week_1 R_programming

Emmanuel Titi

30 May 2022

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.

RESEARCH QUESTION

What characteristics does the person clicking the link posses?

Metric for success

Coming up with a list that generalizes in characteristics or patterns that all ad clickers have would form a good foundation in making educated guess or almost precise predictions on other people with similar traits online.

Context

This research would be mostly appropriate in the client is looking to make more targeted ads such that the ads go to a precise group on individuals who fit the criteria we come up with in the at the end.

BASIC DATA ANALYSIS

Here we try to get familiar with our data set ,its shape ,sum of unique values on each columns and much more.This sort of gives us ideas on how to approach our problem solving works.

```
#importing needed dependencies for analysis and cleaning
library(tidyr)
library(tidyverse)
```

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

## v ggplot2 3.3.6     v dplyr   1.0.9
## v tibble  3.1.7     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4

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

```
#pre veiwing
df_ads=read.csv("http://bit.ly/IPAdvertisingData")
head(df_ads ,6)
```

```
##   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(df_ads ,6)
```

```
##      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
```

We start off by first seeing the shape of our data set ie. the number of columns and rows.

```
dim(df_ads)
```

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

Getting to know the data types of our variables is essential ,helps in knowing how to compare their relationship and ultimately being very useful in our plots

```
#checking for data types of variables
sapply(df_ads,class)
```

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

The data type seem to be on point. Lets try and get a summary of our data set

```
summary(df_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
##  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
```

The summary sort of gives us the basic information we need to understand the scope of every variable we have. For instance we can see the that for the Age variable the minimum age of an individual in our data frame is 19 and maximum is 61 . we have a mean of 36 years fo that .

DATA CLEANING

Checking for null values

Null values make our data inconsistent and may make our analysis hard,for that it is necessary to properly deal with them to improve the quality of the data we have.

```
#checking for the count of missing values
sapply(df_ads, function(x) sum(is.na(x)))
```

```
## 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
```

```
sum(is.na(df_ads))
```

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

It appears our data frame has no missing values.

