Online Course Advertising

mmalinda

2/28/2020

1 DS Core Week 12 IP - Online Course Advertising

1.1 Defining the Question

The aim of this analysis is to identify individuals that are most likely to click on advertisements for an online cryptography course on a Kenyan entrepreneur's blog.

1.2 Metrics for Success

The analysis will be successful when the target audience(s) for her advertisements have been identified.

1.3 Context

Targeted advertising is a form of online advertising that uses information collected on the specific traits, interests, and preferences of a consumer to select which ads to place on a website. Irrelevant online advertisements for unwanted products or services can lead to brand erosion and loss of advertising revenue as customers are more likely to block all ads if they feel that they are intruding on their browsing experience. It is therefore important to target advertisments to the right individuals so that the customer feels that the ads are organic, while also ensuring that they do not make consumers feel uncomfortable about the way their data has been collected or shared. The first step to effective targeted advertising is identifying the target audience for the product or service, as this analysis will do.

1.4 Experimental Design

1. Business understanding
2. Data understanding
3. Data exploration
4. Data cleaning
5. Data analysis
6. Evaluation and conclusion

1.5 Data Relevance

The data is from advertising a related course on the same blog, so it is relevant and the source is reliable.

1.6 Data Exploration

```
library("data.table")

library("dplyr")
```

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

## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

1.6.1 Loading the data

```
df <- read.csv('advertising.csv')
head(df, n=10)
```

```
##    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
## 7                     88.91  33    53852.85               208.36
## 8                     66.00  48    24593.33               131.76
## 9                     74.53  30    68862.00               221.51
## 10                    69.88  20    55642.32               183.82
##                          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
## 7              Enhanced dedicated support      Brandonstad    0    Myanmar
## 8                Reactive local challenge Port Jefferybury    1  Australia
## 9          Configurable coherent function       West Colin    1    Grenada
## 10    Mandatory homogeneous architecture       Ramirezton    1      Ghana
##              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
## 7  2016-01-28 20:59:32             0
## 8  2016-03-07 01:40:15             1
## 9  2016-04-18 09:33:42             0
## 10 2016-07-11 01:42:51             0
```

1.6.2 Dataset Description

```
dim(df)
```

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

The dataframe has 1000 records and 10 variables

```
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
##  Adaptive 24hour Graphic Interface      :  1   Lisamouth      :  3
##  Adaptive asynchronous attitude         :  1   Williamsport   :  3
##  Adaptive context-sensitive application :  1   Benjaminchester:  2
##  Adaptive contextually-based methodology:  1   East John      :  2
##  Adaptive demand-driven knowledgebase   :  1   East Timothy   :  2
##  Adaptive uniform capability            :  1   Johnstad       :  2
##  (Other)                                :994   (Other)        :986
##       Male                 Country                   Timestamp   Clicked.on.Ad
##  Min.   :0.000   Czech Republic:  9   2016-01-01 02:52:10:  1   Min.   :0.0
##  1st Qu.:0.000   France        :  9   2016-01-01 03:35:35:  1   1st Qu.:0.0
##  Median :0.000   Afghanistan   :  8   2016-01-01 05:31:22:  1   Median :0.5
##  Mean   :0.481   Australia     :  8   2016-01-01 08:27:06:  1   Mean   :0.5
##  3rd Qu.:1.000   Cyprus        :  8   2016-01-01 15:14:24:  1   3rd Qu.:1.0
##  Max.   :1.000   Greece        :  8   2016-01-01 20:17:49:  1   Max.   :1.0
##                  (Other)       :950   (Other)            :994
```

```
str(df)
```

```
## '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           : Factor w/ 1000 levels "Adaptive 24hour Graphic Interface",..: 92 465 56
##  $ City                    : Factor w/ 969 levels "Adamsbury","Adamside",..: 962 904 112 940 806 283
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Country                 : Factor w/ 237 levels "Afghanistan",..: 216 148 185 104 97 159 146 13 83
##  $ Timestamp               : Factor w/ 1000 levels "2016-01-01 02:52:10",..: 440 475 368 57 768 690
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
```

The output above gives a simple summary of the data. The daily time spent on the site seems to be in minutes and seconds. The values range from 32.60 to 91.43. The median is 68.22 and the mean is 65, so the values are likely to be close to normally distributed. The ages range from 19 to 61 years old, with a median of 35 and a mean of 36.01. The values are also likely to be close to normally distributed. The area income ranges from 13996 to 79485, with a median of 57012 and a mean of 55000. The values are not likely to be close to normally distributed due to this difference. The daily internet usage ranges from 104.8 to 270.0, with a median of 183.1 and a mean of 180.0. The values are likely to be close to normally distributed. The ad topic line is a categrocial feature, with a different value for each record. The feature can therefore be dropped. City is a categorical feature with high cardinality (the highest frequency is 3 records). The feature male is categorical (binary) with a mean of 0.481, which means there are more records from individuals that are not male. Country is a categorical feature with high cardinality (the highest frequency is 9 records). Time stamp has high cardinality and can be split into year, month, day, hour, minute, and second. The clicked on ad variable is categorical (binary) with a mean of 0.5, which means that the variable of interest is balanced in this dataset.

Apart from city names which are fictional, there are no apparent anomalies.

```
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 this dataset.

```
df[duplicated(df),]
```

```
##  [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)
```

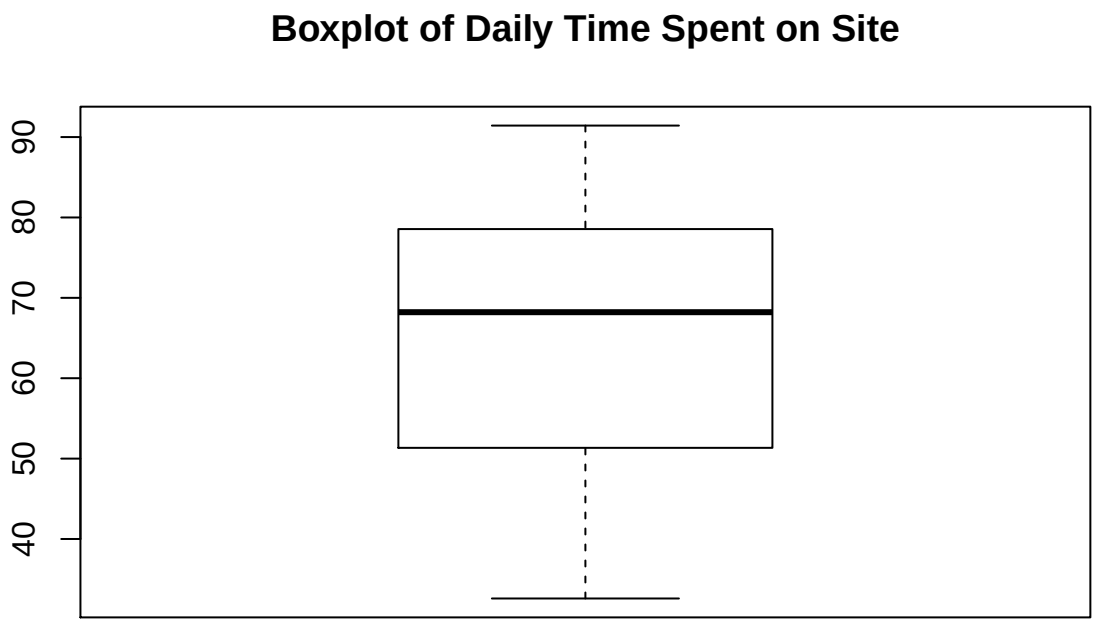
The dataset does not have duplicated records.

1.7 Univariate Analysis

First, a function for mode will be created since R does not have a built in function.

```
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

```
boxplot(df$Daily.Time.Spent.on.Site, main = "Boxplot of Daily Time Spent on Site")
```




```
library(e1071)
paste("mode:", getmode(df$Daily.Time.Spent.on.Site))
```

```
## [1] "mode: 62.26"
```

```
paste("variance:", var(df$Daily.Time.Spent.on.Site))
```

```
## [1] "variance: 251.337094854855"
```

```
paste("std dev:", sd(df$Daily.Time.Spent.on.Site))
```

```
## [1] "std dev: 15.8536145675002"
```

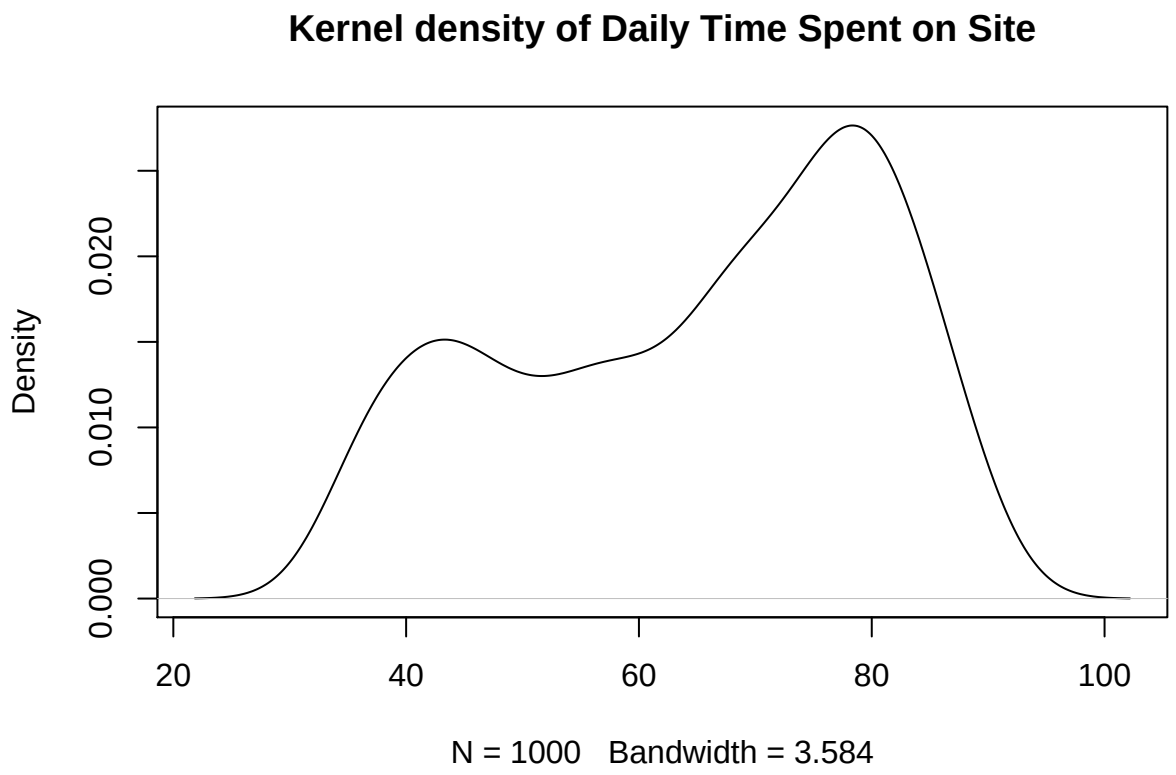
```r
paste("kurtosis:", kurtosis(df$Daily.Time.Spent.on.Site))
```

```
## [1] "kurtosis: -1.09986382635506"
```

```r
paste("skewness:", skewness(df$Daily.Time.Spent.on.Site))
```

