CryptoCourseDataAnalysis

Joy Machuka

8/27/2021

```r
# tinytex::install_tinytex()
```

##Defining 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.

##Metric of success

Our analysis will be considered successful if we are able to analyze the data and get the most likely to take the course statistics.

##Context An entrepreneur who practices online teaching has opted to use ads to advertise her course to the public. To achieve this she needs to use who are more likely to click her ads so sh can target them in the creation process. She uses data collected from past course ads. She uses a data scientist to do this.

##Experimental Design Data preparation Data Cleaning Univariate Analysis Bivariate Analysis Modelling Recommendation Conclusion

##Data Preparation

```r
#load dataset
crypto <- read.csv('http://bit.ly/IPAdvertisingData')
head(crypto)
```

```
##   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
```

```r
#preview the dataset
View(crypto)
```

```r
#viewing first 6 rows
head(crypto, 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
```

```r
#check the shape of dataset
dim(crypto)
```

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

Dataset has 1000 rows and 10 columns.

```r
#checking the class of our dataset
class(crypto)
```

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

We are working with a dataframe.

```r
#checking column names
names(crypto)
```

```
##  [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 datas types of variables
str(crypto)
```

```
## '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 standardi
##  $ 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"
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
```

We have a mixture of integers, numerics and characters.

```
#searching for information about dataset
?crypto
```

```
## No documentation for 'crypto' in specified packages and libraries:
## you could try '??crypto'
```

No documentation for 'crypto' in specified packages and libraries.

##Data Cleaning

```
#checking for null values per column
colSums(is.na(crypto))
```

```
## 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
```

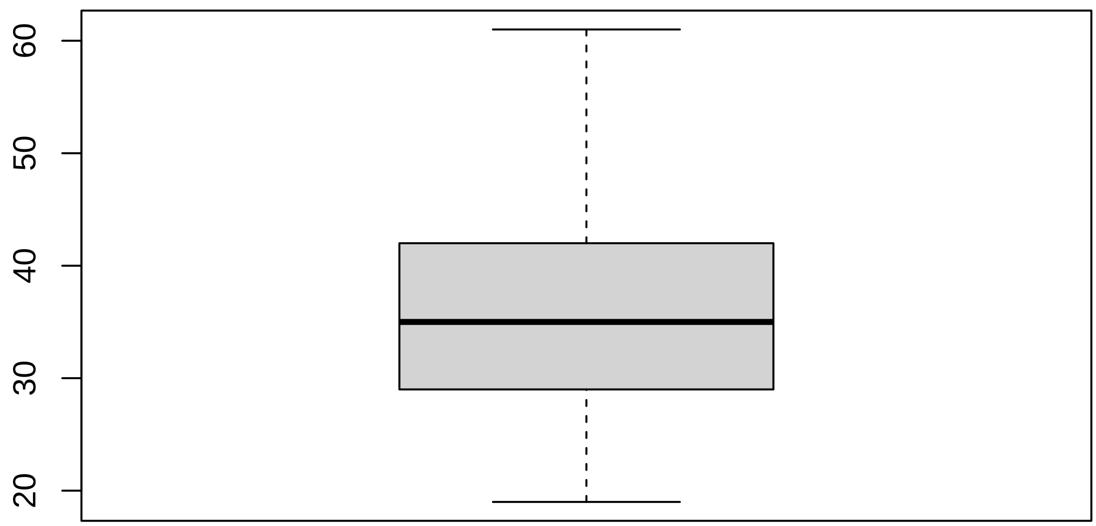
No null values in our data frame.

```
#checking for duplicates
duplicated_rows <- crypto[duplicated(crypto),]
duplicated_rows
```

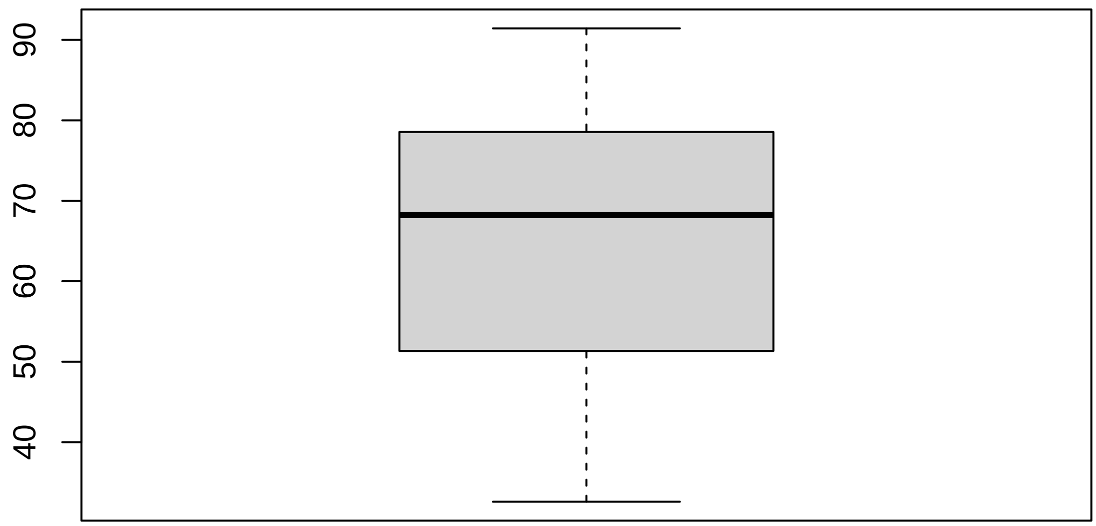
```
##  [1] Daily.Time.Spent.on.Site Age                      Area.Income
##  [4] Daily.Internet.Usage     Ad.Topic.Line            City
##  [7] Male                     Country                  Timestamp
## [10] Clicked.on.Ad
## <0 rows> (or 0-length row.names)
```

There aren't any duplicated rows and missing data from our data frame as seen above.

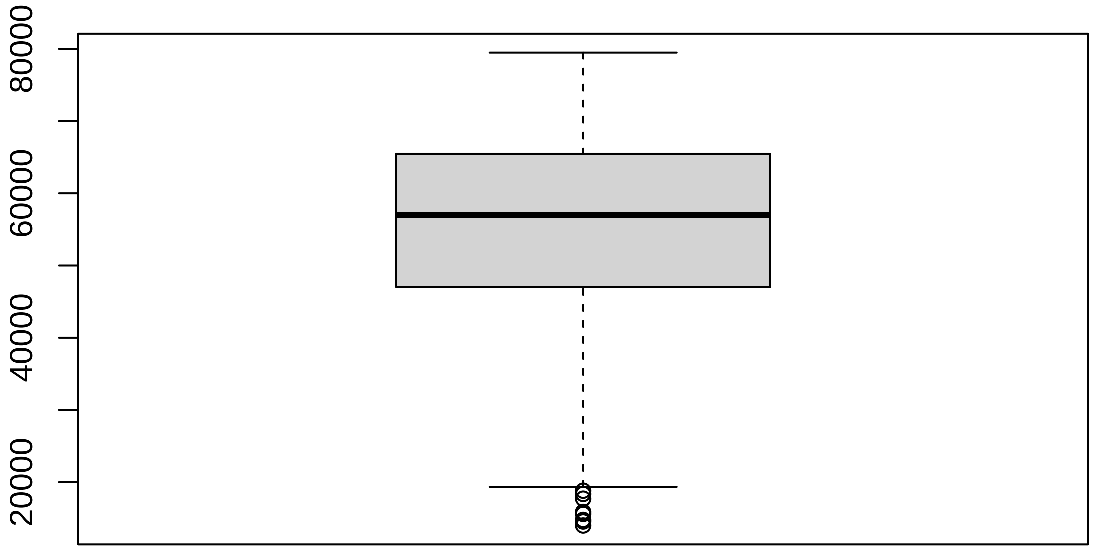
```r
#checking for outliers
boxplot(crypto$Age)
```



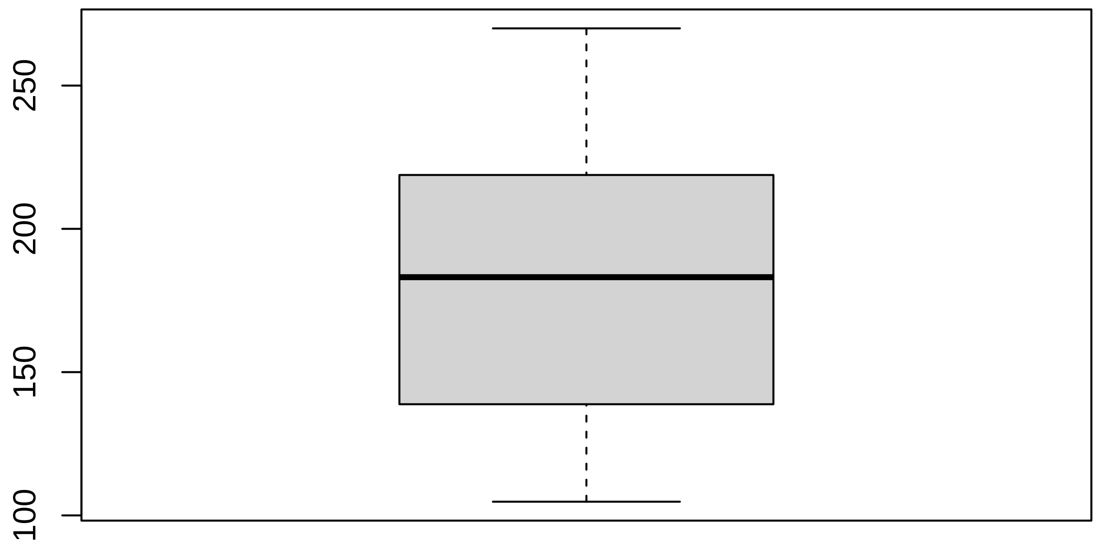

```r
boxplot(crypto$Daily.Time.Spent.on.Site)
```



