Cleaning And EDA with R

2022-05-27

1. Defining the Question

Research 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.

a.) Specifying the question

Determine which kind of people are likely to click on the ads based on characteristics given in the dataset.

b.) The metric for success

Performing univariate and bivariate analysis on the cleaned dataset and later determining the attributes of individuals who are likely to click on our client's ads.

c.) Understanding the context

The dataset provided contains attributes of people who veiw the client's blog and to whether they clicked on a specific ad about her previous cryptography course. Through analysis, we are able to determine the demographics of these people so as to govern strategy of attracting more customers for her new course.

d.) Experimental design taken

i. Reading the data
ii. Checking the data - data understanding
iii. Data cleaning - checking for outliers and anomalies, missing values and duplicates.
iv. Performing univariate analysis on the dataset.
v. Performing bivariate analysis on the dataset.
vi. Conclusion.
vii. Recommendations.

e.) Data appropriateness to answer the given question.

This is an analysis problem which can be solved by understanding the preferences of the client's viewers. We need to determine the kind of viewer that is most likely to click on a cryptography course ad and most likely sign up for it as well.

2. Loading the dataset

```
# load Tibble library
library(tibble)

# load the dataset as dataframe

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

# convert dataframe to tibble and set options to show all columns

df <- as_tibble(df)

#check data structure of our dataset

class(df)
```

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

3. Checking the data

```
# preview the first 6 rows of the dataset
head(df)
```

```
## # A tibble: 6 x 10
##   Daily.Time.Spent~   Age Area.Income Daily.Internet.~ Ad.Topic.Line City   Male
##               <dbl> <int>       <dbl>            <dbl> <chr>         <chr> <int>
## 1              69.0    35      61834.             256. Cloned 5thge~ Wrig~     0
## 2              80.2    31      68442.             194. Monitored na~ West~     1
## 3              69.5    26      59786.             236. Organic bott~ Davi~     0
## 4              74.2    29      54806.             246. Triple-buffe~ West~     1
## 5              68.4    35      73890.             226. Robust logis~ Sout~     0
## 6              60.0    23      59762.             227. Sharable cli~ Jami~     1
## # ... with 3 more variables: Country <chr>, Timestamp <chr>,
## #   Clicked.on.Ad <int>
```

```
# preview the last 6 rows of the dataset
tail(df)
```

```
## # A tibble: 6 x 10
##   Daily.Time.Spent~   Age Area.Income Daily.Internet.~ Ad.Topic.Line City   Male
##               <dbl> <int>       <dbl>            <dbl> <chr>         <chr> <int>
## 1              43.7    28      63127.             173. Front-line b~ Nich~     0
## 2              73.0    30      71385.             209. Fundamental ~ Duff~     1
## 3              51.3    45      67782.             134. Grass-roots ~ New ~     1
## 4              51.6    51      42416.             120. Expanded int~ Sout~     1
## 5              55.6    19      41921.             188. Proactive ba~ West~     0
## 6              45.0    26      29876.             178. Virtual 5thg~ Ronn~     0
## # ... with 3 more variables: Country <chr>, Timestamp <chr>,
## #   Clicked.on.Ad <int>
```

```
# get number of records
nrow(df); ncol(df)
```

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

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

The dataset contains 1000 customer attributes and 10 variables.

```
# get datatypes of each column using class
sapply(df, class)
```

```
## Daily.Time.Spent.on.Site                      Age              Area.Income
##                "numeric"                "integer"                "numeric"
##     Daily.Internet.Usage            Ad.Topic.Line                     City
##                "numeric"              "character"              "character"
##                     Male                  Country                Timestamp
##                "integer"              "character"              "character"
##            Clicked.on.Ad
##                "integer"
```

```
# inspect variable classes
str(df)
```

```
## tibble [1,000 x 10] (S3: tbl_df/tbl/data.frame)
##  $ Daily.Time.Spent.on.Site: num [1:1000] 69 80.2 69.5 74.2 68.4 ...
##  $ Age                     : int [1:1000] 35 31 26 29 35 23 33 48 30 20 ...
##  $ Area.Income             : num [1:1000] 61834 68442 59786 54806 73890 ...
##  $ Daily.Internet.Usage    : num [1:1000] 256 194 236 246 226 ...
##  $ Ad.Topic.Line           : chr [1:1000] "Cloned 5thgeneration orchestration" "Monitored national s
##  $ City                    : chr [1:1000] "Wrightburgh" "West Jodi" "Davidton" "West Terrifurt" ...
##  $ Male                    : int [1:1000] 0 1 0 1 0 1 0 1 1 1 ...
##  $ Country                 : chr [1:1000] "Tunisia" "Nauru" "San Marino" "Italy" ...
##  $ Timestamp               : chr [1:1000] "2016-03-27 00:53:11" "2016-04-04 01:39:02" "2016-03-13 20
##  $ Clicked.on.Ad           : int [1:1000] 0 0 0 0 0 0 0 1 0 0 ...
```

```
# create function to convert categorical columns to factor datatype

tofactor <- function(column){
  as.factor(column)
}

# columns to be converted to factors

df$Male <- tofactor(df$Male)
df$Clicked.on.Ad <- tofactor(df$Clicked.on.Ad)
df$City <- tofactor(df$City)
df$Country <- tofactor(df$Country)



# convert timestamp to datetime
```

```
df$Timestamp <- as.Date(df$Timestamp, format = '%Y-%m-%d %H:%M:%S')

# check datatypes again
sapply(df, class)
```

```
## Daily.Time.Spent.on.Site                      Age              Area.Income
##                "numeric"                "integer"                "numeric"
##     Daily.Internet.Usage            Ad.Topic.Line                     City
##                "numeric"              "character"                 "factor"
##                     Male                  Country                Timestamp
##                 "factor"                 "factor"                   "Date"
##            Clicked.on.Ad
##                 "factor"
```

4. Data Cleaning

a.) Outliers

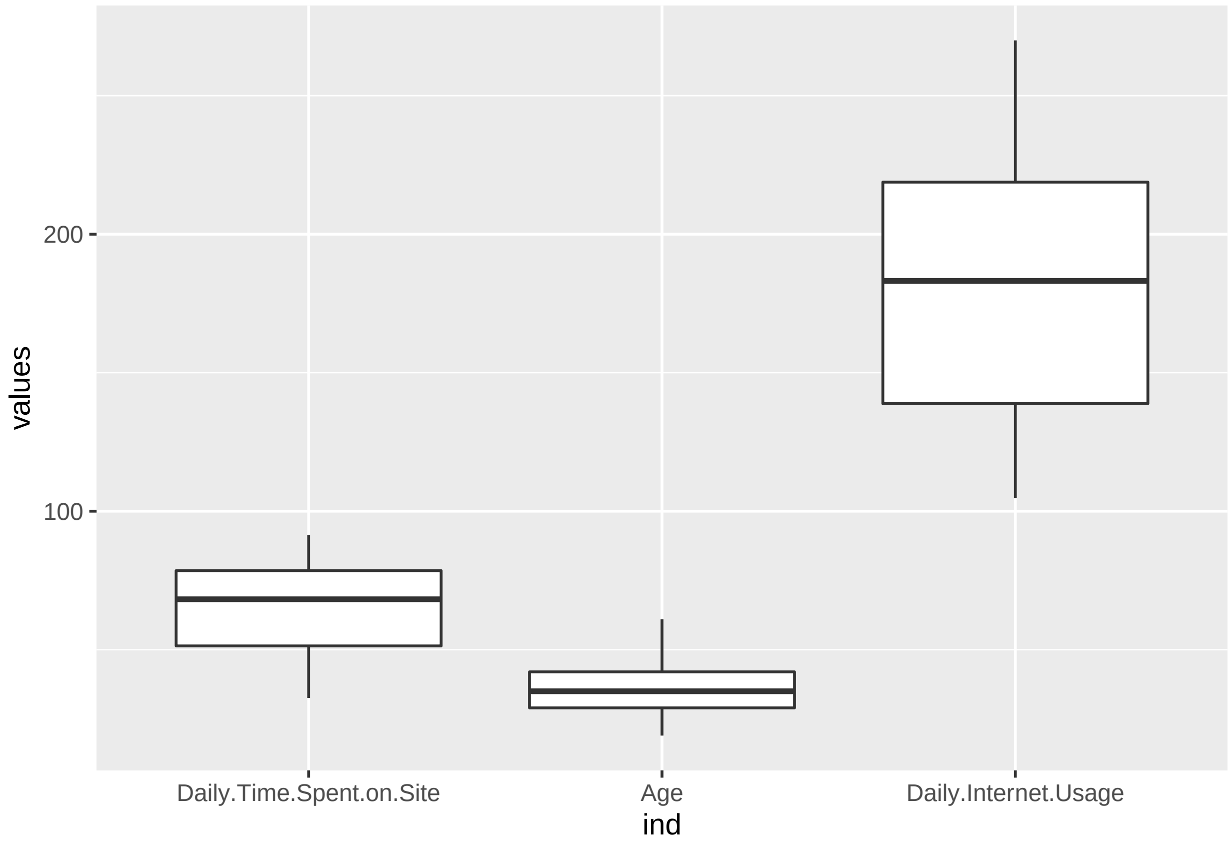
```
# check for outliers

# select only numerical variables
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
```

```
data_num <- select_if(df, is.numeric)
data_num
```

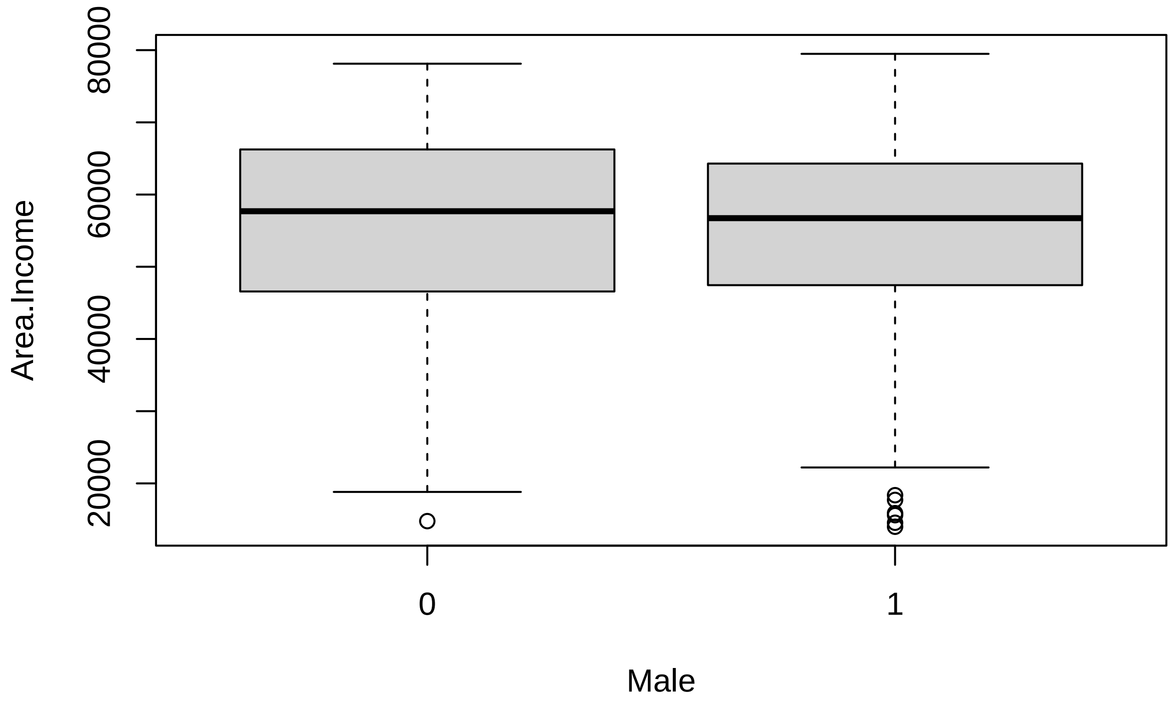
```
## # A tibble: 1,000 x 4
##    Daily.Time.Spent.on.Site   Age Area.Income Daily.Internet.Usage
##                       <dbl> <int>       <dbl>                <dbl>
##  1                     69.0    35      61834.                 256.
##  2                     80.2    31      68442.                 194.
##  3                     69.5    26      59786.                 236.
##  4                     74.2    29      54806.                 246.
##  5                     68.4    35      73890.                 226.
##  6                     60.0    23      59762.                 227.
##  7                     88.9    33      53853.                 208.
##  8                     66      48      24593.                 132.
##  9                     74.5    30      68862                  222.
## 10                     69.9    20      55642.                 184.
## # ... with 990 more rows
```

```r
# boxplot for the numerical values except for Area.Income column which is on another scale
library(ggplot2)
ggplot(stack(select(data_num, -Area.Income)), aes(x = ind, y = values)) + geom_boxplot()
```



There are no outliers for Daily.Time.Spent.on.Site, Age and Daily.Internet.Usage columns.

```r
# boxplot for Area.Income
boxplot(Area.Income~Male, data=df)
```



There are outliers in the Area Income column especially for the Male gender. We shall check if the outliers are valid income figures.

