Identifying Individuals Likely to Click on Ads

Kibuye Joshua

2022-03-19

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

Plotting relevant Univariate and Bivariate to identify trends in the dataset. ### c) Understanding the context We are looking at factors contributing to people clicking ads ### d) Recording the Experimental Design 1. Data loading 2. Data Cleaning 3. Exploratory data analysis ### e) Data Relevance ## 2. Reading the Data

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

3. Checking the Data

```
#Viewing the dataset
View(df)
```

```
# Determining the no. of records in our dataset
dim(df)
```

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

There are 1000 records and 10 variables

```
# Previewing the top of our dataset
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 the bottom of our 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
```

```
# Checking whether each column has an appropriate datatype
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"
```

Timestamp should be changed to date time format

4. External Data Source Validation

Making sure your data matches something outside of the dataset is very important. It allows you to ensure that the measurements are roughly in line with what they should be and it serves as a check on what other things might be wrong in your dataset. External validation can often be as simple as checking your data against a single number, as we will do here.

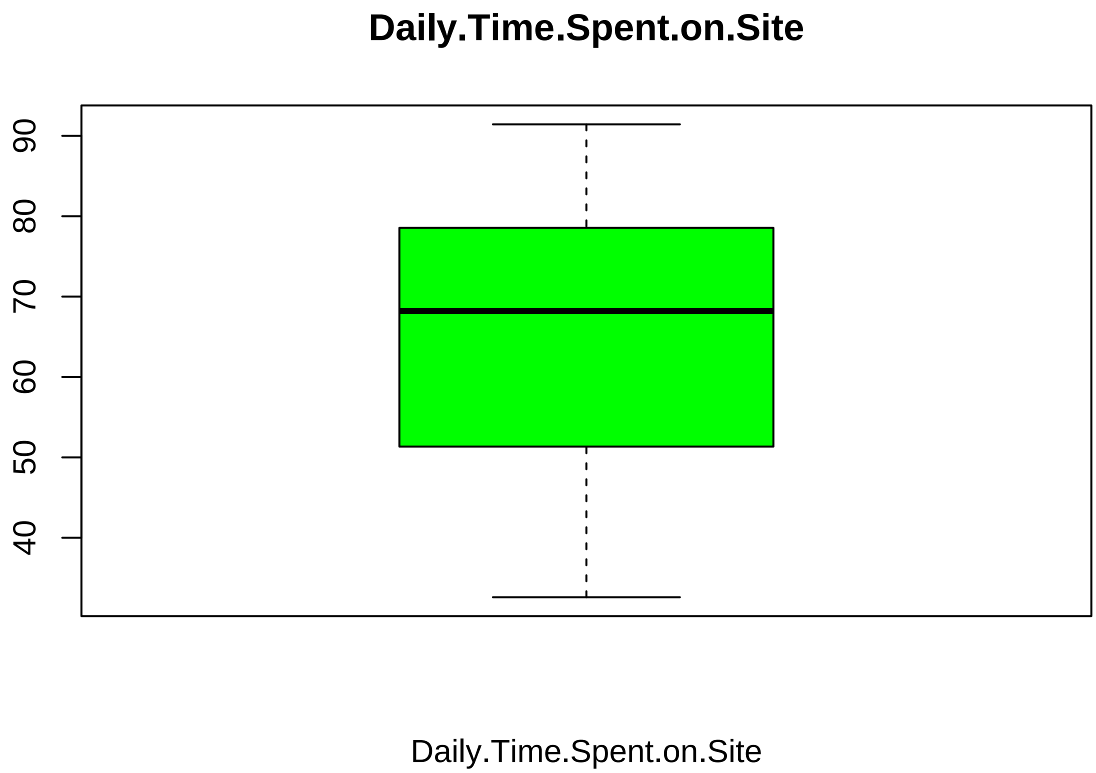
5. Tidying the Dataset

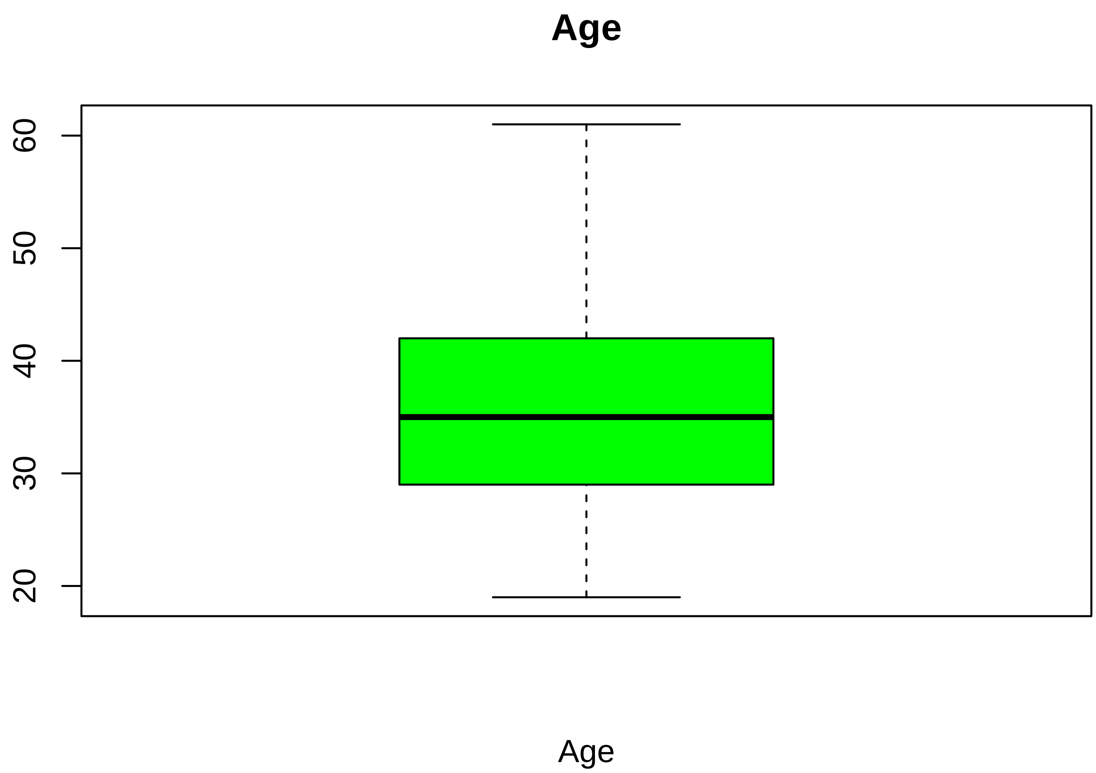
```r
#Defining numerical columns
library(dplyr)
```

