#1. Defining the question

1.1 Specifying the data analytic objective

Predict which individuals are most likely to click on ads from a cryptography course website

1.2 Defining the metric of success

For this study, we will perform conclusive Exploratory Data Analysis to enable us identify individuals who are most likely to click on ads

1.3 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. Using the data previously collected, she is looking to do a study to identify which individuals are most likely to click on her ads.

1.4 Recording the Experimental Design

1. Loading the data
2. Checking the data
3. Tidying the data
4. Univariate Analysis
5. Bivariate Analysis
6. Challenging the solution
7. Recommendations
8. Follow up questions

2. Loading the data set

```
#Loading data
ad <- read.csv('http://bit.ly/IPAdvertisingData')
#Reading head 5
head(ad)
```

```
##   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
```

```
spaceless <- function(x) {colnames(x) <- gsub(" ", "_", colnames(x));x}
advert <- spaceless(ad)
head(advert)
```

```
##   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
```

Checking the summary

```
summary(advert)
```

```
##  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
```

From the data summary we get the measures of central tendency (median, mean, mode and quantile)

Checking the top and bottom columns

```
tail(advert)
```

```
##      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
```

```
head(advert)
```

```
##   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
```

Checking the class

```
class(advert)

## [1] "data.frame"
```

Structure of the dataset

```
str(advert)

## '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 ...

#Datatypes
sapply(advert, 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"
```

#3. Cleaning the dataset

##3.1 Finding missing values

```
colSums(is.na(advert))
```

```
## 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
```

No missing data was found

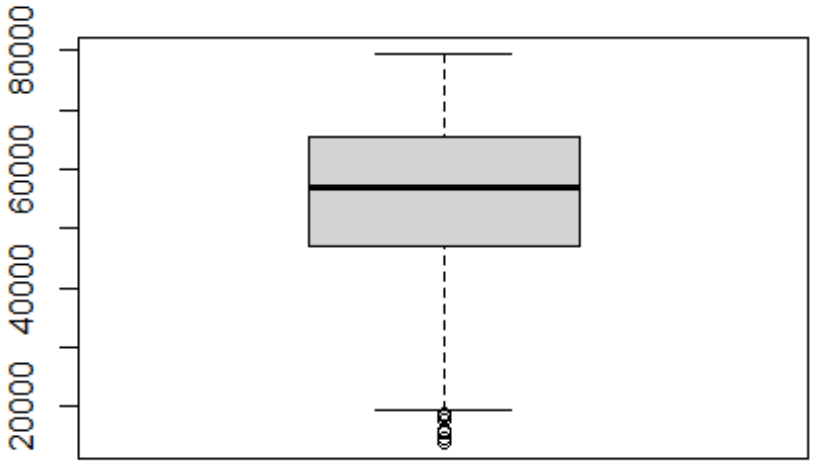
3.2 Checking for duplicates

```
sum(duplicated(advert))
```