b.) Missing values

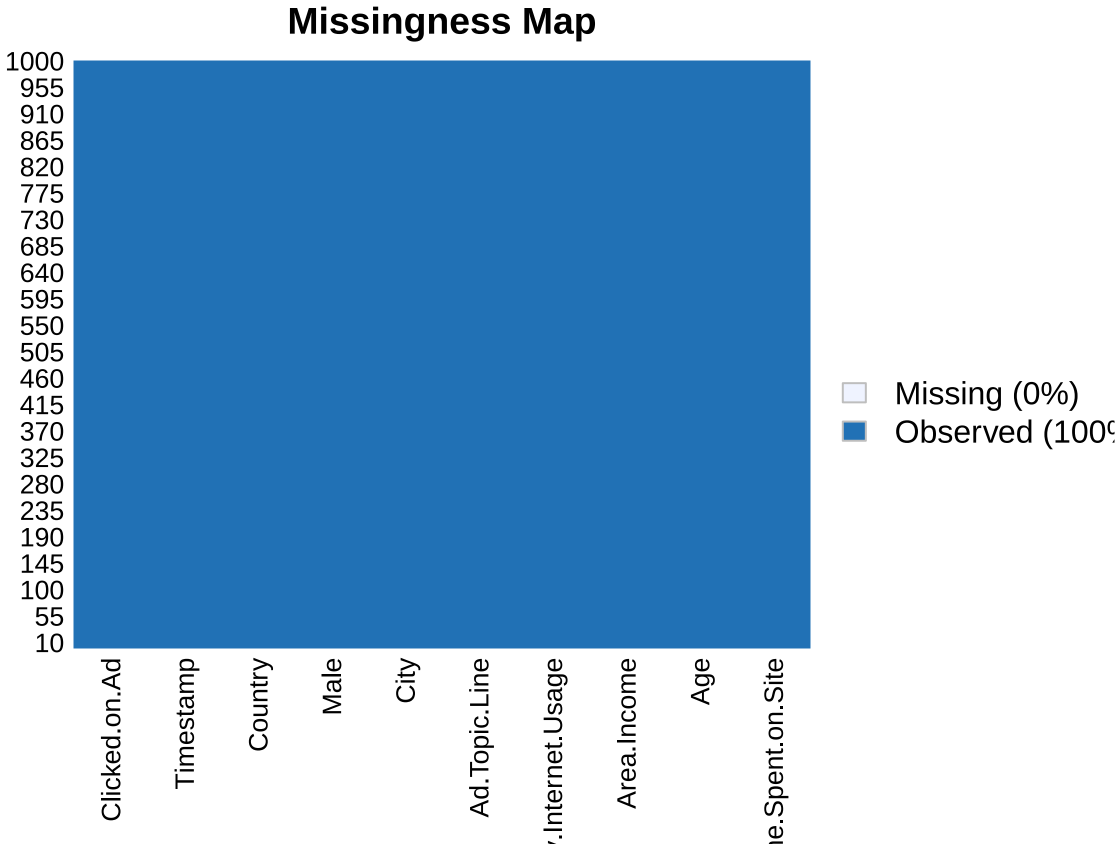
```
# import Amelia and Rcpp libraries

library(Amelia,Rcpp)
```

```
## Loading required package: Rcpp
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.8.0, built: 2021-05-26)
## ## Copyright (C) 2005-2022 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```
#checking missing data visualization
missmap(df)
```

```
## Warning: Unknown or uninitialised column: ‘arguments‘.
## Unknown or uninitialised column: ‘arguments‘.
## Warning: Unknown or uninitialised column: ‘imputations‘.
```



There are no missing values in our dataframe.

```
# confirming we have no missing values

colSums(is.na(df))
```

```
## Daily.Time.Spent.on.Site                      Age              Area.Income
##                        0                        0                        0
##     Daily.Internet.Usage            Ad.Topic.Line                     City
##                        0                        0                        0
##                     Male                  Country                Timestamp
##                        0                        0                        0
##            Clicked.on.Ad
##                        0
```

c.) Duplicate values

```
# check for duplicates
anyDuplicated(df)
```

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

There are no duplicated values in our dataset, otherwise we would have to drop them.

5. Univariate Analysis

```
# check univariate summary of our dataset.
summary(df)
```

```
##  Daily.Time.Spent.on.Site      Age         Area.Income    Daily.Internet.Usage
##  Min.   :32.60            Min.   :19.00   Min.   :13996   Min.   :104.8
##  1st Qu.:51.36            1st Qu.:29.00   1st Qu.:47032   1st Qu.:138.8
##  Median :68.22            Median :35.00   Median :57012   Median :183.1
##  Mean   :65.00            Mean   :36.01   Mean   :55000   Mean   :180.0
##  3rd Qu.:78.55            3rd Qu.:42.00   3rd Qu.:65471   3rd Qu.:218.8
##  Max.   :91.43            Max.   :61.00   Max.   :79485   Max.   :270.0
##
##  Ad.Topic.Line                      City     Male             Country
##  Length:1000        Lisamouth      :  3   0:519   Czech Republic:  9
##  Class :character   Williamsport   :  3   1:481   France        :  9
##  Mode  :character   Benjaminchester:  2           Afghanistan   :  8
##                     East John      :  2           Australia     :  8
##                     East Timothy   :  2           Cyprus        :  8
##                     Johnstad       :  2           Greece        :  8
##                     (Other)        :986           (Other)       :950
##    Timestamp          Clicked.on.Ad
##  Min.   :2016-01-01   0:500
##  1st Qu.:2016-02-17   1:500
##  Median :2016-04-07
##  Mean   :2016-04-09
##  3rd Qu.:2016-05-31
##  Max.   :2016-07-24
##
```

From the table above we are able to get the mean, median, quantiles and range of all numerical variables.

The average daily time spent on the site is 65. The most time a client spent online is 91.43 and the least time is 32.60.

The average area income is 55000, the highest being 79485 and the lowest being 13996.

The average age of clients in our dataset is 36. Max age is 61 and min age is 19.

The average daily internet usage is 180, highest being 270 and the lowest being 104.8.

The data was colected from 1st January 2016 to 24th July, 2016.

Our dataset is balanced as it contains an equal share of people who clicked on the ad and those who did not.

```
# let's define a function that will print results for numeric columns

library(dplyr)

printing <- function(func){

   print(paste("Daily.Time.Spent.on.Site: ", func(df$Daily.Time.Spent.on.Site)))
   print(paste("Daily.Internet.Usage: ", func(df$Daily.Internet.Usage)))
   print(paste("Age: ", func(df$Age)))
   print(paste("Area.Income: ", func(df$Area.Income)))
}
```

```
# use above function to print range
paste("The range of each column is as shown below:-")
```

```
## [1] "The range of each column is as shown below:-"
```

```
printing(range)
```

```
## [1] "Daily.Time.Spent.on.Site:  32.6"  "Daily.Time.Spent.on.Site:  91.43"
## [1] "Daily.Internet.Usage:  104.78" "Daily.Internet.Usage:  269.96"
## [1] "Age:  19" "Age:  61"
## [1] "Area.Income:  13996.5" "Area.Income:  79484.8"
```

```
# other columns whose range is obtainable
print(paste("Ad.Topic.Line: ", range(df$Ad.Topic.Line)))
```

```
## [1] "Ad.Topic.Line:  Adaptive 24hour Graphic Interface"
## [2] "Ad.Topic.Line:  Visionary reciprocal circuit"
```

```
print(paste("Timestamp: ", range(df$Timestamp)))
```

```
## [1] "Timestamp:  2016-01-01" "Timestamp:  2016-07-24"
```

```
paste("The interquartile range of each column is shown beow:-")
```

```
## [1] "The interquartile range of each column is shown beow:-"
```

```
# use IQR function
printing(IQR)
```

```
## [1] "Daily.Time.Spent.on.Site:  27.1875"
## [1] "Daily.Internet.Usage:  79.9625"
## [1] "Age:  13"
## [1] "Area.Income:  18438.8325"
```

The interquatile range explains the range where the bulk of the values lie.

```
# mode for all columns which is not included in the summary
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

paste("The mode of the columns is as shown below:-")
```

```
## [1] "The mode of the columns is as shown below:-"
```

```r
printing(getmode)
```

```
## [1] "Daily.Time.Spent.on.Site:  62.26"
## [1] "Daily.Internet.Usage:  167.22"
## [1] "Age:  31"
## [1] "Area.Income:  61833.9"
```

```r
# add the rest of the columns whosemode can be obtained
print(paste("Male: ", getmode(df$Male)))
```

```
## [1] "Male:  0"
```

```r
print(paste("Clicked.on.Ad: ", getmode(df$Clicked.on.Ad)))
```

```
## [1] "Clicked.on.Ad:  0"
```

```r
print(paste("Ad.Topic.Line: ", getmode(df$Ad.Topic.Line)))
```

```
## [1] "Ad.Topic.Line:  Cloned 5thgeneration orchestration"
```

```r
print(paste("Country: ", getmode(df$Country)))
```

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

```r
print(paste("City: ", getmode(df$City)))
```

```
## [1] "City:  Lisamouth"
```

```r
print(paste("Timestamp: ", getmode(df$Timestamp)))
```

```
## [1] "Timestamp:  2016-04-04"
```

The mode of the columns as shown above represents the column values that are most repeated for each column.

```r
print("The variance of numerical columns is as shown below:-")
```

```
## [1] "The variance of numerical columns is as shown below:-"
```

```r
printing(var)
```

```
## [1] "Daily.Time.Spent.on.Site:  251.337094854855"
## [1] "Daily.Internet.Usage:  1927.41539618619"
## [1] "Age:  77.1861051051051"
## [1] "Area.Income:  179952405.951775"
```

The variance explains the measure of spread of the data points within the data set.

```r
print("The standard deviation of the numerical columns is as shown below:-")
```

```
## [1] "The standard deviation of the numerical columns is as shown below:-"
```

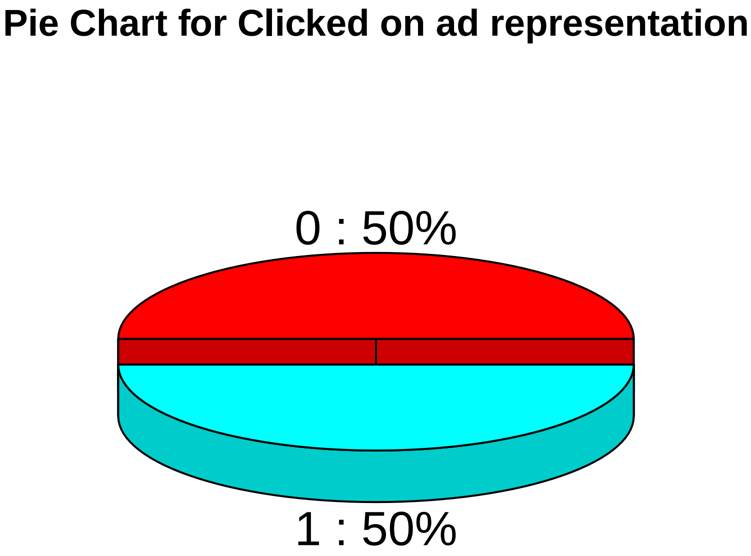
```r
printing(sd)
```

```
## [1] "Daily.Time.Spent.on.Site:  15.8536145675002"
## [1] "Daily.Internet.Usage:  43.9023393019801"
## [1] "Age:  8.78556231012592"
## [1] "Area.Income:  13414.6340222824"
```

The standard deviation measures the distribution of data points around the mean.

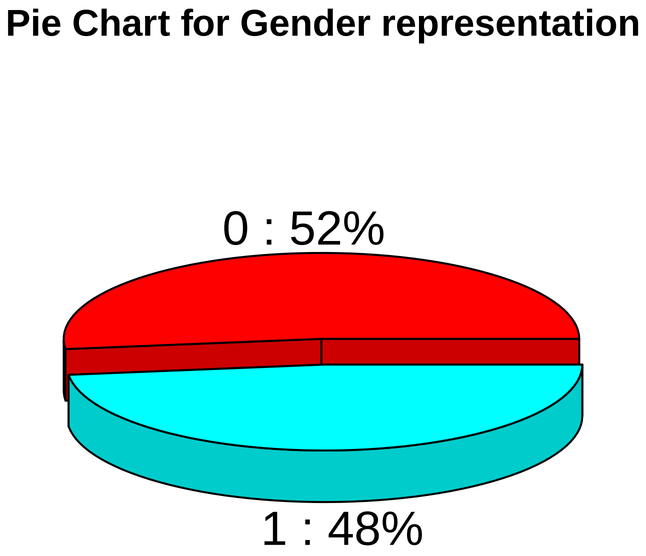
```r
# Pie chart function

# import plotrix for 3D plot
library(plotrix)

piechart <- function(column, title){

  # define labels and values
  slices <- c(tabulate(column))
  lbls <- c(unique(column))

  # convert values to percentage
  pct <- round(slices/sum(slices)*100)

  # add percentages to labels
  lbls <- paste(lbls,":", pct)

  # ad % to labels
  lbls <- paste(lbls,"%",sep="")

  # plot pie chart in 3D
  pie3D(slices,labels=lbls, col=rainbow(length(lbls)),
        explode=0.1, radius=1, start=0, main=title)
}
```

```r
# Clicked.on.ad representation
piechart(df$Clicked.on.Ad, "Pie Chart for Clicked on ad representation")
```


Pie Chart for Clicked on ad representation

Half of the population of the dataset clicked on the ad.

```r
# Gender representation
piechart(df$Male, "Pie Chart for Gender representation")
```



52% are female while 48% of the population is male.