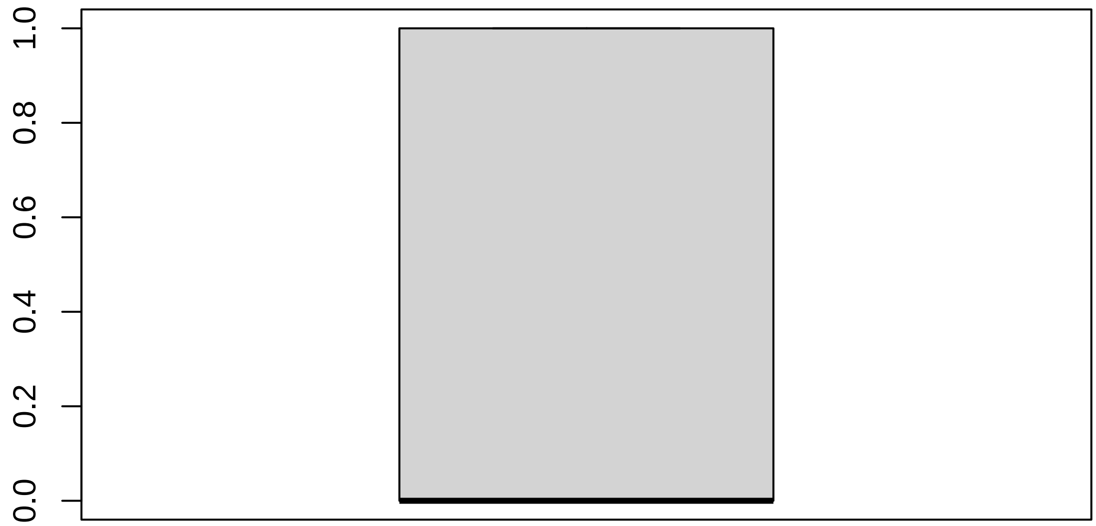

```
boxplot(crypto$Area.Income)
```



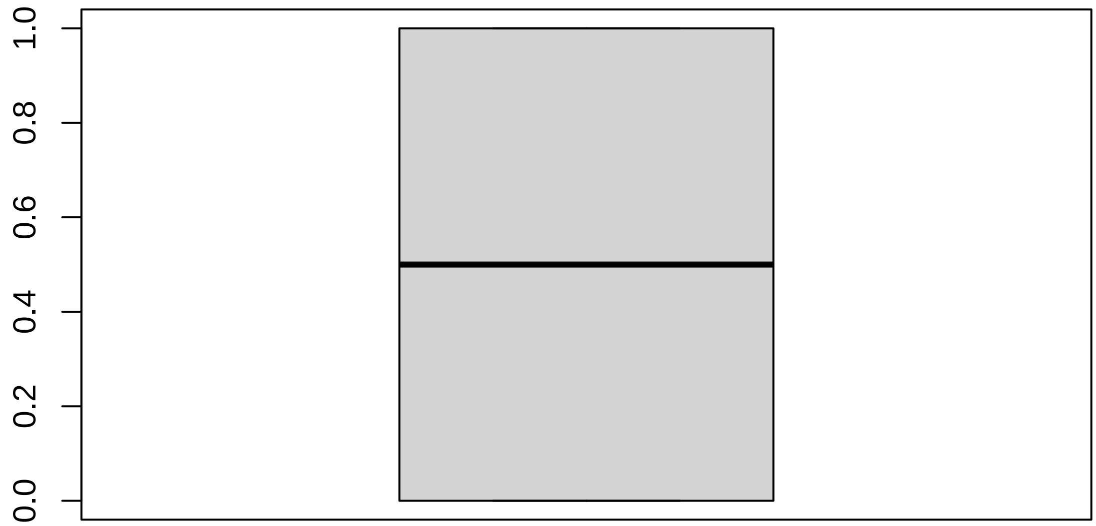

```
boxplot(crypto$Daily.Internet.Usage)
```




```
boxplot(crypto$Male)
```




```
boxplot(crypto$Clicked.on.Ad)
```

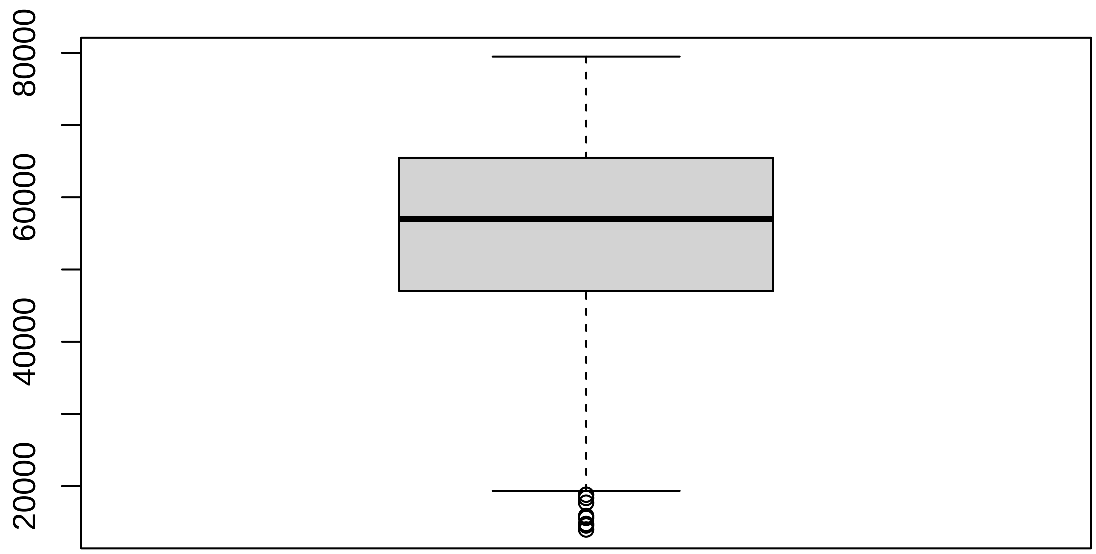



```
# boxplot(crypto$Ad.Topic.Line)
# boxplot(crypto$City)
# boxplot(crypto$Country)
# boxplot(crypto$Timestamp)
```

We have outliers only on the Income column which we are going to keep because it is viable an realistic in the real world

```
# par(mfrow = c(2, 2))  # Set up a 2 x 2 plotting space
#
# # Create the loop.vector (all the columns)
# loop.vector <- 1:10
#
# for (i in loop.vector) { # Loop over loop.vector
# # store data in column.i as x
#   x <- crypto[,i]
# # Plot boxplot of x
#   boxplot(x, main = paste("plot", i),
#        xlim = c(0, 2))
# }
```

```
# Viewing the outliers in the Area.Income column since it is the only column with outliers
boxplot(crypto$Area.Income)
```



Viewing the income boxplot individually.

```
#viewing the outlier rows
boxplot.stats(crypto$Area.Income)$out
```

```
## [1] 17709.98 18819.34 15598.29 15879.10 14548.06 13996.50 14775.50 18368.57
```

Looking at the specific outliers and as we said they are viable so we keep them.

```
#prooving further by checking the quantile distribution of income.
quantile(crypto$Area.Income)
```

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

We decided to leave our outliers since they seemed viable and helpful to our analysis after looking at the quantile distribution

##Univariate Analysis

```
#summary statistics of all the columns
summary(crypto)
```

```
##  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
```

Area income ranges between 13996.5 and 79484.6

```
num = crypto[,c(1,2,3,4,7,10)]
summary(num)
```

```
##  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
##       Male        Clicked.on.Ad
##  Min.   :0.000   Min.   :0.0
##  1st Qu.:0.000   1st Qu.:0.0
##  Median :0.000   Median :0.5
##  Mean   :0.481   Mean   :0.5
##  3rd Qu.:1.000   3rd Qu.:1.0
##  Max.   :1.000   Max.   :1.0
```

```
#Mean
mean(crypto$Daily.Time.Spent.on.Site)
```

```
## [1] 65.0002
```

```
mean(crypto$Age)
```

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

```
mean(crypto$Area.Income)
```

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

```r
mean(crypto$Daily.Internet.Usage)
```

```
## [1] 180.0001
```

```r
mean(crypto$Male)
```

```
## [1] 0.481
```

```r
mean(crypto$Clicked.on.Ad)
```

```
## [1] 0.5
```

On average, the daily time spent on the site is 65 The average age of the user is 36 years. The average area income is 55000.

```r
# Median
median(crypto$Daily.Time.Spent.on.Site)
```

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

```r
median(crypto$Age)
```

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

```r
median(crypto$Area.Income)
```

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

```r
median(crypto$Daily.Internet.Usage)
```

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

```r
median(crypto$Male)
```