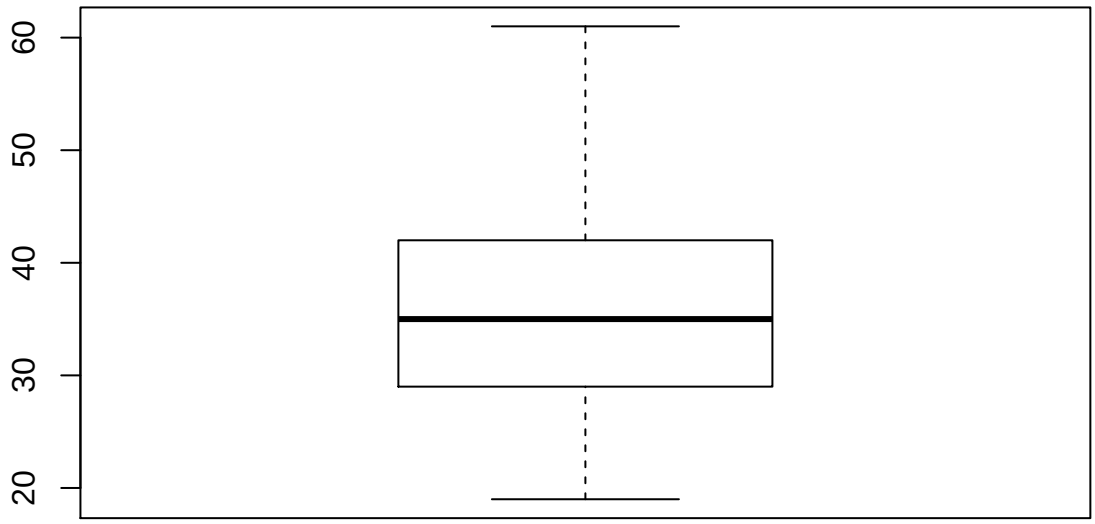
```
## [1] "skewness: -0.370645950169329"
```

```r
plot(density(df$Daily.Time.Spent.on.Site), main = "Kernel density of Daily Time Spent on Site")
```



The time spent on the site is left skewed, i.e. most people spend more time on the site. The variance is high.

```r
boxplot(df$Age, main = "Boxplot of Age")
```




```
paste("mode:", getmode(df$Age))
```

```
## [1] "mode: 31"
```

```
paste("variance:", var(df$Age))
```

```
## [1] "variance: 77.1861051051051"
```

```
paste("std dev:", sd(df$Age))
```

```
## [1] "std dev: 8.78556231012592"
```

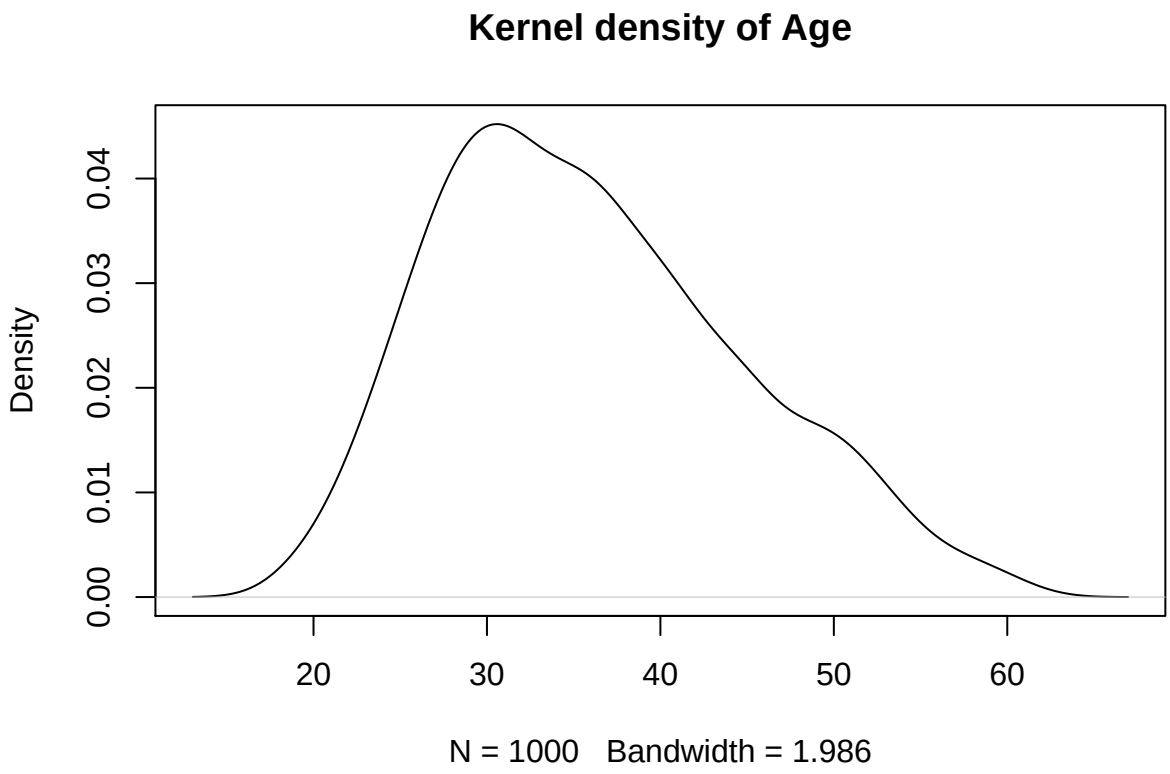
```
paste("kurtosis:", kurtosis(df$Age))
```

```
## [1] "kurtosis: -0.409706599977131"
```

```
paste("skewness:", skewness(df$Age))
```

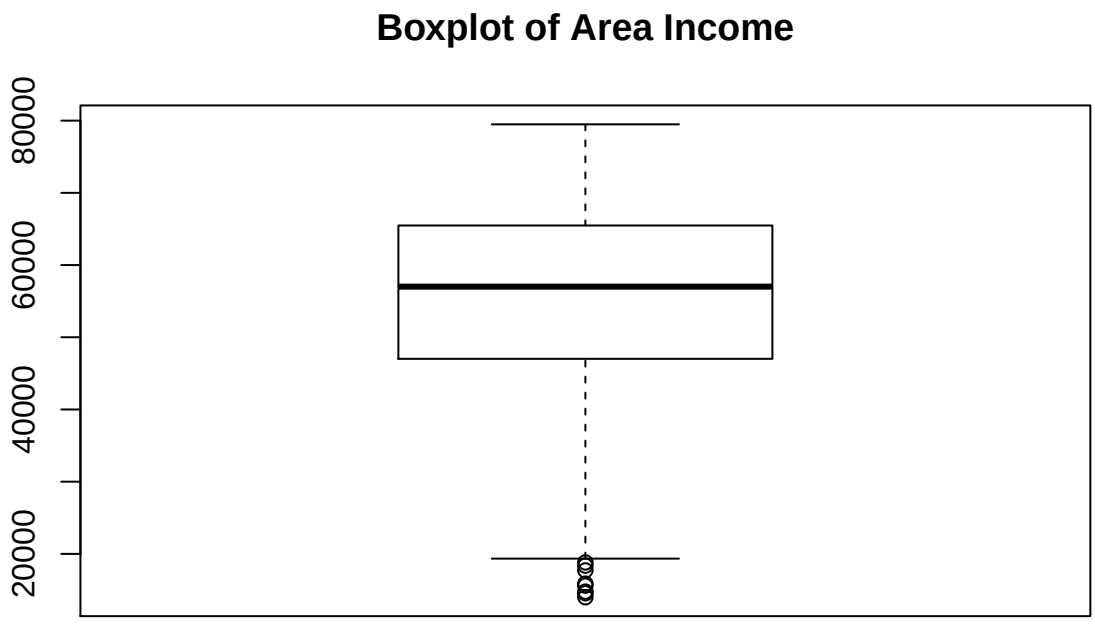
```
## [1] "skewness: 0.477705221630714"
```

```r
plot(density(df$Age), main = "Kernel density of Age")
```



Since the data is right-skewed, most visitors to the site are younger. There is high variance in age.

```r
boxplot(df$Area.Income, main = "Boxplot of Area Income")
```




```
paste("mode:", getmode(df$Area.Income))
```

```
## [1] "mode: 61833.9"
```

```
paste("variance:", var(df$Area.Income))
```

```
## [1] "variance: 179952405.951775"
```

```
paste("std dev:", sd(df$Area.Income))
```

```
## [1] "std dev: 13414.6340222824"
```

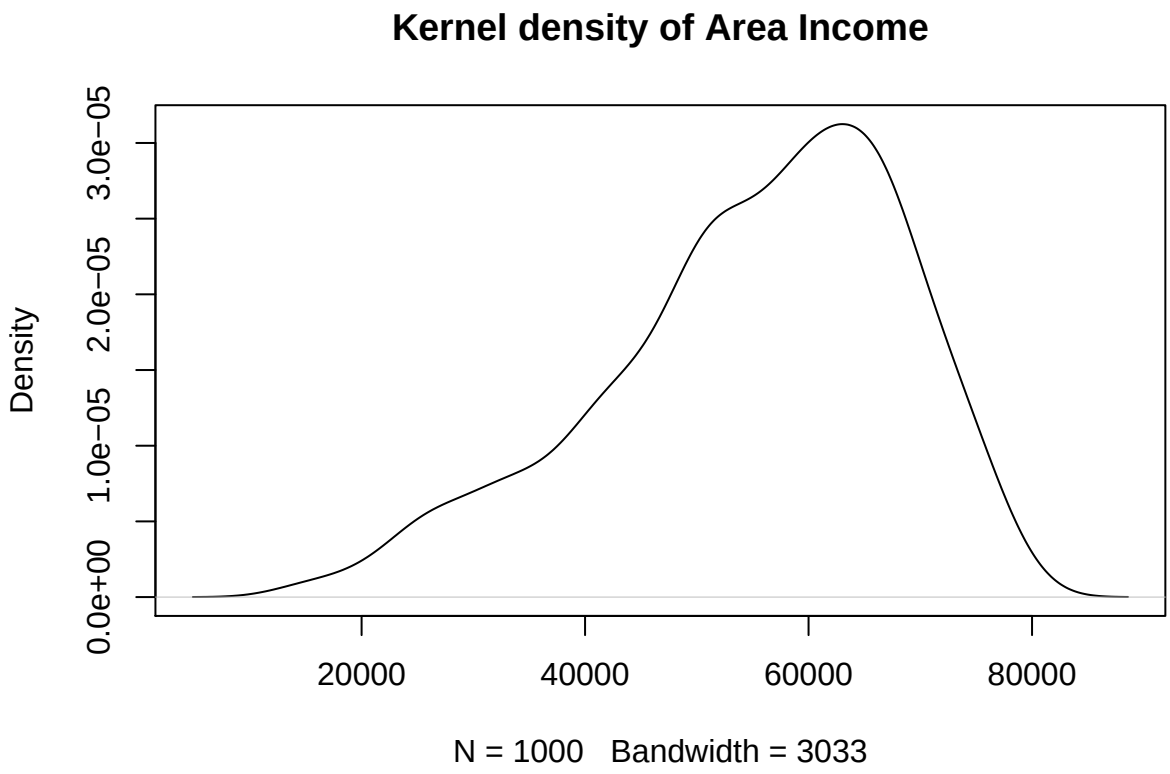
```
paste("kurtosis:", kurtosis(df$Area.Income))
```

```
## [1] "kurtosis: -0.111092431809917"
```

```
paste("skewness:", skewness(df$Area.Income))
```

```
## [1] "skewness: -0.648422850205901"
```

```r
plot(density(df$Area.Income), main = "Kernel density of Area Income")
```