Checking for duplicates

```
sum(duplicated(df_ads))
```

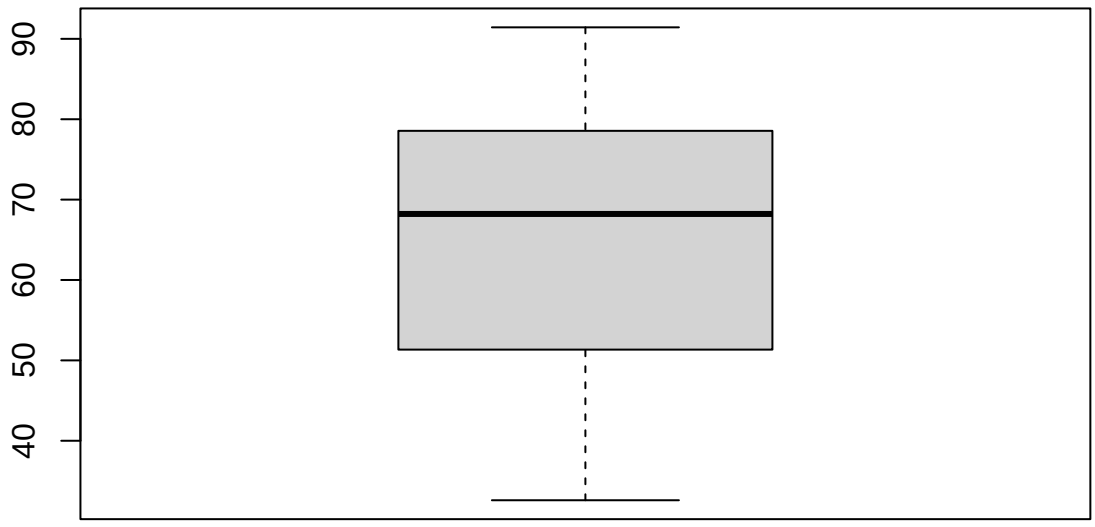
```
## [1] 0
```

There are also zero duplicates

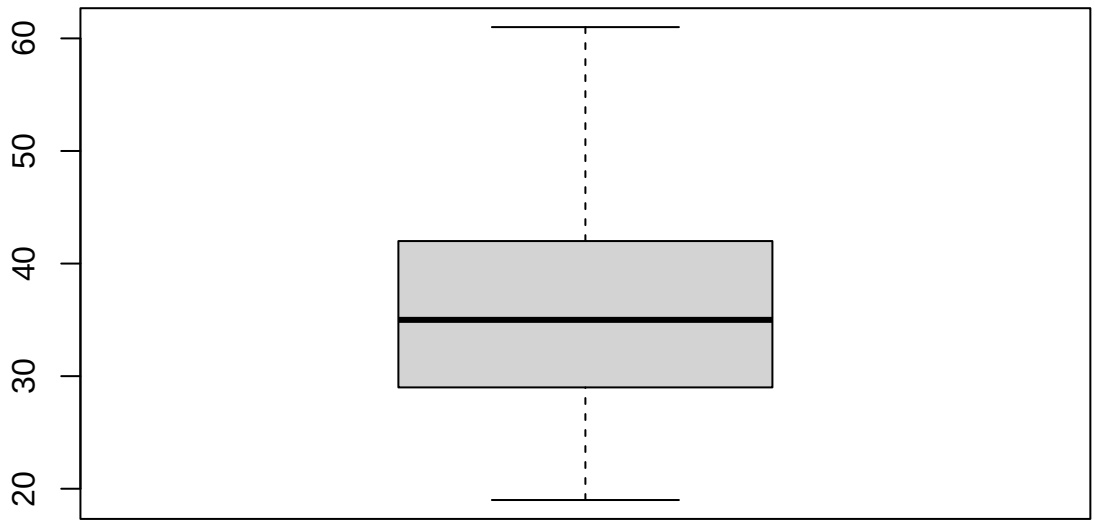
Checking for outliers **Box**

plots are a great way of visualizing outliers

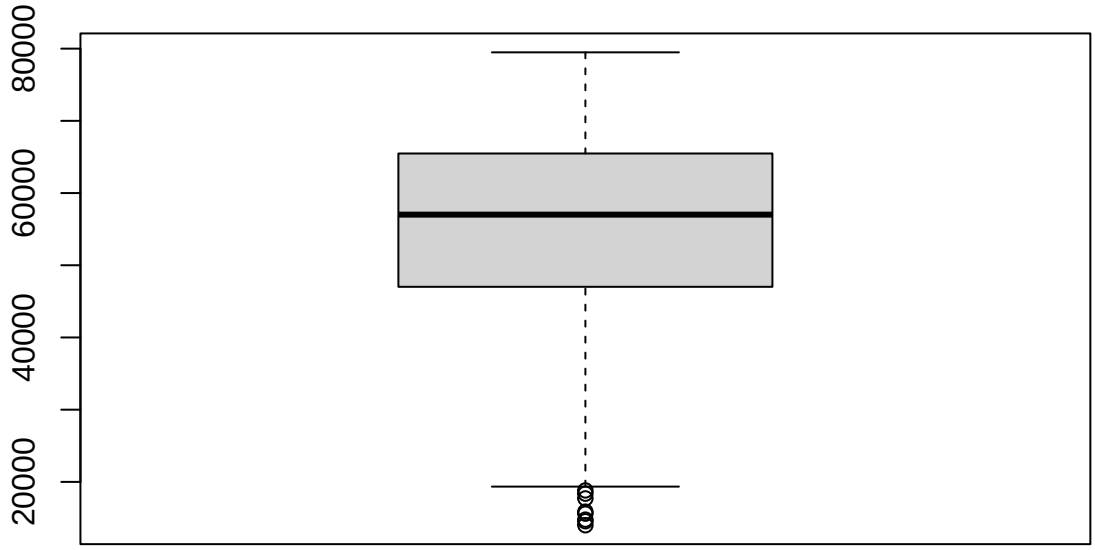
```
boxplot(df_ads$Daily.Time.Spent.on.Site)
```