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

```r
median(crypto$Clicked.on.Ad)
```

```
## [1] 0.5
```

```r
#mode of the columns
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
getmode(crypto$Daily.Time.Spent.on.Site)
```

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

```
getmode(crypto$Age)
```

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

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

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

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

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

```
getmode(crypto$Male)
```

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

```
getmode(crypto$Clicked.on.Ad)
```

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

```
getmode(crypto$City)
```

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

```
getmode(crypto$Country)
```

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

The age that is the most common of the users is 31 years The City with the most repeat users is Lisamouth The country that's most repeated is Czech Republic The most common gender in the data is Female

```
#Variance
var(crypto$Daily.Time.Spent.on.Site)
```

```
## [1] 251.3371
```

```
var(crypto$Age)
```

```
## [1] 77.18611
```

```
var(crypto$Area.Income)
```

```
## [1] 179952406
```

```r
var(crypto$Daily.Internet.Usage)
```

```
## [1] 1927.415
```

```r
var(crypto$Male)
```

```
## [1] 0.2498889
```

```r
var(crypto$Clicked.on.Ad)
```

```
## [1] 0.2502503
```

```r
# Standard Deviation
sd(crypto$Daily.Time.Spent.on.Site)
```

```
## [1] 15.85361
```

```r
sd(crypto$Age)
```

```
## [1] 8.785562
```

```r
sd(crypto$Area.Income)
```

```
## [1] 13414.63
```

```r
sd(crypto$Daily.Internet.Usage)
```

```
## [1] 43.90234
```

```r
sd(crypto$Male)
```

```
## [1] 0.4998889
```

```r
sd(crypto$Clicked.on.Ad)
```

```
## [1] 0.5002502
```

```r
#Quantiles
quantile(crypto$Daily.Time.Spent.on.Site)
```

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

```r
quantile(crypto$Age)
```

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

```r
quantile(crypto$Area.Income)
```

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

```r
quantile(crypto$Daily.Internet.Usage)
```

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

```r
quantile(crypto$Male)
```

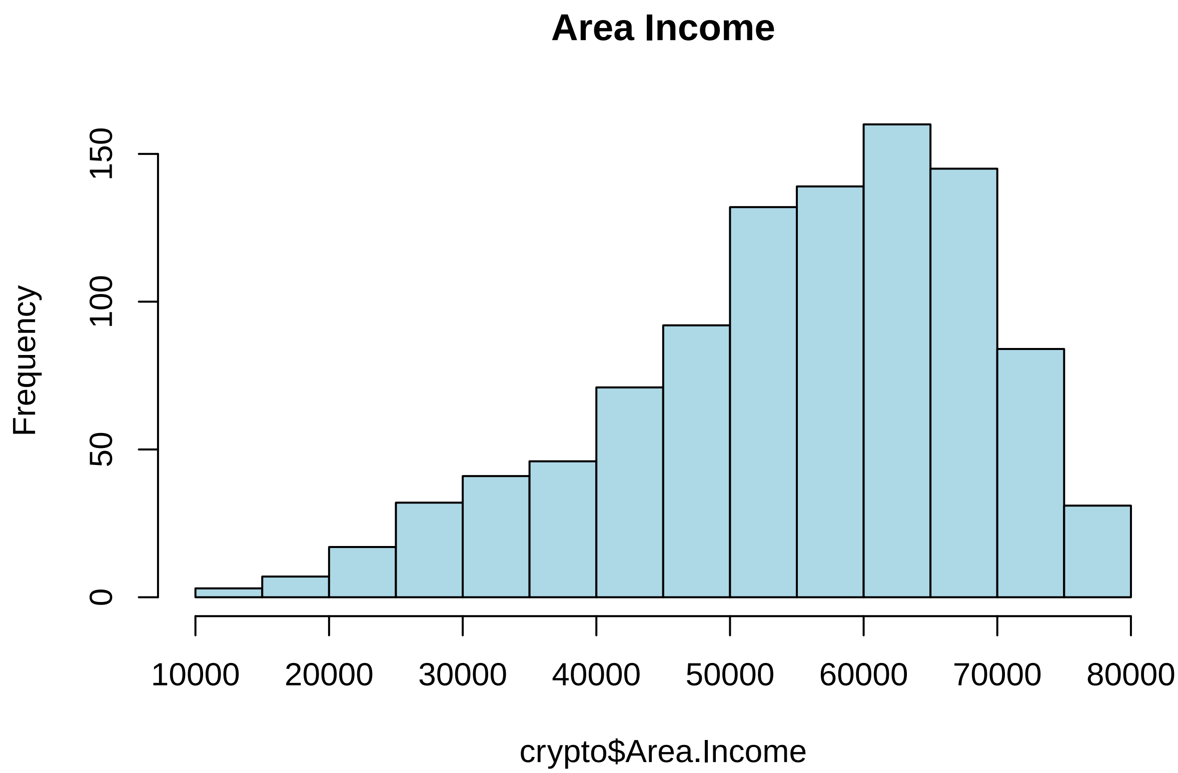
```
##   0%  25%  50%  75% 100%
##    0    0    0    1    1
```

```r
quantile(crypto$Clicked.on.Ad)
```

```
##   0%  25%  50%  75% 100%
##  0.0  0.0  0.5  1.0  1.0
```

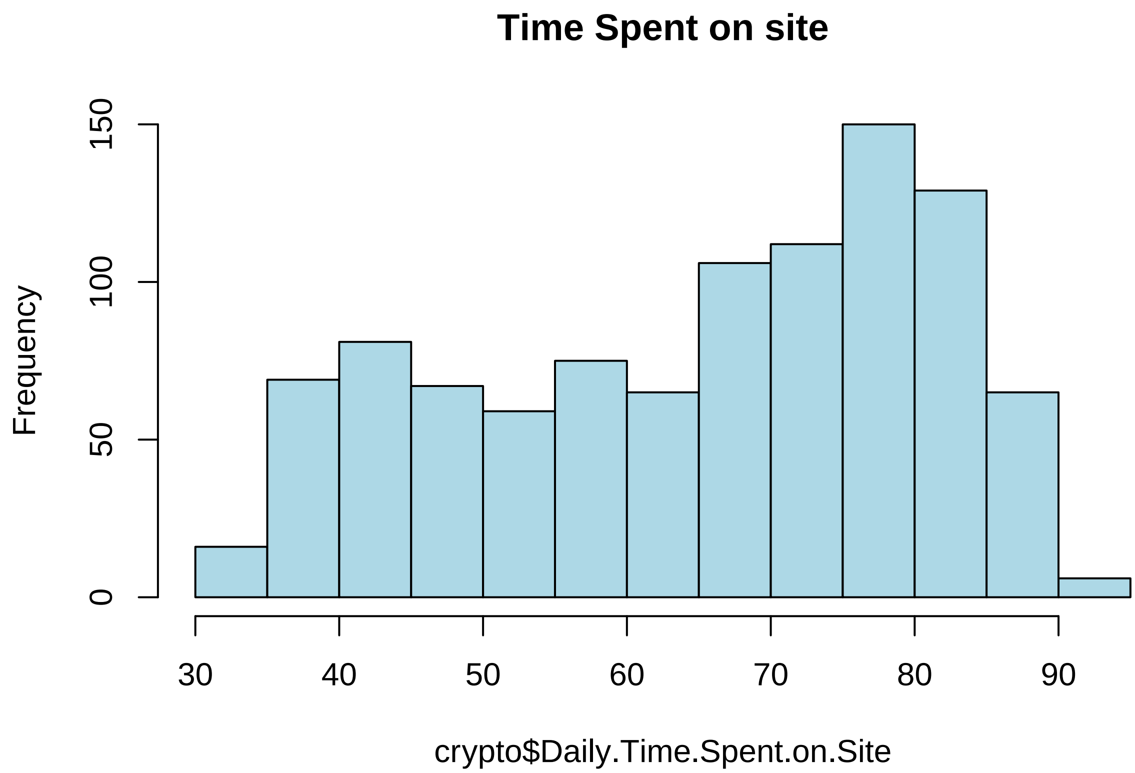
```r
# par(mfrow = c(2, 5))  # Set up a 2 x 2 plotting space
#
# # Create the loop.vector (all the columns)
# loop.vector <- 1:10
#
# for (i in loop.vector) { # Loop over loop.vector
#
# # store data in column.i as x
#   x <- crypto[,i]
#
# # Plot histogram of x
#   hist(x,
#        main = paste("histogram", i),
#        xlab = "Scores",
#        xlim = c(0, 100))
# }
```

```r
hist(crypto$Area.Income, breaks = 10, main = "Area Income",col = "lightblue")
```