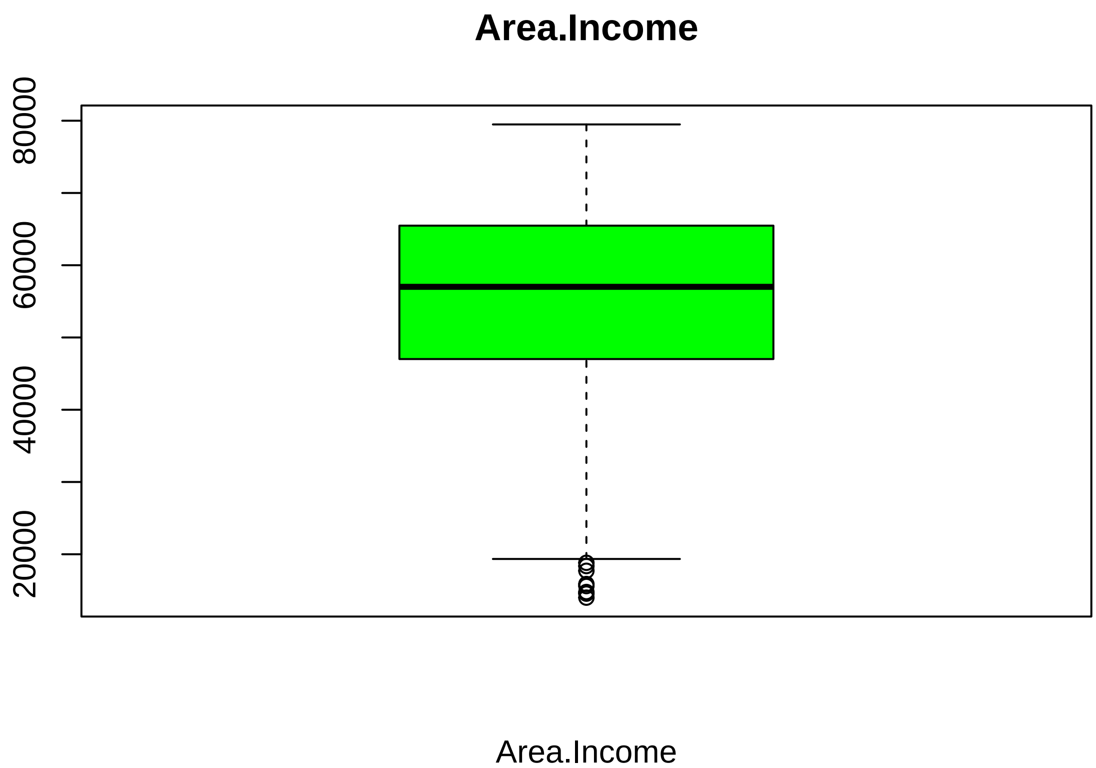
```
##
## Attaching package: 'dplyr'

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

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

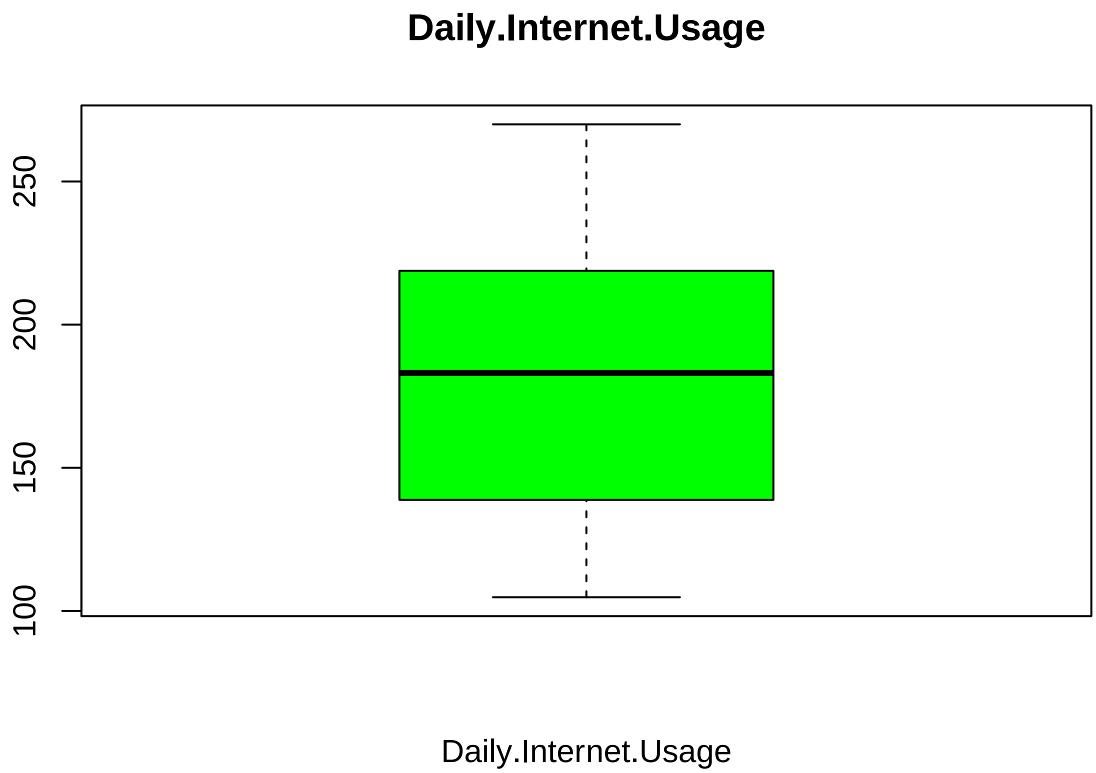
```r
numeric <- df%>%select_if(is.numeric)