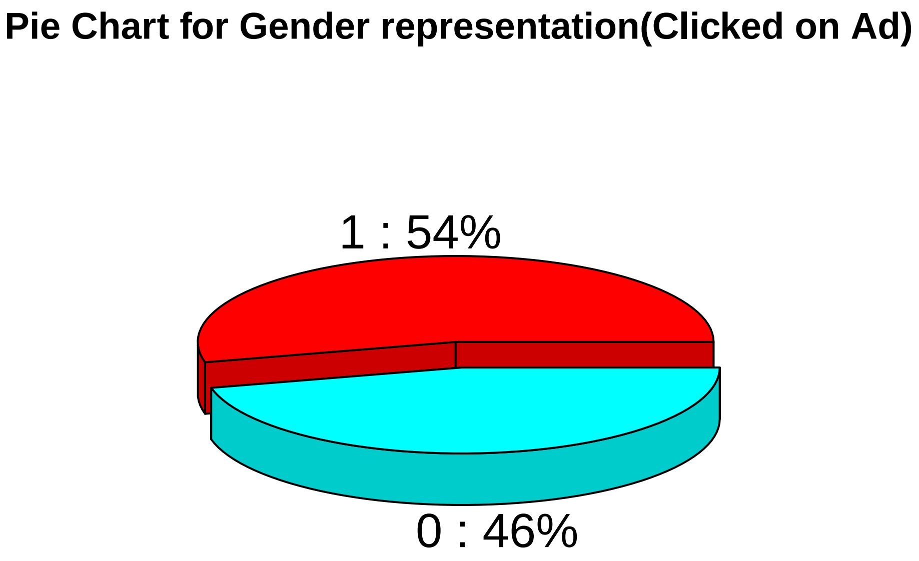
```r
# filter data to only have those who clicked on ad
data <- df %>% filter(Clicked.on.Ad == 1)
data
```

```
## # A tibble: 500 x 10
##    Daily.Time.Spen~   Age Area.Income Daily.Internet.~ Ad.Topic.Line City  Male
##               <dbl> <int>       <dbl>            <dbl> <chr>         <fct> <fct>
##  1             66      48      24593.             132. Reactive loc~ Port~ 1
##  2             47.6    49      45633.             122. Centralized ~ West~ 0
##  3             69.6    48      51637.             113. Centralized ~ West~ 1
##  4             43.0    33      30976              144. Grass-roots ~ West~ 0
##  5             63.4    23      52182.             141. Persistent d~ New ~ 1
##  6             55.4    37      23937.             129. Customizable~ West~ 0
##  7             54.7    36      31088.             118. Grass-roots ~ Jess~ 1
##  8             74.6    40      23822.             136. Advanced 24/~ Mill~ 1
##  9             41.5    52      32636.             165. Mandatory di~ Sout~ 0
## 10             41.4    41      68962.             167. Exclusive ne~ Harp~ 0
## # ... with 490 more rows, and 3 more variables: Country <fct>,
## #   Timestamp <date>, Clicked.on.Ad <fct>
```

```r
# filter data to only those who didn't click on the ad
data_ <- df %>% filter(Clicked.on.Ad == 0)
data_
```

```
## # A tibble: 500 x 10
##    Daily.Time.Spen~   Age Area.Income Daily.Internet.~ Ad.Topic.Line City  Male
##               <dbl> <int>       <dbl>            <dbl> <chr>         <fct> <fct>
##  1             69.0    35      61834.             256. Cloned 5thge~ Wrig~ 0
##  2             80.2    31      68442.             194. Monitored na~ West~ 1
##  3             69.5    26      59786.             236. Organic bott~ Davi~ 0
##  4             74.2    29      54806.             246. Triple-buffe~ West~ 1
##  5             68.4    35      73890.             226. Robust logis~ Sout~ 0
##  6             60.0    23      59762.             227. Sharable cli~ Jami~ 1
##  7             88.9    33      53853.             208. Enhanced ded~ Bran~ 0
##  8             74.5    30      68862              222. Configurable~ West~ 1
##  9             69.9    20      55642.             184. Mandatory ho~ Rami~ 1
## 10             83.1    37      62491.             231. Team-oriente~ East~ 1
## # ... with 490 more rows, and 3 more variables: Country <fct>,
## #   Timestamp <date>, Clicked.on.Ad <fct>
```

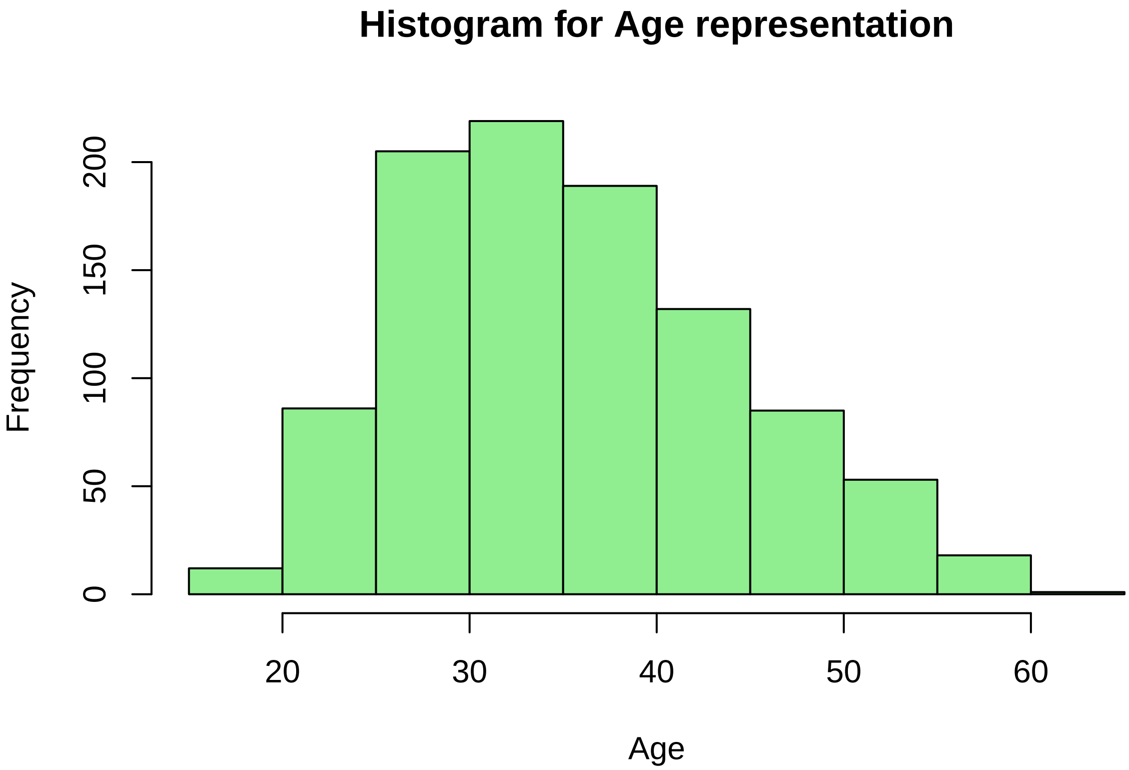
```r
# Gender representation of those who clicked on ad
piechart(data$Male, "Pie Chart for Gender representation(Clicked on Ad)")
```



More female clicked on the ad than male.

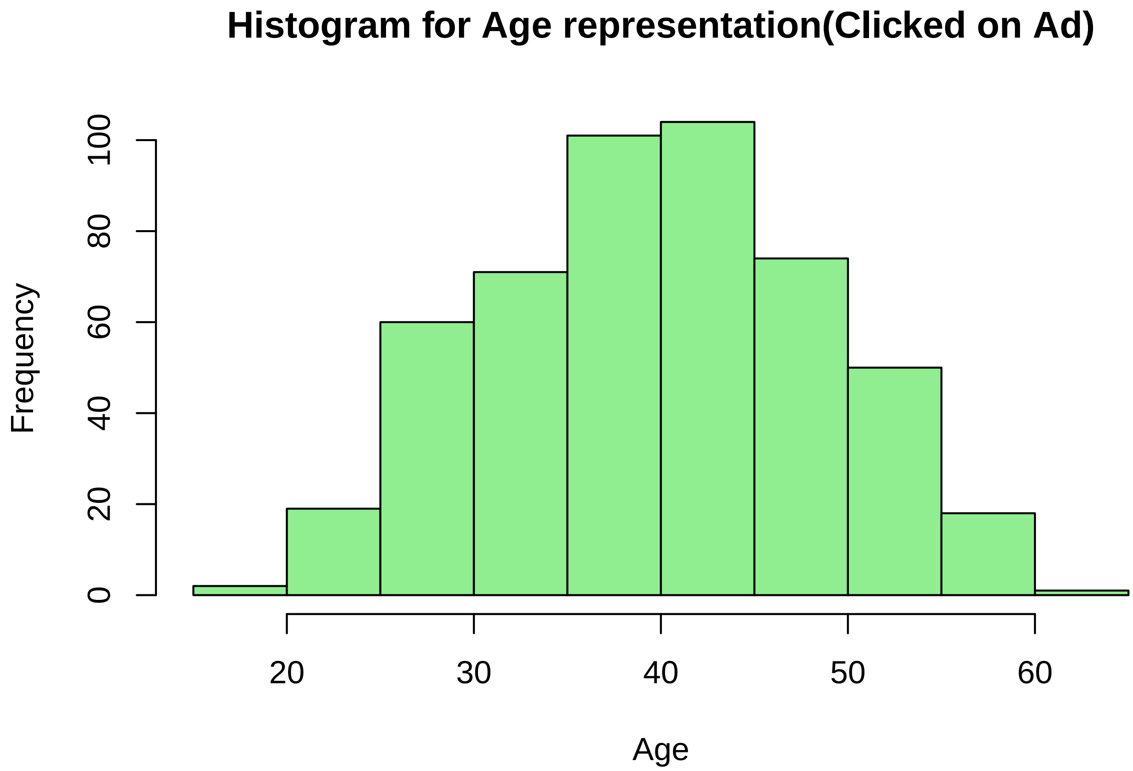
```r
# histograms function
histogram <- function(column,title, xlab){

  hist(column, main= title, xlab=xlab, ylab = "Frequency", col = "lightgreen")

}
```

```r
# age representation
histogram(df$Age, "Histogram for Age representation", "Age")
```



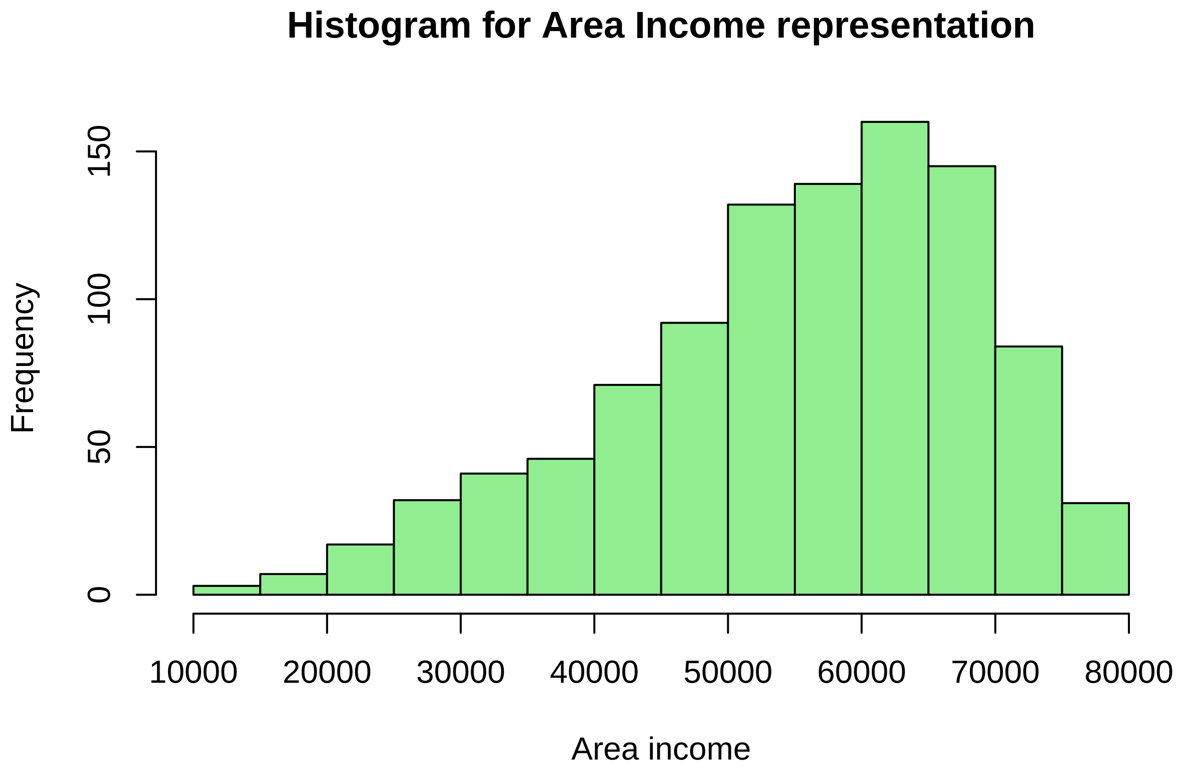
Age representation in the dataset is skewed to the right. Most people lie between the age of 25 and 40.

```r
# age representation of those who clicked on ad
histogram(data$Age, "Histogram for Age representation(Clicked on Ad)", "Age")
```