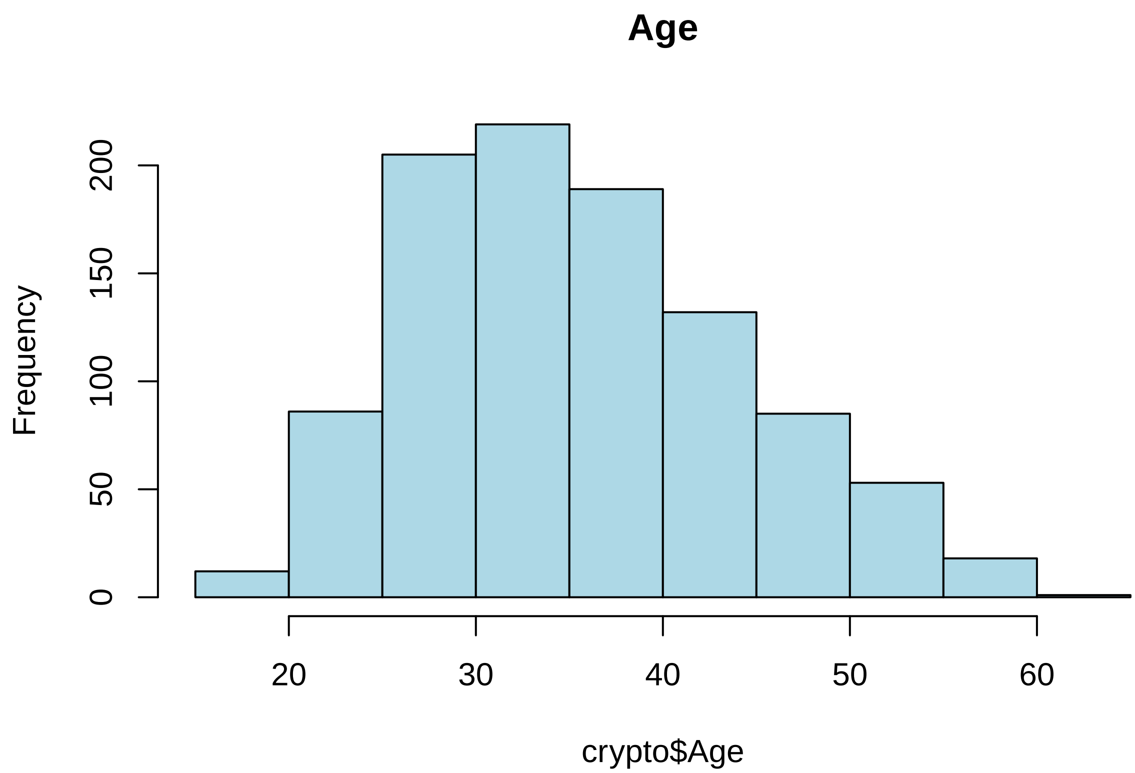
The areas that have an income between 40000 and 70000 have the most clicks on the ads

```
hist(crypto$Daily.Time.Spent.on.Site, breaks = 10, main = "Time Spent on site",col = "lightblue")
```



Averagely 75-85 is the most time spent with high frequencies

```
hist(crypto$Age, breaks = 10, main = "Age",col = "lightblue")
```

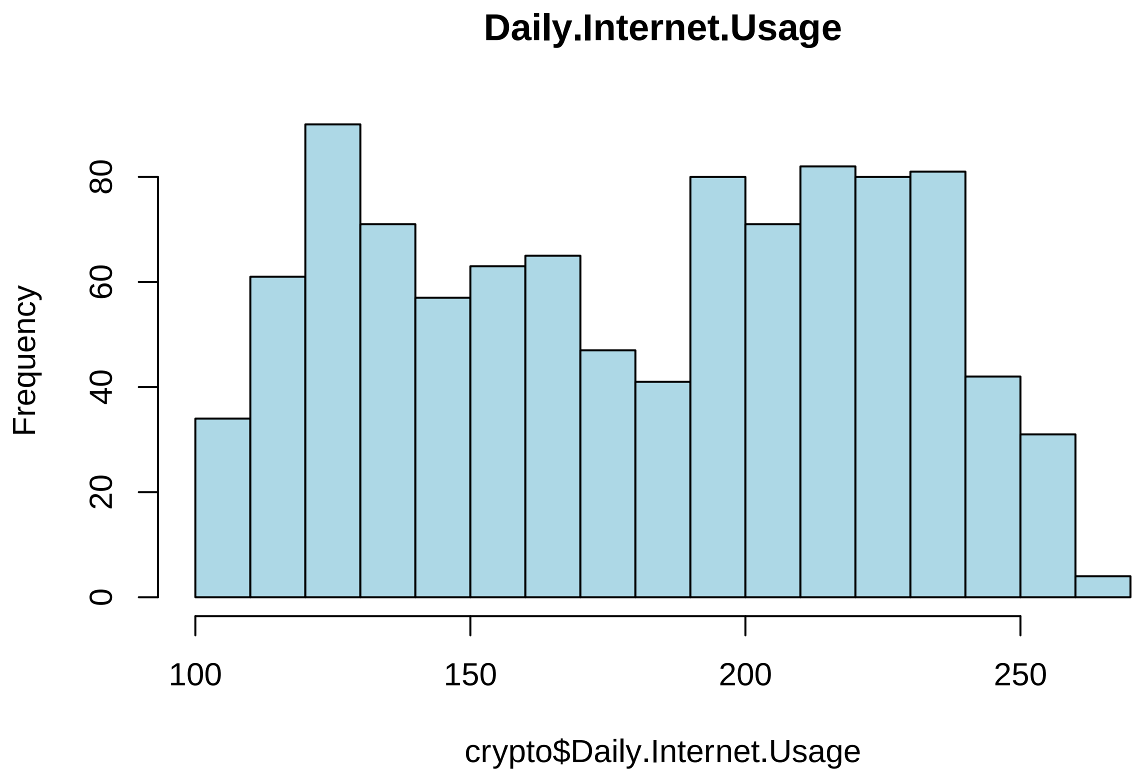


25-40 age seems to be the area with most frequencies.

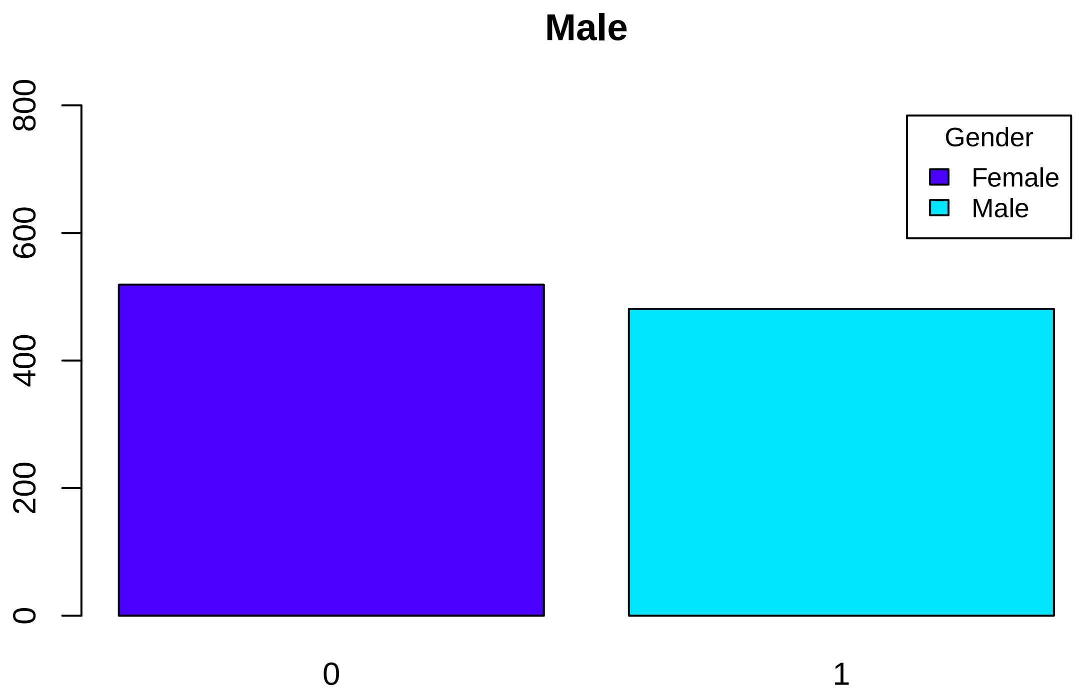
```
names(crypto)
```

```
## [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"
```

```
hist(crypto$Daily.Internet.Usage, breaks = 20, main = "Daily.Internet.Usage",col = "lightblue")
```

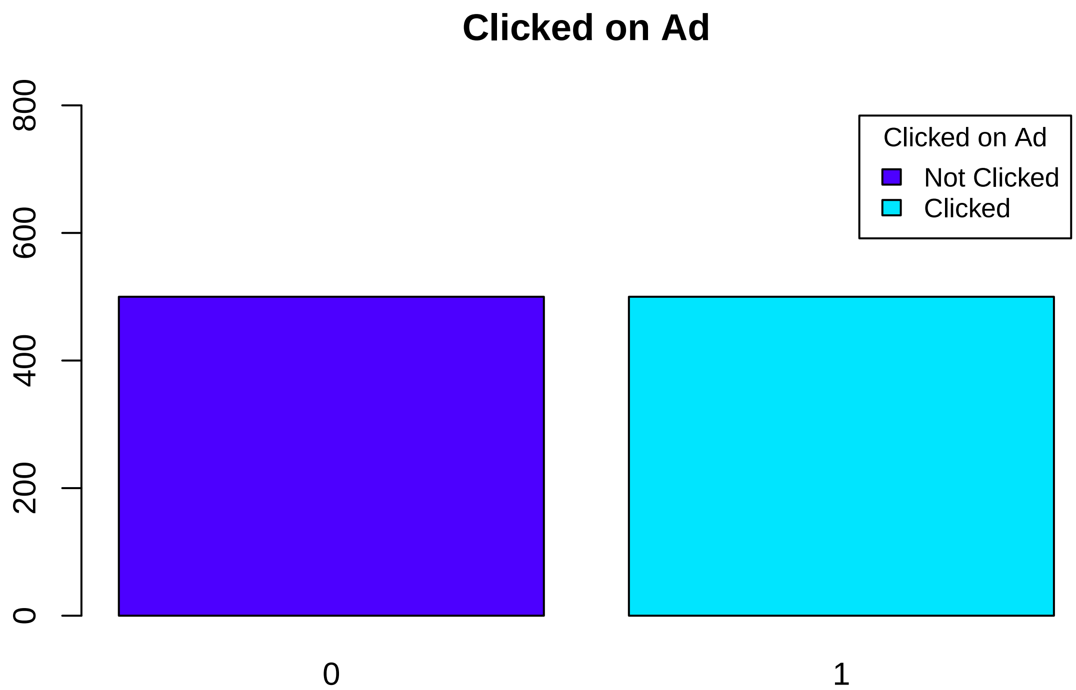



```r
male <- table(crypto$Male)
barplot(male,main = "Male",col = topo.colors(2),ylim = c(0, 800))
legend("topright",inset = .02, title="Gender",
       c("Female","Male"), fill=topo.colors(2), cex=0.8)
```



