min(data$Age)
```

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

```
min(data$Daily.Time.Spent.on.Site)
```

```
## [1] 32.6
```

```
min(data$Area.Income)
```

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

```
min(data$Daily.Internet.Usage)
```

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

Finding the maximum values in the numeric columns

```
max(data$Age)
```

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

```
max(data$Daily.Time.Spent.on.Site)
```

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

```
max(data$Area.Income)
```

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

```
max(data$Daily.Internet.Usage)
```

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

Finding the range in the numeric columns

```
range(data$Age)
```

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

```
range(data$Daily.Time.Spent.on.Site)
```

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

```
range(data$Area.Income)
```

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

```
range(data$Daily.Internet.Usage)
```

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

- The youngest respondent is 19 and the oldest 61 years of age.
- The least time spent on her site is 32 minutes and the highest 91 minutes.
- The lowest income earner among the respondents earns 13,996 while the highest earns 79,484.

- Daily internet usage ranges from 105 - 270

#finding the stardard deviations of the columns

```
sd(data$Age)

## [1] 8.785562

sd(data$Daily.Time.Spent.on.Site)

## [1] 15.85361

sd(data$Area.Income)

## [1] 13414.63

sd(data$Daily.Internet.Usage)

## [1] 43.90234
```

#getting the quantiles in our columns

```
quantile(data$Age)

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

quantile(data$Daily.Time.Spent.on.Site)

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

quantile(data$Area.Income)

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

quantile(data$Daily.Internet.Usage)

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

Frequency Distribution

Finding the frequency distribution in the age column

```
table(data$Age)

##
## 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
44
##  6  6  6 13 19 21 27 37 33 48 48 39 60 38 43 39 39 50 36 37 30 36 32 26 23
21
## 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61
## 30 18 13 16 18 20 12 15 10  9  7  2  6  4  2  4  1
```

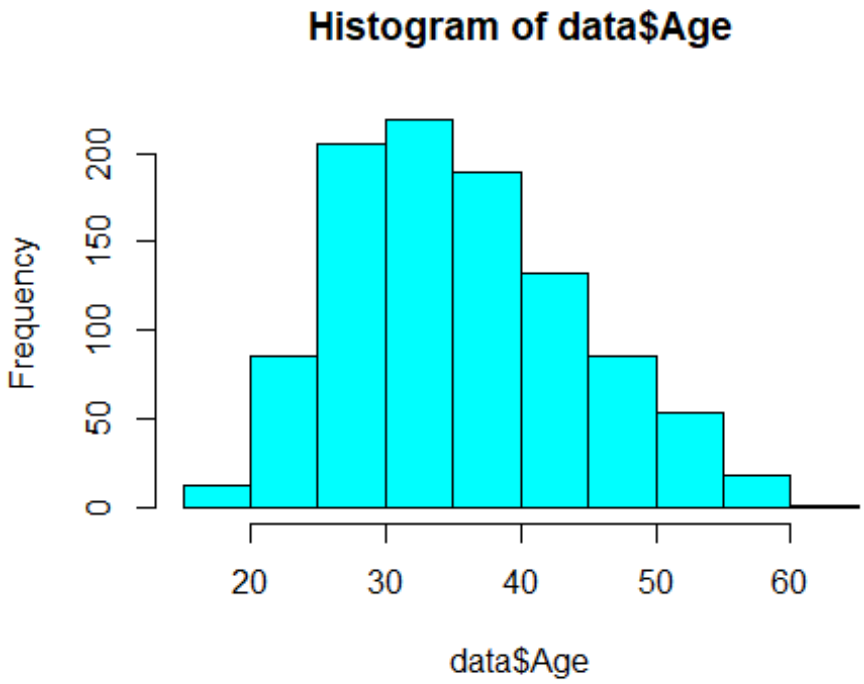
```
summary(data)

##  Daily.Time.Spent.on.Site      Age         Area.Income
Daily.Internet.Usage
##  Min.   :32.60            Min.   :19.00   Min.   :13996   Min.   :104.8
##  1st Qu.:51.36            1st Qu.:29.00   1st Qu.:47032   1st Qu.:138.8
##  Median :68.22            Median :35.00   Median :57012   Median :183.1
##  Mean   :65.00            Mean   :36.01   Mean   :55000   Mean   :180.0
##  3rd Qu.:78.55            3rd Qu.:42.00   3rd Qu.:65471   3rd Qu.:218.8
##  Max.   :91.43            Max.   :61.00   Max.   :79485   Max.   :270.0
##  Ad.Topic.Line          City                Male         Country
##  Length:1000        Length:1000        Min.   :0.000   Length:1000
##  Class :character   Class :character   1st Qu.:0.000   Class :character
##  Mode  :character   Mode  :character   Median :0.000   Mode  :character
##                                        Mean   :0.481
##                                        3rd Qu.:1.000
##                                        Max.   :1.000
##   Timestamp         Clicked.on.Ad
##  Length:1000        Min.   :0.0
##  Class :character   1st Qu.:0.0
##  Mode  :character   Median :0.5
##                     Mean   :0.5
##                     3rd Qu.:1.0
##                     Max.   :1.0
```

Most respondents fall between the age bracket of 24-40years. The age with the highest number of readers is 31 which has a total of 60 people in total.