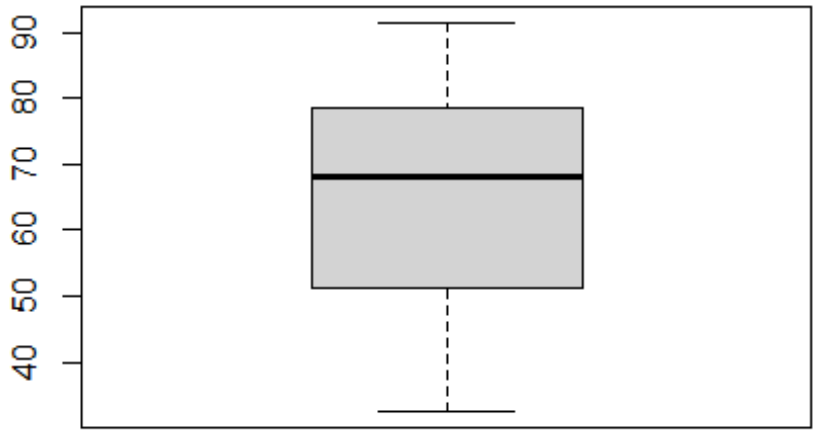
```
## [1] 0
```

3.3 Checking for outliers

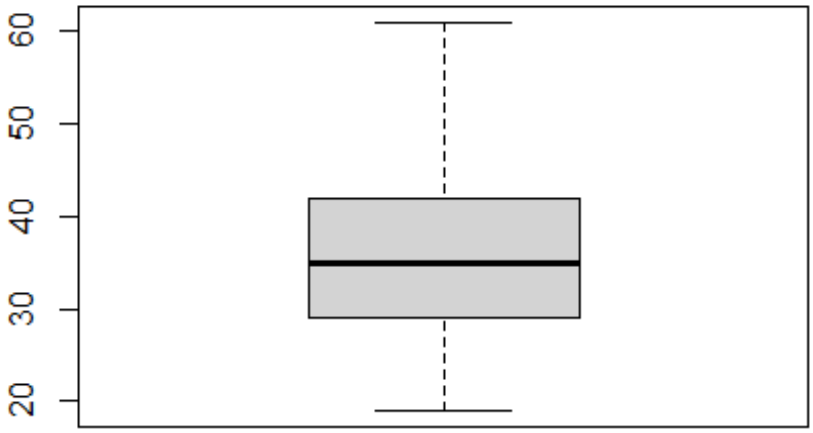
```
# Area Income


boxplot(advert$Area.Income)
```



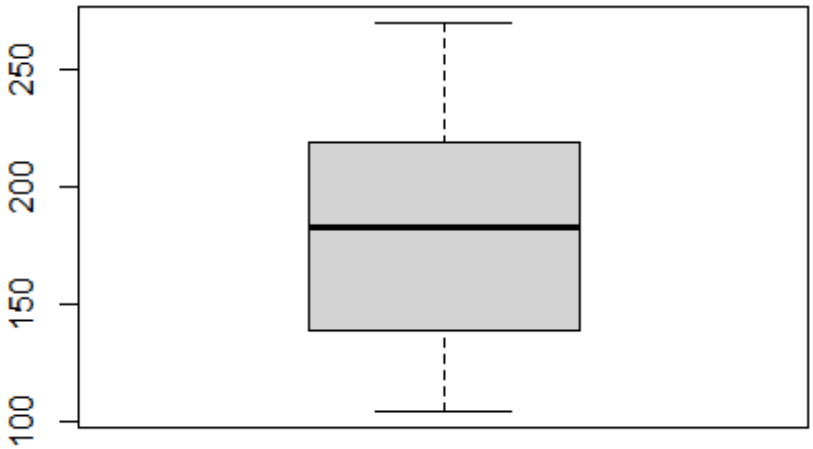
```r
# Time spent on site
boxplot(advert$Daily.Time.Spent.on.Site)
```




```r
# Age
boxplot(advert$Age)
```

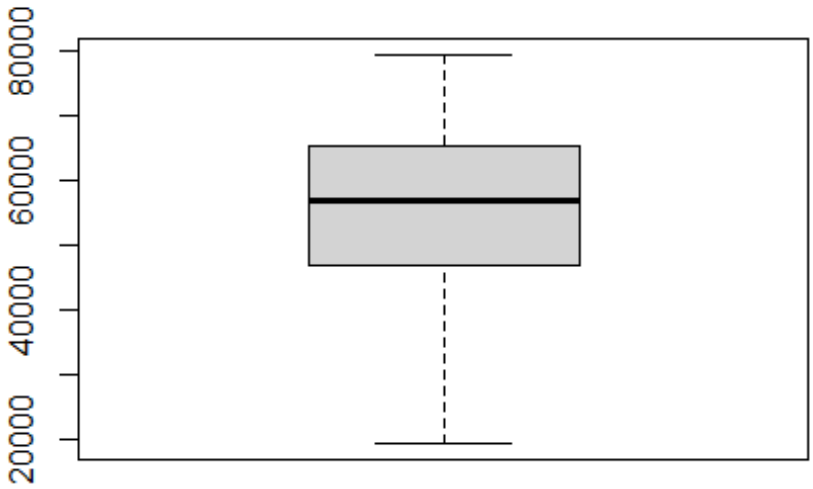



```
# Daily internet usage
boxplot(advert$Daily.Internet.Usage)
```



