Advertisment Analysis

Wangechi Mungai

18/03/2022

1.Defining the Question

a) Specifying the Question

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.

b) Defining the Metric for Success

The analysis will be appraised successful by accessing the correlation and covariance between the target variable which is click on ad and the other variables. The variable which will have a correlation of 0.3 & covariance of 2 and above, will be considered as the variable which influences on the target variable.

c) Understanding the context

The Kenyan Entrepreneur would like to identifying the factors that influence the clicking of ads which is vital for the her.

d) Recording the Experimental Design

The following are the experimental design i took in order to complete this project:

1.Importing all the necessary libraries

2.Loading the dataset

3.Reading, cleaning the dataset

4.Performing Exploratory Data Analysis

e)Reading the Data

```
df <- read.csv("http://bit.ly/IPAdvertisingData")
```

f)Checking the Data

```
head(df)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    68.95  35    61833.90               256.09
## 2                    80.23  31    68441.85               193.77
## 3                    69.47  26    59785.94               236.50
## 4                    74.15  29    54806.18               245.89
## 5                    68.37  35    73889.99               225.58
## 6                    59.99  23    59761.56               226.74
##                           Ad.Topic.Line           City Male    Country
## 1    Cloned 5thgeneration orchestration    Wrightburgh    0    Tunisia
## 2    Monitored national standardization      West Jodi    1      Nauru
## 3      Organic bottom-line service-desk       Davidton    0 San Marino
## 4 Triple-buffered reciprocal time-frame West Terrifurt    1      Italy
## 5         Robust logistical utilization   South Manuel    0    Iceland
## 6       Sharable client-driven software      Jamieberg    1     Norway
##             Timestamp Clicked.on.Ad
## 1 2016-03-27 00:53:11             0
## 2 2016-04-04 01:39:02             0
## 3 2016-03-13 20:35:42             0
## 4 2016-01-10 02:31:19             0
## 5 2016-06-03 03:36:18             0
## 6 2016-05-19 14:30:17             0
```

```
#previewing tail of dataset
tail(df)
```

```
##      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
```

```
#taking a glance of the dataset
library(dplyr)
```

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

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

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

```
glimpse(df)
```

```
## Rows: 1,000
## Columns: 10
## $ Daily.Time.Spent.on.Site <dbl> 68.95, 80.23, 69.47, 74.15, 68.37, 59.99, 88.~
## $ Age                      <int> 35, 31, 26, 29, 35, 23, 33, 48, 30, 20, 49, 3~
## $ Area.Income              <dbl> 61833.90, 68441.85, 59785.94, 54806.18, 73889~
## $ Daily.Internet.Usage     <dbl> 256.09, 193.77, 236.50, 245.89, 225.58, 226.7~
## $ Ad.Topic.Line            <chr> "Cloned 5thgeneration orchestration", "Monito~
## $ City                     <chr> "Wrightburgh", "West Jodi", "Davidton", "West~
## $ Male                     <int> 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, ~
## $ Country                  <chr> "Tunisia", "Nauru", "San Marino", "Italy", "I~
## $ Timestamp                <chr> "2016-03-27 00:53:11", "2016-04-04 01:39:02",~
## $ Clicked.on.Ad            <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, ~
```

```
#previewing the shape of the dataset
dim(df)
```

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

The dataset contains 1000 rows and 10 columns

```
#previewing the descriptive statistics of dataset
summary(df)
```

```
##  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 datatypes of the columns
sapply(df, 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"
```

All datatypes are correct except timestamp which should be change to date datatype in data cleaning.

```
#checking for any null values
colSums(is.na(df))
```

```
## 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

```
#checking for duplicate values
anyDuplicated(df)
```

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

There are no duplicates in the dataset

g)Data Cleaning

```
#coverting timestamp column data type to date datatype
df$Timestamp <- as.Date(df$Timestamp)
class(df$Timestamp)
```

```
## [1] "Date"
```

```
#converting column names to lower case
colnames(df) = tolower(colnames(df))
colnames(df)
```

```
##  [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"
```

```
#checking for outliers for numerical cols
num_cols <- unlist(lapply(df, is.numeric))         # Identify numeric columns
num_cols
```

```
## daily.time.spent.on.site                      age              area.income
##                     TRUE                     TRUE                     TRUE
##     daily.internet.usage            ad.topic.line                     city
##                     TRUE                    FALSE                    FALSE
##                     male                  country                timestamp
##                     TRUE                    FALSE                    FALSE
##            clicked.on.ad
##                     TRUE
```