```

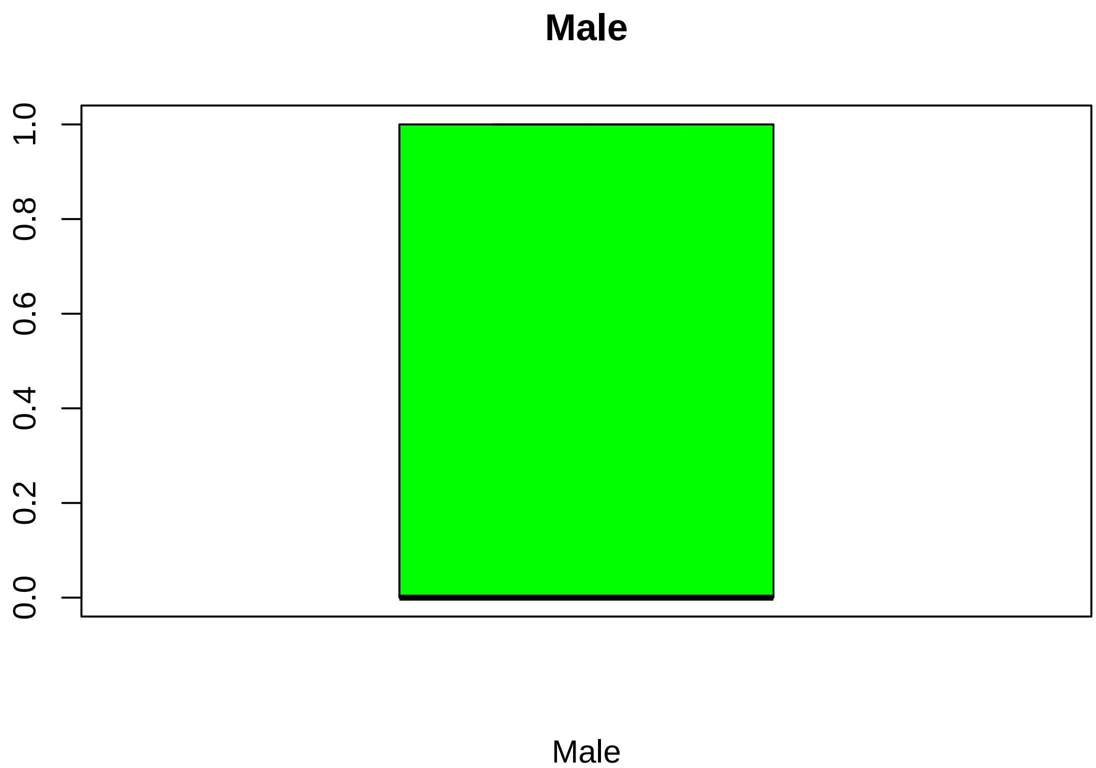
```r
for(i in 1:6) {
    boxplot(numeric[,i], main=names(numeric)[i], xlab=names(numeric)[i],col = "green")}
```

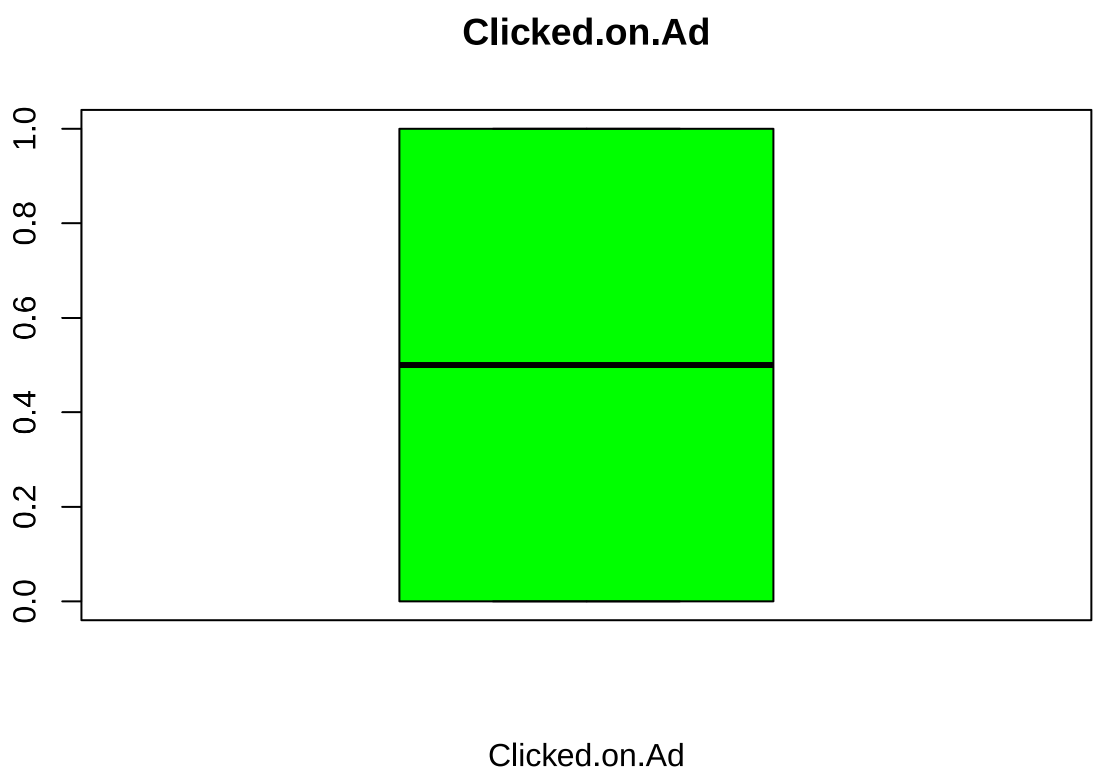

Daily.Time.Spent.on.Site
90
80
70
60
50
40
Daily.Time.Spent.on.Site


Age
60
50
40
30
20
Age


Area.Income
80000
60000
40000
20000
Area.Income


Daily.Internet.Usage
250
200
150
100
Daily.Internet.Usage


Male
1.0
0.8
0.6
0.4
0.2
0.0
Male



The outliers exist in the area income column. However, we do not drop them as they are true values.