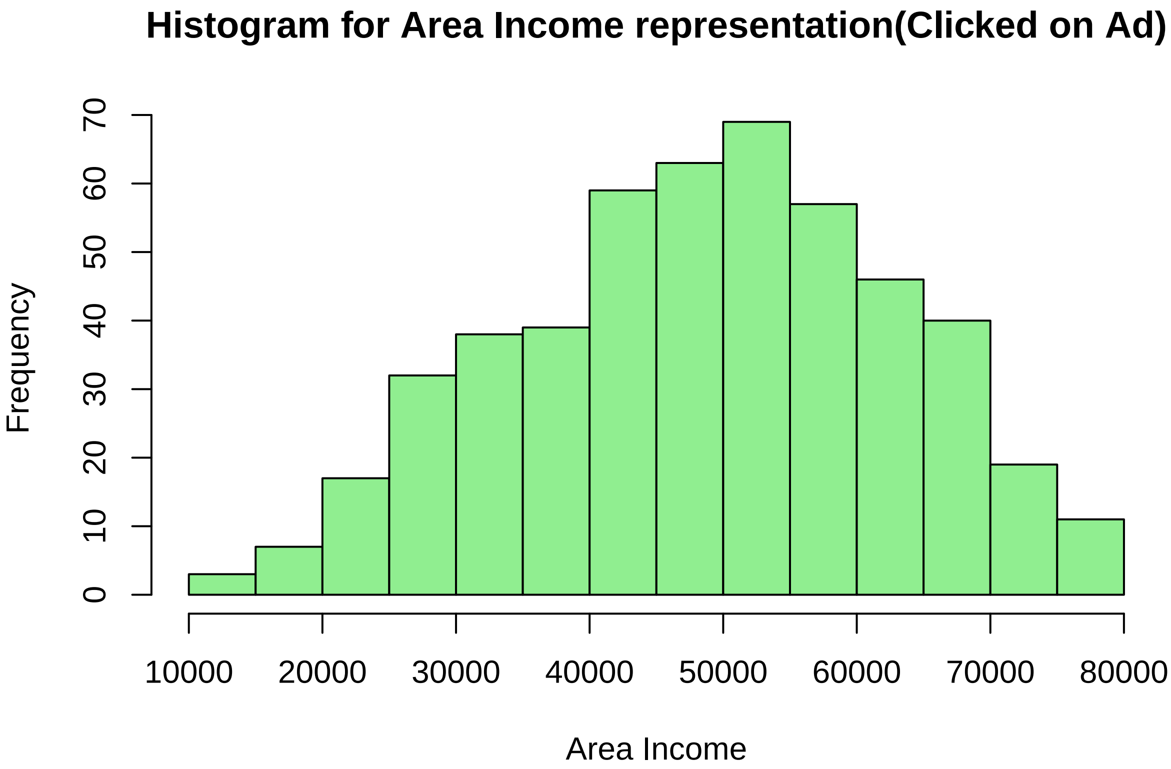
The data seems normally distributed. Those who clicked on the ad are aged between 35 and 45.

```
# area income representation
histogram(df$Area.Income, "Histogram for Area Income representation", "Area income")
```



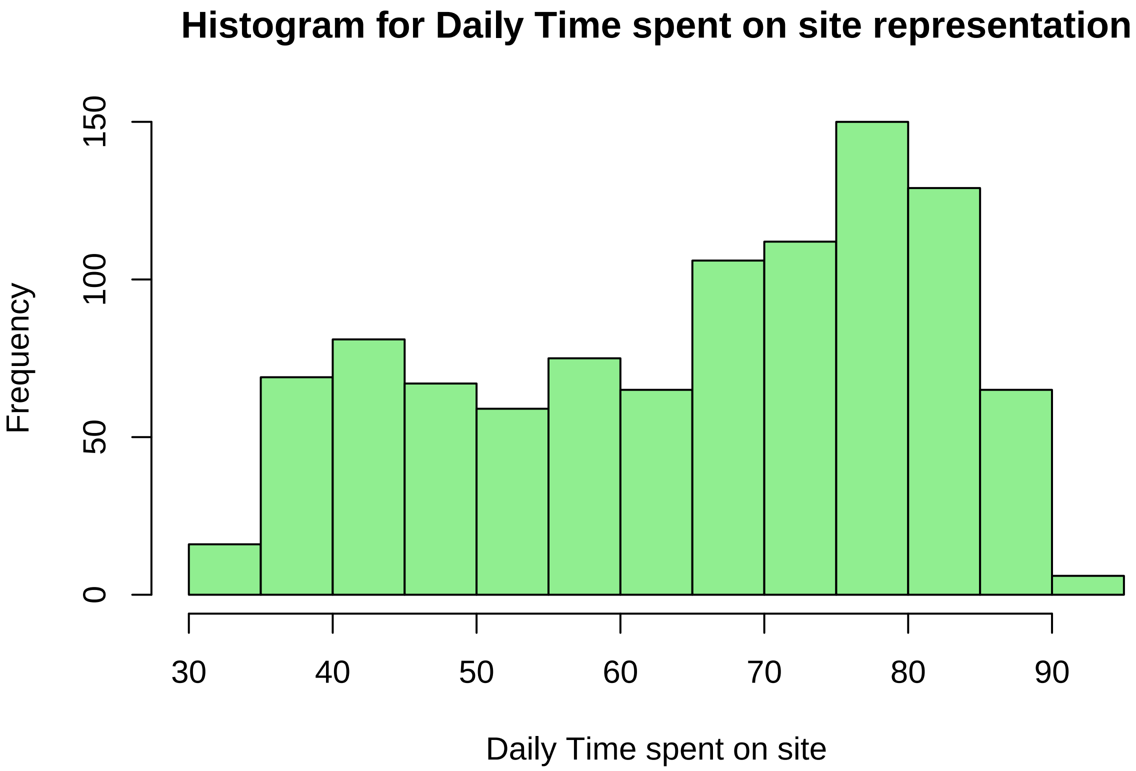
Area Income representation is skewed to the left, most people earning between 50k and 70k.

```r
# Area representation of those who clicked on ad
histogram(data$Area.Income, "Histogram for Area Income representation(Clicked on Ad)", "Area Income")
```



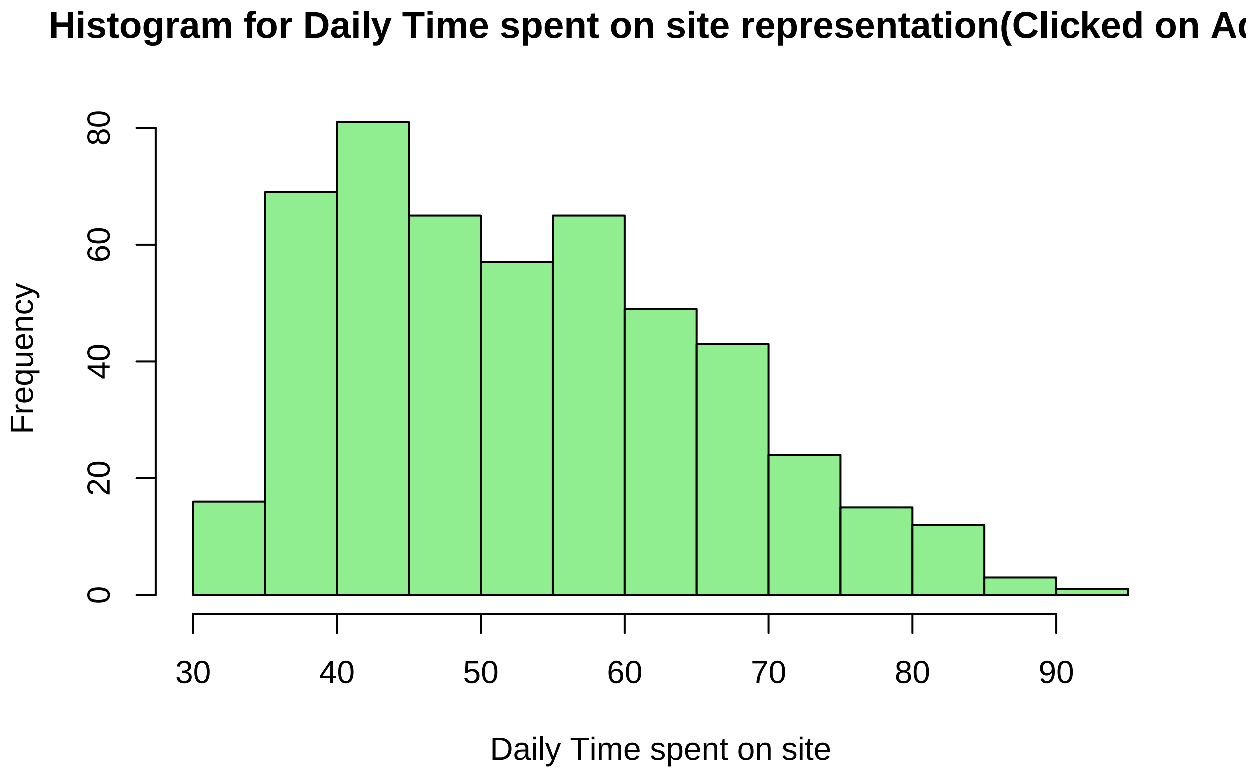
Those who clicked on ad earn between 40k and 60k.

```
# Daily time spent on site representation
histogram(df$Daily.Time.Spent.on.Site, "Histogram for Daily Time spent on site representation", "Daily
```



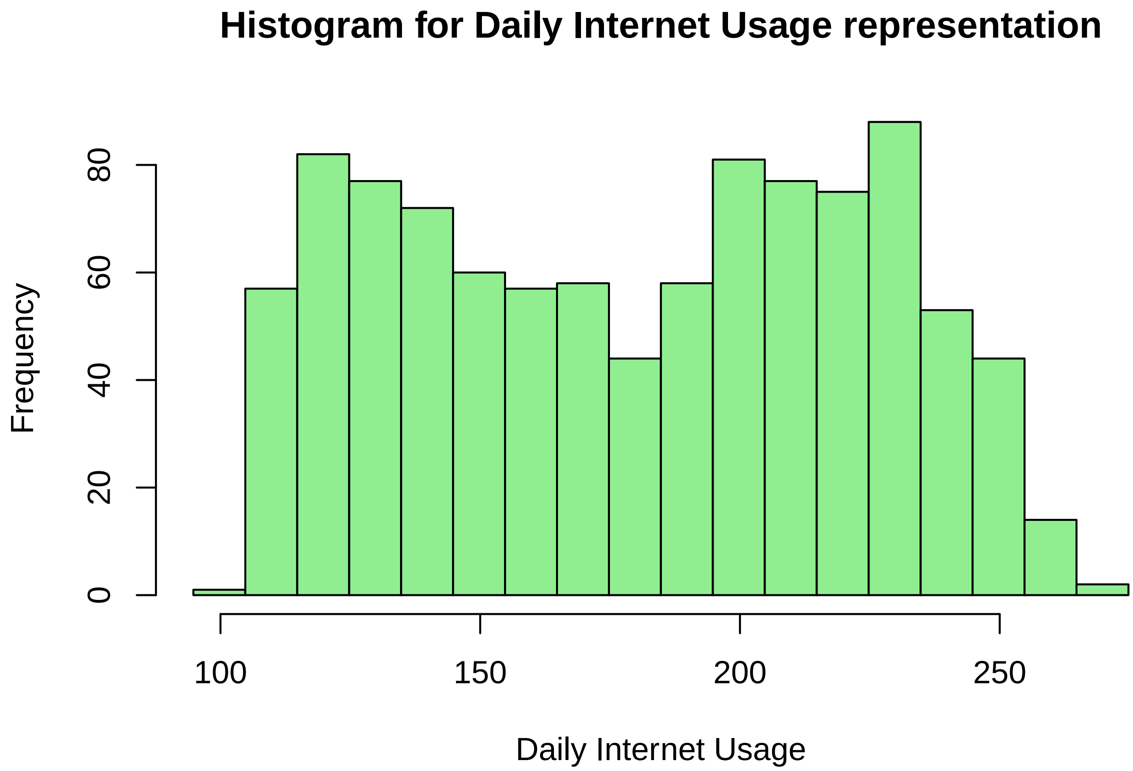
Most people spend between 65 to 85 minutes on site.

```
# Daily time spent on site representation of those who clicked on ad
histogram(data$Daily.Time.Spent.on.Site, "Histogram for Daily Time spent on site representation(Clicked
```