```
#displayig the numerical columns
df_num <- df[ , num_cols]
head(df_num, 5)
```

```
##   daily.time.spent.on.site age area.income daily.internet.usage male
## 1                    68.95  35    61833.90               256.09    0
## 2                    80.23  31    68441.85               193.77    1
## 3                    69.47  26    59785.94               236.50    0
## 4                    74.15  29    54806.18               245.89    1
## 5                    68.37  35    73889.99               225.58    0
##   clicked.on.ad
## 1             0
## 2             0
## 3             0
## 4             0
## 5             0
```

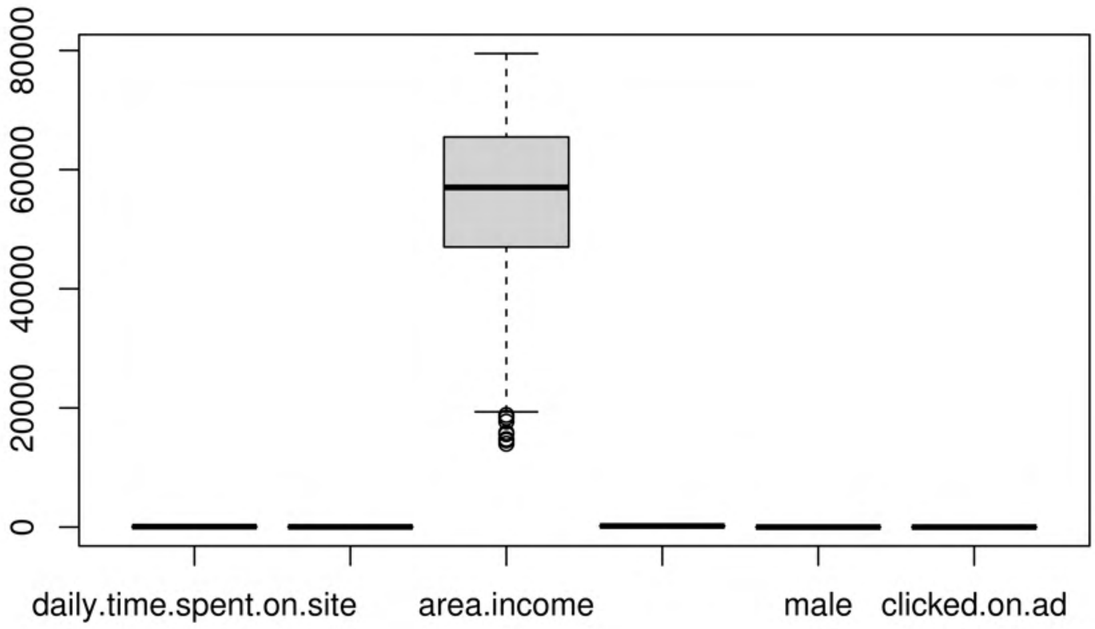
```
outlier <- function(x){
  out <- boxplot.stats(x)$out
  return((length(out)/  1000)*100)
}
```

```
# Get outlier count per column
sapply(df[,c("daily.time.spent.on.site","age","area.income","daily.internet.usage","male","clicked.on.a
```

```
## daily.time.spent.on.site                      age              area.income
##                      0.0                      0.0                      0.8
##     daily.internet.usage                     male            clicked.on.ad
##                      0.0                      0.0                      0.0
```

only the area income outliers has outliers.

```
 #visualizing the outiers
   boxplot(df_num)
```



there are a number of outliers in area income column but will not be removed since they are neccesary for our analysis

h) Exploratory Data Analysis Univariate Analysis