```
#Checking for missing data
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

```
#Checking for duplicates
sum(duplicated(df))
```

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

There are no duplicates in the dataset.

```
#Changing column names to lowercase
names(df) <- tolower(names(df))
```

```r
#Changing TimeStamp to datetime
df$timestamp <- as.Date(df$timestamp)
```

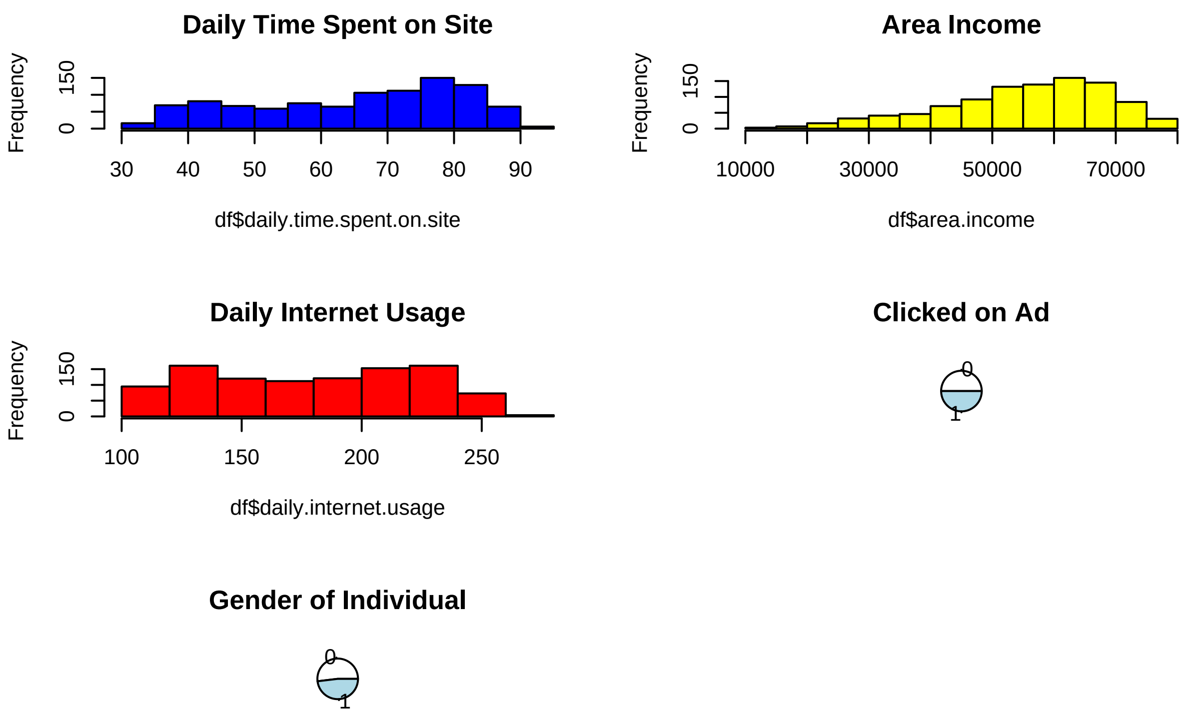
```r
#Checking if changes have applied
head(df)
```

```
##   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             0
## 2 2016-04-04             0
## 3 2016-03-13             0
## 4 2016-01-10             0
## 5 2016-06-03             0
## 6 2016-05-19             0
```