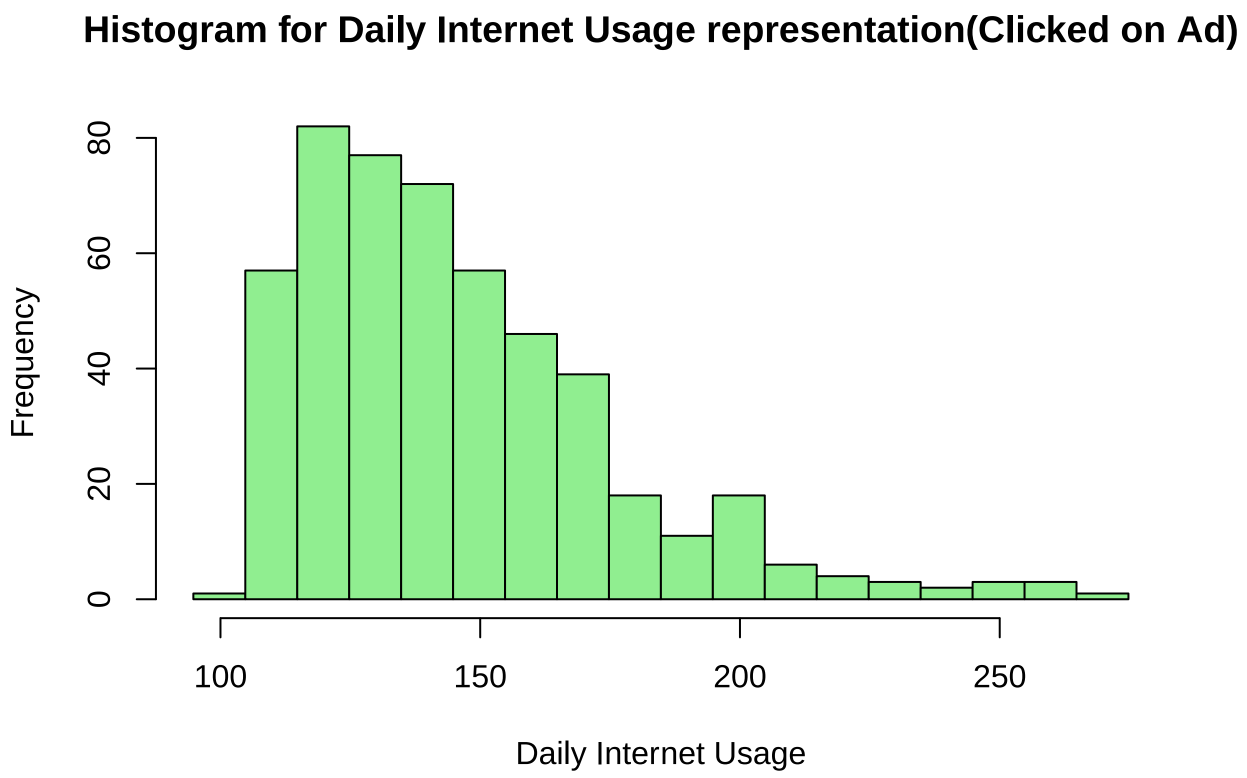
Most people who click on the ad spend between 35 to 60 minutes on site.

```
# Daily internet usage representation

# customizing bin width
bin_width <- 10

nbins <- seq(min(df$Daily.Internet.Usage) - bin_width,
             max(df$Daily.Internet.Usage) + bin_width, by = bin_width)

# plot histogram
hist(df$Daily.Internet.Usage, breaks = nbins, main = "Histogram for Daily Internet Usage representation
```



Most people use internet bundles between 115 to 145MBs and between 195 and 235MBs.

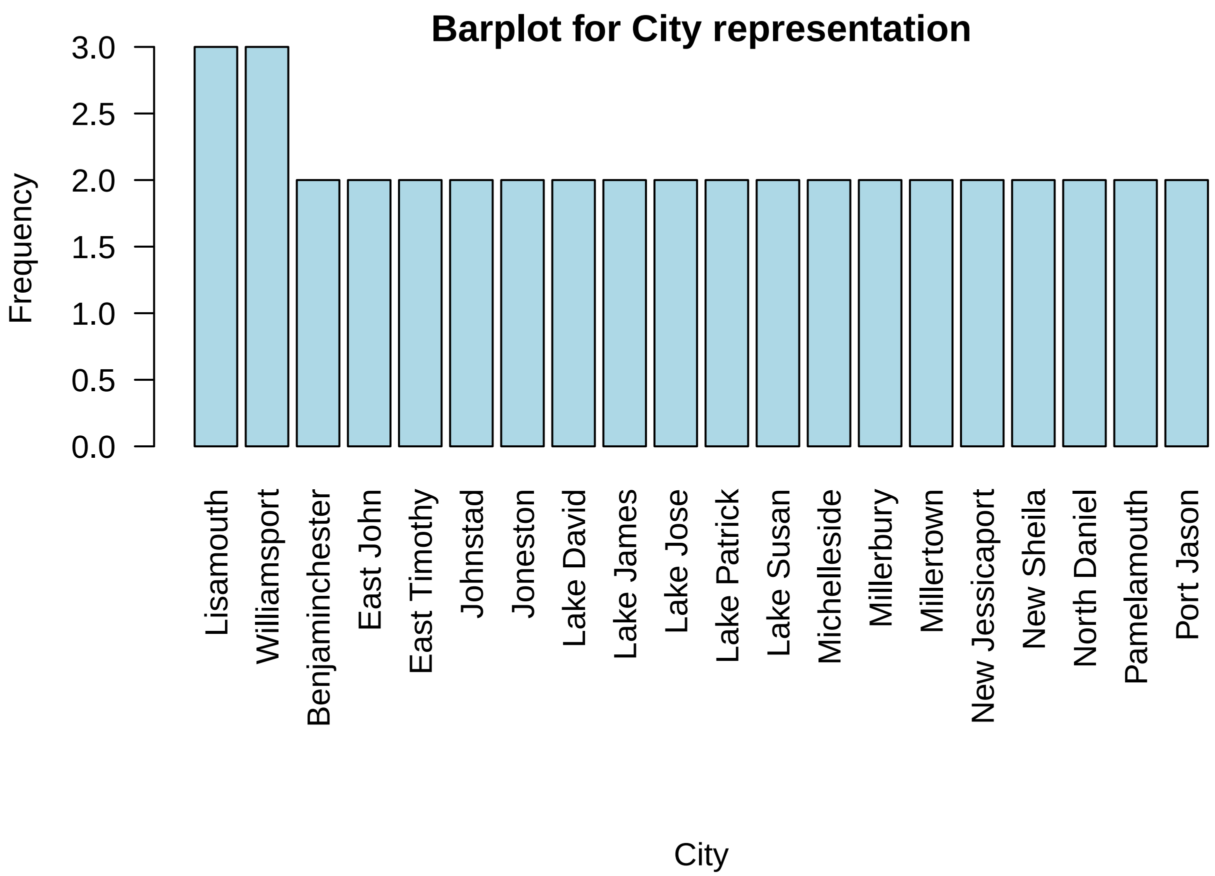
```
# Daily internet usage representation of those who clicked on ad

hist(data$Daily.Internet.Usage, breaks = nbins, main = "Histogram for Daily Internet Usage representati
```



Most people who click on ads use internet data between 105 and 185MBs.

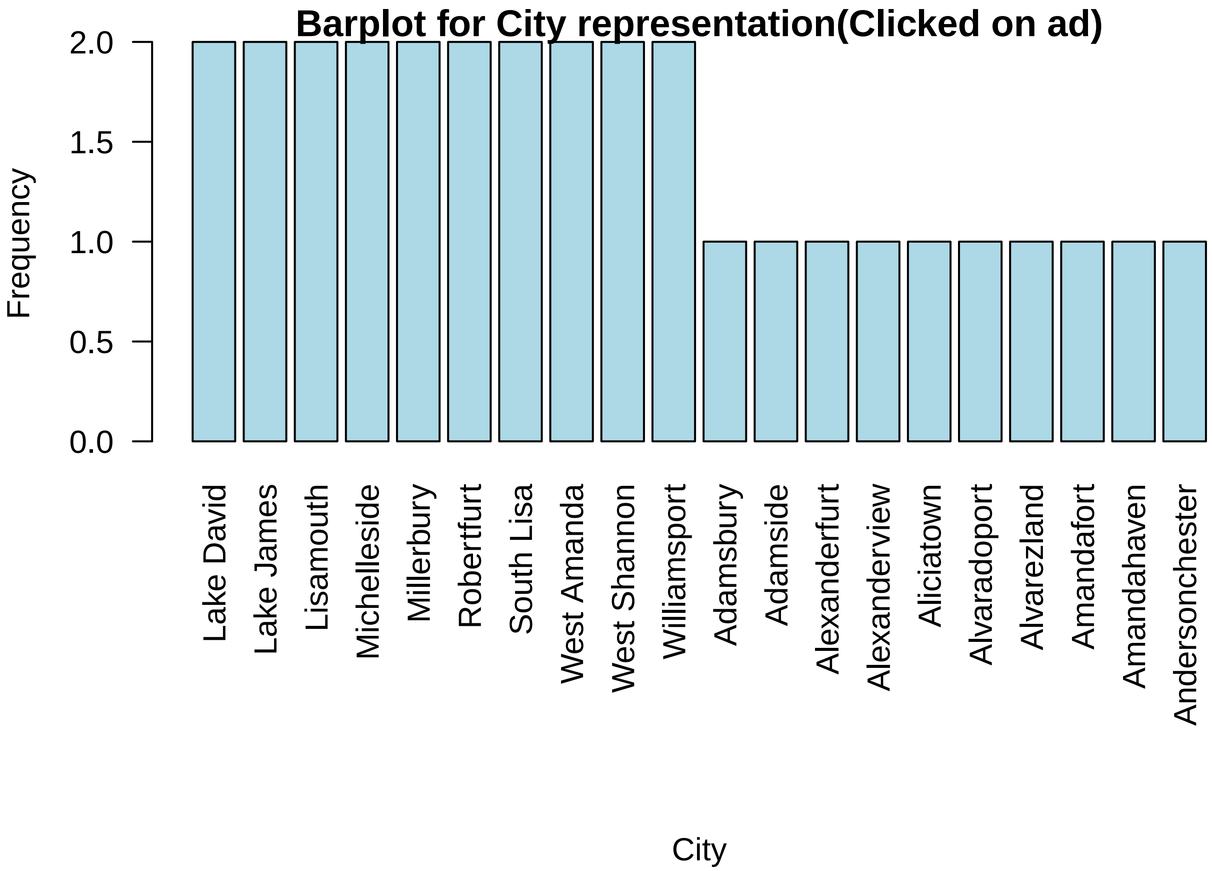
```r
# barplot function
bar <- function(column, title, xlab){

  # create frequency table
  freq <- table(column)

  # sort frequency table
  sorted_freq <- (freq[order(freq,decreasing=TRUE)])

  # adjust margins of frequency table
  par(mar=c(11,4,1,0))

  # plot bar graph for first 20 values with the highest count
  barplot(sorted_freq[1:20], main=title, ylab="Frequency", las=2, col="lightblue")
  title(xlab = xlab, line=10)
}
```

```r
# City representation
bar(df$City, "Barplot for City representation", "City")
```



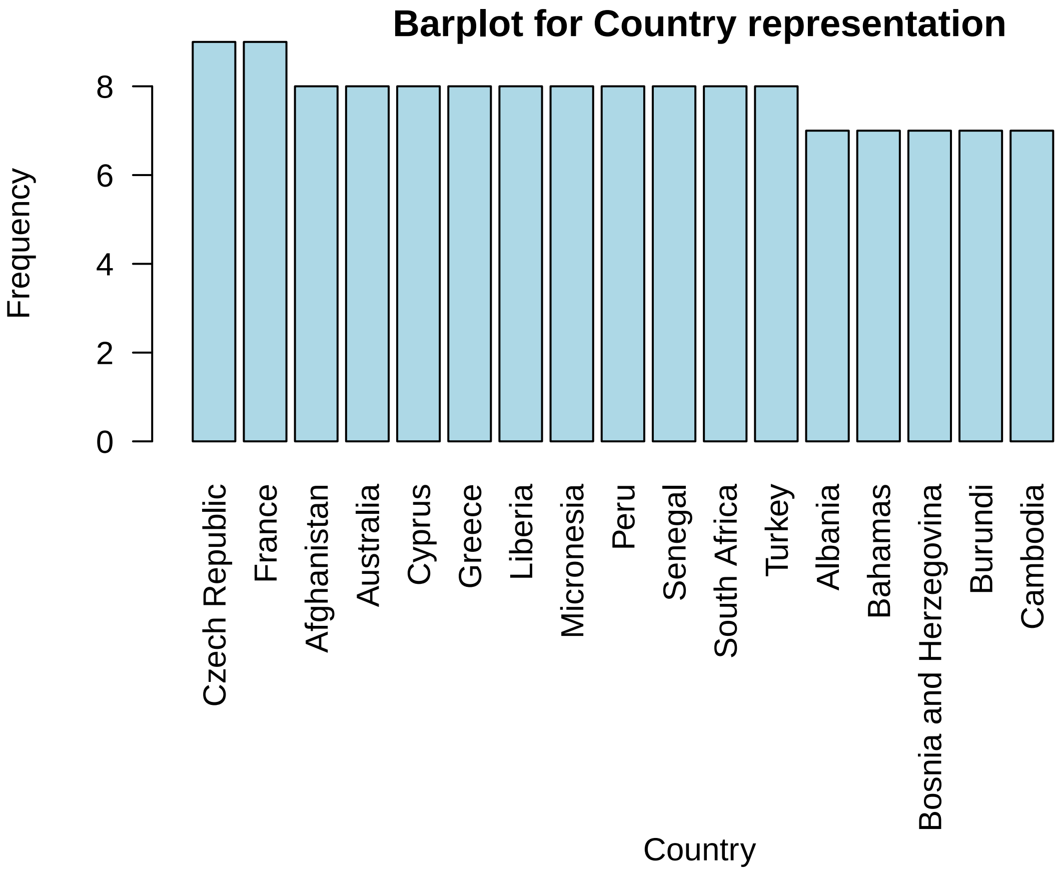
Most people come from Lisamouth and Williamsport cities both with 3 counts each.

```
# City representation of those who clicked on ad

bar(data$City, "Barplot for City representation(Clicked on ad)", "City")
```