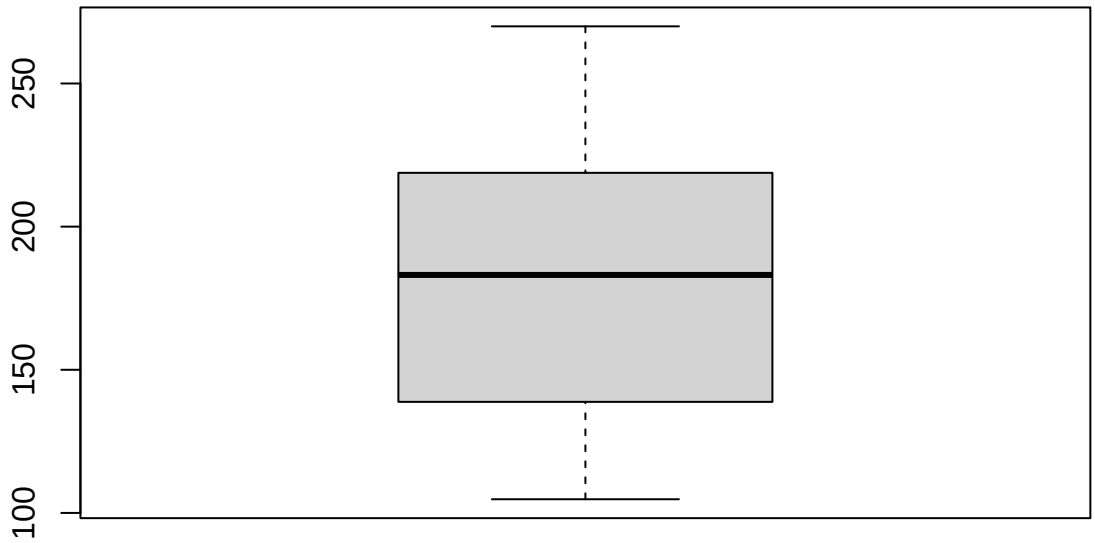
```
boxplot(df_ads$Age)
```




```
boxplot(df_ads$Area.Income)
```




```
boxplot(df_ads$Daily.Internet.Usage)
```



The area income appears to have outliers ,basically values below the 20000 mark

```
#count of values with less than 20000 in the area income column
sum(df_ads$Area.Income<20000)
```

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

This make 1% of our total data frame dropping the 10 rows would be safer than imputing which may change overall distribution of the data

```
new_df<-df_ads[!(df_ads$Area.Income<20000),]
dim(new_df)
```

```
## [1] 990  10
```

UNIVARIATE ANALYSIS

Mean

```
#this can be used to confirm our calculations on measures if central tendancy

summary(new_df)
```

```
##  Daily.Time.Spent.on.Site      Age          Area.Income    Daily.Internet.Usage
##  Min.   :32.60            Min.   :19.00   Min.   :20593   Min.   :104.8
```

```
##  1st Qu.:51.32            1st Qu.:29.00   1st Qu.:47366   1st Qu.:138.7
##  Median :68.44            Median :35.00   Median :57295   Median :183.5
##  Mean   :65.06            Mean   :35.98   Mean   :55385   Mean   :180.0
##  3rd Qu.:78.59            3rd Qu.:42.00   3rd Qu.:65557   3rd Qu.:218.9
##  Max.   :91.43            Max.   :61.00   Max.   :79485   Max.   :270.0
##  Ad.Topic.Line          City                Male          Country
##  Length:990         Length:990         Min.   :0.0000   Length:990
##  Class :character   Class :character   1st Qu.:0.0000   Class :character
##  Mode  :character   Mode  :character   Median :0.0000   Mode  :character
##                                        Mean   :0.4798
##                                        3rd Qu.:1.0000
##                                        Max.   :1.0000
##   Timestamp         Clicked.on.Ad
##  Length:990         Min.   :0.0000
##  Class :character   1st Qu.:0.0000
##  Mode  :character   Median :0.0000
##                     Mean   :0.4949
##                     3rd Qu.:1.0000
##                     Max.   :1.0000
```

```
#lets calculate the ,ean /avarages of the numeric variables
#Daily.Time.Spent.on.Site
sum(new_df$Daily.Time.Spent.on.Site)/length(new_df$Daily.Time.Spent.on.Site)
```

```
## [1] 65.05808
```

```
#Age
sum(new_df$Age)/length(new_df$Age)
```

```
## [1] 35.98384
```

```
#Area.Income
sum(new_df$Area.Income)/length(new_df$Area.Income)
```

```
## [1] 55384.82
```

```
#Daily.Internet.Usage
sum(new_df$Daily.Internet.Usage)/length(new_df$Daily.Internet.Usage)
```

```
## [1] 180.0282
```

Mode

Mode is the most recurring number in a given set of numbers

```
#Daily.Time.Spent.on.Site
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
v<-new_df$Daily.Time.Spent.on.Site
result_v <- getmode(v)
print(result_v)
```

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

```
#Age
getmode <- function(p) {
   uniqv <- unique(p)
   uniqv[which.max(tabulate(match(p, uniqv)))]
}
p<-new_df$Age
result_p <- getmode(p)
print(result_p)
```