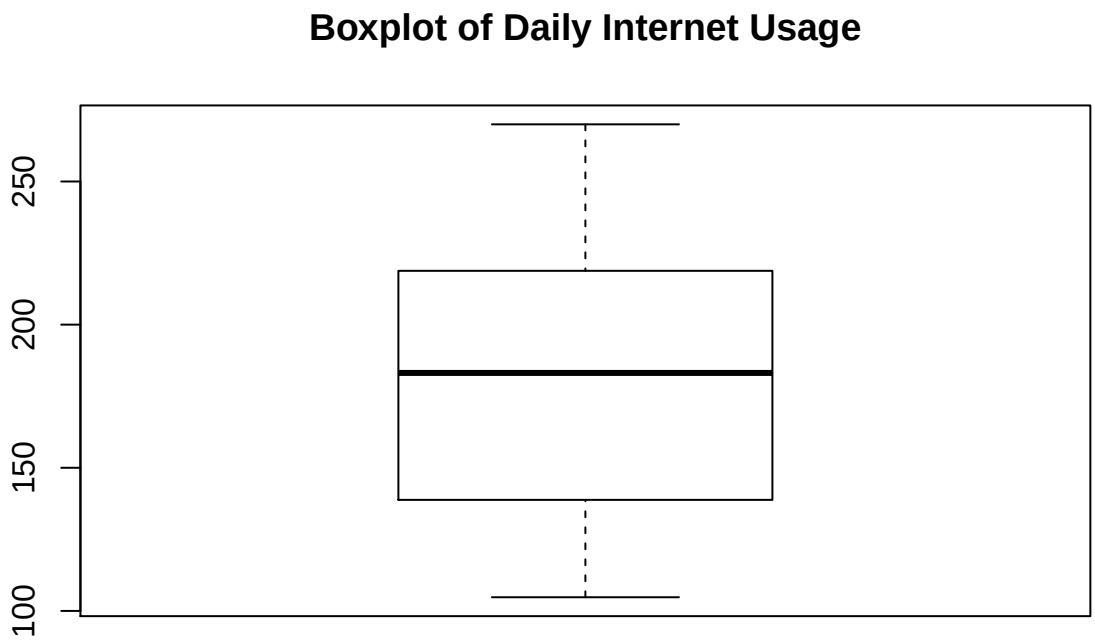
The data is left skewed, with outliers on the lower end. The variance is high. Looking at these outliers can tell us where the majority of visitors to the site fall in terms of income.

```r
boxplot.stats(df$Area.Income)$out
```

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

The outliers have area income under 19000.

```r
boxplot(df$Daily.Internet.Usage, main = "Boxplot of Daily Internet Usage")
```




```
paste("mode:", getmode(df$Daily.Internet.Usage))
```

```
## [1] "mode: 167.22"
```

```
paste("variance:", var(df$Daily.Internet.Usage))
```

```
## [1] "variance: 1927.41539618619"
```

```
paste("std dev:", sd(df$Daily.Internet.Usage))
```

```
## [1] "std dev: 43.9023393019801"
```

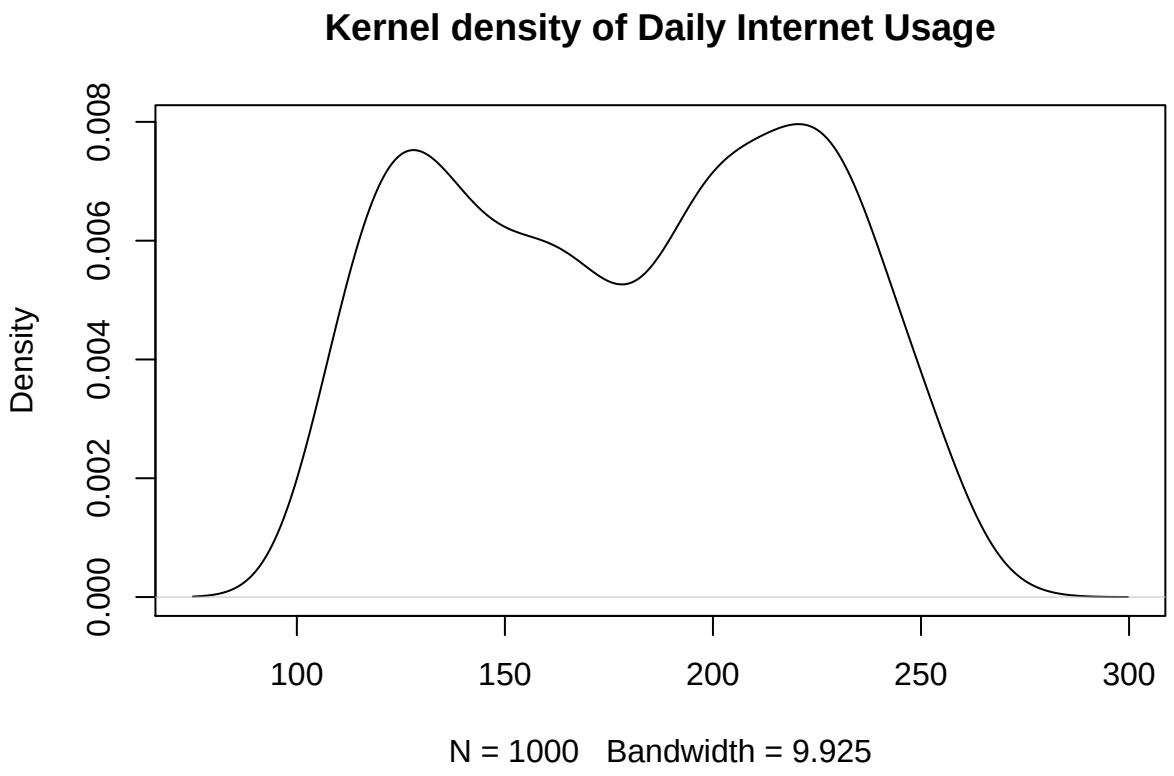
```
paste("kurtosis:", kurtosis(df$Daily.Internet.Usage))
```

```
## [1] "kurtosis: -1.27575249371253"
```

```
paste("skewness:", skewness(df$Daily.Internet.Usage))
```

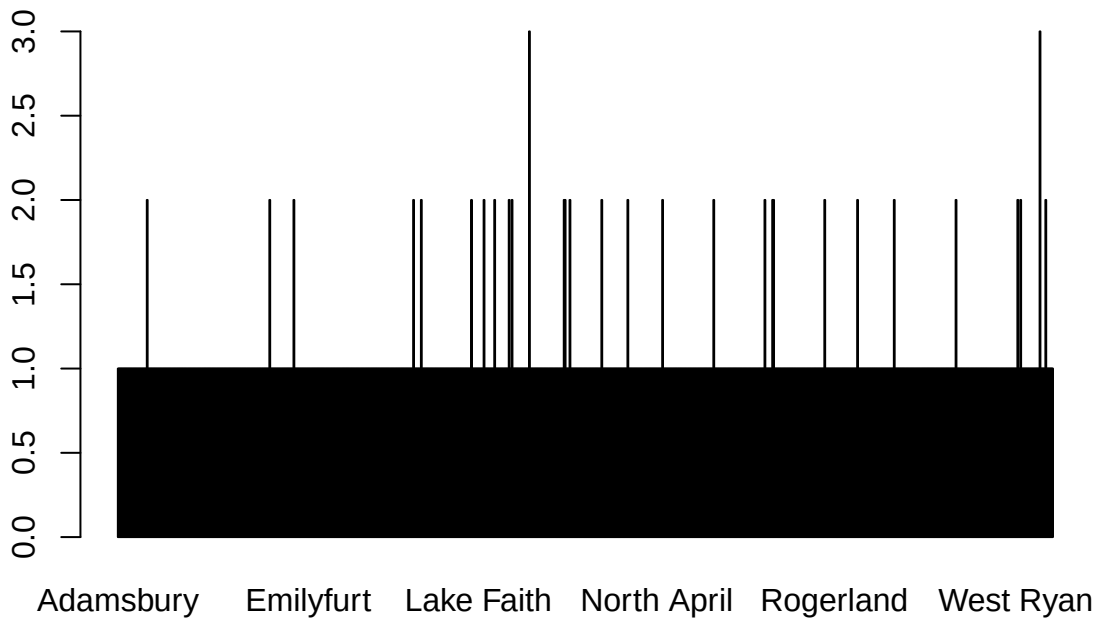
```
## [1] "skewness: -0.0334368136557063"
```

```r
plot(density(df$Daily.Internet.Usage), main = "Kernel density of Daily Internet Usage")
```



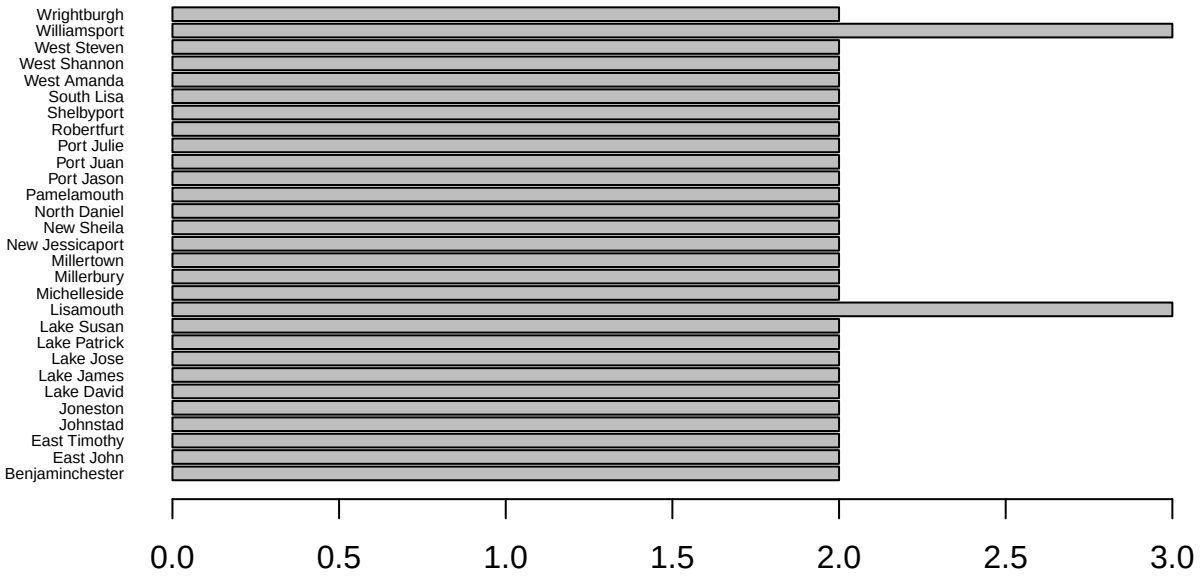
The mode is lower than the mean and median, and variance is high.

```r
city_freq <- table(df$City)
barplot(city_freq)
```