##3.4 Removing outliers

```
outlier <- 47032 - 1.5 * IQR(advert$Area.Income)
advert$Area.Income[advert$Area.Income < outlier]<- outlier

boxplot(advert$Area.Income)
```



We remove outliers by limiting extreme values in the statistical data to reduce the effect of possibly spurious outliers

4. Exploratory Data Analysis

4.1 Univariate Analysis

Measures of Central Tendency

```
#Selecting the numeric columns
num <- subset(advert, select = -c(Ad.Topic.Line, City,  Male,   Country,
Country, Timestamp))
#Getting the measures of central tendency
summary(num)

##  Daily.Time.Spent.on.Site      Age         Area.Income
Daily.Internet.Usage
##  Min.   :32.60           Min.   :19.00   Min.   :19374   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   :55025   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
##  Clicked.on.Ad
##  Min.   :0.0
##  1st Qu.:0.0
##  Median :0.5
##  Mean   :0.5
##  3rd Qu.:1.0
##  Max.   :1.0
```

Distribution of data

```
#install.packages("moments")

library(moments)

head(num)

##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
Clicked.on.Ad
## 1                    68.95  35    61833.90               256.09
0
## 2                    80.23  31    68441.85               193.77
0
## 3                    69.47  26    59785.94               236.50
0
## 4                    74.15  29    54806.18               245.89
0
## 5                    68.37  35    73889.99               225.58
0
## 6                    59.99  23    59761.56               226.74
0

#Checking for skewness
paste("Daily Time_Spent_Skewness: ", paste
(skewness(advert$Daily.Time.Spent.on.Site), collapse = ','))

## [1] "Daily Time_Spent_Skewness:  -0.371202614867441"

paste("Income_Skewness: ", paste (skewness(advert$Area.Income), collapse =
','))

## [1] "Income_Skewness:  -0.620048077753174"

paste("Age_Skewness: ", paste (skewness(advert$Age), collapse = ','))

## [1] "Age_Skewness:  0.478422676206608"

paste("Daily_Internet_Usage_Skewness: ", paste
(skewness(advert$Daily.Internet.Usage), collapse = ','))

## [1] "Daily_Internet_Usage_Skewness:  -0.0334870316434409"
```

```r
#Checking for kurtosis
paste("Daily Time_Spent_Kurtosis: ", paste
(kurtosis(advert$Daily.Time.Spent.on.Site), collapse = ','))
```

```
## [1] "Daily Time_Spent_Kurtosis:  1.90394215401081"
```

```r
paste("Income_Kurtosis: ", paste (kurtosis(advert$Area.Income), collapse =
','))
```

```
## [1] "Income_Kurtosis:  2.78988148954566"
```

```r
paste("Age_Kurtosis: ", paste (kurtosis(advert$Age), collapse = ','))
```