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

```
#Area.Income
getmode <- function(k) {
   uniqv <- unique(k)
   uniqv[which.max(tabulate(match(k, uniqv)))]
}
k<-new_df$Area.Income
result_k <- getmode(k)
print(result_k)
```

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

```
#Daily.Internet.Usage
getmode <- function(s) {
   uniqv <- unique(s)
   uniqv[which.max(tabulate(match(s, uniqv)))]
}
s<-new_df$Daily.Internet.Usage
result_s <- getmode(s)
print(result_s)
```

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

Variance

Variance is the squared sums of deviations from the mean in a certain variable.

```
#Daily.Time.Spent.on.Site
mn1=sum(new_df$Daily.Time.Spent.on.Site)/length(new_df$Daily.Time.Spent.on.Site)
diff1<-(new_df$Daily.Time.Spent.on.Site-(mn1))
vr1<-sum(diff1^2/new_df$Daily.Time.Spent.on.Site-1)
vr1
```

```
## [1] 3832.963
```

```
#Area.Income
mn2=sum(new_df$Area.Income)/length(new_df$Area.Income)
diff2<-(new_df$Area.Income-(mn2))
vr2<-sum(diff2^2/new_df$Area.Income-1)
vr2
```

```
## [1] 4123306
```

```
#Age
mn3=sum(new_df$Age)/length(new_df$Agee)
diff3<-(new_df$Age-(mn3))
vr3<-sum(diff3^2/new_df$Age-1)
vr3
```

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

```
#Daily.Internet.Usage
mn4=sum(new_df$Daily.Internet.Usage)/length(new_df$Daily.Internet.Usage)
diff4<-(new_df$Daily.Daily.Internet.Usage-(mn4))
vr4<-sum(diff4^2/new_df$Daily.Internet.Usage-1)
vr4
```

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

Standard Deviation

The standard deviation is a summary measure of the differences of each observation from the mean

```
#Daily.Time.Spent.on.Site SD
sqrt(vr1)
```

```
## [1] 61.91092
```

```
#Age SD
sqrt(vr3)
```

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

```
#Area.Income SD
sqrt(vr2)
```

```
## [1] 2030.593
```

```
#Daily.Internet.UsageSD
sqrt(vr4)
```

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

Predictive univariate analysis

We analyse the properties of single variables and see their contributions towards predicting individuals most likely to click on our ads. Lets start by seeing number of people who clicked the ads.

```
#we change the data type of Clicked.on.Ad to character to syplify our analysis
new_df_Clicked.on.Ad <- transform(new_df,
                                  Clicked.on.Ad = as.character(Clicked.on.Ad))
ads_view<-new_df%>%
  count(Clicked.on.Ad,sort = TRUE)%>%
  view()
ads_view
```

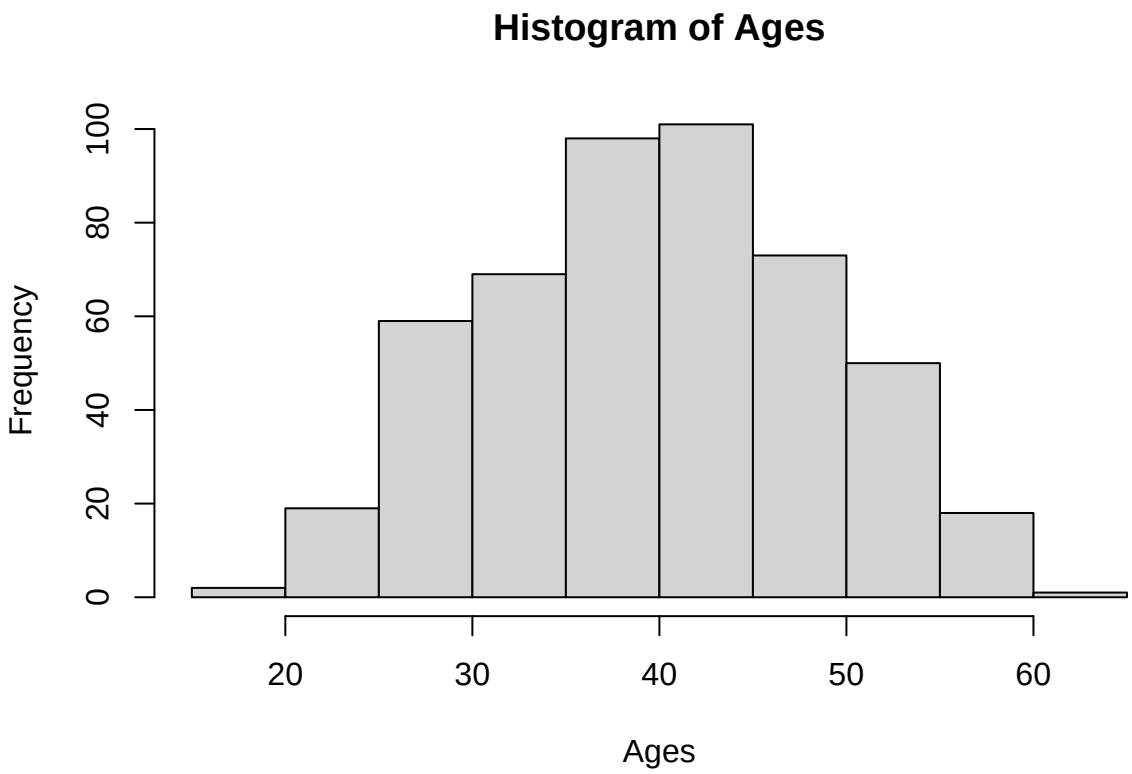
```
##   Clicked.on.Ad   n
## 1             0 500
## 2             1 490
```