```
# describing our columns

psych::describe(df)
```

```
## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning Inf
```

```
## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf
```

```
##                          vars    n     mean       sd   median  trimmed      mad
## daily.time.spent.on.site    1 1000    65.00    15.85    68.22    65.74    17.92
## age                         2 1000    36.01     8.79    35.00    35.51     8.90
## area.income                 3 1000 55000.00 13414.63 57012.30 56038.94 13316.62
## daily.internet.usage        4 1000   180.00    43.90   183.13   179.99    58.61
## ad.topic.line*              5 1000   500.50   288.82   500.50   500.50   370.65
## city*                       6 1000   487.32   279.31   485.50   487.51   356.57
## male                        7 1000     0.48     0.50     0.00     0.48     0.00
## country*                    8 1000   116.41    69.94   114.50   115.82    89.70
## timestamp                   9 1000      NaN       NA       NA      NaN       NA
## clicked.on.ad              10 1000     0.50     0.50     0.50     0.50     0.74
##                               min      max    range  skew kurtosis     se
## daily.time.spent.on.site    32.60    91.43    58.83 -0.37    -1.10   0.50
## age                         19.00    61.00    42.00  0.48     -0.41   0.28
## area.income              13996.50 79484.80 65488.30 -0.65     -0.11 424.21
## daily.internet.usage       104.78   269.96   165.18 -0.03     -1.28   1.39
## ad.topic.line*               1.00  1000.00   999.00  0.00     -1.20   9.13
## city*                        1.00   969.00   968.00  0.00     -1.19   8.83
## male                         0.00     1.00     1.00  0.08     -2.00   0.02
## country*                     1.00   237.00   236.00  0.08     -1.23   2.21
## timestamp                     Inf     -Inf     -Inf    NA        NA     NA
## clicked.on.ad                0.00     1.00     1.00  0.00     -2.00   0.02
```

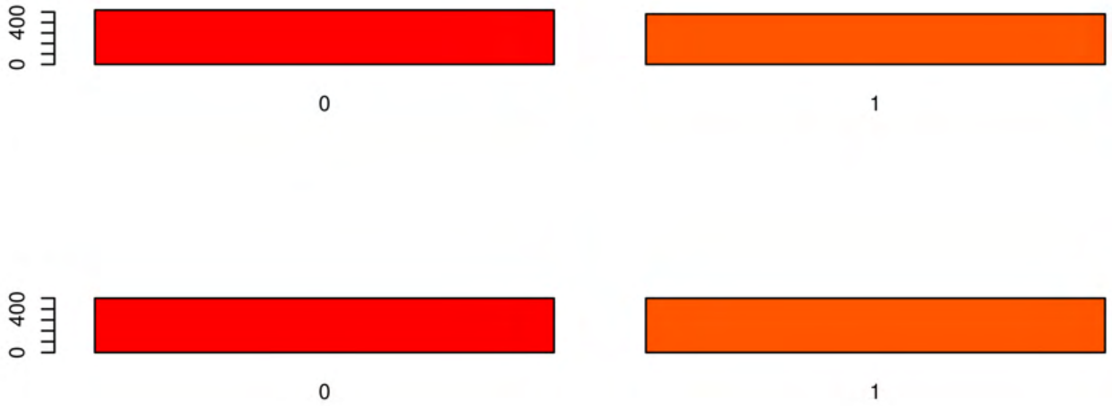
the variables are fairly skewed

```
# Frequency distribution of the categorical variables
sapply(df[, c("male","clicked.on.ad")], table)
```

```
##   male clicked.on.ad
## 0  519           500
## 1  481           500
```

The male were 481 while those who are not male were 519. Those who clicked on ads and those who did not click had an equal number which is 500

```
# Creating histogram plots to visually view the categorical variables
par(mfrow=c(3,1))
categ <- c("male","clicked.on.ad")
for(i in categ) {
    counts <- table(df[,i])
    name <- names(df)[i]
    barplot(counts, main=name, col = heat.colors(5))}
```



```
#finding mean of age column
x <-mean(df$age)
print(paste(x, "is the mean for age column"))
```

```
## [1] "36.009 is the mean for age column"
```

```
#median
y <-median(df$age)
print(paste(y, "is the median for age column"))
```

```
## [1] "35 is the median for age column"
```

```
#mode
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
z <- getmode(df$age)
print(paste(z, "is the mode for age column"))
```

```
## [1] "31 is the mode for age column"
```

```r
#minimun
a <-min(df$age)
print(paste(a, "is the minimum value for age column"))
```

```
## [1] "19 is the minimum value for age column"
```

```r
#maximum
b <-max(df$age)
print(paste(b, "is the maximum value for age column"))
```

```
## [1] "61 is the maximum value for age column"
```

```r
#range

c <-range(df$age)
print(paste(c, "is the range for age column"))
```

```
## [1] "19 is the range for age column" "61 is the range for age column"
```

```r
#quarntile
c <-range(df$age)
print(paste(c, "is the quarntile for age column"))
```

```
## [1] "19 is the quarntile for age column" "61 is the quarntile for age column"
```

```r
#standard deviation
d <-sd(df$age)
print(paste(c, "is the standard dev for age column"))
```

```
## [1] "19 is the standard dev for age column"
## [2] "61 is the standard dev for age column"
```

```r
#variance
e <-var(df$age)
print(paste(e, "is the var for age column"))
```

```
## [1] "77.1861051051051 is the var for age column"
```

```r
#finding mean of daily.time.spent.on.site column
x <-mean(df$daily.time.spent.on.site)
print(paste(x, "is the mean for daily.time.spent.on.site column"))
```