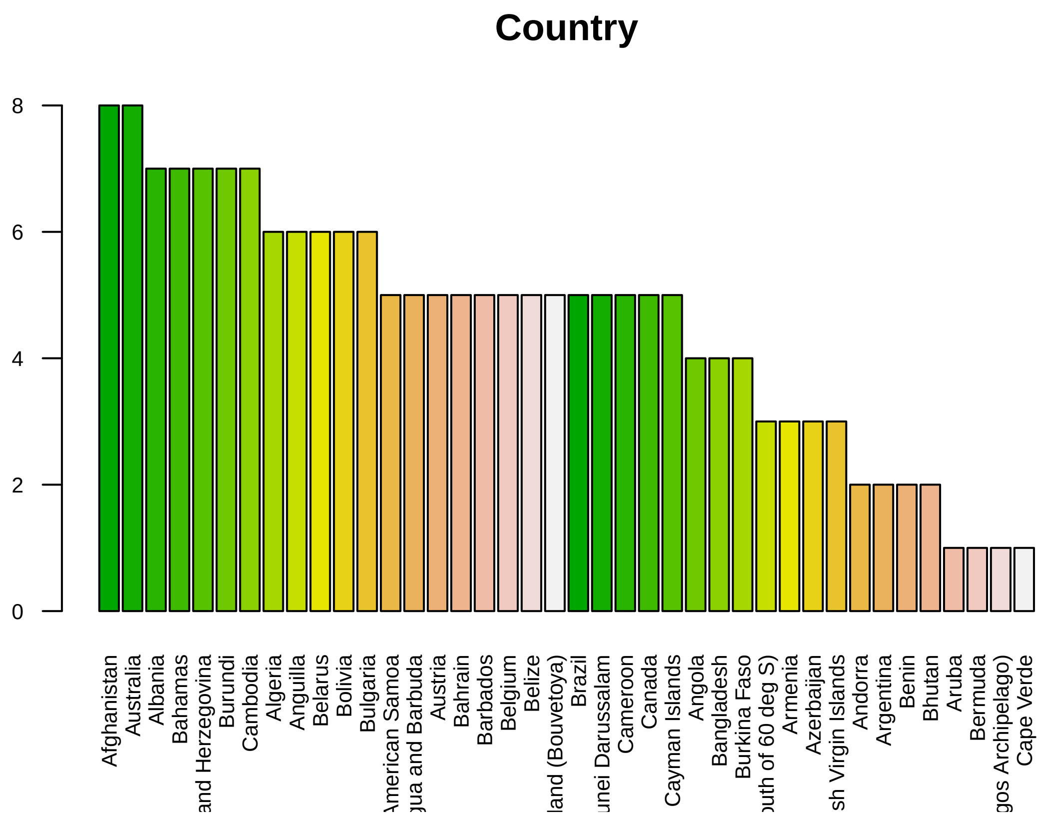
More females than males engage with the ads.

```
clicked <- table(crypto$Clicked.on.Ad)
barplot(clicked,main = "Clicked on Ad",col = topo.colors(2), ylim = c(0,800))
legend("topright",inset = .02, title="Clicked on Ad",
       c("Not Clicked","Clicked"), fill=topo.colors(2), cex=0.8)
```



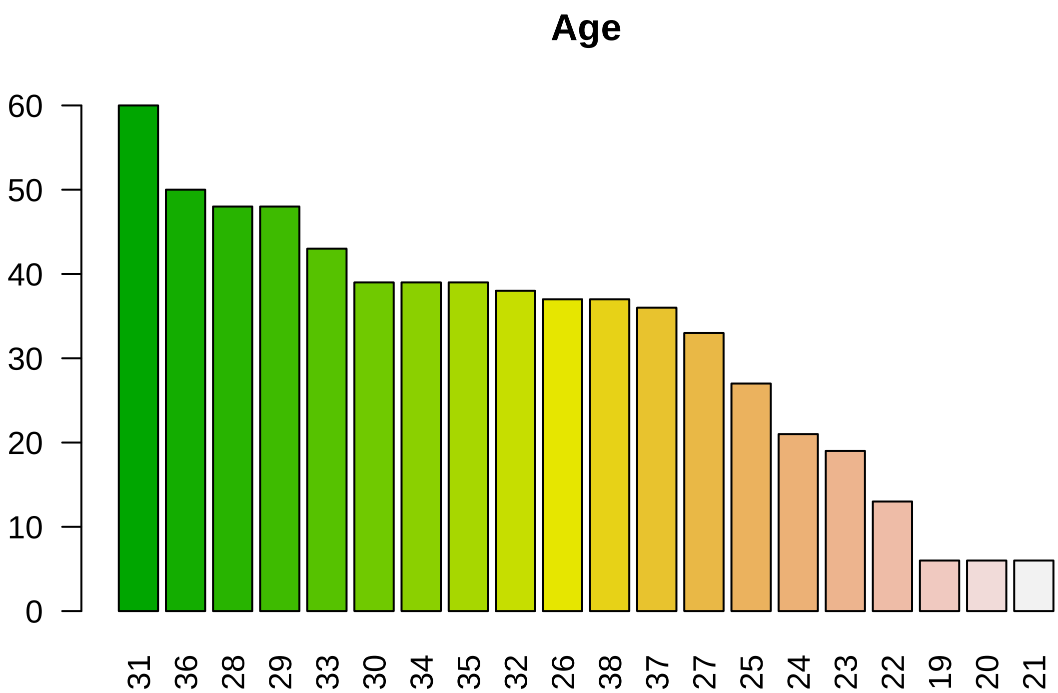
Difference between clicked and not clicked ads is not very significant.

```
par(las=2, cex.axis=0.7)
country <- table(crypto$Country)
barplot(sort(country[1:40], decreasing = TRUE), main = "Country",col = terrain.colors(20))
```



Afhanistan, Australia and Albania have the most engagement. With Cape Verde being the least.

```
par(las=2)
age <- table(crypto$Age)
barplot(sort(age[1:20], decreasing = TRUE), main = "Age",col = terrain.colors(20))
```



Ages 31, 36, 28, 29 and 33 are more actively involved as seen above.

##Bivariate Analysis

```
# install.packages('dplyr')
```

```
# install.packages('ggplot2')
```