We see that slightly less than half of the total individuals viewed our ad.Let make a derived data frame from this and explore the individuals that viewed the add

```
ad_viewers<-new_df[new_df$Clicked.on.Ad==1,]
head(ad_viewers,4)
```

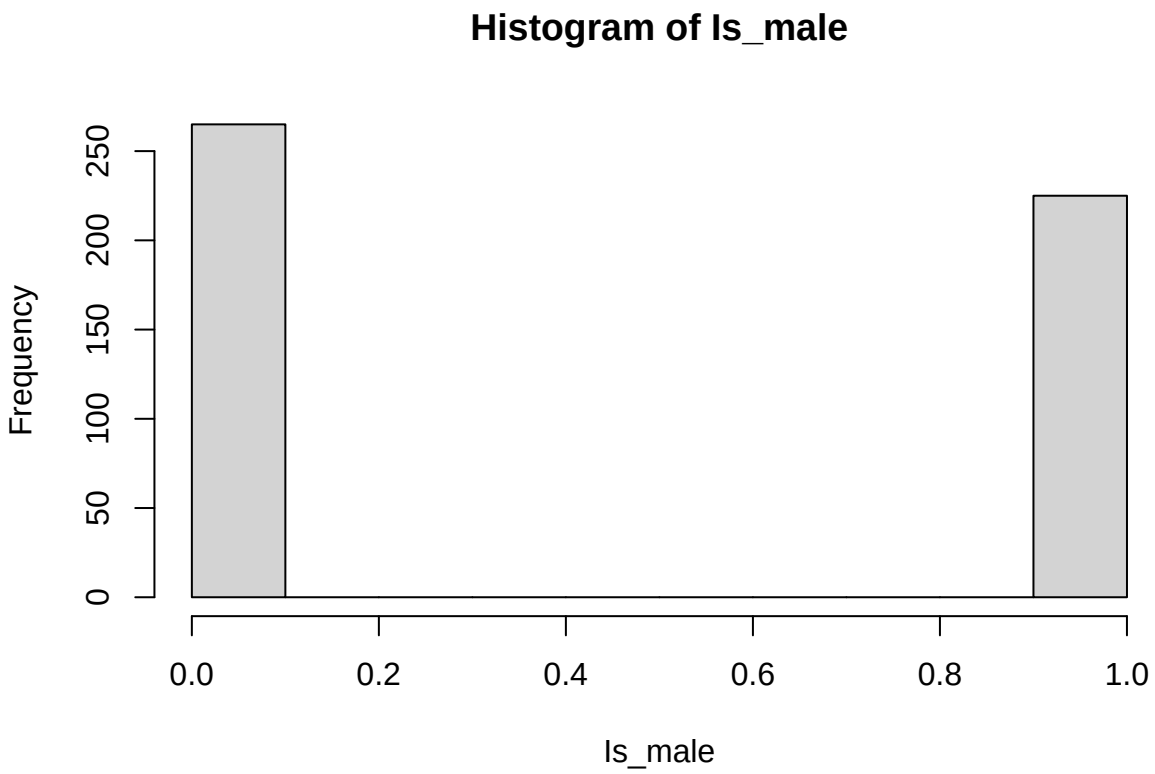
```
##    Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 8                     66.00  48    24593.33               131.76
## 11                    47.64  49    45632.51               122.02
## 13                    69.57  48    51636.92               113.12
## 15                    42.95  33    30976.00               143.56
##                           Ad.Topic.Line             City Male   Country
## 8               Reactive local challenge Port Jefferybury    1 Australia
## 11        Centralized neutral neural-net  West Brandonton    0     Qatar
## 13 Centralized content-based focus group   West Katiefurt    1     Egypt
## 15         Grass-roots coherent extranet     West William    0  Barbados
##              Timestamp Clicked.on.Ad
## 8  2016-03-07 01:40:15             1
## 11 2016-03-16 20:19:01             1
## 13 2016-06-03 01:14:41             1
## 15 2016-03-24 09:31:49             1
```

```
Ages=ad_viewers$Age
hist(Ages)
```