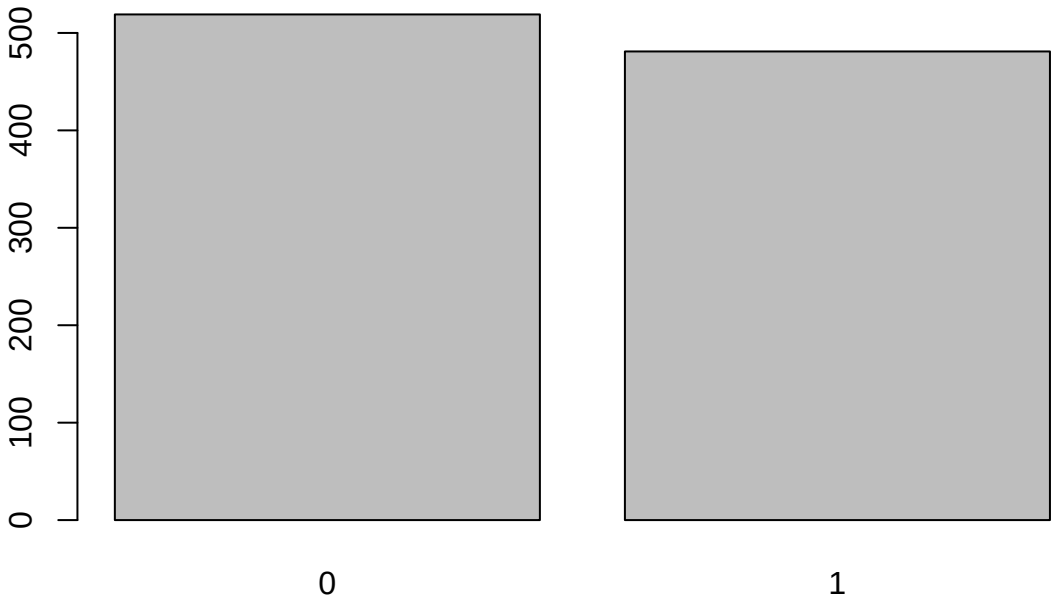
There are few cities with visitor frequency higher than 1. These can be examined to see which cities have more than one visitor.

```
barplot(city_freq[city_freq > 1], horiz = TRUE, las =1, cex.names=0.5)
```



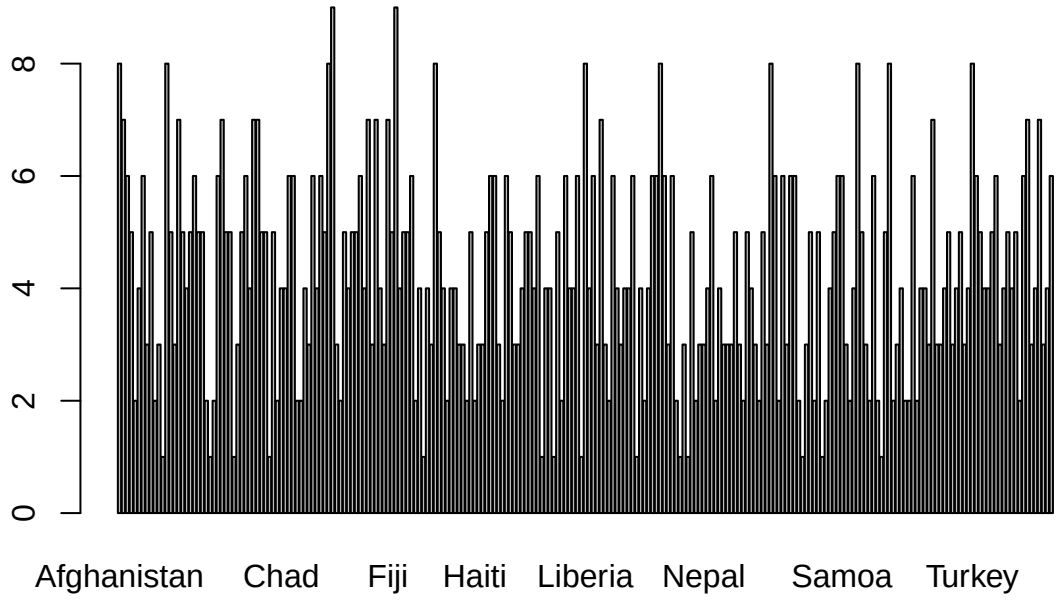
Since these are fictional cities, we cannot comment on the region/country. However, from the variance we can see that the city is unlikely to be an important factor.

```
sex_freq <- table(df$Male)
barplot(sex_freq)
```



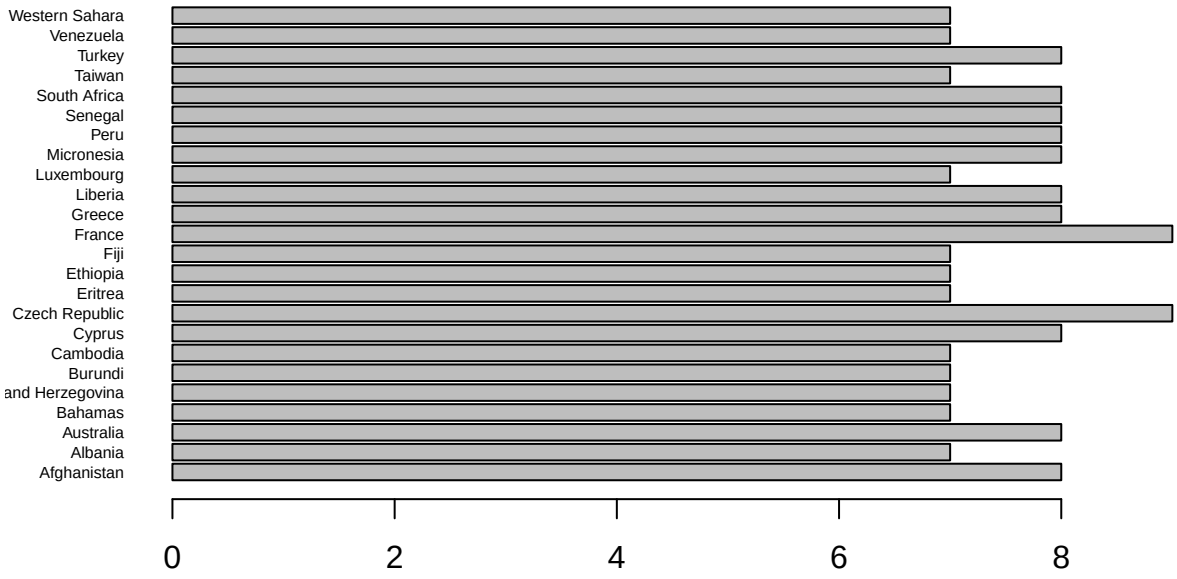
There are more female than male visitors to the site.

```
country_freq <- table(df$Country)
barplot(country_freq)
```



There are few countries with more than 6 visitors to the site.

```
barplot(country_freq[country_freq > 6], horiz = TRUE, las =1, cex.names=0.5)
```



There is no specific region with the most visitors to the site.

```
topic_freq <- table(df$Ad.Topic.Line)
barplot(topic_freq)
```

This feature has no variance, so it can be dropped.

```
df <- select(df, -Ad.Topic.Line)
names(df)
```

```
## [1] "Daily.Time.Spent.on.Site" "Age"
## [3] "Area.Income"              "Daily.Internet.Usage"
## [5] "City"                     "Male"
## [7] "Country"                  "Timestamp"
## [9] "Clicked.on.Ad"
```

The timestamp feature should be split into year, month, day, hour, and second.

```
head(df$Timestamp)
```

```
## [1] 2016-03-27 00:53:11 2016-04-04 01:39:02 2016-03-13 20:35:42
## [4] 2016-01-10 02:31:19 2016-06-03 03:36:18 2016-05-19 14:30:17
## 1000 Levels: 2016-01-01 02:52:10 2016-01-01 03:35:35 ... 2016-07-24 00:22:16
```

The format is YYYY-MM-DD HH:MM:SS. This can be broken down into separate variables.

```
df$day <- format(as.POSIXct(strptime(df$Timestamp,"%Y-%m-%d %H:%M:%S",tz="")) ,format = "%d")
head(df$day)
```

```
## [1] "27" "04" "13" "10" "03" "19"
```

```
df$month <- format(as.POSIXct(strptime(df$Timestamp,"%Y-%m-%d %H:%M:%S",tz="")) ,format = "%m")
head(df$month)
```

```
## [1] "03" "04" "03" "01" "06" "05"
```

```
df$year <- format(as.POSIXct(strptime(df$Timestamp,"%Y-%m-%d %H:%M:%S",tz="")) ,format = "%Y")
head(df$year)
```

```
## [1] "2016" "2016" "2016" "2016" "2016" "2016"
```

```
df$hour <- format(as.POSIXct(strptime(df$Timestamp,"%Y-%m-%d %H:%M:%S",tz="")) ,format = "%H")
head(df$hour)
```

```
## [1] "00" "01" "20" "02" "03" "14"
```

```
df$min <- format(as.POSIXct(strptime(df$Timestamp,"%Y-%m-%d %H:%M:%S",tz="")) ,format = "%M")
head(df$min)
```

```
## [1] "53" "39" "35" "31" "36" "30"
```

```
df$sec <- format(as.POSIXct(strptime(df$Timestamp,"%Y-%m-%d %H:%M:%S",tz="")) ,format = "%S")
head(df$sec)
```

```
## [1] "11" "02" "42" "19" "18" "17"
```

```
paste(head(df$year), head(df$month), head(df$day), head(df$hour), head(df$min), head(df$sec))
```

```
## [1] "2016 03 27 00 53 11" "2016 04 04 01 39 02" "2016 03 13 20 35 42"
## [4] "2016 01 10 02 31 19" "2016 06 03 03 36 18" "2016 05 19 14 30 17"
```

```
names(df)
```

```
##  [1] "Daily.Time.Spent.on.Site" "Age"
##  [3] "Area.Income"              "Daily.Internet.Usage"
##  [5] "City"                     "Male"
##  [7] "Country"                  "Timestamp"
##  [9] "Clicked.on.Ad"            "day"
## [11] "month"                    "year"
## [13] "hour"                     "min"
## [15] "sec"
```

The column was successfully split. To analyse the date and time data we need to convert these columns to numeric data type.

```
df[,10:15] <- sapply(df[,10:15],as.numeric)
class(df$day)
```

```
## [1] "numeric"
```

The Timestamp column can now be dropped

```r
df <- select(df, -Timestamp)
names(df)
```

```
##  [1] "Daily.Time.Spent.on.Site" "Age"
##  [3] "Area.Income"              "Daily.Internet.Usage"
##  [5] "City"                     "Male"
##  [7] "Country"                  "Clicked.on.Ad"
##  [9] "day"                      "month"
## [11] "year"                     "hour"
## [13] "min"                      "sec"
```