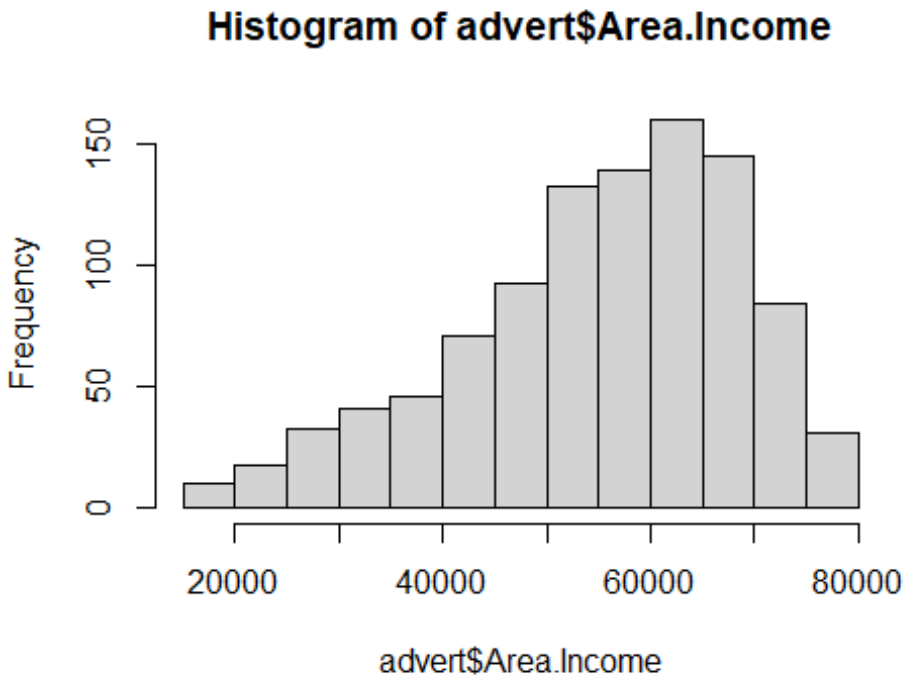
```
## [1] "Age_Kurtosis:  2.59548176807726"
```

```r
paste("Daily_Internet_Usage_Kurtosis: ", paste
(kurtosis(advert$Daily.Internet.Usage), collapse = ','))
```

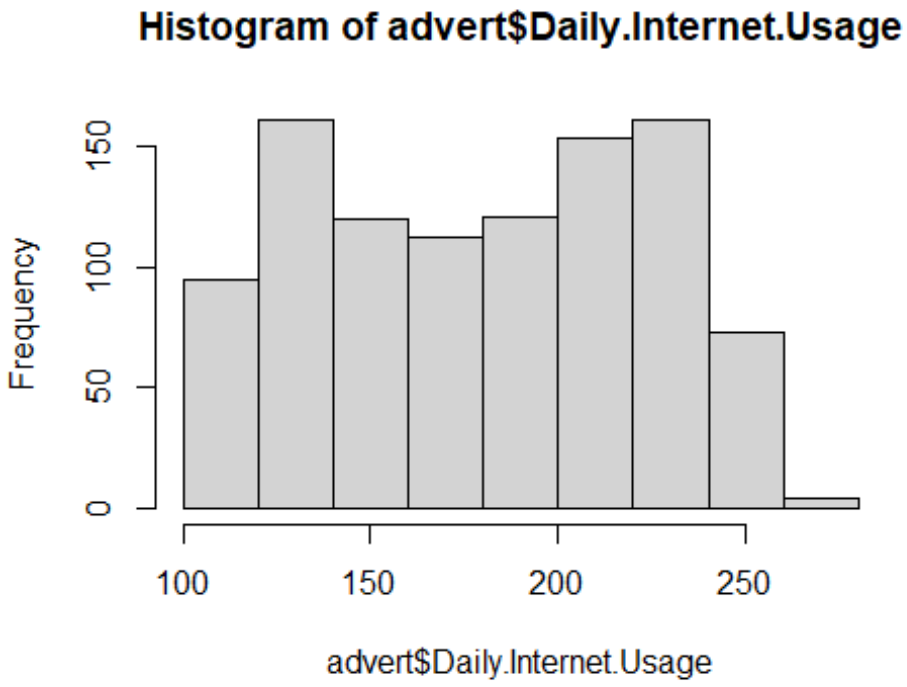
```
## [1] "Daily_Internet_Usage_Kurtosis:  1.72770118094819"
```

```r
hist(advert$Age)
```