The age column distribution is normal most of the ad viewers age are uniformly distributed about the mean on either sides

```
new_df_Male<- transform(new_df,
                        Male = as.character(Male))
Is_male<-ad_viewers$Male
hist(Is_male)
```



Here we can conclude that most of the ad viewers were female as shown by the higher number of o count which represents not_male

```
ads_view<-new_df%>%
  count(Age,sort = TRUE)%>%
  view()
head(ads_view,10)
```

```
##    Age  n
## 1   31 60
## 2   36 49
## 3   28 48
## 4   29 48
## 5   33 42
## 6   34 39
## 7   35 39
## 8   30 38
## 9   26 37
## 10  32 37
```

Individuals between 26-40 are more frequent on clicking the ads from the frequency table above.

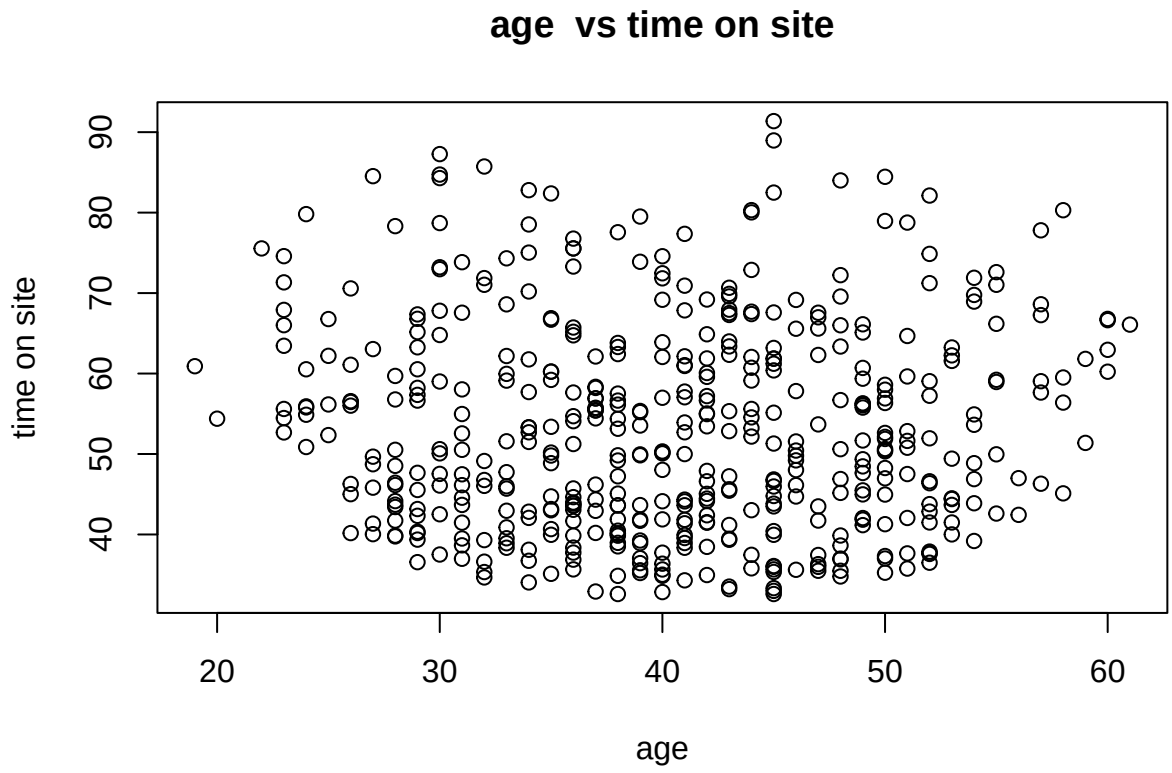
Bivariate Analysis

We can now see the relationship between our numerical variables by looking at their covariance coefficients and p-values, scatter plots helps with the visualizations.