1.8 Bivariate analyis

1.8.1 Correlation matrix

```r
num_cols <- Filter(is.numeric, df)
cor(num_cols)
```

```
##                          Daily.Time.Spent.on.Site          Age  Area.Income
## Daily.Time.Spent.on.Site             1.0000000000 -0.331513343  0.310954413
## Age                                 -0.3315133428  1.000000000 -0.182604955
## Area.Income                          0.3109544125 -0.182604955  1.000000000
## Daily.Internet.Usage                 0.5186584753 -0.367208560  0.337495533
## Male                                -0.0189508546 -0.021044064  0.001322359
## Clicked.on.Ad                       -0.7481165641  0.492531266 -0.476254628
## day                                 -0.0112173604 -0.038161625 -0.026523412
## month                               -0.0109195620  0.023689247 -0.050216130
## year                                           NA           NA           NA
## hour                                 0.0008949812 -0.049905128  0.034572917
## min                                 -0.0218149343 -0.030467212  0.001157562
## sec                                  0.0361737515 -0.009499007  0.007935967
##                          Daily.Internet.Usage         Male Clicked.on.Ad
## Daily.Time.Spent.on.Site           0.51865848 -0.018950855  -0.748116564
## Age                               -0.36720856 -0.021044064   0.492531266
## Area.Income                        0.33749553  0.001322359  -0.476254628
## Daily.Internet.Usage               1.00000000  0.028012326  -0.786539176
## Male                               0.02801233  1.000000000  -0.038027466
## Clicked.on.Ad                     -0.78653918 -0.038027466   1.000000000
## day                               -0.01253076 -0.013252632  -0.005269365
## month                              0.01752985  0.005219737   0.016095459
## year                                       NA           NA           NA
## hour                               0.07434699  0.058552057  -0.047431029
## min                                0.01060475  0.057699607   0.022969162
## sec                                0.03534903  0.029416421  -0.031512939
##                                  day        month year          hour
## Daily.Time.Spent.on.Site -0.011217360 -0.010919562   NA  0.0008949812
## Age                      -0.038161625  0.023689247   NA -0.0499051285
## Area.Income              -0.026523412 -0.050216130   NA  0.0345729170
## Daily.Internet.Usage     -0.012530762  0.017529853   NA  0.0743469886
## Male                     -0.013252632  0.005219737   NA  0.0585520575
## Clicked.on.Ad            -0.005269365  0.016095459   NA -0.0474310291
```

```
## day                          1.000000000 -0.017273510   NA -0.0170644864
## month                       -0.017273510  1.000000000   NA -0.0137476053
## year                                  NA           NA    1            NA
## hour                        -0.017064486 -0.013747605   NA  1.0000000000
## min                          0.037559426 -0.089898643   NA -0.0211057370
## sec                          0.022899053  0.030837283   NA  0.0122824175
##                                      min          sec
## Daily.Time.Spent.on.Site -0.021814934  0.036173752
## Age                      -0.030467212 -0.009499007
## Area.Income               0.001157562  0.007935967
## Daily.Internet.Usage      0.010604748  0.035349033
## Male                      0.057699607  0.029416421
## Clicked.on.Ad             0.022969162 -0.031512939
## day                       0.037559426  0.022899053
## month                    -0.089898643  0.030837283
## year                                  NA           NA
## hour                     -0.021105737  0.012282417
## min                       1.000000000 -0.036879768
## sec                      -0.036879768  1.000000000
```

Daily time spent on the site and daily internet usage are strongly negatively correlated to whether the visitor clicked on the ad. Age and area income are moderately correlated (positively and negatively, respectively). The rest do not have a strong correlation to whether the person clicked on the ad.

The year column has null values for correlation, which we will look into.

```
unique(df$year)
```

```
## [1] 2016
```

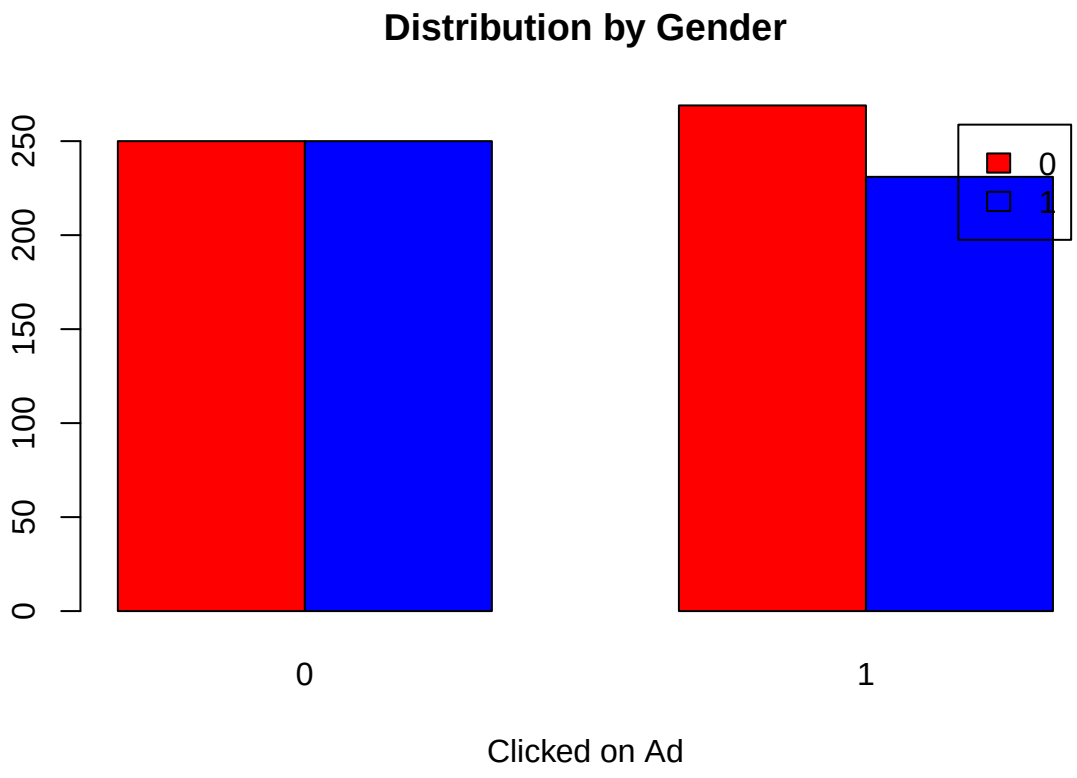
The column has only one unique value and hence can be dropped.

```
df <- select(df, -year)
names(df)
```

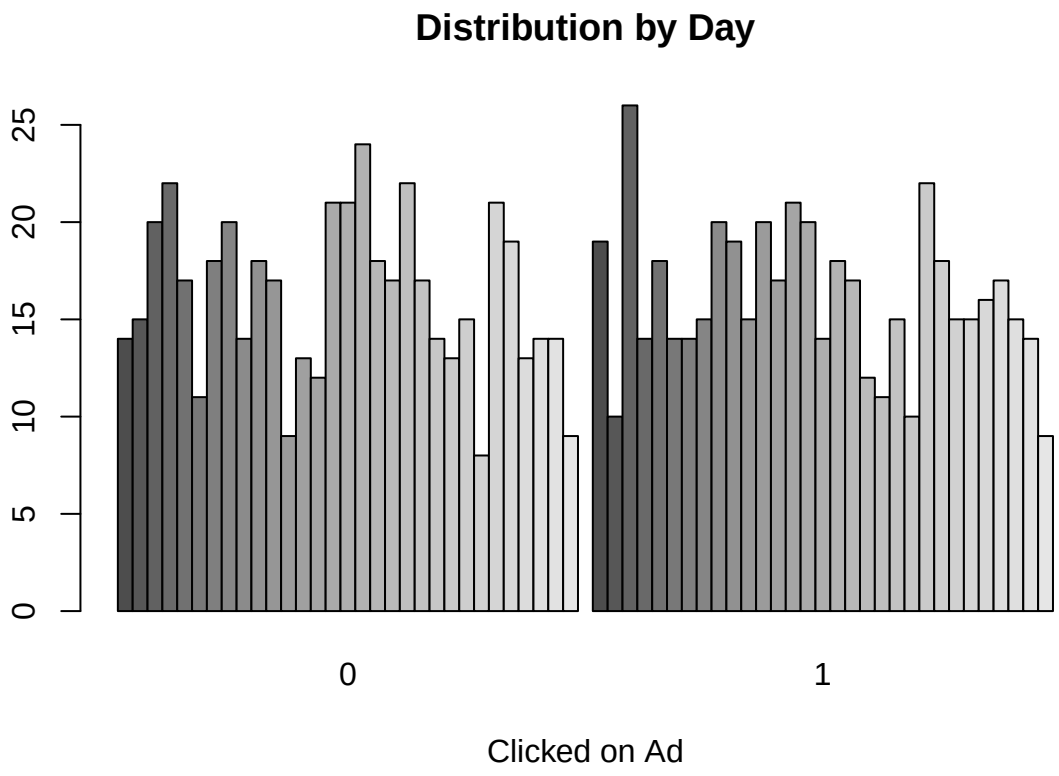
```
##  [1] "Daily.Time.Spent.on.Site" "Age"
##  [3] "Area.Income"              "Daily.Internet.Usage"
##  [5] "City"                     "Male"
##  [7] "Country"                  "Clicked.on.Ad"
##  [9] "day"                      "month"
## [11] "hour"                     "min"
## [13] "sec"
```

1.8.2 Bivariate plots

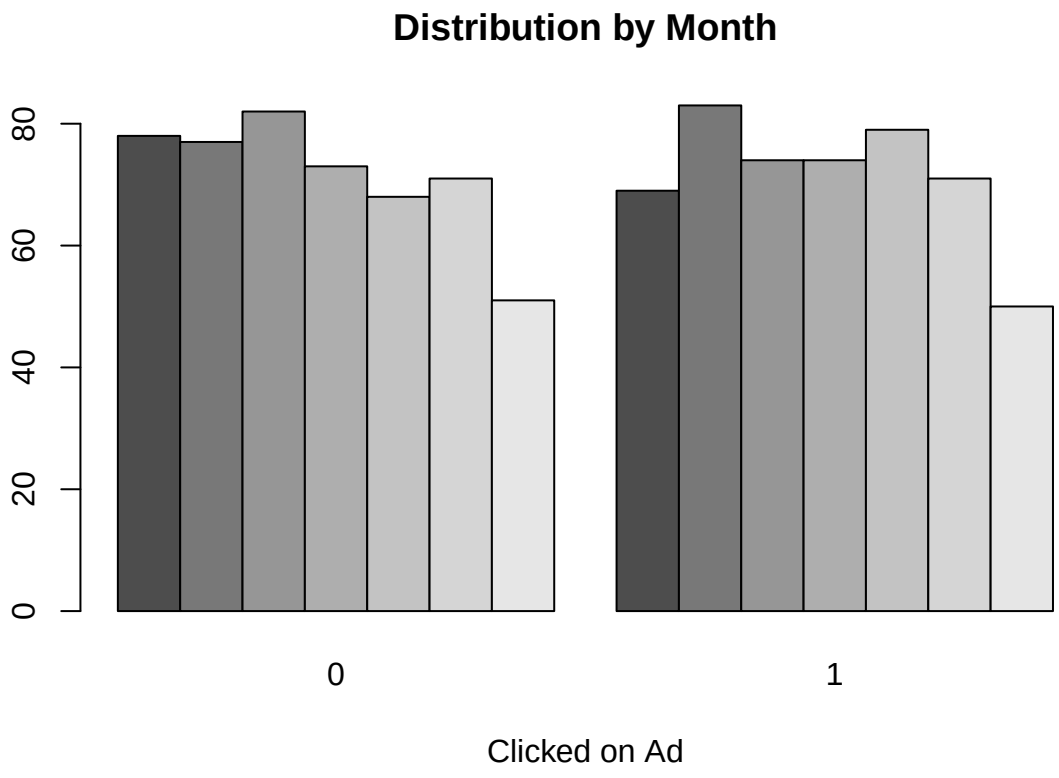
```
gender <- table(df$Male, df$Clicked.on.Ad)
barplot(gender, main="Distribution by Gender",
  xlab="Clicked on Ad", col=c("red","blue"),
  legend = rownames(gender), beside=TRUE)
```