```
## [1] "65.0002 is the mean for daily.time.spent.on.site column"
```

```r
#median
y <-median(df$daily.time.spent.on.site)
print(paste(y, "is the median for daily.time.spent.on.site column"))
```

```
## [1] "68.215 is the median for daily.time.spent.on.site column"
```

```r
#mode
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
z <- getmode(df$daily.time.spent.on.site)
print(paste(z, "is the mode for daily.time.spent.on.site column"))
```

```
## [1] "62.26 is the mode for daily.time.spent.on.site column"
```

```r
#minimun
a <-min(df$daily.time.spent.on.site)
print(paste(a, "is the minimum value for daily.time.spent.on.site column"))
```

```
## [1] "32.6 is the minimum value for daily.time.spent.on.site column"
```

```r
#maximum
b <-max(df$daily.time.spent.on.site)
print(paste(b, "is the maximum value for daily.time.spent.on.site column"))
```

```
## [1] "91.43 is the maximum value for daily.time.spent.on.site column"
```

```r
#range

c <-range(df$daily.time.spent.on.site)
print(paste(c, "is the range for daily.time.spent.on.site column"))
```

```
## [1] "32.6 is the range for daily.time.spent.on.site column"
## [2] "91.43 is the range for daily.time.spent.on.site column"
```

```r
#quarntile
c <-range(df$daily.time.spent.on.site)
print(paste(c, "is the quarntile for daily.time.spent.on.site column"))
```

```
## [1] "32.6 is the quarntile for daily.time.spent.on.site column"
## [2] "91.43 is the quarntile for daily.time.spent.on.site column"
```

```r
#standard deviation
d <-sd(df$daily.time.spent.on.site)
print(paste(c, "is the standard dev for daily.time.spent.on.site column"))
```

```
## [1] "32.6 is the standard dev for daily.time.spent.on.site column"
## [2] "91.43 is the standard dev for daily.time.spent.on.site column"
```

```r
#variance
e <-var(df$daily.time.spent.on.site)
print(paste(e, "is the var for daily.time.spent.on.site column"))
```

```
## [1] "251.337094854855 is the var for daily.time.spent.on.site column"
```

```
#finding mean of daily.internet.usage
x <-mean(df$daily.internet.usage)
print(paste(x, "is the mean for daily.internet.usage column"))
```

```
## [1] "180.0001 is the mean for daily.internet.usage column"
```

```
#median
y <-median(df$daily.internet.usage)
print(paste(y, "is the median for daily.internet.usage column"))
```

```
## [1] "183.13 is the median for daily.internet.usage column"
```

```
#mode
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
z <- getmode(df$daily.internet.usage)
print(paste(z, "is the mode for daily.internet.usage column"))
```

```
## [1] "167.22 is the mode for daily.internet.usage column"
```

```
#minimun
a <-min(df$daily.internet.usage)
print(paste(a, "is the minimum value for daily.internet.usage column"))
```

```
## [1] "104.78 is the minimum value for daily.internet.usage column"
```

```
#maximum
b <-max(df$daily.internet.usage)
print(paste(b, "is the maximum value for daily.internet.usage column"))
```

```
## [1] "269.96 is the maximum value for daily.internet.usage column"
```

```
#range

c <-range(df$daily.internet.usage)
print(paste(c, "is the range for daily.internet.usage column"))
```

```
## [1] "104.78 is the range for daily.internet.usage column"
## [2] "269.96 is the range for daily.internet.usage column"
```

```
#quarntile
c <-range(df$daily.internet.usage)
print(paste(c, "is the quarntile for daily.internet.usage column"))
```

```
## [1] "104.78 is the quarntile for daily.internet.usage column"
## [2] "269.96 is the quarntile for daily.internet.usage column"
```

```r
#standard deviation
d <-sd(df$daily.internet.usage)
print(paste(c, "is the standard dev for daily.internet.usage column"))
```

```
## [1] "104.78 is the standard dev for daily.internet.usage column"
## [2] "269.96 is the standard dev for daily.internet.usage column"
```

```r
#variance
e <-var(df$daily.internet.usage)
print(paste(e, "is the var for daily.internet.usage column"))
```

```
## [1] "1927.41539618619 is the var for daily.internet.usage column"
```

```r
#finding mean of clicked.on.ad
x <-mean(df$clicked.on.ad)
print(paste(x, "is the mean for clicked.on.ad column"))
```