Scatter plots

```
plot(x = ad_viewers$Age, y = ad_viewers$Daily.Time.Spent.on.Site,
    xlab = "age",
    ylab = "time on site",

    main = "age  vs time on site"
)
```




```
#similarly lets calculate their correlation coeficients

cor(ad_viewers$Age, ad_viewers$Daily.Time.Spent.on.Site, method = "pearson")
```

```
## [1] -0.01349623
```

The two variable have weak negative correlation .In this specific data set ,as age increases there is a slight decrease in time spent online,but in this case the decrease is too small almost insignificant.

```
#corelation test
cor.test(ad_viewers$Age, ad_viewers$Daily.Time.Spent.on.Site, method = "pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  ad_viewers$Age and ad_viewers$Daily.Time.Spent.on.Site
## t = -0.29817, df = 488, p-value = 0.7657
```

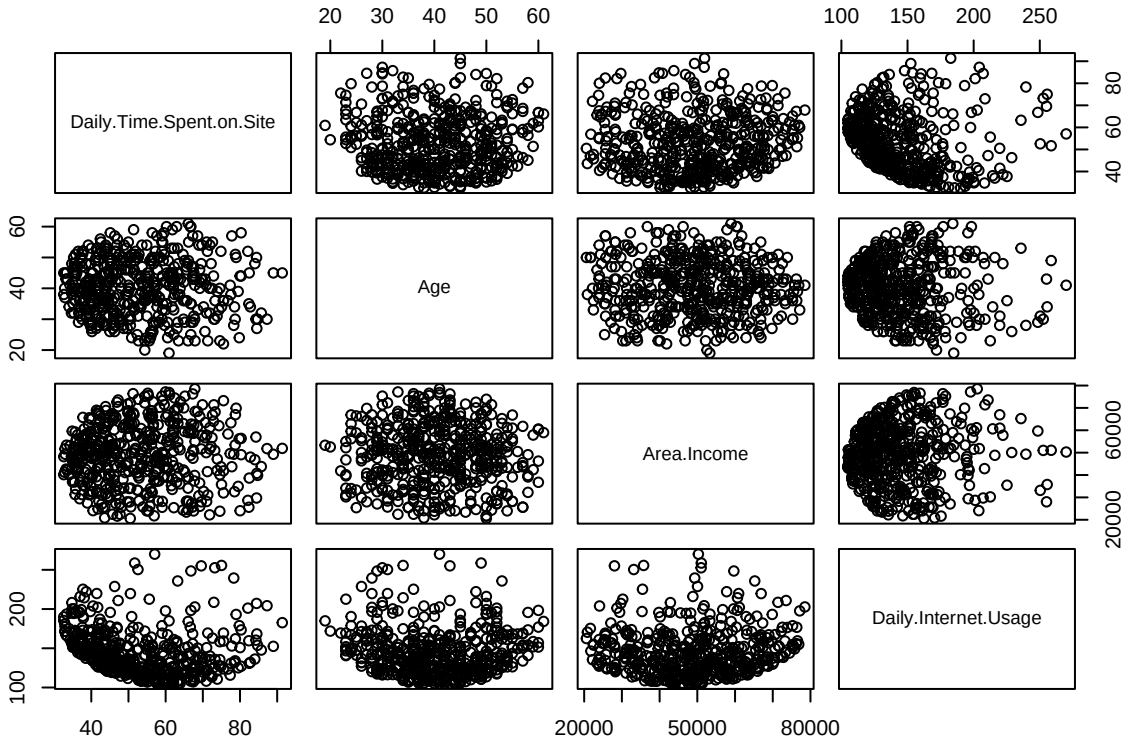
```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1019560  0.0751753
## sample estimates:
##         cor
## -0.01349623
```