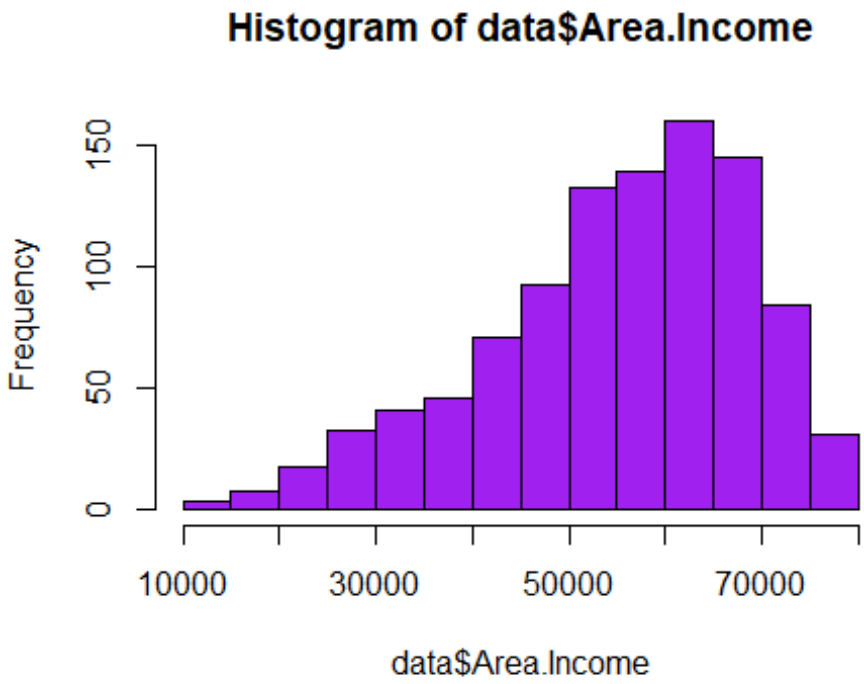
Histogram

```
hist(data$Age, col  = "Cyan")
```



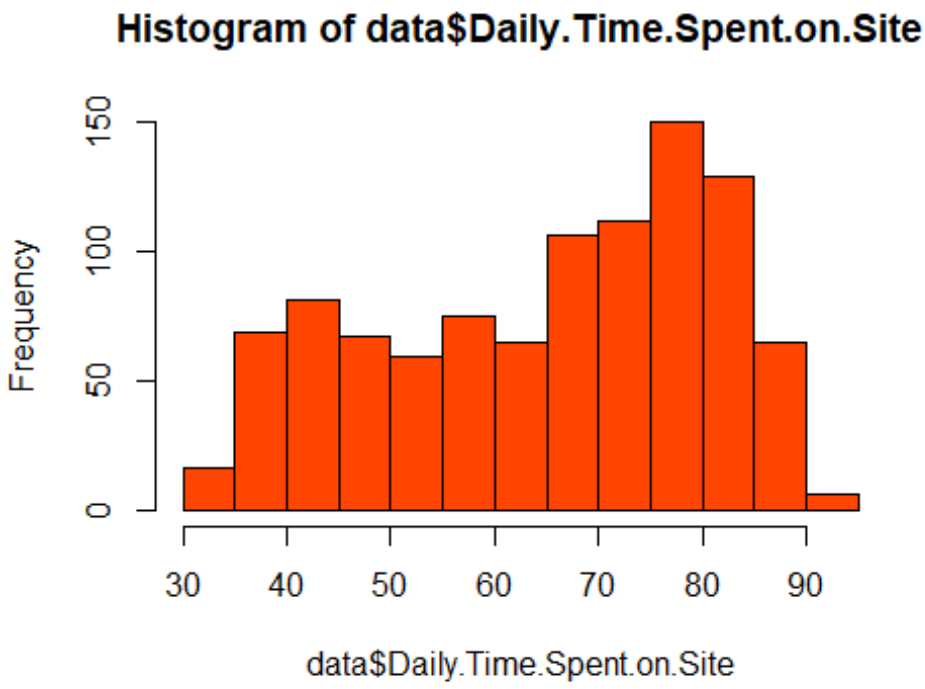
#Most respondents fall in the age bracket of 25-40yrs.

```
hist(data$Area.Income, col = "Purple")
```