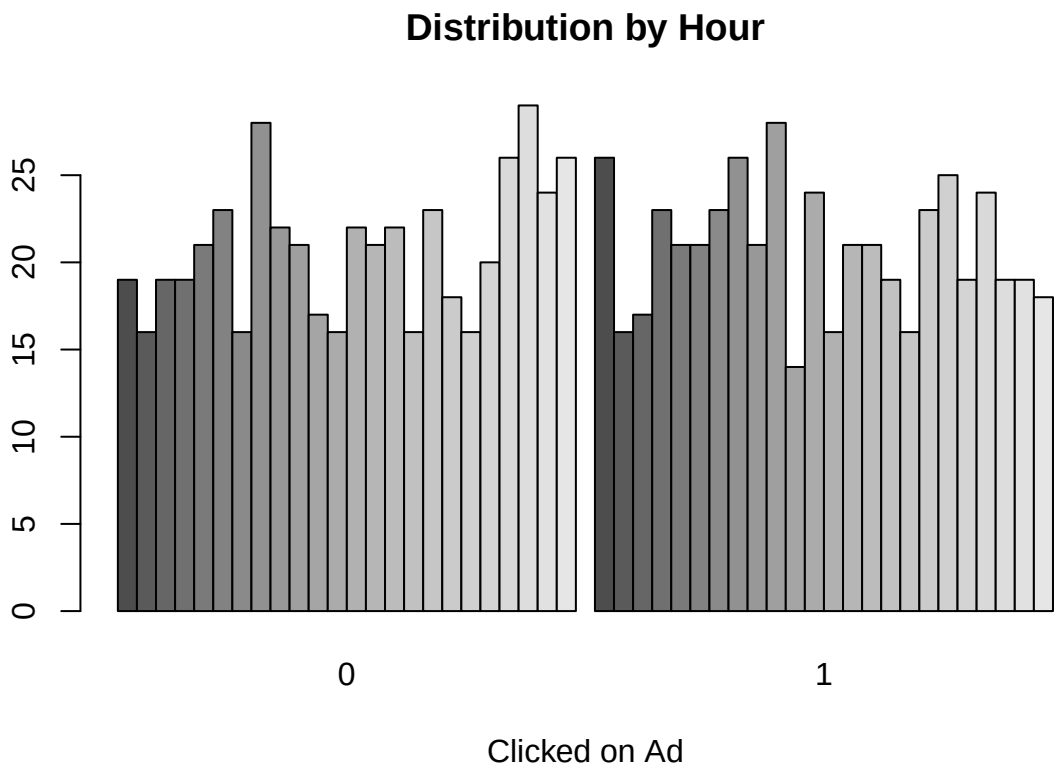

```r
day <- table(df$day, df$Clicked.on.Ad)
barplot(day, main="Distribution by Day",
  xlab="Clicked on Ad",
 beside=TRUE)
```



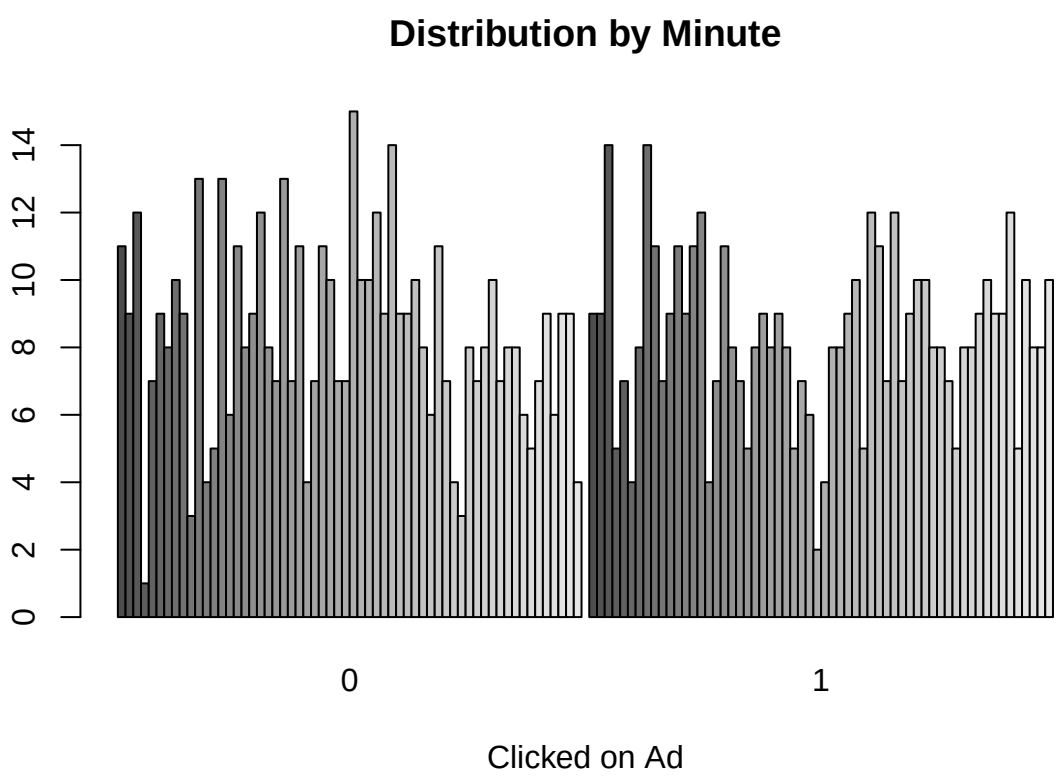

```
month <- table(df$month, df$Clicked.on.Ad)
barplot(month, main="Distribution by Month",
  xlab="Clicked on Ad",
 beside=TRUE)
```



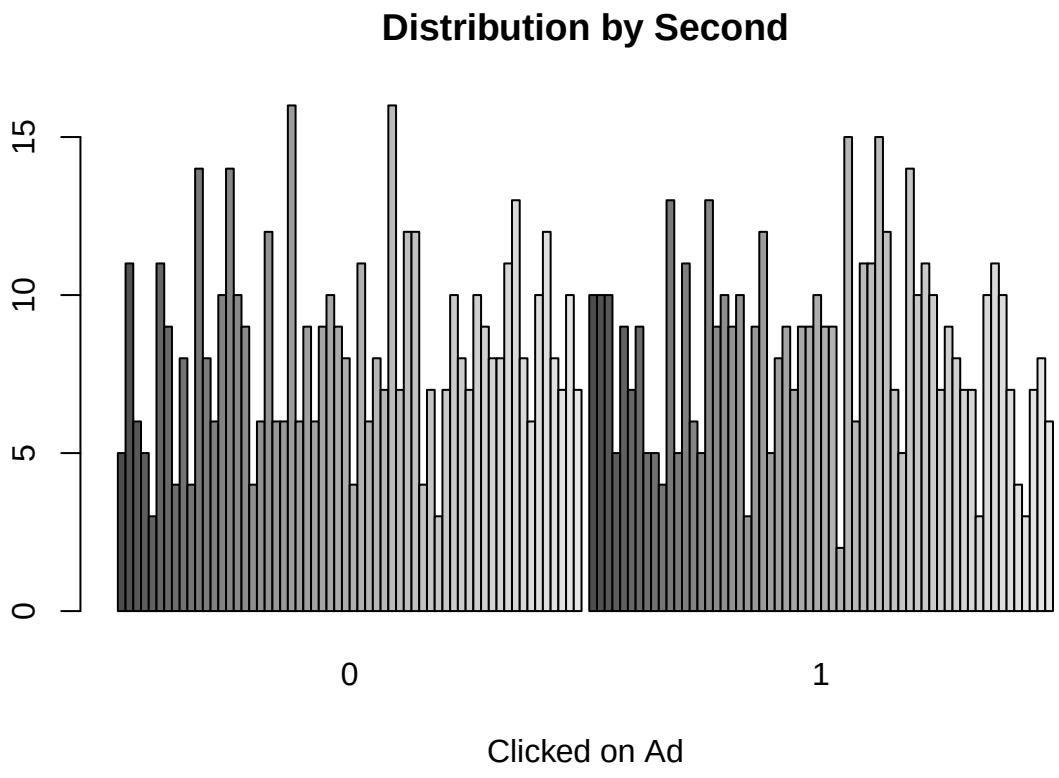

```
hour <- table(df$hour, df$Clicked.on.Ad)
barplot(hour, main="Distribution by Hour",
  xlab="Clicked on Ad",
 beside=TRUE)
```



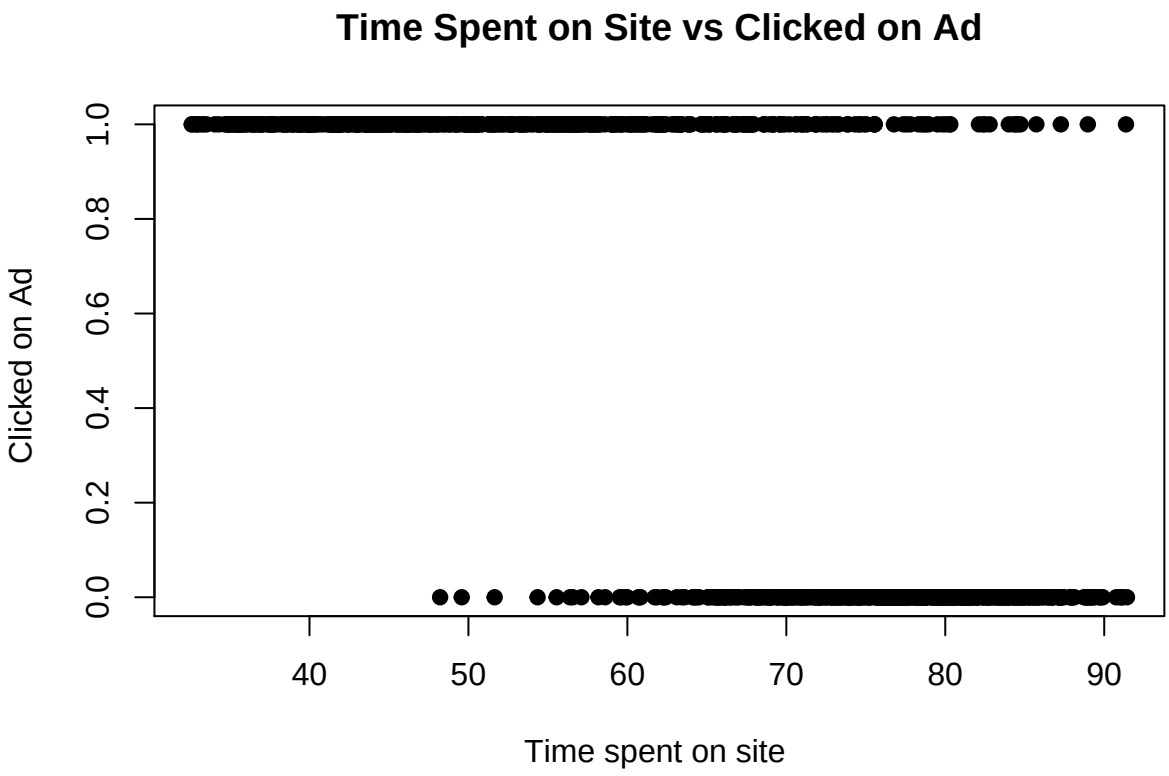

```
minute <- table(df$min, df$Clicked.on.Ad)
barplot(minute, main="Distribution by Minute",
  xlab="Clicked on Ad",
 beside=TRUE)
```