```
## [1] "0.5 is the mean for clicked.on.ad column"
```

```r
#median
y <-median(df$clicked.on.ad)
print(paste(y, "is the median for clicked.on.ad column"))
```

```
## [1] "0.5 is the median for clicked.on.ad column"
```

```r
#mode
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
z <- getmode(df$clicked.on.ad)
print(paste(z, "is the mode for clicked.on.ad column"))
```

```
## [1] "0 is the mode for clicked.on.ad column"
```

```r
#minimun
a <-min(df$clicked.on.ad)
print(paste(a, "is the minimum value for clicked.on.ad column"))
```

```
## [1] "0 is the minimum value for clicked.on.ad column"
```

```r
#maximum
b <-max(df$clicked.on.ad)
print(paste(b, "is the maximum value for clicked.on.ad column"))
```

```
## [1] "1 is the maximum value for clicked.on.ad column"
```

```
#range

c <-range(df$clicked.on.ad)
print(paste(c, "is the range for clicked.on.ad column"))
```

```
## [1] "0 is the range for clicked.on.ad column"
## [2] "1 is the range for clicked.on.ad column"
```

```
#quarntile
c <-range(df$clicked.on.ad)
print(paste(c, "is the quarntile for clicked.on.ad column"))
```

```
## [1] "0 is the quarntile for clicked.on.ad column"
## [2] "1 is the quarntile for clicked.on.ad column"
```

```
#standard deviation
d <-sd(df$clicked.on.ad)
print(paste(c, "is the standard dev for clicked.on.ad column"))
```

```
## [1] "0 is the standard dev for clicked.on.ad column"
## [2] "1 is the standard dev for clicked.on.ad column"
```

```
#variance
e <-var(df$clicked.on.ad)
print(paste(e, "is the var for clicked.on.ad column"))
```

```
## [1] "0.25025025025025 is the var for clicked.on.ad column"
```

```
#finding mean of male

x <-mean(df$male)
print(paste(x, "is the mean for male column"))
```

```
## [1] "0.481 is the mean for male column"
```

```
#median
y <-median(df$male)
print(paste(y, "is the median for male column"))
```

```
## [1] "0 is the median for male column"
```

```
#mode
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
z <- getmode(df$male)
print(paste(z, "is the mode for male column"))
```

```
## [1] "0 is the mode for male column"
```

```r
#minimun
a <-min(df$male)
print(paste(a, "is the minimum value for male column"))
```

```
## [1] "0 is the minimum value for male column"
```

```r
#maximum
b <-max(df$male)
print(paste(b, "is the maximum value for male column"))
```

```
## [1] "1 is the maximum value for male column"
```

```r
#range

c <-range(df$male)
print(paste(c, "is the range for male column"))
```

```
## [1] "0 is the range for male column" "1 is the range for male column"
```

```r
#quarntile
c <-range(df$male)
print(paste(c, "is the quarntile for male column"))
```

```
## [1] "0 is the quarntile for male column" "1 is the quarntile for male column"
```

```r
#standard deviation
d <-sd(df$male)
print(paste(c, "is the standard dev for male column"))
```

```
## [1] "0 is the standard dev for male column"
## [2] "1 is the standard dev for male column"
```

```r
#variance
e <-var(df$male)
print(paste(e, "is the var for male column"))
```

```
## [1] "0.249888888888889 is the var for male column"
```

```r
#finding mean of area.income

x <-mean(df$area.income)
print(paste(x, "is the mean for area.income column"))
```

```
## [1] "55000.00008 is the mean for area.income column"
```

```r
#median
y <-median(df$area.income)
print(paste(y, "is the median for area.income column"))
```

```
## [1] "57012.3 is the median for area.income column"
```

```r
#mode
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
z <- getmode(df$area.income)
print(paste(z, "is the mode for area.income column"))
```

```
## [1] "61833.9 is the mode for area.income column"
```

```r
#minimun
a <-min(df$area.income)
print(paste(a, "is the minimum value for area.income column"))
```

```
## [1] "13996.5 is the minimum value for area.income column"
```

```r
#maximum
b <-max(df$area.income)
print(paste(b, "is the maximum value for area.income column"))
```

```
## [1] "79484.8 is the maximum value for area.income column"
```

```r
#range

c <-range(df$area.income)
print(paste(c, "is the range for area.income column"))
```

```
## [1] "13996.5 is the range for area.income column"
## [2] "79484.8 is the range for area.income column"
```

```r
#quarntile
c <-range(df$area.income)
print(paste(c, "is the quarntile for area.income column"))
```