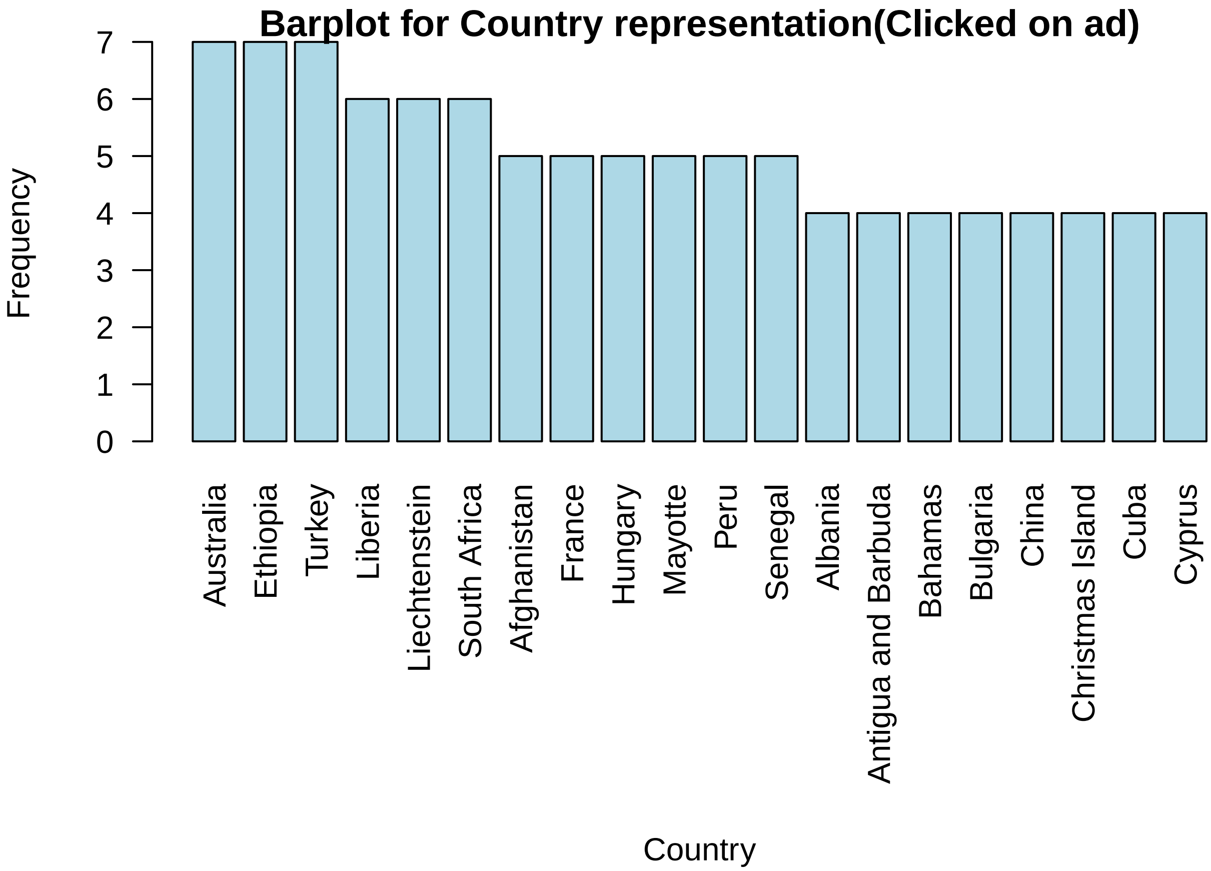
Most people who click the ads are in 10 cities namely: - Lake David - Lake James - Lisamouth - Michelleside - Millerbury - Robertfurt - South Lisa - West Amanda - West Shannon - Williamsport

```
# Country representation
bar(df$Country, "Barplot for Country representation", "Country")
```



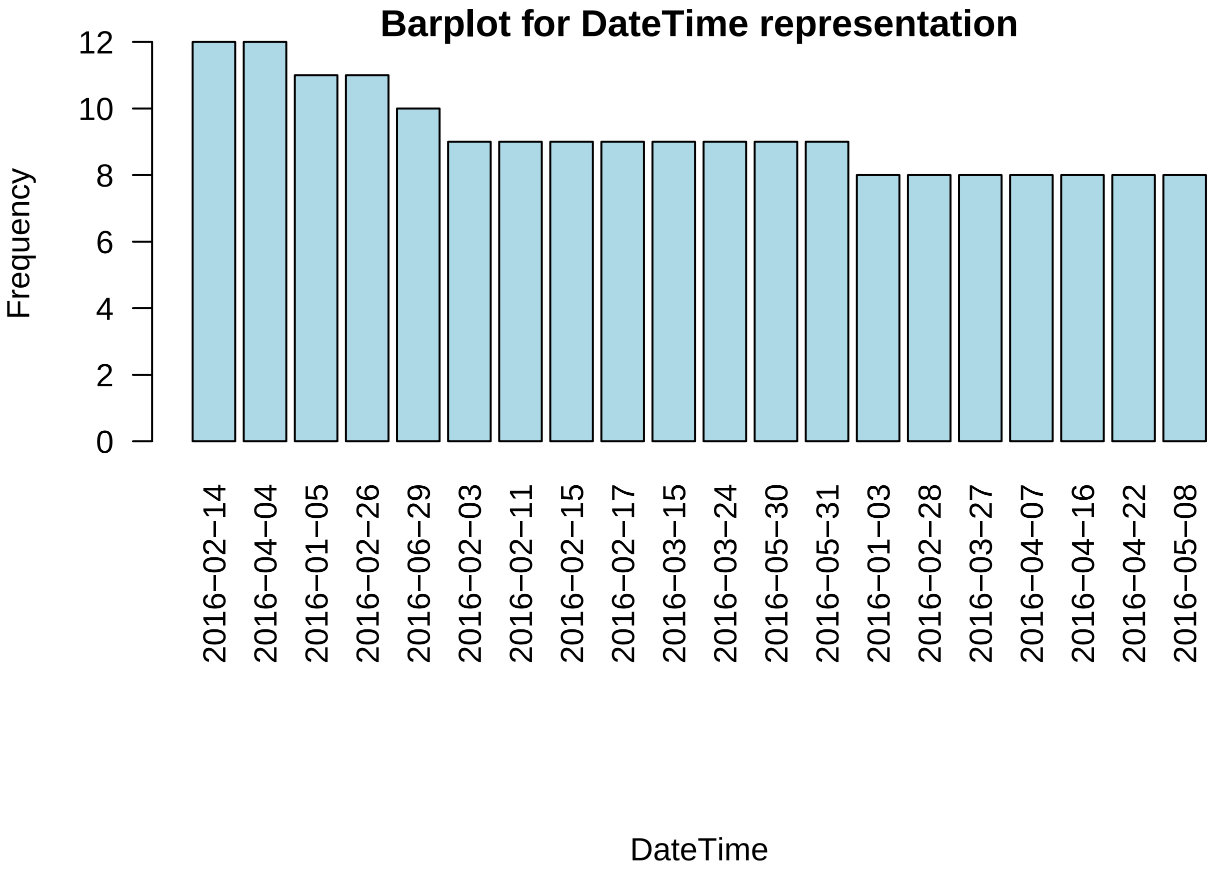
Czech Republic and France have the highest representation of the population in our dataset both with 9 counts.

```r
# Country representation of those who clicked on ad

bar(data$Country, "Barplot for Country representation(Clicked on ad)", "Country")
```



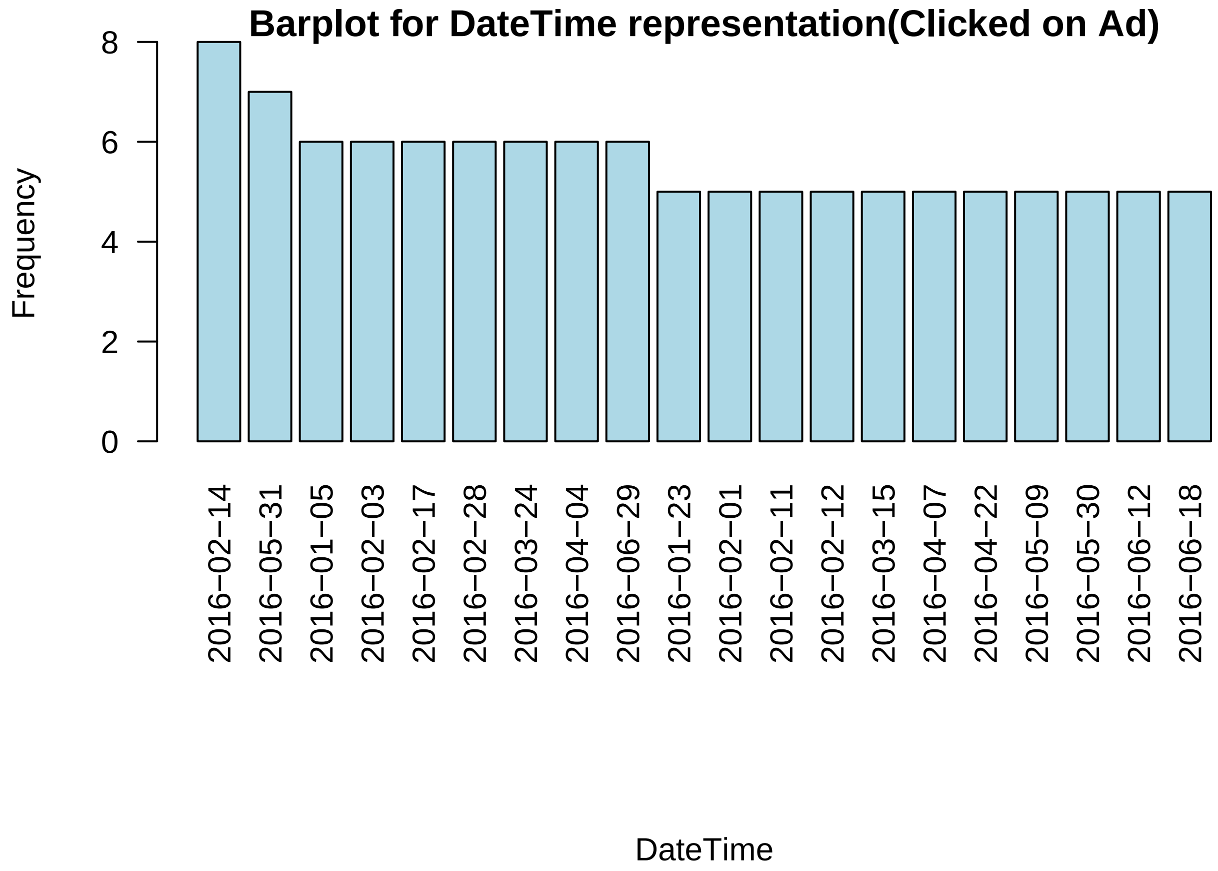
Most people who clicked on the ad are in 3 countries namely: Australia, Ethopia and Turkey.

```
# Timestamp representation
bar(df$Timestamp, "Barplot for DateTime representation", "DateTime")
```



Data was collected most for 2016-02-14 and 2016-04-04 each with a frequency of 12.

```
# Timestamp representation for those who clicked on ad
bar(data$Timestamp, "Barplot for DateTime representation(Clicked on Ad)", "DateTime")
```



2016-02-14 recorded the highest number of people who clicked on the ad.