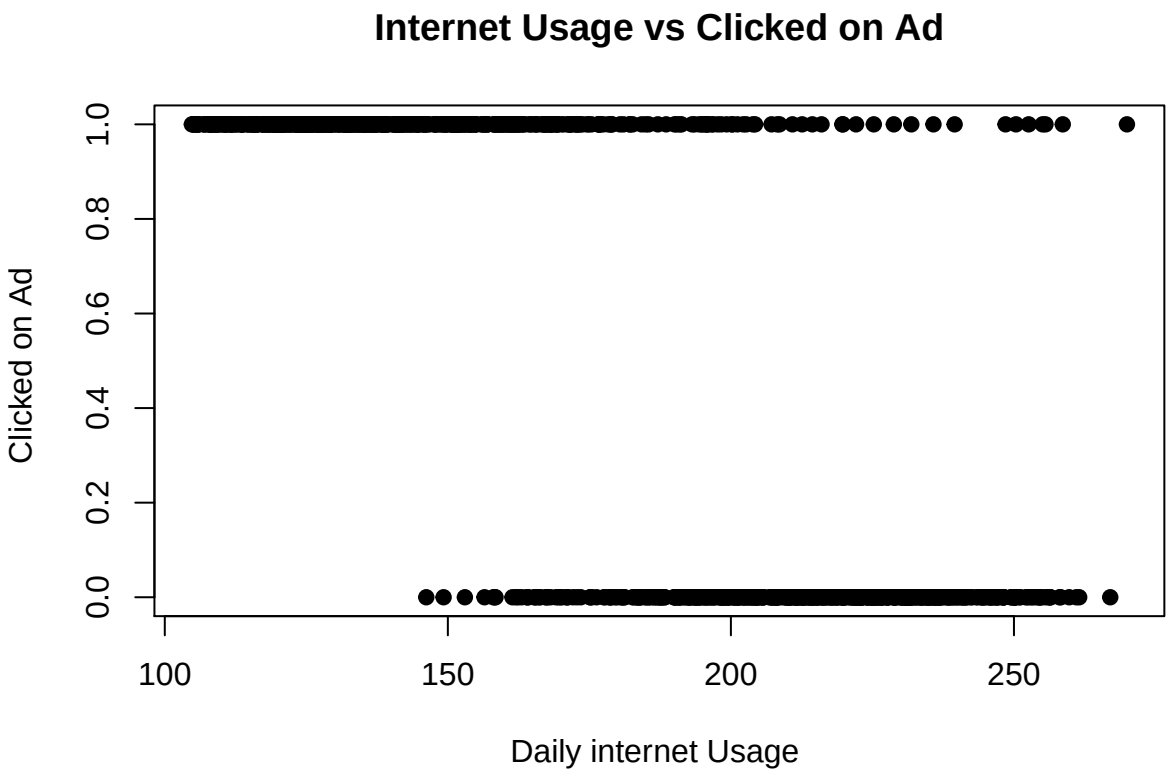

```
sec <- table(df$sec, df$Clicked.on.Ad)
barplot(sec, main="Distribution by Second",
  xlab="Clicked on Ad",
 beside=TRUE)
```



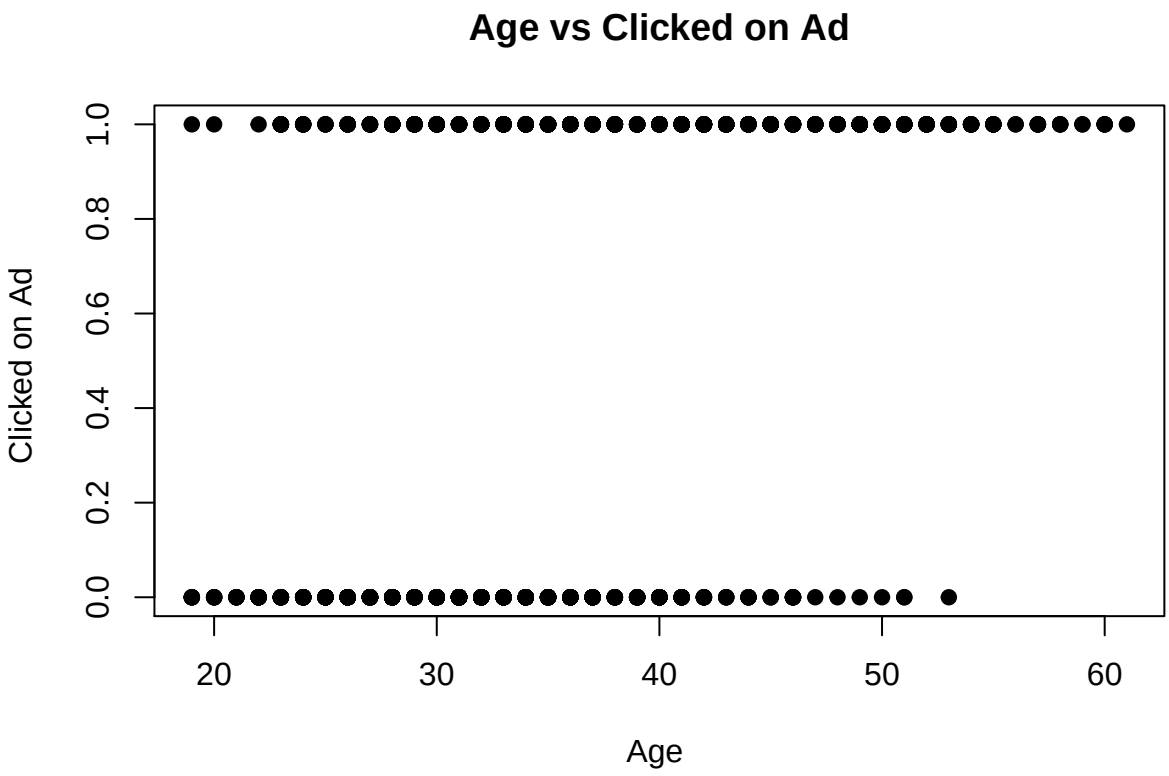

```
plot(df$Daily.Time.Spent.on.Site, df$Clicked.on.Ad, main="Time Spent on Site vs Clicked on Ad",
   xlab="Time spent on site ", ylab="Clicked on Ad ", pch=19)
```



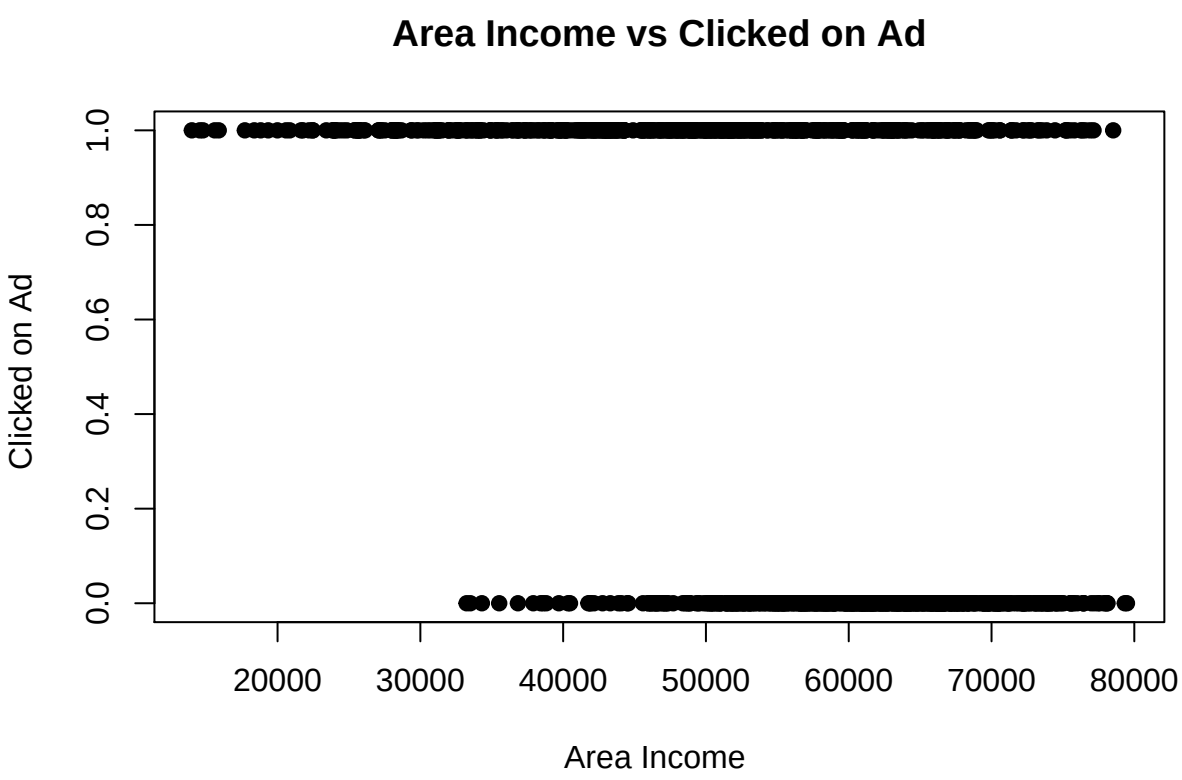

```
plot(df$Daily.Internet.Usage, df$Clicked.on.Ad, main="Internet Usage vs Clicked on Ad",
   xlab="Daily internet Usage ", ylab="Clicked on Ad ", pch=19)
```




```
plot(df$Age, df$Clicked.on.Ad, main="Age vs Clicked on Ad",
   xlab="Age ", ylab="Clicked on Ad ", pch=19)
```




```
plot(df$Area.Income, df$Clicked.on.Ad, main="Area Income vs Clicked on Ad",
   xlab="Area Income ", ylab="Clicked on Ad ", pch=19)
```



1.9 Conclusion

The analysis was partially successful since it provided extra information on which visitors to the site are likely to click on the ads.