```
## [1] "13996.5 is the quarntile for area.income column"
## [2] "79484.8 is the quarntile for area.income column"
```

```r
#standard deviation
d <-sd(df$area.income)
print(paste(c, "is the standard dev for area.income column"))
```

```
## [1] "13996.5 is the standard dev for area.income column"
## [2] "79484.8 is the standard dev for area.income column"
```

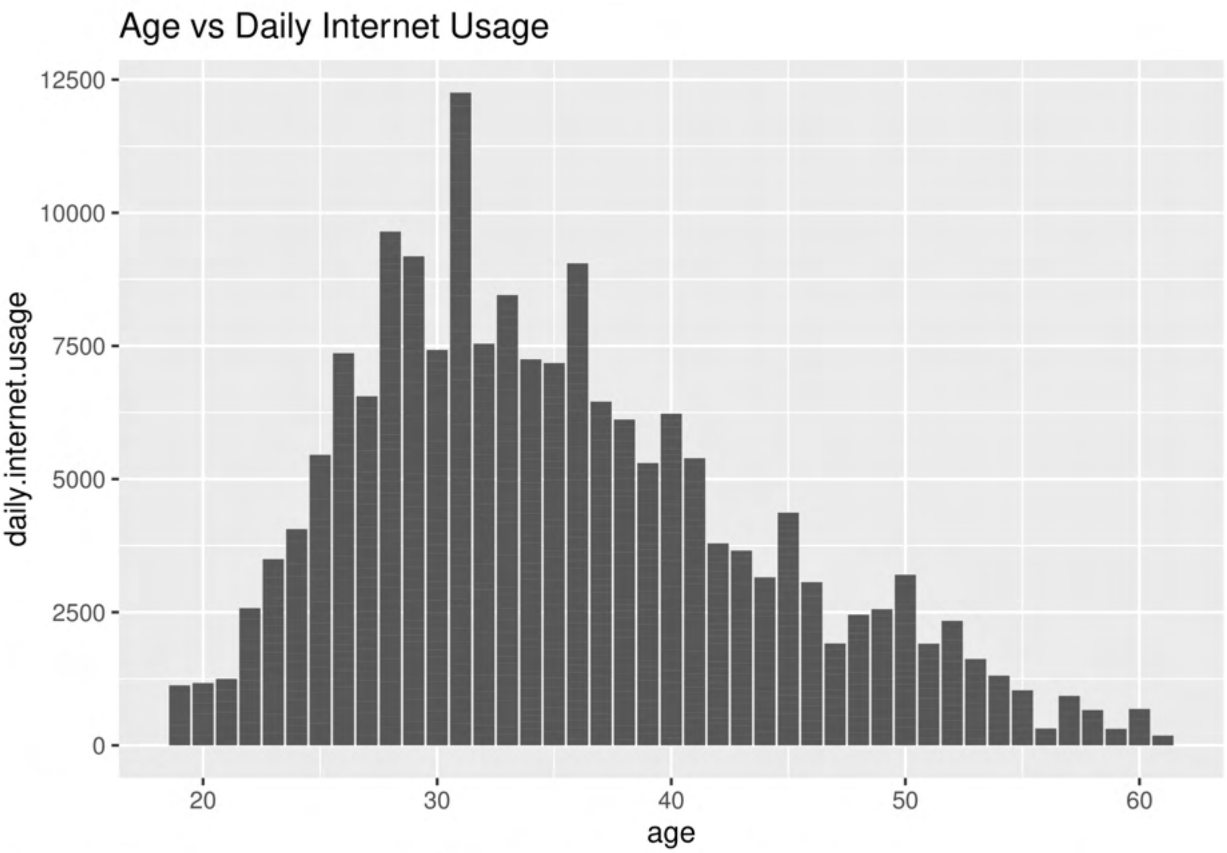
```r
#variance
e <-var(df$area.income)
print(paste(e, "is the var for area.income column"))
```