6. Bivariate analysis

```
# check covariance in numerical variables with covariance matrix
cov(data_num)
```

```
##                          Daily.Time.Spent.on.Site          Age  Area.Income
## Daily.Time.Spent.on.Site                251.33709    -46.17415     66130.81
## Age                                     -46.17415     77.18611    -21520.93
## Area.Income                           66130.81091 -21520.92580 179952405.95
## Daily.Internet.Usage                    360.99188   -141.63482    198762.53
##                          Daily.Internet.Usage
## Daily.Time.Spent.on.Site             360.9919
## Age                                 -141.6348
## Area.Income                       198762.5315
## Daily.Internet.Usage                1927.4154
```

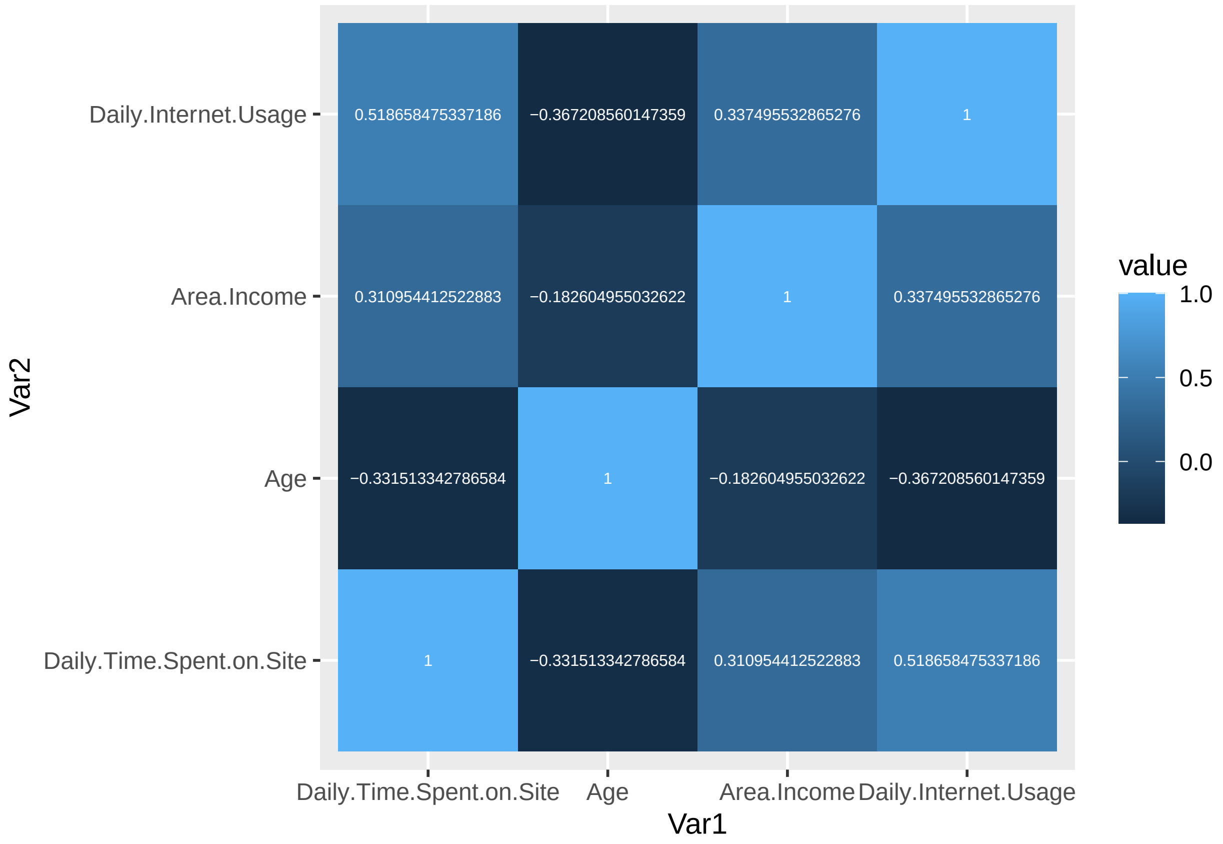
Covariance is the measure of how two random variables vary together. A high negative covariance indicates negative correlation while a high positive covariance indicates positive correlation. A value close to zero indicates weak covariance.

We can say the following have positive covariance indicating positive correlation:- * Area.Income and Daily.Time.Spent.On.Site * Area.Income and Daily.Internet.Usage * Daily.Time.Spent.On.Site and Daily.Internet.Usage

The following have negative covariance indicating negative correlation:- Age and Area.Income Age and Daily.Internet.Usage

In order to measure the strength of this relationship, we shall use a correlation matrix.

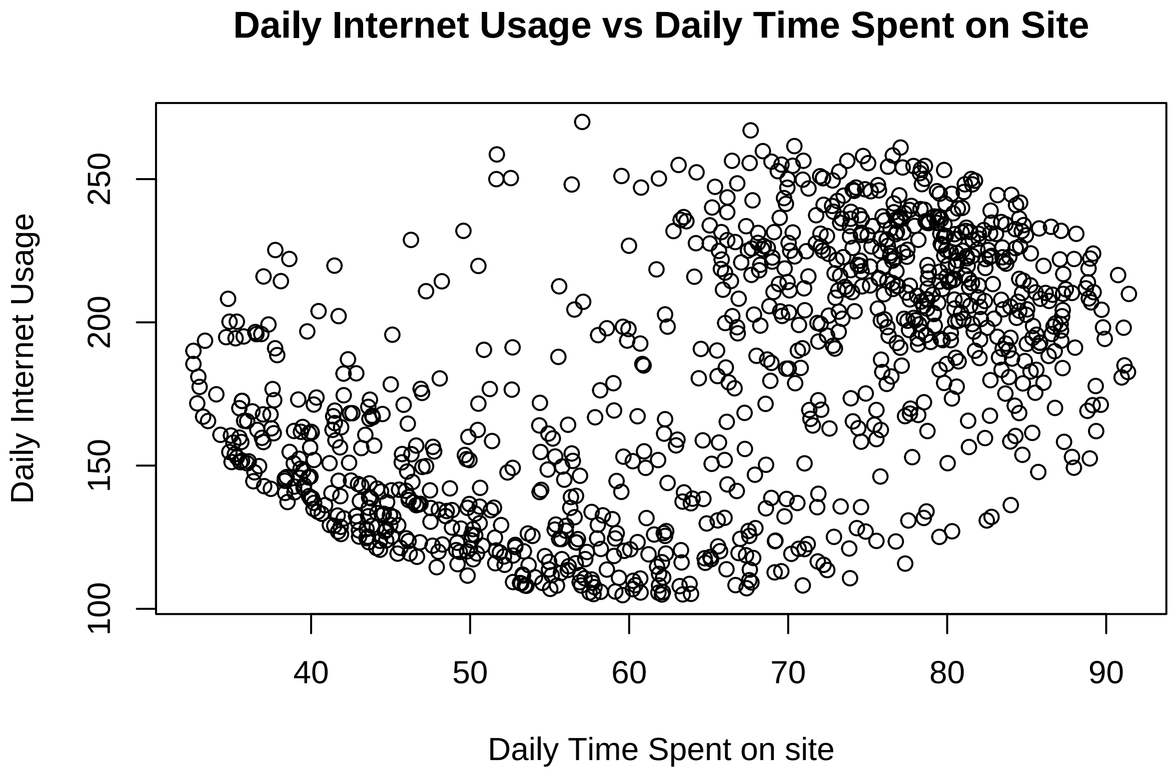
```
# correlation matrix

cormat <- cor(data_num)

# melt correlation matrix to enable heatmap plot
library(reshape2)
melted_cormat <- melt(cormat)

# plot heatmap
library(ggplot2)
ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
geom_tile() + geom_text(aes(Var2, Var1, label = value), color = "white", size = 2)
```



The correlation matrix tries to explain the covariance between variables. There is a strong positive correlation between Daily.Time.Spent.On.Site and Daily.Internet.Usage since time spent on site depends on internet accessibility.

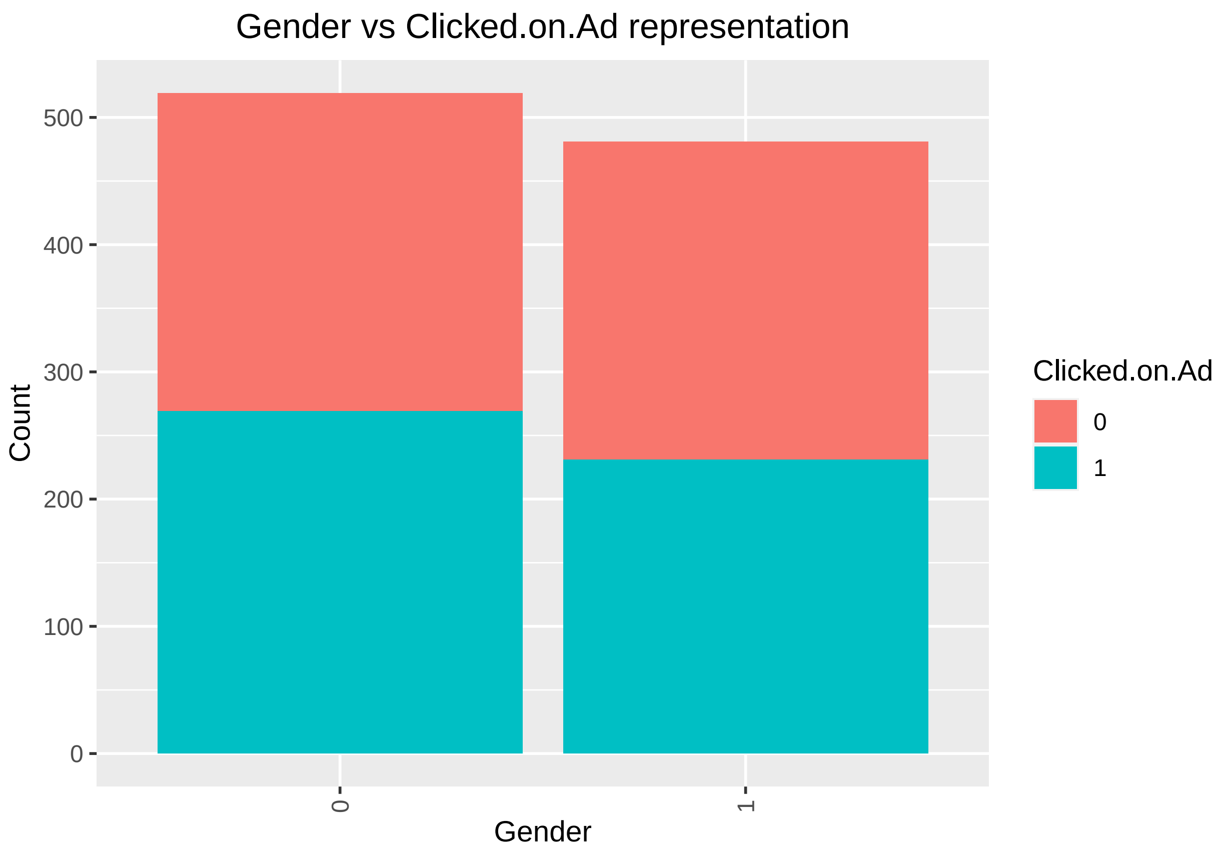
All other variables are weakly correlated.

```
# scatterplot for the two correlated variables

plot(df$Daily.Time.Spent.on.Site, df$Daily.Internet.Usage, xlab="Daily Time Spent on site", ylab="Daily
```



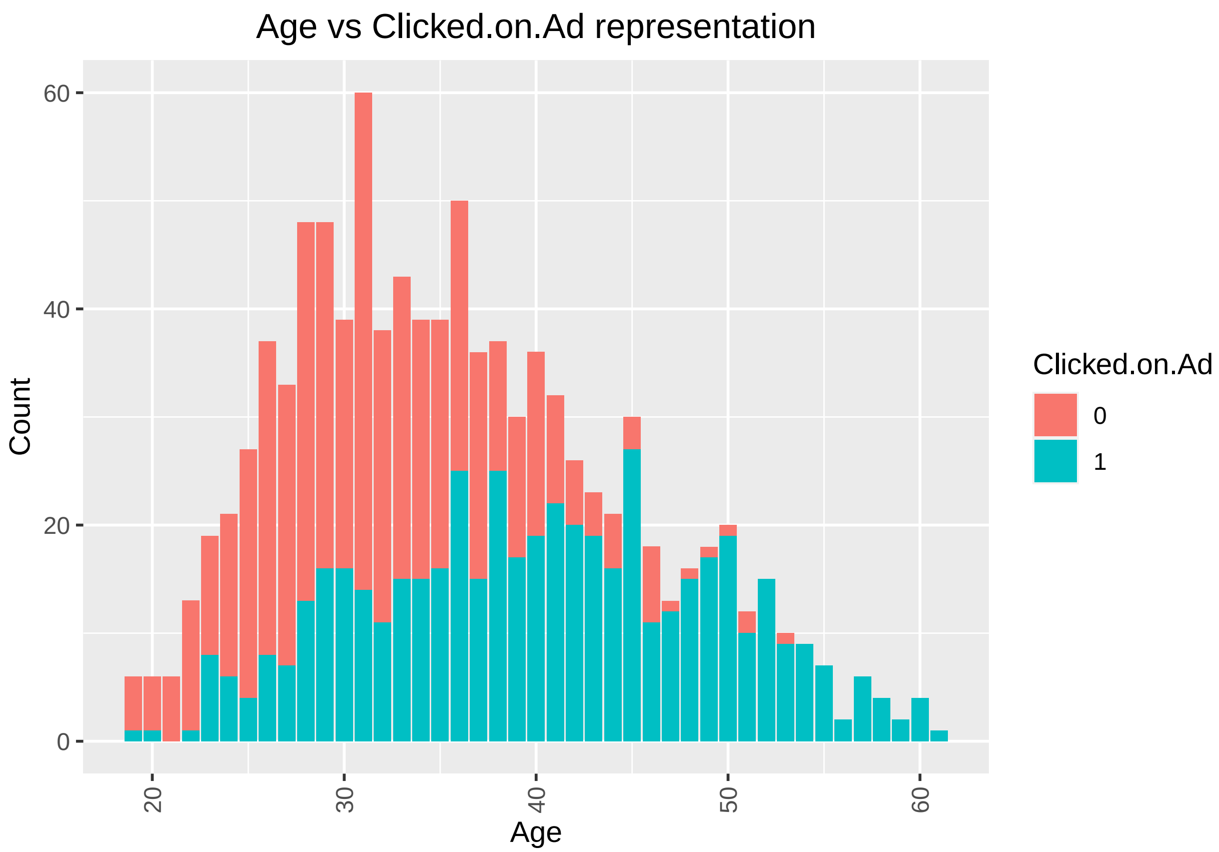
```
# create funtion to plot stacked barchart
stacked <- function(column1, column2, title, value1, value2, legend){

    # vectorize the two columns to plot
    attribute1 <- c(column1)
    attribute2 <- c(column2)

    # create dataframe of the two columns
    data <- data.frame(attribute1, attribute2)

    # plot stacked bargraph
    ggplot(data=data) + geom_bar(aes(fill=attribute2, x=attribute1)) + ggtitle(title) + xlab(value1) + y


}
```

```
# Gender against clicking on ad

stacked(df$Male, df$Clicked.on.Ad, "Gender vs Clicked.on.Ad representation", "Gender", "Count", "Clicke
```