6. Exploratory Analysis

6.1 Univariate analysis

```r
#Plotting univariate
par(mfrow=c(3,2))
hist(df$daily.time.spent.on.site, main = "Daily Time Spent on Site", col = "blue")
hist(df$area.income, main = "Area Income", col = "yellow")
hist(df$daily.internet.usage, main = "Daily Internet Usage", col = "red")
pie(table(df$clicked.on.ad), main = "Clicked on Ad")
pie(table(df$male), main = "Gender of Individual")
```



Measures of Dispersion and Central Tendencies

```
#Checking the mean age
mean(df$age)
```

```
## [1] 36.009
```

```
mean(df$area.income)
```

```
## [1] 55000
```

6.2 Bivariate Analysis

```
#Finding the covariance of numerical columns
cov(numeric)
```

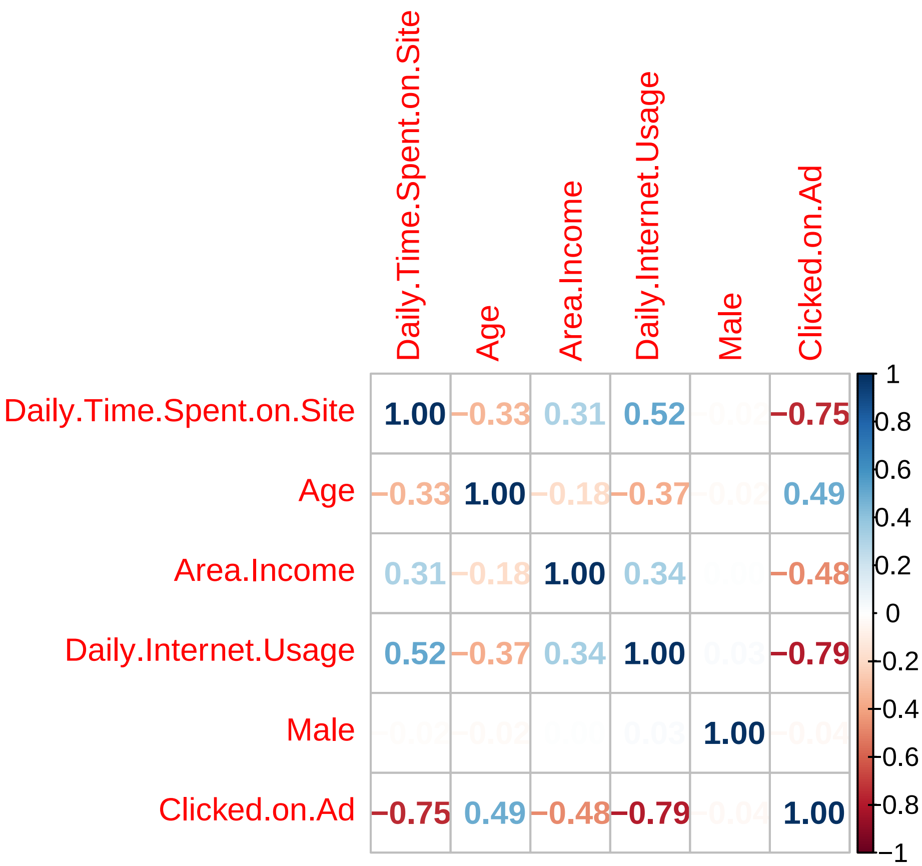
```
##                          Daily.Time.Spent.on.Site           Age   Area.Income
## Daily.Time.Spent.on.Site              251.3370949 -4.617415e+01  6.613081e+04
## Age                                   -46.1741459  7.718611e+01 -2.152093e+04
## Area.Income                         66130.8109082 -2.152093e+04  1.799524e+08
## Daily.Internet.Usage                  360.9918827 -1.416348e+02  1.987625e+05
## Male                                   -0.1501864 -9.242142e-02  8.867509e+00
## Clicked.on.Ad                          -5.9331431  2.164665e+00 -3.195989e+03
```

```
##                               Daily.Internet.Usage        Male Clicked.on.Ad
## Daily.Time.Spent.on.Site              3.609919e+02 -0.15018639 -5.933143e+00
## Age                                  -1.416348e+02 -0.09242142  2.164665e+00
## Area.Income                           1.987625e+05  8.86750903 -3.195989e+03
## Daily.Internet.Usage                  1.927415e+03  0.61476667 -1.727409e+01
## Male                                  6.147667e-01  0.24988889 -9.509510e-03
## Clicked.on.Ad                        -1.727409e+01 -0.00950951  2.502503e-01
```

```r
#Correlation plot
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
corr <-cor(numeric)
corrplot(corr, method = "number")
```