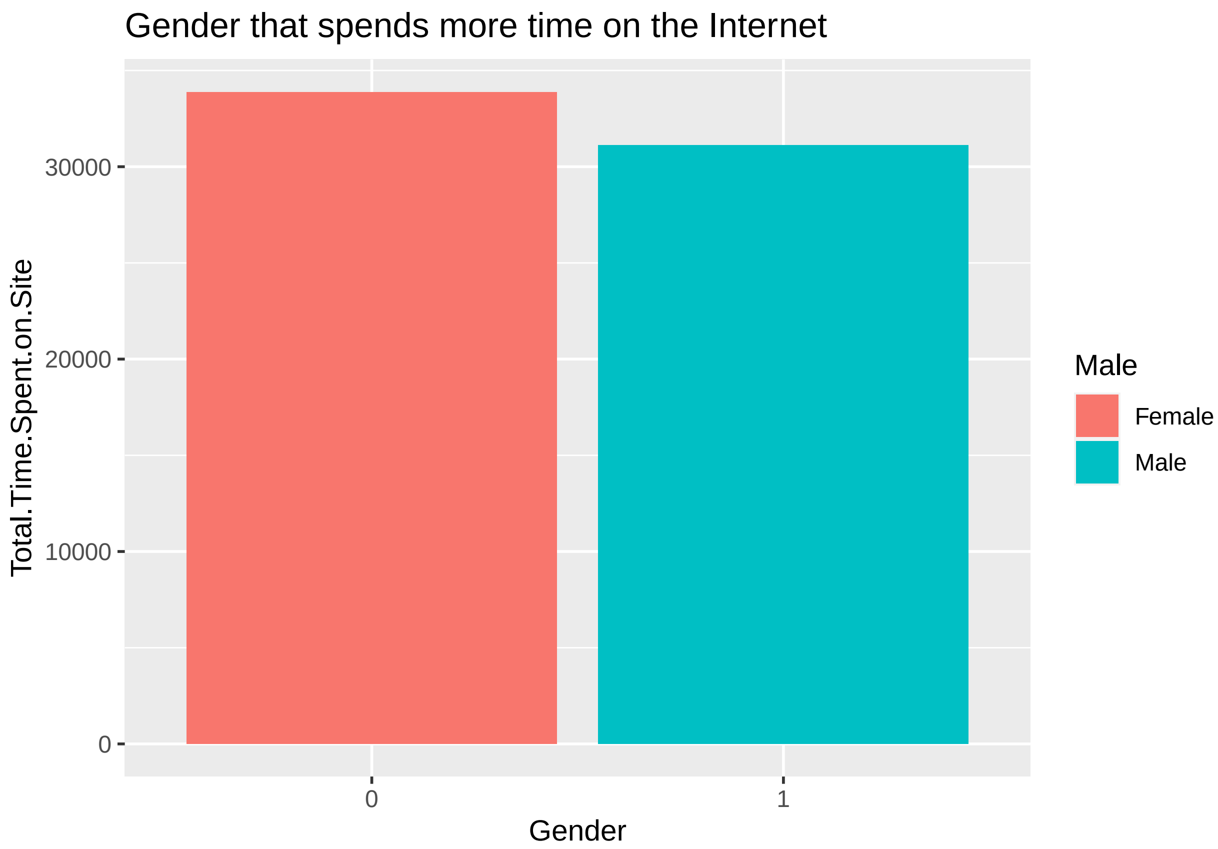
```
library("dplyr")
```

```
##
## 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
```

```
library("ggplot2")
# group by gender/Male
by_time <- crypto %>%
  group_by(Male) %>%
  summarise(Total.Time.Spent.on.Site = sum(Daily.Time.Spent.on.Site))
by_time
```

```
## # A tibble: 2 x 2
##    Male Total.Time.Spent.on.Site
##   <int>                    <dbl>
## 1     0                   33885.
## 2     1                   31115.
```

```
p <- ggplot(by_time, aes(x = factor(Male), y = Total.Time.Spent.on.Site, fill = factor(Male)))+geom_bar
p + scale_fill_discrete(name = "Male", labels = c("Female","Male"))+ labs(title="Gender that spends more
```



Females spend more time on the internet than males.

```
#separating clicked ads
clicked_ad <- crypto[crypto$Clicked.on.Ad == 1,]
```

```
#countries with more clicked ads
library("dplyr")
country <- crypto %>% group_by(Country) %>% summarise(clicked.ad =sum(Clicked.on.Ad[Clicked.on.Ad == 1]
head(country)
```

```
## # A tibble: 6 x 2
##   Country        clicked.ad
##   <chr>               <int>
## 1 Afghanistan             5
## 2 Albania                 4
## 3 Algeria                 3
```

```
## 4 American Samoa          3
## 5 Andorra                 2
## 6 Angola                  1
```

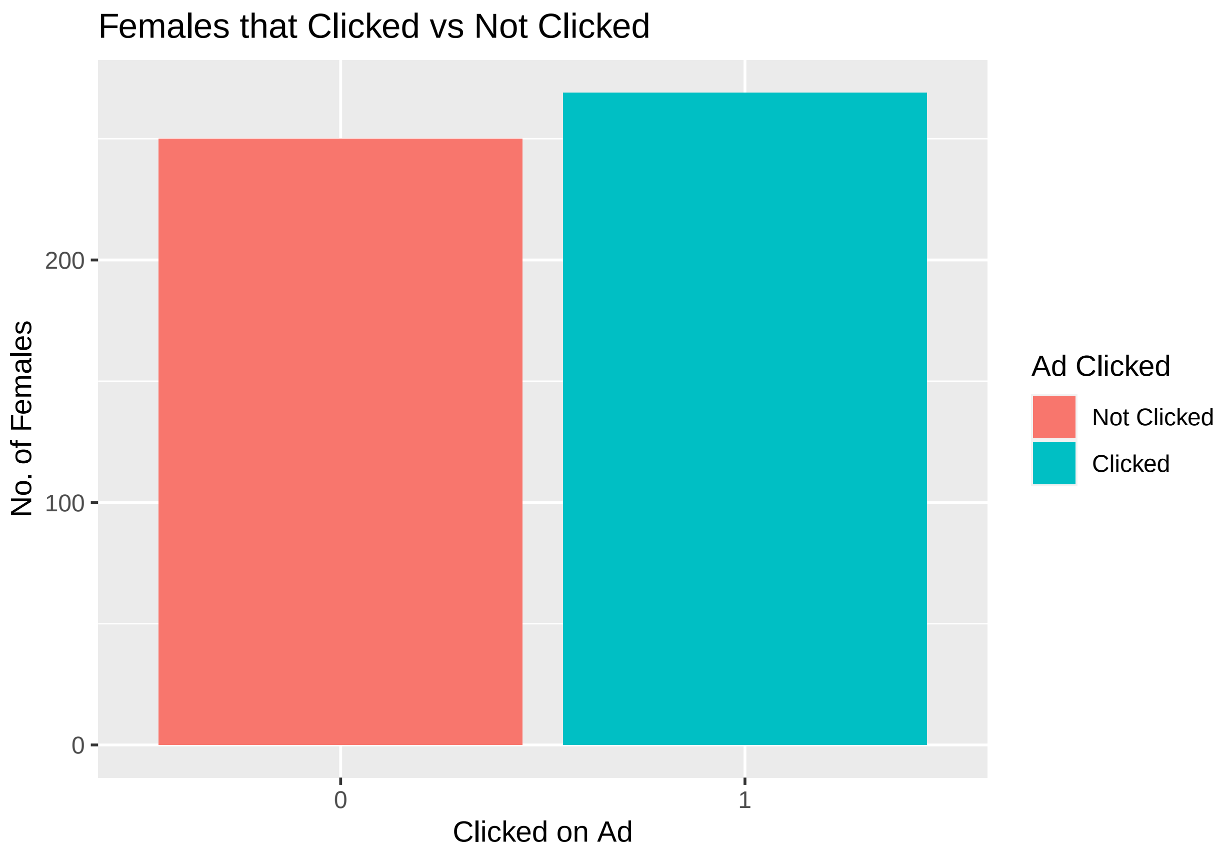
Afghanistan, Albania and Algeria have most clicked ads.

```
# c <- ggplot(rows, aes(x = reorder(Country,clicked.ad), y=clicked.ad)) + geom_col() +coord_flip() +  g
# c + labs(title="Country with Highest Clicks on Ads", x="Countries", y="Clicked Ads")
```

```
# Females that click on ads
gender <- crypto %>% group_by(Clicked.on.Ad) %>% summarise(gender = length(Male[Male == 0]))
gender
```

```
## # A tibble: 2 x 2
##   Clicked.on.Ad gender
##           <int>  <int>
## 1             0    250
## 2             1    269
```

```
females <- ggplot(gender, aes(x = factor(Clicked.on.Ad), y = gender, fill=factor(Clicked.on.Ad))) + geom
females + scale_fill_discrete(name = "Ad Clicked", labels = c("Not Clicked","Clicked"))+ labs(title="Fem
```