The main points were as follows: * The more time that visitors spent on the site, the less likely they are to click on the ad. * Users who spend more time on the internet daily are less likely to click on the ad. * Male visitors are less likely to click on the ad, but the difference does not appear to be significant. * The ads were most popular in February and May, on the 3rd, 23rd and between 9th to 15th days of the month, at midnight, 7am or 9am, in the 3rd or 8th minute of the hour, and in the second half of the minute. They had a noticeable dip in engagement in July, at the end of the month, at 10am and 12pm, and in the middle of the hour. * Younger visitors are less likely to click on the ad. * Visitors from higher income areas are less likely to click on the add.

1.10 Recommendations

The business owner will be advised to target ads towards a more mature audience with lower income. The ads should not be targeted to visitors that spend a lot of time on the site, and those who spend a lot of time on the internet daily, as they would be most likely to block all future ads from the site. The ads should not be targeted to either sex. In terms of time and day, the ad will have better engagement if posted in the middle of the month, late at night or between 7am and 9am, at the beginning of the hour, in the second half of the minute, and before July.

1.11 Binary Classification models

Since the data is not normally distributed, non-parametric (decision trees and K-nearest neighbours) models will be used for this analysis.

As seen in the EDA, city and country columns have high cardinality and low variance, so they can be dropped for modelling.

```
df <- df[,-c(5,7)]
names(df)
```

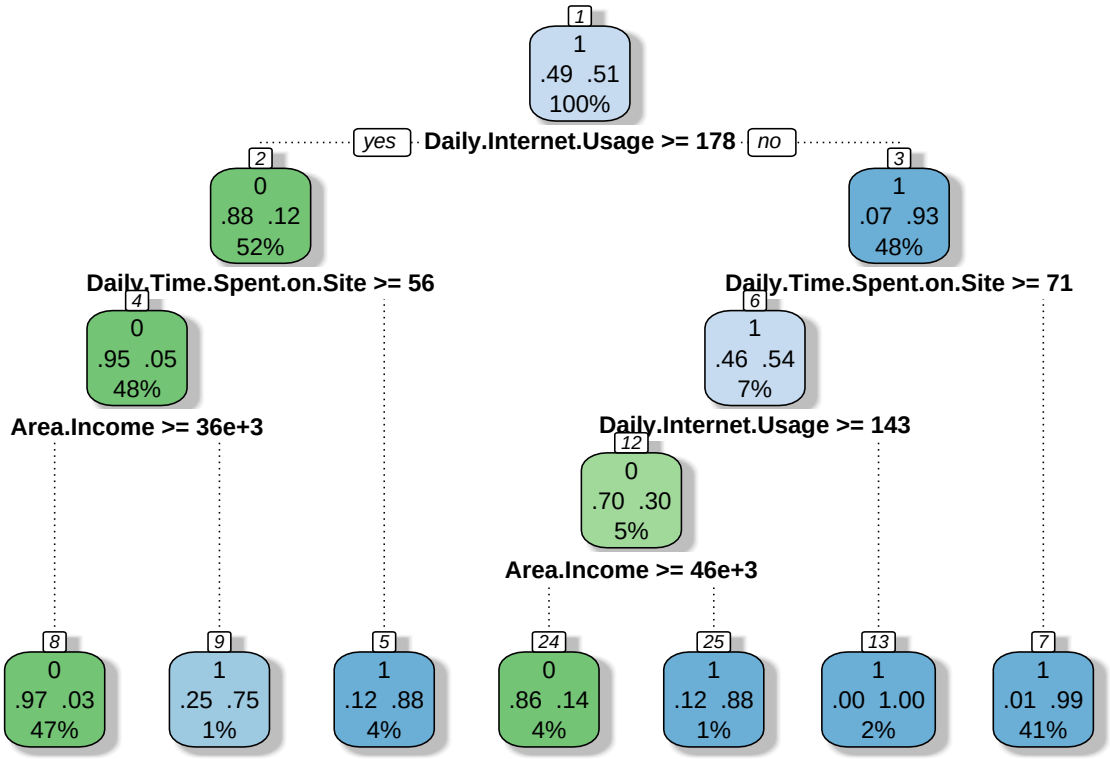
```
##  [1] "Daily.Time.Spent.on.Site" "Age"
##  [3] "Area.Income"              "Daily.Internet.Usage"
##  [5] "Male"                     "Clicked.on.Ad"
##  [7] "day"                      "month"
##  [9] "hour"                     "min"
## [11] "sec"
```

1.11.1 Decision Tree Model

```
#data splicing
set.seed(0)
train <- sample(1:nrow(df),size = ceiling(0.80*nrow(df)),replace = FALSE)

# training set
ad_train <- df[train,]

# test set
ad_test <- df[-train,]
```

```
set.seed(0)

ad_tree <- rpart(Clicked.on.Ad~.,
data= ad_train,
method = "class")
```

```
fancyRpartPlot(ad_tree, caption = NULL)
```



The feature that splits the data best is daily internet usage, followed by daily time spent on the site and area income.

```
ad_pred <-predict(ad_tree, ad_test , type = 'class')

#Calculating accuracy
t <- table(ad_test$Clicked.on.Ad, ad_pred)
paste(t)
```

```
## [1] "101" "5"   "5"   "89"
```

```
accuracy_Test <- sum(diag(t)) / sum(t)
print(paste('Accuracy for test', accuracy_Test))
```

```
## [1] "Accuracy for test 0.95"
```

95% accuracy is satisfactory for the decision tree model.

```
ad_tree$variable.importance
```

```
##     Daily.Internet.Usage Daily.Time.Spent.on.Site                     Age
##              275.2266674              221.7532413              111.3592356
##              Area.Income                     hour                      min
##              109.7334245               11.4342340                8.1523266
##                      day                      sec
##                1.2821242                0.8516135
```

From the decision tree, we see that the most important features for determining whether a potential customer will click on the advertisement for the course are: daily internet usage, daily time spent on the site, age, and area income. The time and date of clicking on the ad are not very important for this prediction.

1.11.2 K-Nearest Neighbour

The data will first be normalized to ensure that all features are on the same scale.

```
# Creating normalization function
normalize <- function(x) {
return ((x - min(x)) / (max(x) - min(x)))
}

# Making a copy of the data
df_norm <- copy(df)

df_norm <- as.data.frame(lapply(df_norm, normalize))
summary(df_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       Clicked.on.Ad      day
##  Min.   :0.0000       Min.   :0.000   Min.   :0.0   Min.   :0.0000
##  1st Qu.:0.2061       1st Qu.:0.000   1st Qu.:0.0   1st Qu.:0.2333
##  Median :0.4743       Median :0.000   Median :0.5   Median :0.4667
##  Mean   :0.4554       Mean   :0.481   Mean   :0.5   Mean   :0.4828
##  3rd Qu.:0.6902       3rd Qu.:1.000   3rd Qu.:1.0   3rd Qu.:0.7333
##  Max.   :1.0000       Max.   :1.000   Max.   :1.0   Max.   :1.0000
##      month             hour             min              sec
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.1667   1st Qu.:0.2609   1st Qu.:0.2373   1st Qu.:0.2542
##  Median :0.5000   Median :0.5217   Median :0.5085   Median :0.5085
##  Mean   :0.4695   Mean   :0.5070   Mean   :0.4924   Mean   :0.5050
##  3rd Qu.:0.6667   3rd Qu.:0.7826   3rd Qu.:0.7288   3rd Qu.:0.7458
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
```

The data will be split into 80/20 train/test sets.

```
#data splicing
set.seed(0)
train <- sample(1:nrow(df_norm),size = ceiling(0.80*nrow(df_norm)),replace = FALSE)

# training set
ad_train_norm <- df_norm[train,]

# test set
ad_test_norm <- df_norm[-train,]
```

```r
library(class)
require(class)

#Training the model
model <- knn(train= ad_train_norm, test= ad_test_norm, cl= ad_train_norm$Clicked.on.Ad, k=17)

# Confustion Matrix
paste("Confusion matrix")
```

```
## [1] "Confusion matrix"
```

```r
table(ad_test_norm$Clicked.on.Ad, model)
```

```
##    model
##       0   1
##   0 106   0
##   1   0  94
```

From the confusion matrix, the model predicts whether a customer clicks on the ad with 100% accuracy.

```r
set.seed(0)
ctrl <- trainControl(method="repeatedcv",repeats = 10)
knnFit <- train(Clicked.on.Ad ~ ., data = ad_train_norm, method = "knn", trControl = ctrl, preProcess =
```

```
## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.
```

```r
knnFit
```

```
## k-Nearest Neighbors
##
## 800 samples
##  10 predictor
##
## Pre-processing: centered (10), scaled (10)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 720, 720, 720, 720, 720, 720, ...
## Resampling results across tuning parameters:
##
##   k   RMSE       Rsquared   MAE
##    5  0.2039809  0.8319847  0.07520417
##    7  0.2017850  0.8354748  0.08026563
##    9  0.2012682  0.8371098  0.08302500
##   11  0.2016230  0.8368118  0.08578788
##   13  0.2011531  0.8377979  0.08778984
##   15  0.2003830  0.8393650  0.08960156
##   17  0.1990696  0.8417693  0.09048366
##   19  0.1991426  0.8419469  0.09194276
##   21  0.1998944  0.8412265  0.09368696
##   23  0.2005451  0.8406537  0.09561639
```

```
##   25  0.2009820  0.8404856  0.09756500
##   27  0.2005855  0.8417107  0.09854001
##   29  0.2001013  0.8428588  0.09936595
##   31  0.1998630  0.8437930  0.10053075
##   33  0.2002716  0.8434799  0.10225145
##   35  0.2007780  0.8431060  0.10387044
##   37  0.2014401  0.8425424  0.10555236
##   39  0.2022398  0.8417450  0.10697420
##   41  0.2029668  0.8410211  0.10840433
##   43  0.2032395  0.8410943  0.10951090
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 17.
```

```
# Creating the confucion matrix
tb <- table(model,ad_test_norm$Clicked.on.Ad)

# Checking the accuracy
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
accuracy(tb)
```

```
## [1] 100
```

```
library(gmodels)
CrossTable(x = ad_test_norm$Clicked.on.Ad, y = model,
           prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  200
##
##
##                              | model
## ad_test_norm$Clicked.on.Ad |         0 |         1 | Row Total |
## ---------------------------|-----------|-----------|-----------|
##                          0 |       106 |         0 |       106 |
##                            |     1.000 |     0.000 |     0.530 |
##                            |     1.000 |     0.000 |           |
##                            |     0.530 |     0.000 |           |
## ---------------------------|-----------|-----------|-----------|
##                          1 |         0 |        94 |        94 |
##                            |     0.000 |     1.000 |     0.470 |
##                            |     0.000 |     1.000 |           |
##                            |     0.000 |     0.470 |           |
```

```
## --------------------------|-----------|-----------|-----------|
##               Column Total |       106 |        94 |       200 |
##                            |     0.530 |     0.470 |           |
## --------------------------|-----------|-----------|-----------|
##
##
```

The KNN model predicts whether a user clicked on the ad with 100% accuracy on test data.