6.3 Multivariate Analysis

```r
# Converting the target as a factor

df$clicked.on.ad = factor(df$clicked.on.ad, levels = c(0,1))

# checking the variable datatypes

sapply(df, class)
```

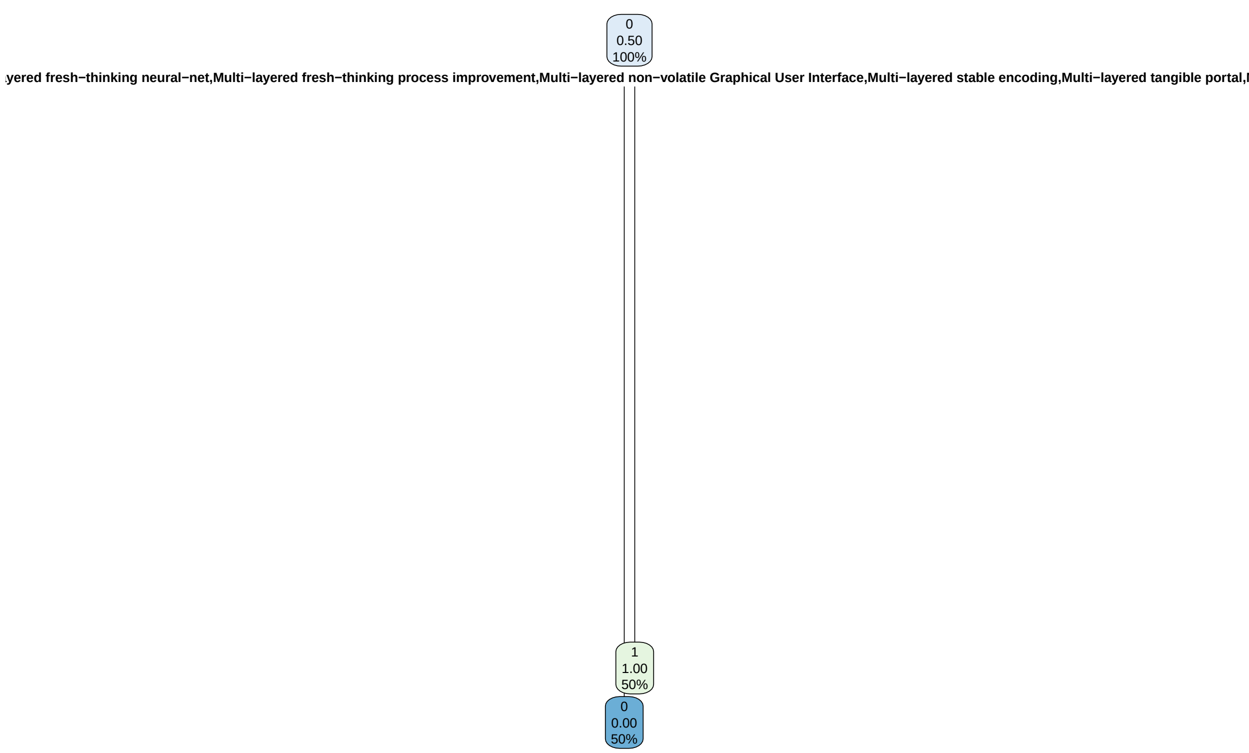
```
## daily.time.spent.on.site                     age            area.income
##                 "numeric"               "integer"              "numeric"
##      daily.internet.usage           ad.topic.line                   city
##                 "numeric"             "character"            "character"
##                      male                 country              timestamp
##                 "integer"             "character"                 "Date"
##              clicked.on.ad
##                  "factor"
```