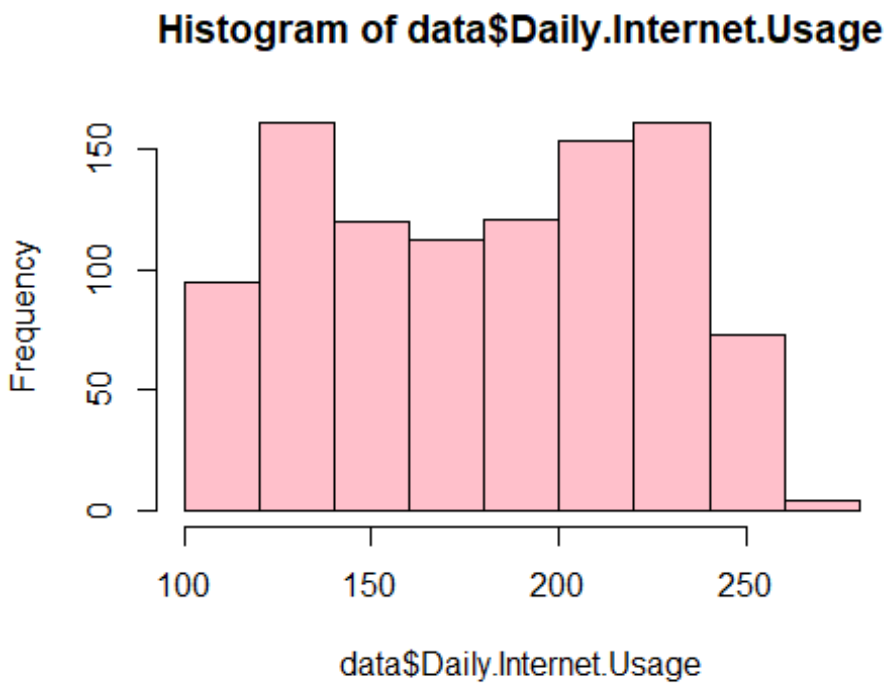


#Majority of the respondents earn between 50K - 70K

```
hist(data$Daily.Time.Spent.on.Site, col = "orangered")
```




```
hist(data$Daily.Internet.Usage, col = "pink")
```



Bivariate Analysis

Covariance

```
cov(data$Age, data$Daily.Time.Spent.on.Site)
```

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

#There is a negative relationship between the age and the time spent on site which means as the age increases, the daily time spent on the site decreases.

```
cov(data$Age, data$Daily.Internet.Usage)
```

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

#There is a negative relationship between the age and the daily internet usage as well.

```
cov(data$Area.Income,data$Daily.Time.Spent.on.Site)
```

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

#There is a strong positive relationship between the income and daily time spent on site variables. This means the higher the income, the more the time spent on site and the lower the income, the less the time spent on site.

```
cov(data$Age,data$Area.Income)
```

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

#There is a negative correlation between the age and income variables.

Correlation matrix

```
cor(data$Age, data$Daily.Time.Spent.on.Site)

## [1] -0.3315133

cor(data$Age,data$Daily.Internet.Usage)

## [1] -0.3672086

cor(data$Area.Income,data$Daily.Internet.Usage)

## [1] 0.3374955

cor(data$Area.Income,data$Daily.Time.Spent.on.Site)

## [1] 0.3109544

cor(data$Age,data$Area.Income)

## [1] -0.182605

cor(data[, c("Age","Daily.Time.Spent.on.Site","Daily.Internet.Usage")])

##                                Age Daily.Time.Spent.on.Site
## Age                         1.0000000               -0.3315133
## Daily.Time.Spent.on.Site -0.3315133                1.0000000
## Daily.Internet.Usage     -0.3672086                0.5186585
##                          Daily.Internet.Usage
## Age                                -0.3672086
## Daily.Time.Spent.on.Site            0.5186585
## Daily.Internet.Usage                1.0000000

cor(data[,unlist(lapply(data, is.numeric))])

##                          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
```

There are negative correlations between the following variables 1.Area Income and Daily Time Spent on Site 2.Male and Daily Time Spent on Site 3.Clicking on the Advert and Daily

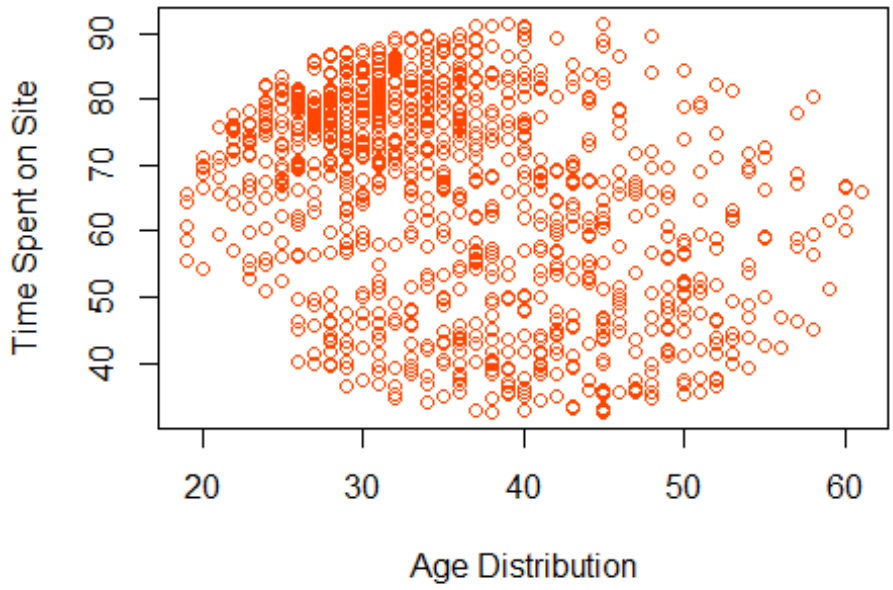
Time Spent on Site. 4.Area Income and Age 5.Daily Internet Usage and Age 6.Male and Age 7.Area Income and Age 8.Area Income and Clicking on the Advert 9.Daily Internet usage and Clicking on the advert. 10.Male and Clicking on the Advert