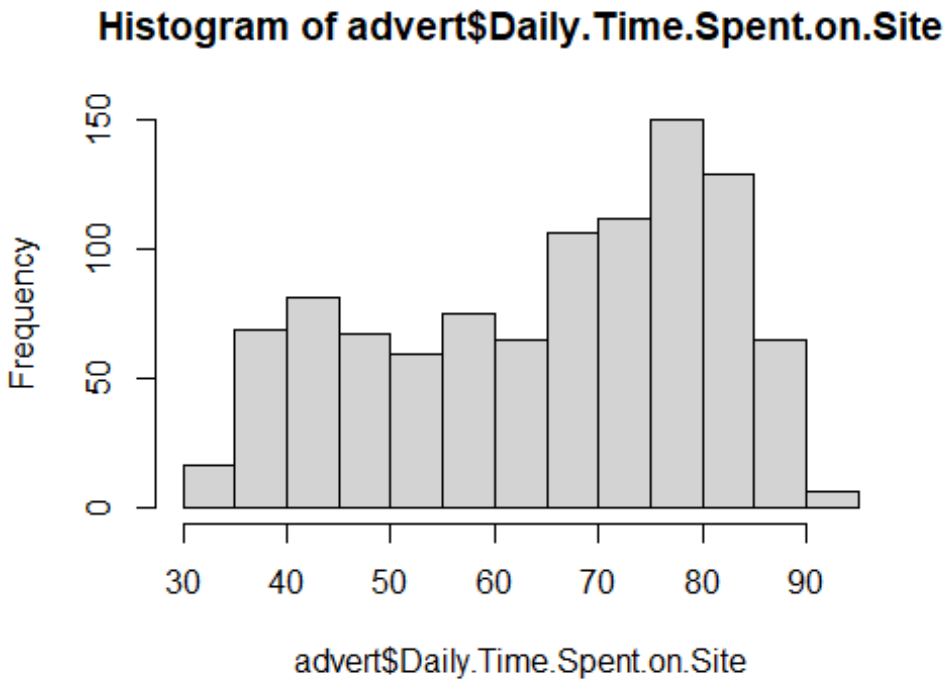

```r
hist(advert$Area.Income)
```




```
hist(advert$Daily.Internet.Usage)
```

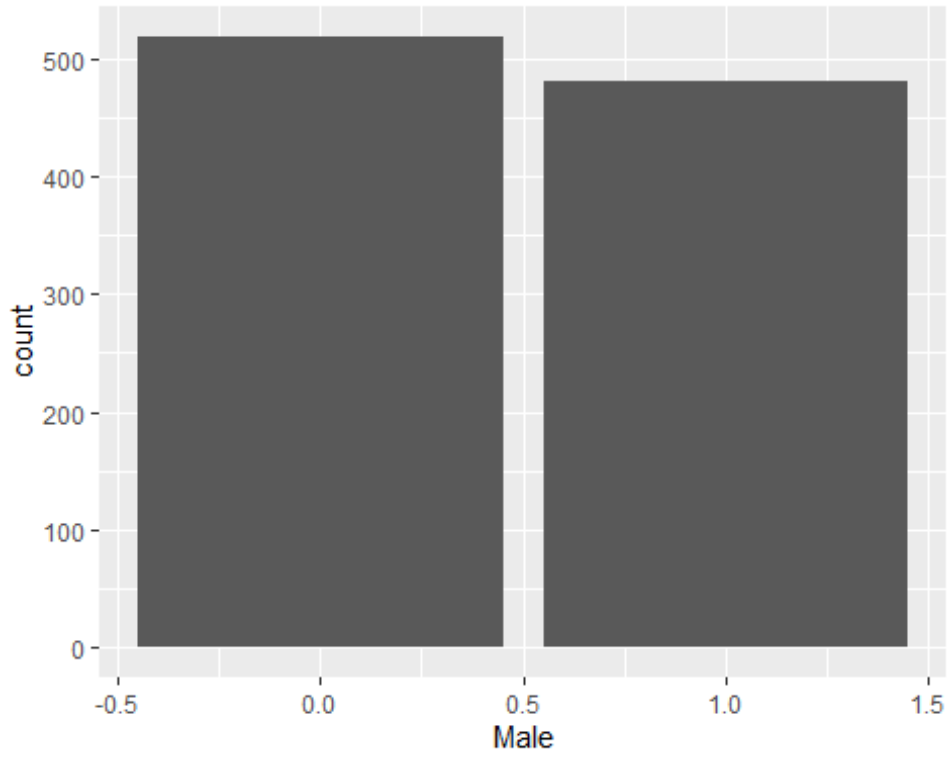



```
hist(advert$Daily.Time.Spent.on.Site)
```