For the two variables the p values is also high meaning that there is little evidence of relationship or difference between the two

```
#correlation matrix
#install.packages("corrplot")
#library("corrplot")
cor(ad_viewers[1:4])
```

```
##                           Daily.Time.Spent.on.Site         Age Area.Income
## Daily.Time.Spent.on.Site               1.00000000 -0.01349623  0.03239681
## Age                                   -0.01349623  1.00000000 -0.03395276
## Area.Income                            0.03239681 -0.03395276  1.00000000
## Daily.Internet.Usage                  -0.18497592 -0.05200544  0.04116358
##                           Daily.Internet.Usage
## Daily.Time.Spent.on.Site          -0.18497592
## Age                               -0.05200544
## Area.Income                        0.04116358
## Daily.Internet.Usage               1.00000000
```

```
pairs(ad_viewers[1:4])
```



From the matrix and the pair plots we can see that all of our numerical variables have very little correlation due to their very small correlation coefficients.

Models

we will now create a predictive model that will aid in predicting likenesses of any random person clicking on the add,

XG Boost is our model of choice,lets start by prepairing our data .

Feature engineering

```
#install.packages("xgboost")
require(xgboost)
```

```
## Loading required package: xgboost
```

```
##
## Attaching package: 'xgboost'
```

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

To start off we encode our categorical variables

```
#install.packages("mltools")
#install.packages("data.table")
#library(mltools)
#library(data.table)
```

```
sample <- sample(c(TRUE, FALSE), nrow(new_df), replace=TRUE, prob=c(0.7,0.3))
```

```
train  <- new_df[sample, ]
train<-train%>%select(-Ad.Topic.Line,-Country,-City,-Timestamp)
test   <- new_df[!sample, ]
test<-test%>%select(-Ad.Topic.Line,-Country,-City,-Timestamp)
```

```
X_train = train%>%select(-Clicked.on.Ad)
# independent variables for train
Y_train <-as.numeric(train$Clicked.on.Ad)

X_test = train%>%select(-Clicked.on.Ad)
Y_test <-as.numeric(train$Clicked.on.Ad)
```

We now run the model

```
library(xgboost)
model<-xgboost(data=as.matrix(X_train) ,
               label=(Y_train),
               nrounds =20,
               verbose=1,
               )
```

```
## [1]  train-rmse:0.360482
## [2]  train-rmse:0.265168
## [3]  train-rmse:0.200003
## [4]  train-rmse:0.154209
## [5]  train-rmse:0.124709
## [6]  train-rmse:0.108322
## [7]  train-rmse:0.096029
## [8]  train-rmse:0.082178
## [9]  train-rmse:0.070262
## [10] train-rmse:0.065316
## [11] train-rmse:0.061633
## [12] train-rmse:0.051657
## [13] train-rmse:0.043642
## [14] train-rmse:0.037463
## [15] train-rmse:0.032297
## [16] train-rmse:0.031220
## [17] train-rmse:0.026889
## [18] train-rmse:0.025672
## [19] train-rmse:0.023710
## [20] train-rmse:0.023151
```

```
attributes(model)
```

```
## $names
## [1] "handle"         "raw"            "niter"          "evaluation_log"
## [5] "call"           "params"         "callbacks"      "feature_names"
## [9] "nfeatures"
##
## $class
## [1] "xgb.Booster"
```

```
xgb.importance(model=model)
```

```
##                     Feature         Gain       Cover Frequency
## 1:     Daily.Internet.Usage 0.7049519694 0.222619627 0.2156863
## 2: Daily.Time.Spent.on.Site 0.1991539363 0.289287170 0.3088235
## 3:              Area.Income 0.0706179670 0.348051914 0.3186275
## 4:                      Age 0.0250137503 0.138963037 0.1446078
## 5:                     Male 0.0002623771 0.001078252 0.0122549
```

we can see that daily internet usage contributes almost 75% influence on the likeliness of clicking

Predictions and confusion matrix

```
pred<-predict(model,as.matrix(X_test))
pred<-as.numeric(pred)
```

```
library(caret)
```

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

```
##
## Attaching package: 'caret'
```

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

```
pred<-as.integer(pred>0.5)
confusionMatrix(table(pred, Y_test))
```

```
## Confusion Matrix and Statistics
##
##     Y_test
## pred   0   1
##    0 342   0
##    1   0 353
##
##                Accuracy : 1
##                  95% CI : (0.9947, 1)
##     No Information Rate : 0.5079
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
```

```
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.4921
##          Detection Rate : 0.4921
##    Detection Prevalence : 0.4921
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

CONCLUTION

From our finding we can hence conclude :

- Individuals between 26-40 are more frequent on clicking the ads.
- Females are more likely to click on our ads.