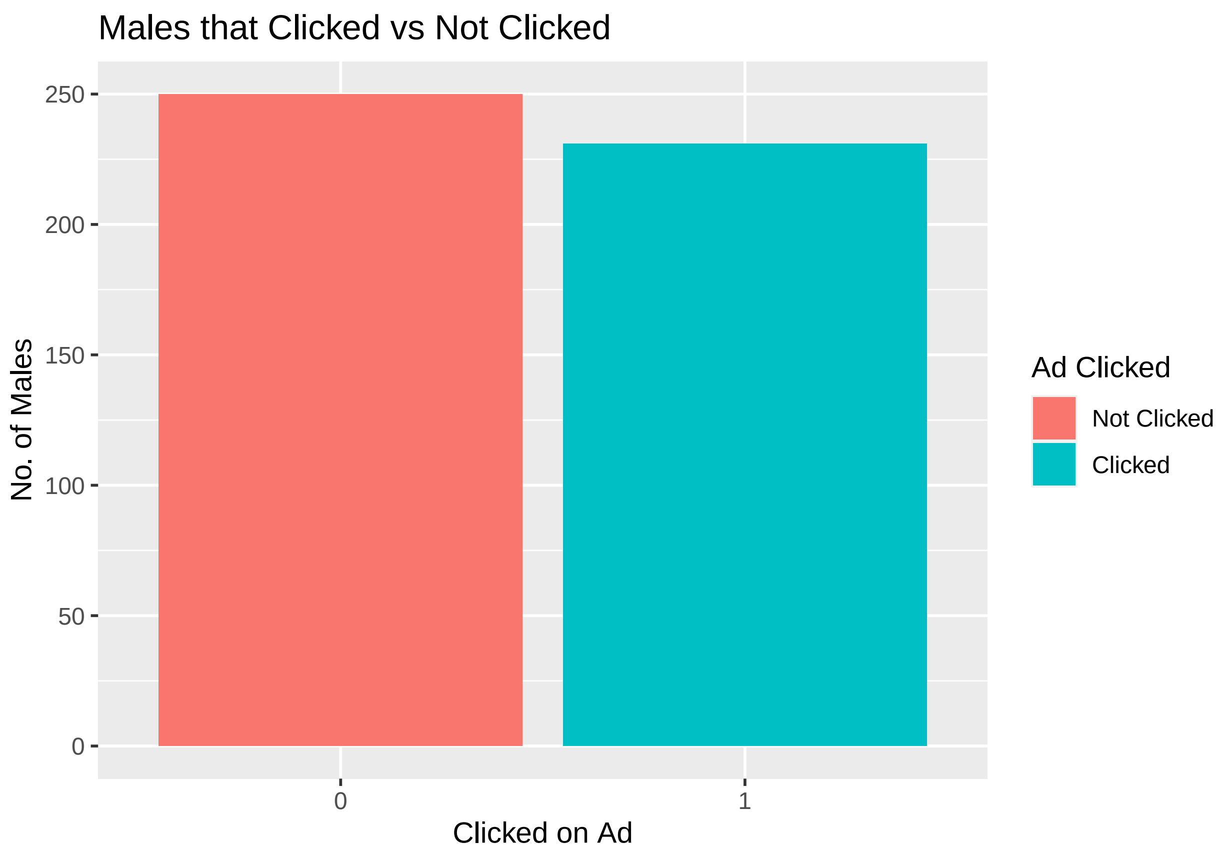


Most females were clicking the ads.

```r
# Males that clicked on ads
males <- crypto %>% group_by(Clicked.on.Ad) %>% summarise(gender = length(Male[Male == 1]))
males
```

```
## # A tibble: 2 x 2
##   Clicked.on.Ad gender
##           <int>  <int>
## 1             0    250
## 2             1    231
```

```r
males <- ggplot(males, aes(x = factor(Clicked.on.Ad), y = gender, fill=factor(Clicked.on.Ad))) + geom_ba
males + scale_fill_discrete(name = "Ad Clicked", labels = c("Not Clicked","Clicked"))+ labs(title="Males
```



Most males did not click on the ads.

```r
str(crypto)
```

```
## '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 standardi
##  $ 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"
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
```

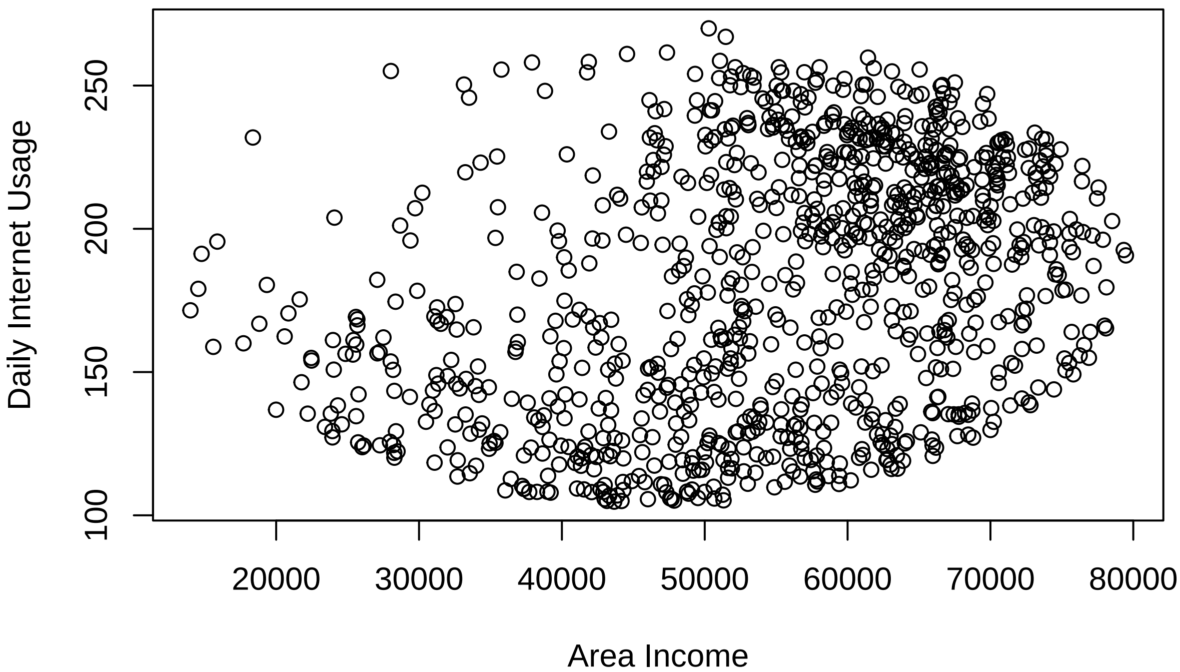
```
head(num,4)
```

```
##   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
##   Clicked.on.Ad
## 1             0
## 2             0
## 3             0
## 4             0
```

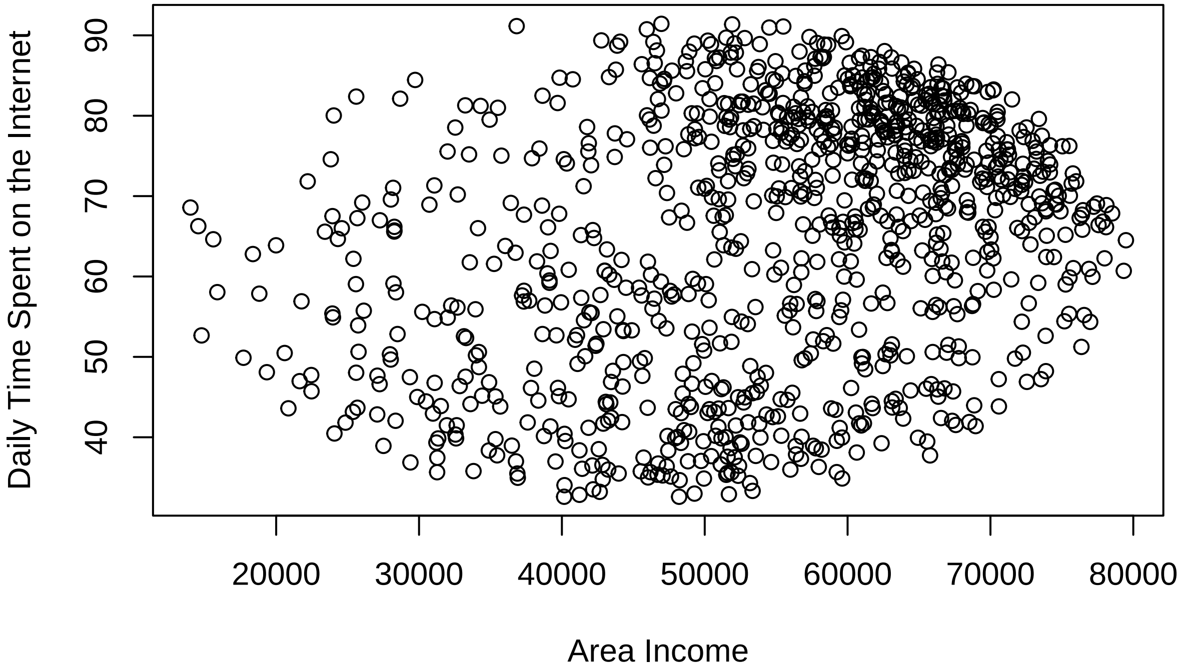
```
# Covariance
covariance = cov(num)
View(round(covariance,2))
```

```
# Correlation Matrix
correlation_matrix = cor(num)
View(round(correlation_matrix,2))
```

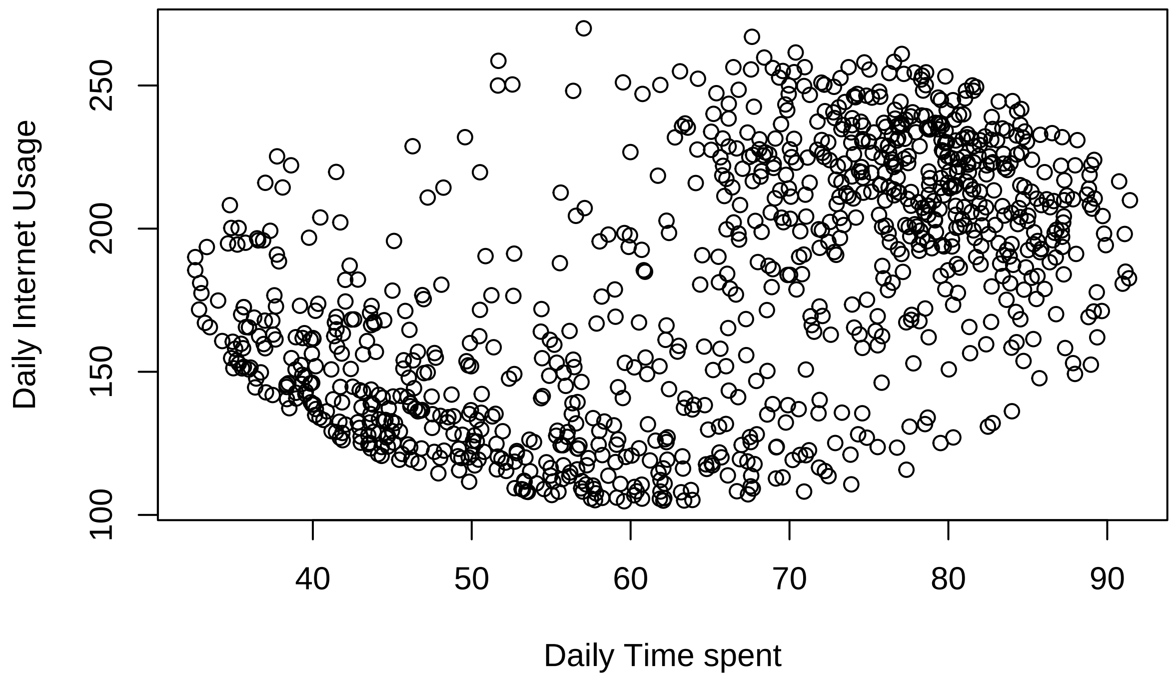
```
# Scatter Plot
area.income <- crypto$Area.Income
internet.usage <- crypto$Daily.Internet.Usage
time.spent <- crypto$Daily.Time.Spent.on.Site

plot(area.income, internet.usage, xlab="Area Income",ylab = "Daily Internet Usage")
```




```
plot(area.income,time.spent,xlab = "Area Income",ylab = "Daily Time Spent on the Internet")
```




```
plot(time.spent,internet.usage, xlab="Daily Time spent", ylab="Daily Internet Usage")
```