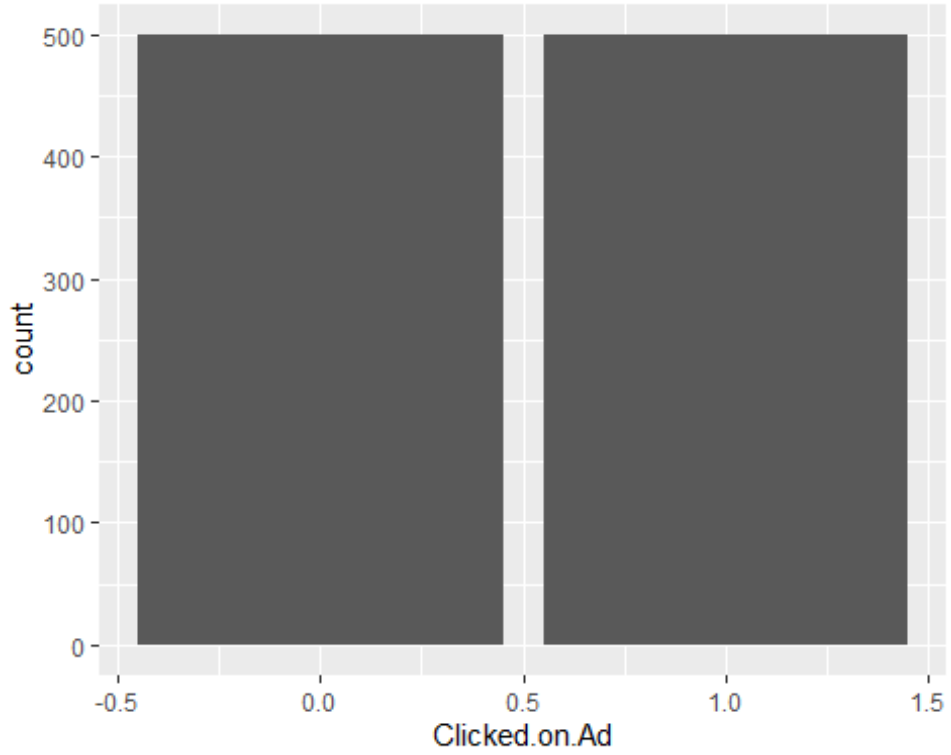
Categorical Data

```
#Which gender is mainly active on the blog?
library(ggplot2)
ggplot(data = advert) +
  geom_bar(mapping = aes(x = Male))
```