There are positive Correlations between the following variables: 1.Age and Clicking on the advert 2.Male and Daily Internet Usage 3.Male and Area Income 4.Daily Time Spent on Site and Daily Internet Usage. 5.Area Income and Daily Time Spent on Site 6.Area Income and Daily Internet Usage 7.Area Income and Male 8.Age and Clicking on the Advert.

Scatter Plots

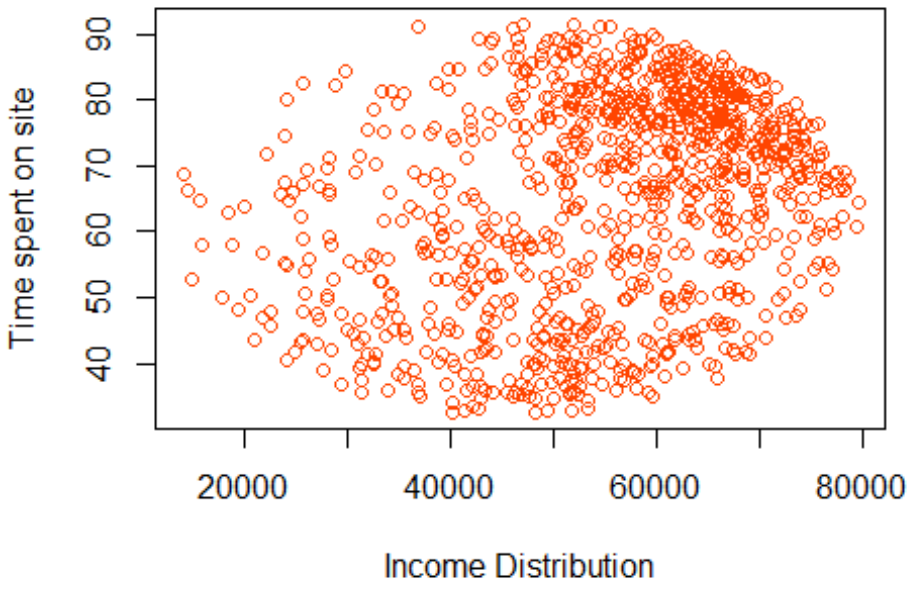
A scatter plot for age and daily time spent on site.

```
plot(data$Age,data$Daily.Time.Spent.on.Site, xlab = "Age Distribution",
     ylab = "Time Spent on Site", col="orangered")
```



Scatter plot for Income Distribution and Daily time spent on site.

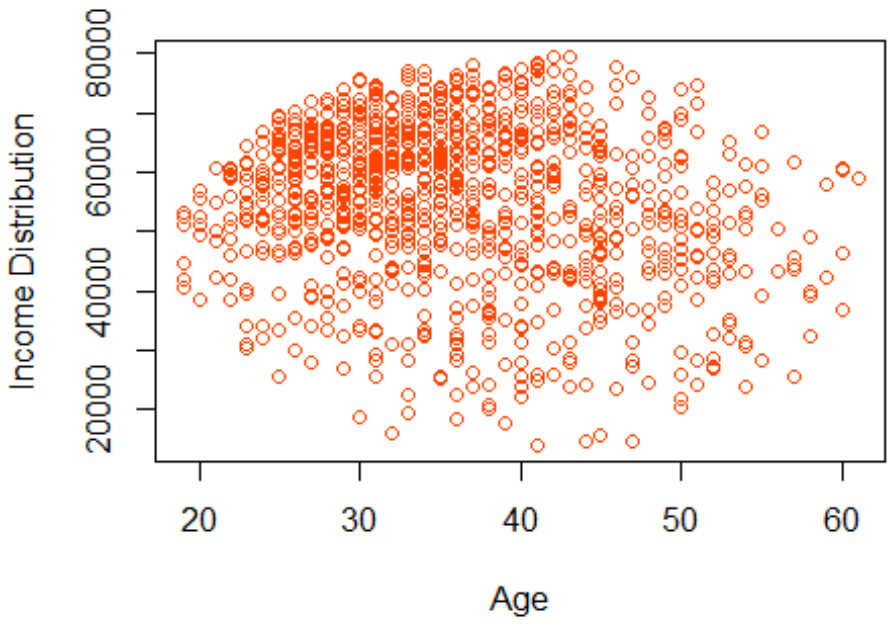
```
plot(data$Area.Income,data$Daily.Time.Spent.on.Site, xlab= "Income
Distribution", ylab = "Time spent on site", col="orangered")
```




```
plot(data$Area.Income,data$Daily.Time.Spent.on.Site, xlab= "Income
Distribution", ylab = "Time spent on site", col="orangered")
```