```
## [1] "179952405.951775 is the var for area.income column"
```

Bivariate analysis

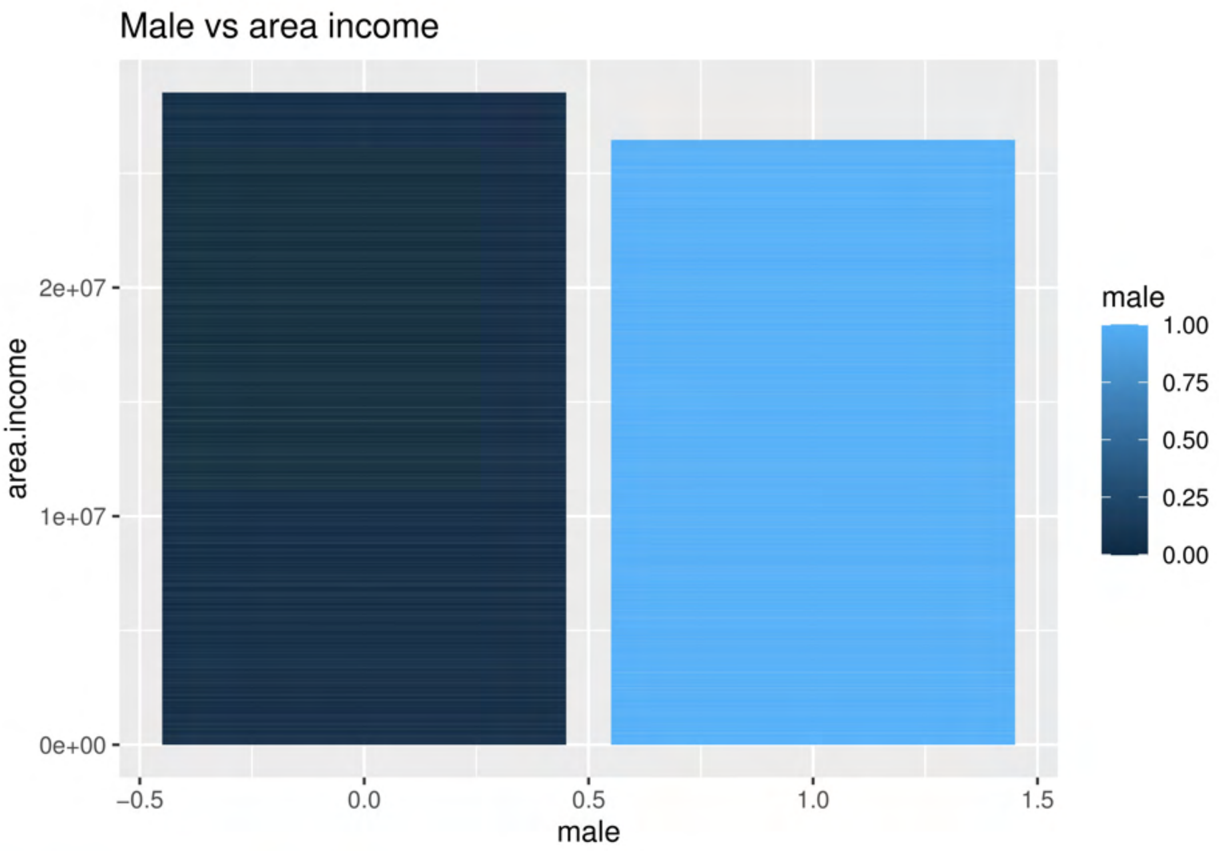
```
library(ggplot2)
```