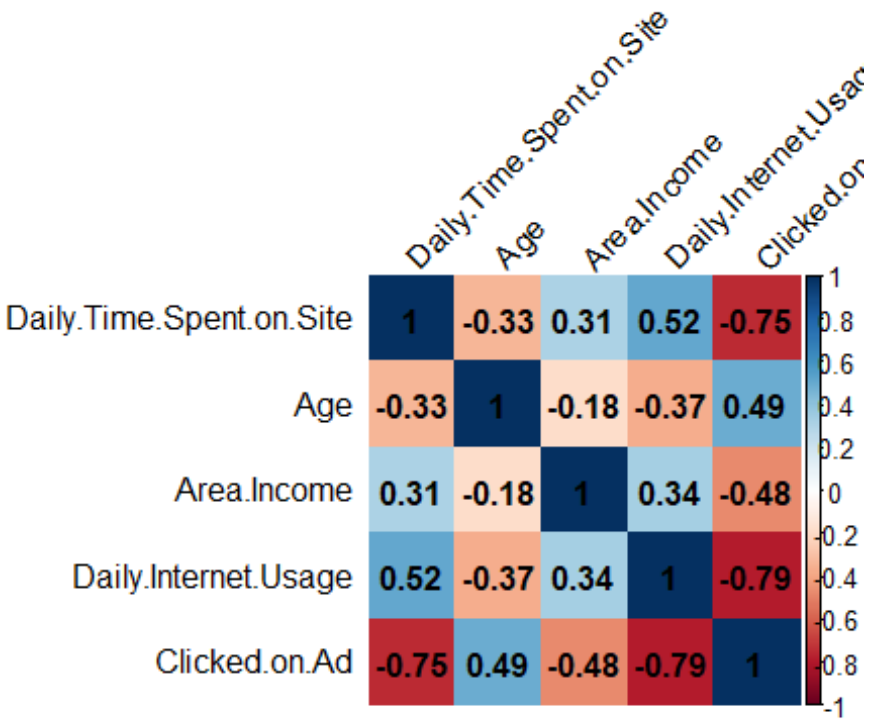
Assuming that if male = 1 then we can conclude that more females frequennt the blog more as compared to males

```
#Do most people clickon ads or not?

ggplot(data = advert) +
  geom_bar(mapping = aes(x = Clicked.on.Ad))
```



4.2 Bivariate Analysis

```
# install.packages("corrplot")

library(corrplot)

## Warning: package 'corrplot' was built under R version 4.0.5

## corrplot 0.84 loaded

#Get the correlation matrix
res = cor(num)
#Plotting a correlation plot

corrplot(res, method="color",addCoef.col = "black",
         tl.col="black", tl.srt=45)
```




```
#Change datattypes
advert$Male <- as.factor(advert$Male)
advert$Clicked.on.Ad <- as.factor(advert$Clicked.on.Ad)
#Checking datatypes
sapply(advert, 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
##                 "factor"               "character"              "character"
##            Clicked.on.Ad
##                 "factor"
```

```
#install.packages("tidyverse")
library(ggplot2)

ggplot(advert,
       aes(x = Clicked.on.Ad,
           fill = Male)) +
  geom_bar(position = "stack")
```

5. Feature Engineering

```
head(advert)
```

```
##   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
```

```
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
```

```
mod.data <- select(advert, -c(5,6,8,9))
head(mod.data)
```

```
##   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
## 6                    59.99  23    59761.56               226.74    1
##   Clicked.on.Ad
## 1             0
## 2             0
## 3             0
## 4             0
## 5             0
## 6             0
```

```
#
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.5
```

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

```
#Create an index for data partitioning
set.seed(42)
index <- createDataPartition(mod.data$Clicked.on.Ad, p = 0.80, list = FALSE)
```

```
#Using the indexes to split data into test and train set
dat.train <- mod.data[index, ]
dat.test <- mod.data[-index, ]
```