##Modelling

```
#importing libraries
library("caret")
```

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

```
library("tidyverse")
```

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

```
## v tibble  3.1.4     v purrr   0.3.4
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

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

```
library("rpart")
library("e1071")
```

```
# Normalize our features
features <- crypto[,c(1,2,3,4,7)]
# The normalization function is created
normalize <-function(x) { (x -min(x))/(max(x)-min(x))}
# Normalization function is applied to the dataframe
crypto_norm <- as.data.frame(lapply(features, normalize))
head(crypto_norm)
```

```
##   Daily.Time.Spent.on.Site       Age Area.Income Daily.Internet.Usage Male
## 1                0.6178820 0.3809524   0.7304725            0.9160310    0
## 2                0.8096209 0.2857143   0.8313752            0.5387456    1
## 3                0.6267211 0.1666667   0.6992003            0.7974331    0
## 4                0.7062723 0.2380952   0.6231599            0.8542802    1
## 5                0.6080231 0.3809524   0.9145678            0.7313234    0
## 6                0.4655788 0.0952381   0.6988280            0.7383460    1
```

```
summary(crypto_norm)
```

```
##  Daily.Time.Spent.on.Site      Age          Area.Income
##  Min.   :0.0000           Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.3189           1st Qu.:0.2381   1st Qu.:0.5044
##  Median :0.6054           Median :0.3810   Median :0.6568
##  Mean   :0.5507           Mean   :0.4050   Mean   :0.6261
##  3rd Qu.:0.7810           3rd Qu.:0.5476   3rd Qu.:0.7860
##  Max.   :1.0000           Max.   :1.0000   Max.   :1.0000
##  Daily.Internet.Usage      Male
##  Min.   :0.0000       Min.   :0.000
##  1st Qu.:0.2061       1st Qu.:0.000
##  Median :0.4743       Median :0.000
##  Mean   :0.4554       Mean   :0.481
##  3rd Qu.:0.6902       3rd Qu.:1.000
##  Max.   :1.0000       Max.   :1.000
```

```
# Generate a random number that is 80% of the total number of rows in dataset
train <- sample(1:nrow(crypto), 0.8 * nrow(crypto))
#training data
crypto_train <- crypto_norm[train,]
crypto_train_target <- as.factor(crypto[train,10])
# testing data
crypto_test <- crypto_norm[-train,]
crypto_test_target <- as.factor(crypto[-train,10])
dim(crypto_train)
```

```
## [1] 800   5
```

```
dim(crypto_test)
```

```
## [1] 200   5
```

KNN Classification

```r
# Applying k-NN classification algorithm.
library(class)
# No. of neighbors are generally square root of total number of instances
neigh <- round(sqrt(nrow(crypto)))+1
knn_model <- knn(crypto_train,crypto_test, cl=crypto_train_target, k=neigh)
# Visualizing classification results
cm_knn <- confusionMatrix(table(crypto_test_target, knn_model))
cm_knn
```

```
## Confusion Matrix and Statistics
##
##                   knn_model
## crypto_test_target   0   1
##                  0  91   2
##                  1   5 102
##
##                Accuracy : 0.965
##                  95% CI : (0.9292, 0.9858)
##     No Information Rate : 0.52
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9298
##
##  Mcnemar's Test P-Value : 0.4497
##
##             Sensitivity : 0.9479
##             Specificity : 0.9808
##          Pos Pred Value : 0.9785
##          Neg Pred Value : 0.9533
##              Prevalence : 0.4800
##          Detection Rate : 0.4550
##    Detection Prevalence : 0.4650
##       Balanced Accuracy : 0.9643
##
##        'Positive' Class : 0
##
```

Decision Trees

```r
# convert the target column to a factor
crypto$Clicked.on.Ad <- as.factor(crypto$Clicked.on.Ad)
features = crypto[,c(1,2,3,4,7,10)]
# Splitting
intrain <- createDataPartition(y = crypto$Clicked.on.Ad, p= 0.8, list = FALSE)
training <- features[intrain,]
testing <- features[-intrain,]
set.seed(42)
myGrid <- expand.grid(mtry = sqrt(ncol(crypto)),
                     splitrule = c("gini", "extratrees"),
                     min.node.size = 20)
dt_model <- train(Clicked.on.Ad ~ .,
               data = training,
               method = "ranger",
```

```r
                tuneGrid = myGrid,
                trControl = trainControl(method='repeatedcv',
                                         number=10,
                                         repeats=3,
                                         search = 'random',
                                        verboseIter = FALSE))
dt_model
```

```
## Random Forest
##
## 800 samples
##   5 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 720, 720, 720, 720, 720, 720, ...
## Resampling results across tuning parameters:
##
##   splitrule   Accuracy   Kappa
##   gini        0.9675000  0.9350000
##   extratrees  0.9704167  0.9408333
##
## Tuning parameter 'mtry' was held constant at a value of 3.162278
##
## Tuning parameter 'min.node.size' was held constant at a value of 20
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were mtry = 3.162278, splitrule
##  = extratrees and min.node.size = 20.
```

```r
# Make predictions and check accuracy
dt_pred <- predict(dt_model,testing )
cm_dt <- confusionMatrix(table(dt_pred, testing$Clicked.on.Ad))
cm_dt
```

```
## Confusion Matrix and Statistics
##
##
## dt_pred  0  1
##       0 95  7
##       1  5 93
##
##                Accuracy : 0.94
##                  95% CI : (0.8975, 0.9686)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.88
##
##  Mcnemar's Test P-Value : 0.7728
##
##             Sensitivity : 0.9500
##             Specificity : 0.9300
```

```
##          Pos Pred Value : 0.9314
##          Neg Pred Value : 0.9490
##              Prevalence : 0.5000
##          Detection Rate : 0.4750
##    Detection Prevalence : 0.5100
##       Balanced Accuracy : 0.9400
##
##        'Positive' Class : 0
##
```

Naive Bayes

```
# split the training into Features and labels for the model
x = training[,1:4]
y = training$Clicked.on.Ad
nb_model <- train(x,y, "nb", trControl = trainControl(method = "repeatedcv",
                       number = 10,
                       repeats = 3),
                      preProcess = c("range"))
# Make prediction
nb_pred <- predict(nb_model , testing)
# Accuracy
cm_nb <- confusionMatrix(table(nb_pred, testing$Clicked.on.Ad))
cm_nb
```

```
## Confusion Matrix and Statistics
##
##
## nb_pred  0  1
##       0 94  5
##       1  6 95
##
##                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 : 1
##
##             Sensitivity : 0.9400
##             Specificity : 0.9500
##          Pos Pred Value : 0.9495
##          Neg Pred Value : 0.9406
##              Prevalence : 0.5000
##          Detection Rate : 0.4700
##    Detection Prevalence : 0.4950
##       Balanced Accuracy : 0.9450
##
##        'Positive' Class : 0
##
```

#Challenging the solution with Support Vector Machines.

```
# Split the Data into Train and Test into 80:20 split
intrain <- createDataPartition(y = crypto$Clicked.on.Ad, p= 0.8, list = FALSE)
training <- features[intrain,]
testing <- features[-intrain,]

set.seed(42)
svm_Linear <- train(Clicked.on.Ad ~ ., data = training, method = "svmLinear",
trControl=trainControl(method = "repeatedcv",
                       number = 10,
                       repeats = 3),
                      preProcess = c("center", "scale"))
# preProcess -> deals with normalization
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
```

```
# Make predictions and check accuracy
test_pred <- predict(svm_Linear, testing)
cm_svmlinear <- confusionMatrix(table(test_pred, testing$Clicked.on.Ad))
cm_svmlinear
```

```
## Confusion Matrix and Statistics
##
##
## test_pred  0  1
##         0 98  7
##         1  2 93
##
##                Accuracy : 0.955
##                  95% CI : (0.9163, 0.9792)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.91
##
##  Mcnemar's Test P-Value : 0.1824
##
##             Sensitivity : 0.9800
##             Specificity : 0.9300
```

```
##          Pos Pred Value : 0.9333
##          Neg Pred Value : 0.9789
##              Prevalence : 0.5000
##          Detection Rate : 0.4900
##    Detection Prevalence : 0.5250
##       Balanced Accuracy : 0.9550
##
##        'Positive' Class : 0
##
```

#Conclusions All models performed well with accuracy scores of above 95%. However, the SVM and Naive BAyes performed the best out of all models.