Scatter plot for Age and Income Distribution

```
plot(data$Age,data$Area.Income, xlab = "Age", ylab ="Income Distribution",
col="orangered")
```



Modelling

K-Nearest-Neighbors

```
df <- data[,c("Clicked.on.Ad","Daily.Time.Spent.on.Site",
"Age","Area.Income","Daily.Internet.Usage")]

#Randomizing our data for better results
random <- runif(1000, 1:4)

## Warning in runif(1000, 1:4): NAs produced

ad_random <- df[order(random),]
head(ad_random)

##    Clicked.on.Ad Daily.Time.Spent.on.Site Age Area.Income
Daily.Internet.Usage
## 1              0                   68.95  35    61833.90
256.09
## 5              0                   68.37  35    73889.99
225.58
## 9              0                   74.53  30    68862.00
221.51
## 13             1                   69.57  48    51636.92
113.12
```

```
## 17             1                    55.39  37    23936.86
129.41
## 21             0                    77.22  30    64802.33
224.44
```

```
normal <- function(x) (
  return( ((x - min(x)) /(max(x)-min(x))) )
)
normal(1:4)
```

```
## [1] 0.0000000 0.3333333 0.6666667 1.0000000
```

```
ad_new <- as.data.frame(lapply(ad_random[1:4], normal))
summary(ad_new)
```

```
##  Clicked.on.Ad Daily.Time.Spent.on.Site      Age          Area.Income
##  Min.   :0.0   Min.   :0.0000           Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0   1st Qu.:0.3189           1st Qu.:0.2381   1st Qu.:0.5044
##  Median :0.5   Median :0.6054           Median :0.3810   Median :0.6568
##  Mean   :0.5   Mean   :0.5507           Mean   :0.4050   Mean   :0.6261
##  3rd Qu.:1.0   3rd Qu.:0.7810           3rd Qu.:0.5476   3rd Qu.:0.7860
##  Max.   :1.0   Max.   :1.0000           Max.   :1.0000   Max.   :1.0000
```

Let's create test and train datasets

```
train <- ad_new[1:800,]
test <- ad_new[801:1000,]
train_sp <- ad_random[1:800,10]
test_sp <- ad_random[801:1000,10]
```

Let's call the class package which contains the KNN algorithm. The table(test_sp, model) is our confusion matrix.

```
library(caret)
```

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

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

```
intrain <- createDataPartition(y = data$Clicked.on.Ad, p= 0.7, list = FALSE)
training <- df[intrain,]
testing <- df[-intrain,]
```

Checking the dimension of the training and testing dataframe.

```
dim(training);
```

```
## [1] 700   5
```

```
dim(testing);
```

```
## [1] 300   5
```

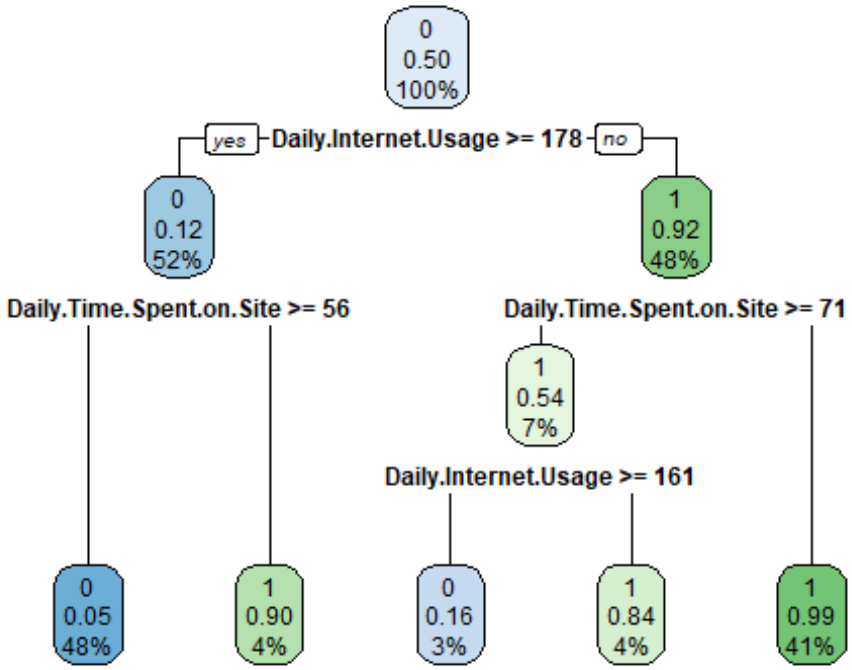
Decision Trees

```
library(rpart.plot)

## Loading required package: rpart

library(mlbench)
library(rpart)

dt <- rpart(Clicked.on.Ad ~ Daily.Time.Spent.on.Site + Age + Area.Income +
Daily.Internet.Usage, data = df, method  ="class")
rpart.plot(dt)
```