6. Decision Trees

```
#Installing packages to be used for modelling
#install.packages("rpart")
library(rpart)
```

```
#install.packages("e1071")
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.0.5
```

```
##
## Attaching package: 'e1071'
```

```
## The following objects are masked from 'package:moments':
##
##     kurtosis, moment, skewness
```

```
#Fitting in the decision tree
TreeFit <- rpart(Clicked.on.Ad ~ ., data = dat.train)

#Factor the Clicked.on.Ad vector in the test dataset
dat.test$Clicked.on.Ad <- factor(dat.test$Clicked.on.Ad)

#Using model to predict
TreePredict <- predict(TreeFit, newdata = dat.test, type = "class")
confusionMatrix(TreePredict, dat.test$Clicked.on.Ad)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 96  6
##          1  4 94
##
##                Accuracy : 0.95
##                  95% CI : (0.91, 0.9758)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9
##
##  Mcnemar's Test P-Value : 0.7518
##
##             Sensitivity : 0.9600
##             Specificity : 0.9400
##          Pos Pred Value : 0.9412
##          Neg Pred Value : 0.9592
##              Prevalence : 0.5000
##          Detection Rate : 0.4800
##    Detection Prevalence : 0.5100
##       Balanced Accuracy : 0.9500
##
##        'Positive' Class : 0
##
```

Decision trees had a 95% accuracy score.

7. KNN

```
#Fitting model to training dataset
#Also we scale and center our data
knnModel <- train(Clicked.on.Ad ~ ., data = dat.train, method = "knn",
preProcess = c("center", "scale"))

#Using the model to predict
knnPredict <- predict(knnModel, newdata = dat.test)

#Printing out the confusion matrix and statistics
confusionMatrix(knnPredict, dat.test$Clicked.on.Ad)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 98  9
##          1  2 91
##
##                Accuracy : 0.945
##                  95% CI : (0.9037, 0.9722)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.89
##
##  Mcnemar's Test P-Value : 0.07044
##
##             Sensitivity : 0.9800
##             Specificity : 0.9100
##          Pos Pred Value : 0.9159
##          Neg Pred Value : 0.9785
##              Prevalence : 0.5000
##          Detection Rate : 0.4900
##    Detection Prevalence : 0.5350
##       Balanced Accuracy : 0.9450
##
##        'Positive' Class : 0
##
```

KNN performs at an accuracy of 94.5%.

```
#Installing and running the kernlab package
#install.packages("kernlab")
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
#controling all the computational overheads using traincontrol
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)

#We fit the model using the linear kernel
#Data is also scaled and centered
svm_Linear <- train(Clicked.on.Ad ~., data = dat.train, method = "svmLinear",
trControl=trctrl,
preProcess = c("center", "scale"),
tuneLength = 10)

# We then check the result of our train() model
svm_Linear
```

```
## Support Vector Machines with Linear Kernel
##
## 800 samples
##   5 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (5), scaled (5)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 720, 720, 720, 720, 720, 720, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9708333  0.9416667
##
## Tuning parameter 'C' was held constant at a value of 1
```

```
#We then predict
test_pred <- predict(svm_Linear, newdata = dat.test)
test_pred
```

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

```
#Print the confusion matrix and statistics
confusionMatrix(table(test_pred, dat.test$Clicked.on.Ad))
```

```
## Confusion Matrix and Statistics
##
##
## test_pred  0  1
##         0 97  5
##         1  3 95
##
##                Accuracy : 0.96
##                  95% CI : (0.9227, 0.9826)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.92
##
##  Mcnemar's Test P-Value : 0.7237
##
##             Sensitivity : 0.9700
##             Specificity : 0.9500
##          Pos Pred Value : 0.9510
##          Neg Pred Value : 0.9694
##              Prevalence : 0.5000
##          Detection Rate : 0.4850
##    Detection Prevalence : 0.5100
##       Balanced Accuracy : 0.9600
##
##        'Positive' Class : 0
##
```

As compared to KNN and Decision Trees, the SVM linear kernel model performs the best.It has an accuracy score of 96%

Conclusion

In conclusion, we advice the owner of the blog to use an SVM model with a linear kernel to predict whether users of the blog will click on an ad or not