7. Implementing the Solution

7.1 Decision Tree Classifier

```r
# Using decision tree
# Fitting the model
# Specifying the target and predictor variables
library(rpart)
m <- rpart(clicked.on.ad ~ . ,
    data = df,
    method = "class")
```

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



```r
# Making predictions
# Printing the confusion matrix
p <- predict(m, df, type ="class")
table(p, df$clicked.on.ad)
```

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

All the predictions were correct.

```r
# Printing the Accuracy
mean(df$clicked.on.ad == p)
```

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

- The model accuracy is 100% which is good to make predictions.

8. Challenging the Solution

Random Forest Classifier

```r
# Training the model
# Setting seed for randomness
library(caret)
```

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

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

```r
set.seed(12)
model <- train(clicked.on.ad ~. ,
                data = df,
                method = "ranger")
```

```
## Growing trees.. Progress: 83%. Estimated remaining time: 6 seconds.
## Growing trees.. Progress: 76%. Estimated remaining time: 9 seconds.
## Growing trees.. Progress: 87%. Estimated remaining time: 4 seconds.
## Growing trees.. Progress: 72%. Estimated remaining time: 12 seconds.
## Growing trees.. Progress: 93%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 87%. Estimated remaining time: 4 seconds.
## Growing trees.. Progress: 84%. Estimated remaining time: 5 seconds.
## Growing trees.. Progress: 93%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 94%. Estimated remaining time: 1 seconds.
## Growing trees.. Progress: 93%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 73%. Estimated remaining time: 11 seconds.
## Growing trees.. Progress: 81%. Estimated remaining time: 7 seconds.
```

```
## Growing trees.. Progress: 100%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 68%. Estimated remaining time: 14 seconds.
## Growing trees.. Progress: 78%. Estimated remaining time: 8 seconds.
## Growing trees.. Progress: 92%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 67%. Estimated remaining time: 15 seconds.
## Growing trees.. Progress: 90%. Estimated remaining time: 3 seconds.
## Growing trees.. Progress: 69%. Estimated remaining time: 13 seconds.
## Growing trees.. Progress: 89%. Estimated remaining time: 3 seconds.
## Growing trees.. Progress: 97%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 99%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 93%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 92%. Estimated remaining time: 2 seconds.
## Growing trees.. Progress: 94%. Estimated remaining time: 1 seconds.
## Growing trees.. Progress: 86%. Estimated remaining time: 5 seconds.
## Growing trees.. Progress: 87%. Estimated remaining time: 4 seconds.
## Growing trees.. Progress: 100%. Estimated remaining time: 0 seconds.
## Growing trees.. Progress: 95%. Estimated remaining time: 1 seconds.
```

```r
# Printing the model
model
```

```
## Random Forest
##
## 1000 samples
##    9 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 1000, 1000, 1000, 1000, 1000, 1000, ...
## Resampling results across tuning parameters:
##
##   mtry  splitrule   Accuracy   Kappa
##      2  gini        0.4908071  0.0000000000
##      2  extratrees  0.4909132  0.0002110818
##     66  gini        0.9660655  0.9320536162
##     66  extratrees  0.9672887  0.9345101534
##   2209  gini        0.9555424  0.9109851830
##   2209  extratrees  0.9624188  0.9247419148
##
## Tuning parameter 'min.node.size' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were mtry = 66, splitrule = extratrees
##  and min.node.size = 1.
```

We attain an accuracy of 96.24%