Looking for feature importances.

```
data.frame(dt$variable.importance)

##                           dt.variable.importance
## Daily.Internet.Usage                     339.7809
## Daily.Time.Spent.on.Site                 279.0247
## Age                                      126.2649
## Area.Income                              119.2524
```

This decision tree algorithm predicts 957 correct observations out of 1000. This model achieves an accuracy of 95.7 %.

```
library(caret)
set.seed(12)
```

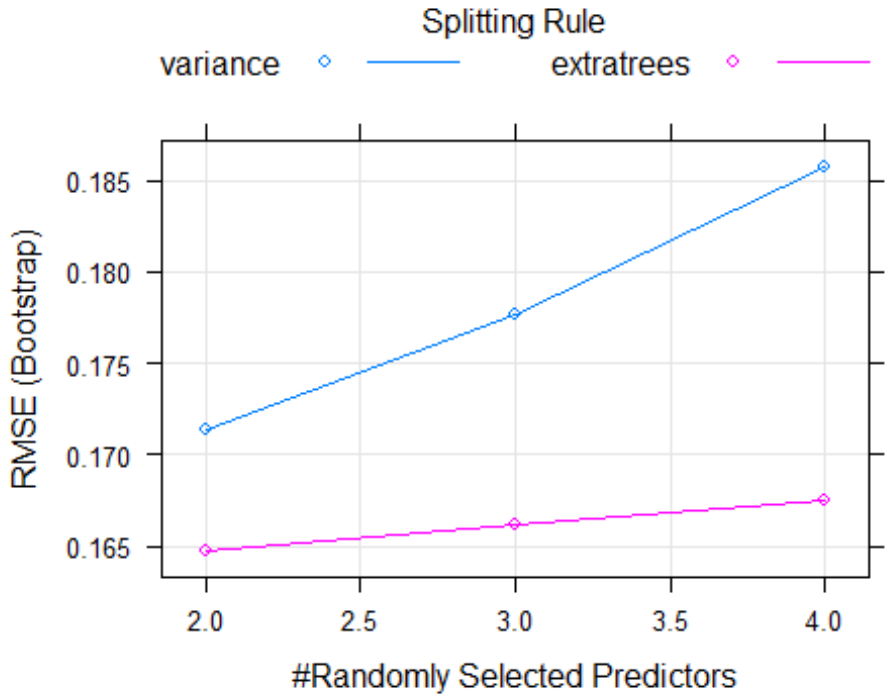
```
model <- train(Clicked.on.Ad ~ Daily.Time.Spent.on.Site + Age + Area.Income +
Daily.Internet.Usage ,data = df,method = "ranger")

## 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.

model

## Random Forest
##
## 1000 samples
##    4 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    0.1713734  0.8825549  0.06331311
##   2     extratrees  0.1647794  0.8923250  0.06987902
##   3     variance    0.1776455  0.8737515  0.05998737
##   3     extratrees  0.1661163  0.8898515  0.06574389
##   4     variance    0.1857072  0.8623095  0.05935058
##   4     extratrees  0.1674823  0.8878285  0.06395853
##
## 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 = 2, splitrule = extratrees
##  and min.node.size = 5.

plot(model)
```



Support Vector Machines

```
library(caret)
intrain <- createDataPartition(y = data$Clicked.on.Ad, p= 0.7, list = FALSE)
training <- data[intrain,]
testing <- data[-intrain,]

dim(training);

## [1] 700  10

dim(testing);

## [1] 300  10
```

Let's factorize our target variable for accurate results.

```
training[["Clicked.on.Ad"]] = factor(training[["Clicked.on.Ad"]])
```

Controlling the computational overheads

```
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)

svm_Linear <- train(Clicked.on.Ad ~ Daily.Time.Spent.on.Site + Age +
Area.Income +Daily.Internet.Usage , data = training, method = "svmLinear",
trControl=trctrl,
preProcess = c("center", "scale"),
tuneLength = 10)
```

Checking the result of our training model.

```
svm_Linear
```

```
## Support Vector Machines with Linear Kernel
##
## 700 samples
##   4 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (4), scaled (4)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 630, 630, 630, 630, 630, 630, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9695238  0.9390476
##
## Tuning parameter 'C' was held constant at a value of 1
```