```
p<-ggplot(data=df, aes(x=age, y=daily.internet.usage)) +
  geom_bar(stat="identity" ) + ggtitle("Age vs Daily Internet Usage")
p
```

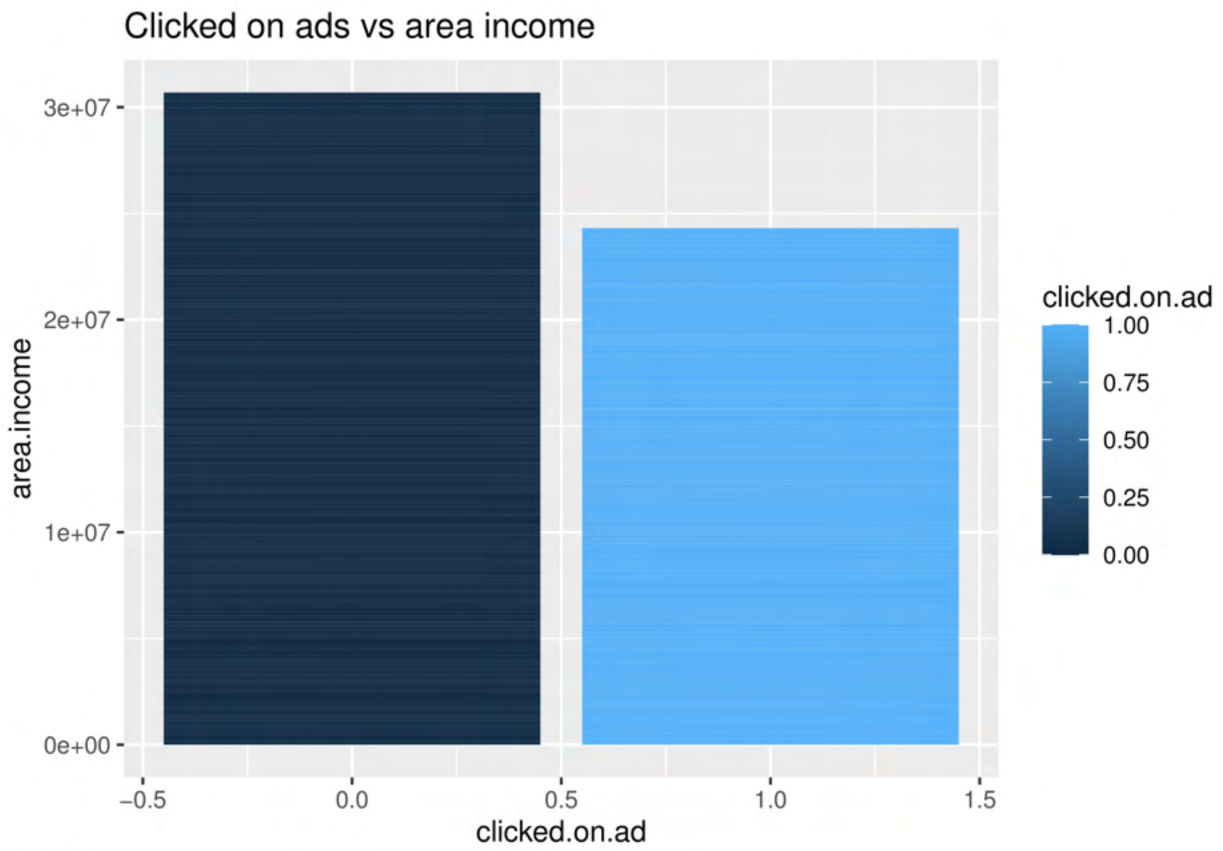


The age between 20 and 45 have the highest number of records of daily internet usage

```
#gender vs area income
p<-ggplot(data=df, aes(x=male, y=area.income,fill=male)) +
  geom_bar(stat="identity") + ggtitle("Male vs area income")
p
```



```
#clicked on ad vs area income
p<-ggplot(data=df, aes(x=clicked.on.ad
, y=area.income, fill=clicked.on.ad)) +
  geom_bar(stat="identity") + ggtitle("Clicked on ads vs area income")
p
```



```
#checking the covariance between age an click on ad
x <- df$age
y <- df$clicked.on.ad
cov(x,y)
```

```
## [1] 2.164665
```

```
cor(x,y)
```

```
## [1] 0.4925313
```

there is a positive linear relationship between the 2 variables as well as some correlation between the variables

```
#checking the covariance and correlation between area income and click on ad
x <- df$area.income
y <- df$clicked.on.ad
cov(x,y)
```

```
## [1] -3195.989
```

```
cor(x,y)
```

```
## [1] -0.4762546
```

there is a negative linear relationship between the 2 variables as well as a very low correlation between the variables

```
#checking the covariance and correlation between male and click on ad
x <- df$male
y <- df$clicked.on.ad
cov(x,y)
```

```
## [1] -0.00950951
```

```
cor(x,y)
```

```
## [1] -0.03802747
```

there is a negative linear relationship between the 2 variables as well as a very low correlation between the variables

```
#checking the covariance and correlation between daily.time.spent.on.site and click on ad
x <- df$daily.time.spent.on.site
y <- df$clicked.on.ad
cov(x,y)
```

```
## [1] -5.933143
```

```
cor(x,y)
```

```
## [1] -0.7481166
```

there is a negative linear relationship between the 2 variables as well as a very low correlation between the variables

```
#checking the covariance and correlation between daily.internet.usage and click on ad
x <- df$daily.internet.usage
y <- df$clicked.on.ad
cov(x,y)
```

```
## [1] -17.27409
```

```
cor(x,y)
```

```
## [1] -0.7865392
```

there is a negative linear relationship between the 2 variables as well as a very low correlation between the variables

Conclusion And Recommendation

From the analysis done, we have realized that Age has a great impact on the click of ads. It has the highest covariance and correlation: 2.164665 & 0.4925313 respectively.While area income had the least covariance and correlation which means it totally has no influence on clicking of ads.

As a data science consultant, I would advice the entrepreneur to focus her ads on those between 20-45 yearsand be creative enough to keep them interested