Predicting our model results using the predict() method.

```
test_pred <- predict(svm_Linear, newdata = testing)
test_pred
```

```
##   [1] 0 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 1 0 1 0 0 1 1 1 1 1 1 0 1 1 1 0
0 1 0
##  [38] 1 1 0 0 0 0 1 0 1 1 1 0 0 0 1 1 0 0 1 0 1 0 1 1 0 1 0 1 0 0 1 1 0 0
1 1 1
##  [75] 1 1 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 1 0 0 0 1 0
1 0 1
## [112] 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 1 0 0 1 1 0 0 0 0
1 0 0
## [149] 0 1 1 0 0 1 0 1 0 1 1 1 0 0 0 0 0 0 0 1 1 1 1 0 1 1 0 0 0 1 1 1 0 1
0 0 1
## [186] 0 1 0 0 0 0 0 1 1 1 0 1 0 1 1 0 1 0 1 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0
0 1 1
## [223] 0 0 0 1 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 0 0 1 0 1 0 1 1 0
0 1 0
## [260] 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 0 1 0 0 1 1 1 1 1 1 1 0 1 0 1 1 0 0 1
1 1 1
## [297] 1 1 1 1
## Levels: 0 1
```

Checking the accuracy of our model using a confusion matrix.

```
confusionMatrix(table(test_pred, testing$Clicked.on.Ad))
```

```
## Confusion Matrix and Statistics
##
##
## test_pred   0   1
```

```
##          0 147   9
##          1   3 141
##
##                Accuracy : 0.96
##                  95% CI : (0.9312, 0.9792)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.92
##
##  Mcnemar's Test P-Value : 0.1489
##
##             Sensitivity : 0.9800
##             Specificity : 0.9400
##          Pos Pred Value : 0.9423
##          Neg Pred Value : 0.9792
##              Prevalence : 0.5000
##          Detection Rate : 0.4900
##    Detection Prevalence : 0.5200
##       Balanced Accuracy : 0.9600
##
##        'Positive' Class : 0
##
```

Our model achieved an accuracy level of 96% which is pretty good.

Naive Bayes

```
# splitting the dataset into the training set and test set
#install.packages('caTools')
library(caTools)
set.seed(123)
split <- sample.split(df$Clicked.on.Ad, SplitRatio = 0.80)
training <- subset(data, split == TRUE)
testing <- subset(data, split == FALSE)

#checking the dimensions of the split
dim(training)

## [1] 800  10

dim(testing)

## [1] 200  10

# Fitting Naive Bayes to the Training set
library(e1071)
classifier = naiveBayes(x = training[-6],
                        y = training$Clicked.on.Ad)
```

```r
# Predicting the Test set results
y_pred = predict(classifier, newdata = testing[-6])
y_pred
```

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

```r
# Making the Confusion Matrix
mt = table(testing[, 6], y_pred)
mt
```

```
##                      y_pred
##                       0 1
##   Amandafort          0 1
##   Amyhaven            1 0
##   Andrewmouth         1 0
##   Ashleychester       1 0
##   Barbershire         1 0
##   Beckton             1 0
##   Bernardton          0 1
##   Birdshire           1 0
##   Blairville          1 0
##   Bradleyburgh        0 1
##   Brandiland          0 1
##   Brandymouth         0 1
##   Brendaburgh         1 0
##   Brownbury           1 0
##   Brownton            0 1
##   Calebberg           1 0
##   Cannonbury          1 0
##   Carterton           1 0
##   Cassandratown       1 0
##   Charlottefort       0 1
##   Chrismouth          0 1
##   Christinetown       0 1
##   Christopherport     0 1
##   Clineshire          1 0
##   Codyburgh           0 1
##   Collinsburgh        1 0
##   Contrerasshire      1 0
```

```
##   Curtisport          0 1
##   Davidside           0 1
##   Davilachester       0 1
##   East Brianberg      1 0
##   East Deborahhaven   1 0
##   East Eric           0 1
##   East Heatherside    0 1
##   East Heidi          0 1
##   East Johnport       1 0
##   East Michele        0 1
##   East Mike           0 1
##   East Ronald         0 1
##   East Tammie         0 1
##   East Timothy        1 0
##   East Toddfort       1 0
##   Ericksonmouth       0 1
##   Erinton             0 1
##   Estradafurt         1 0
##   Fraziershire        0 1
##   Garciaview          0 1
##   Gonzalezburgh       1 0
##   Grahamberg          0 1
##   Gravesport          1 0
##   Greghaven           1 0
##   Hammondport         1 0
##   Helenborough        0 1
##   Hernandezfort       1 0
##   Jeffreyburgh        0 1
##   Jeffreymouth        0 1
##   Joechester          0 1
##   Johnstad            1 0
##   Johnstonmouth       0 1
##   Jonesland           1 0
##   Jonesshire          0 1
##   Josephmouth         0 1
##   Juanport            1 0
##   Katieport           0 1
##   Kellytown           1 0
##   Kingchester         0 1
##   Lake Annashire      1 0
##   Lake Brian          1 0
##   Lake Cassandraport  0 1
##   Lake David          0 1
##   Lake Edward         0 1
##   Lake Jessica        0 1
##   Lake Josetown       1 0
##   Lake Michaelport    1 0
##   Lake Susan          1 0
##   Lawsonshire         0 1
##   Lesliebury          0 1
```

```
##   Lisafort            1 0
##   Lisamouth           1 0
##   Lopezmouth          1 0
##   Lukeport            1 0
##   Mcdonaldfort        1 0
##   Meaganfort          1 0
##   Meghanchester       0 1
##   Melanieton          0 1
##   Melissafurt         1 0
##   Michellefort        0 1
##   Morganport          0 1
##   New Amanda          0 1
##   New Cynthia         1 0
##   New Henry           0 1
##   New James           0 1
##   New Jasmine         1 0
##   New Jeffreychester  1 0
##   New Julie           1 0
##   New Marcusbury      0 1
##   New Patriciashire   1 0
##   New Shane           1 0
##   New Traceystad      1 0
##   New Tyler           1 0
##   Nicholasland        0 1
##   North Brandon       1 0
##   North Brittanyburgh 0 1
##   North Charlesbury   0 1
##   North Jessicaville  0 1
##   North Joshua        1 0
##   North Lauraland     0 1
##   North Lisachester   1 0
##   North Randy         1 0
##   North Ronaldshire   1 0
##   North Tara          1 0
##   North Tylerland     1 0
##   North Wesleychester 1 0
##   Olsonstad           0 1
##   Palmerside          0 1
##   Patriciahaven       1 0
##   Paulhaven           1 0
##   Pearsonfort         1 0
##   Petersonfurt        0 1
##   Port Aprilville     0 1
##   Port Beth           0 1
##   Port Blake          0 1
##   Port Brianfort      1 0
##   Port Christopher    0 1
##   Port Daniel         1 0
##   Port Danielleberg   1 0
##   Port Dennis         0 1
```

```
##   Port Erikhaven      0 1
##   Port Georgebury     0 1
##   Port Gregory        1 0
##   Port James          1 0
##   Port Juan           0 1
##   Port Julie          1 0
##   Port Lawrence       0 1
##   Port Mitchell       0 1
##   Port Rachel         0 1
##   Port Robin          1 0
##   Port Sarahshire     0 1
##   Ramirezside         0 1
##   Randyshire          1 0
##   Reneechester        0 1
##   Richardshire        0 1
##   Rickymouth          1 0
##   Robertfurt          0 1
##   Robertstown         0 1
##   Roberttown          0 1
##   Rogerburgh          0 1
##   Sandersland         1 0
##   Sandraland          1 0
##   Smithside           0 1
##   Smithtown           1 0
##   South Aaron         0 1
##   South Jackieberg    0 1
##   South Jaimeview     1 0
##   South John          0 1
##   South Johnnymouth   0 1
##   South Kyle          0 1
##   South Manuel        1 0
##   South Mark          0 1
##   South Peter         0 1
##   South Stephanieport 1 0
##   Taylorhaven         0 1
##   Taylormouth         0 1
##   Thomasview          1 0
##   Timothyfurt         0 1
##   Timothymouth        0 1
##   Tracyhaven          0 1
##   Turnerchester       0 1
##   Turnerview          1 0
##   Valerieland         1 0
##   Villanuevaton       1 0
##   Wallacechester      1 0
##   Welchshire          0 1
##   Wendyton            1 0
##   West Arielstad      1 0
##   West Brenda         1 0
##   West Chloeborough   0 1
```

```
##   West Colin          1 0
##   West Connor         0 1
##   West Daleborough    1 0
##   West Dennis         1 0
##   West Ericfurt       0 1
##   West Jodi           1 0
##   West Julia          0 1
##   West Mariafort      1 0
##   West Michaelshire   1 0
##   West Michaelstad    1 0
##   West Russell        1 0
##   West Samantha       1 0
##   West Terrifurt      1 0
##   West Tinashire      0 1
##   West Wendyland      1 0
##   West Zacharyborough 1 0
##   Westshire           0 1
##   Whiteport           0 1
##   Whitneyfort         1 0
##   Williamport         1 0
##   Williamsfort        0 1
##   Williamsport        1 0
##   Wrightview          0 1
```

```
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(mt)
```

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

our model achieved an accuracy score of 96.5%

modelling conclusion

SVM and Naive Bayes perfomed well on our dataset with accuracy scores of 96.5% and 96% respectively.

##Conclusion

After our analysis, we conclude that the following insights would help identify an individual who is likely to click on the ad:

1. Daily Time Spent on Site-the higher the time the lower the chances of clicking.
2. Age-The higher the Age the Higher the chance of clicking on the ads
3. Area Income-The higher the income the higher the chances of clicking on the ad
4. Internet Usage-The lower the Internet Usage the higher the chances of clicking the ad.
5. Low income earners are more likely to click on the Ads.

Recommendations

- Since the data shows that most of the respondents fall in the age bracket 25-41, she should tailor make the course to attract older people as well.
- Our client should target people with a higher income since the majority of those wh clicked the ad were low income earners.
- Most people spent about 70-85 which could be quite tiresome hence she should ensue that the courses are not too long to attract more people.

Follow up questions

Did we have the right data?

Yes

Do we need other data to answer our question?

No

Did we have the right question?

Yes