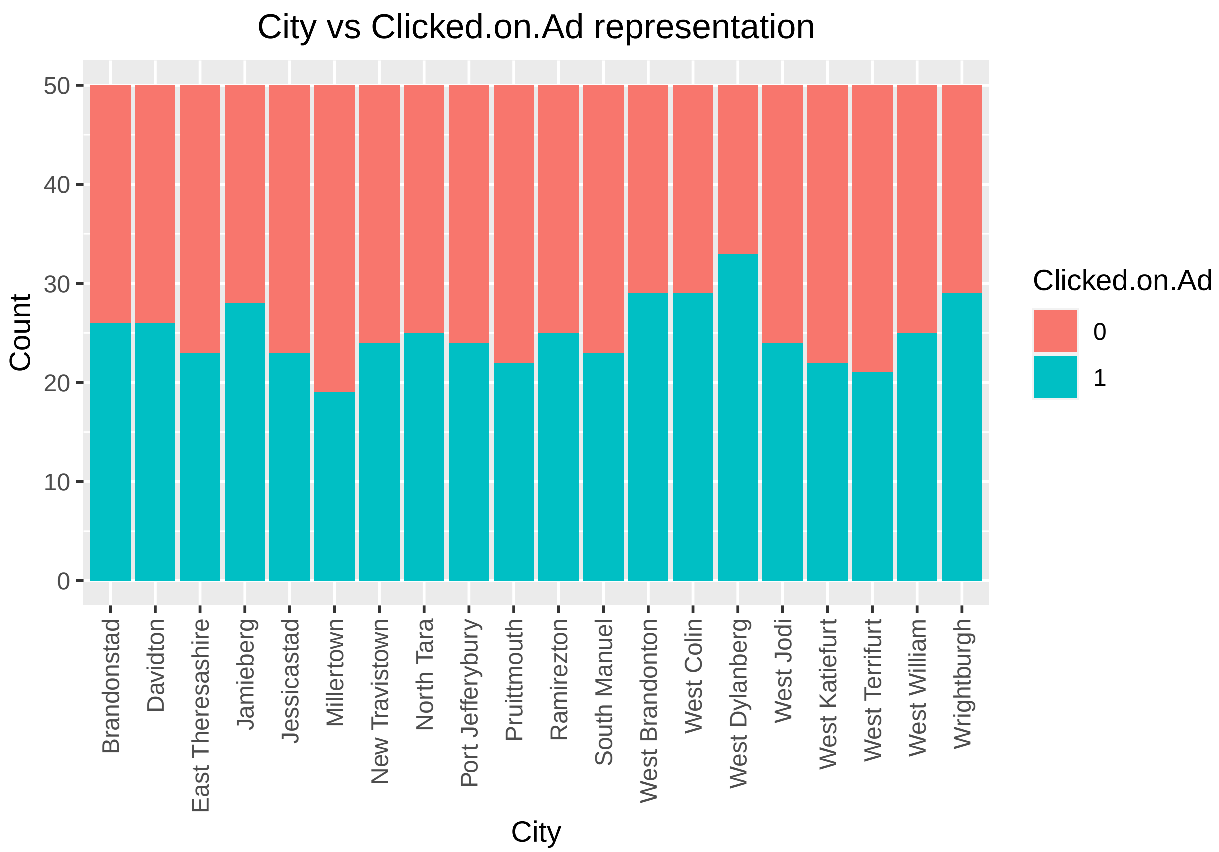
There is almost equal representation of male and female who clicked and those who did not click on the ad. This implies that gender may not influence who clicks on the add.

```
# Age representation vs clicking on ad
stacked(df$Age, df$Clicked.on.Ad, "Age vs Clicked.on.Ad representation", "Age", "Count", "Clicked.on.Ad
```

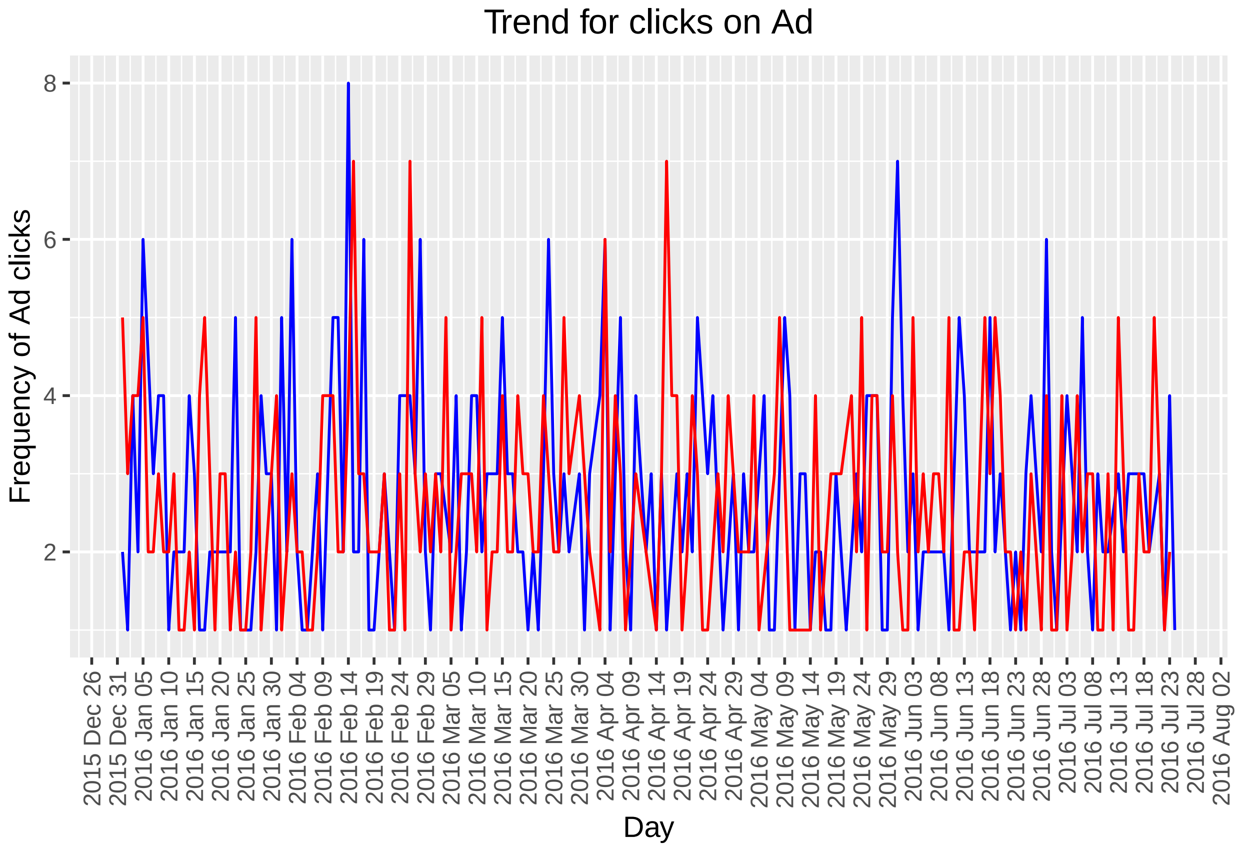


Most people who clicked the ad were 45 years old. Almost all people aged between 47 and 65 clicked on the ad. Most people aged 35 and below seem uninterested in clicking on the ads.

```
# City representation vs clicking on ad
stacked(df$City[1:20], df$Clicked.on.Ad, "City vs Clicked.on.Ad representation", "City", "Count", "Clic
```

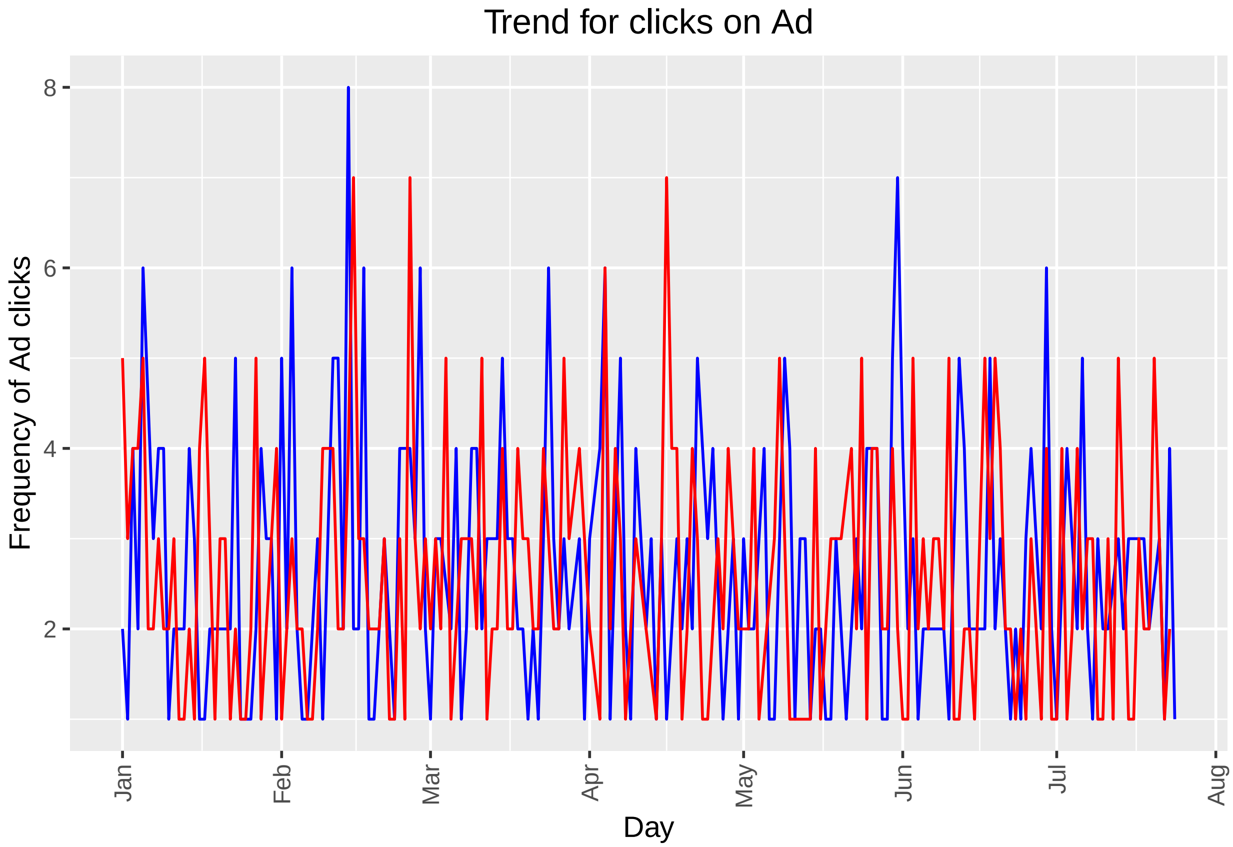


```r
# Date representation


# group days by number of clicks on ad.
times <- aggregate(x=as.factor(data$Clicked.on.Ad), by = list(data$Timestamp), FUN=length)

# group days by number of no clicks on ad.
times_ <- aggregate(x=as.factor(data_$Clicked.on.Ad), by = list(data_$Timestamp), FUN=length)

# plot line graph

ggplot() +
  geom_line(data=times, aes(x=Group.1, y=x, group=1), col = "blue")+
  geom_line(data=times_, aes(x=Group.1, y=x, group=1), col = "red") +
  geom_point() + scale_x_date(date_labels = "%Y %b %d", date_breaks = "5 days") + theme(axis.text.x = el
```



The highest number of ad clicks was recorded in the week of Februrary 2016 between 9th and 14th followed by the first week of June, 2016.

```
ggplot() +
  geom_line(data=times, aes(x=Group.1, y=x, group=1), col = "blue")+
  geom_line(data=times_, aes(x=Group.1, y=x, group=1), col = "red") +
  geom_point() + scale_x_date(date_labels = "%b", date_breaks = "1 month") + theme(axis.text.x = elemen
```



The ad was mostly clicked in early February and towards the beginning of June.

7. Conclusion

According to our analysis, gender contributes the least to who clicks on the ad or not. This is seen from the small change in percentage from the whole dataset to those who clickd on the ad. However, the rest of the attributes to some extent have an impact to this effect. Geographical location, Internet accessibility, Age and Date determine who clicks the ad.

8. Recommendations

From our findings, we are able to deduce the following in an attempt to determine a target market for our client:

i.) More females are likely to click on the ad than male. ii.) Mid Income earners between 40k and 60k are a likely target market. iii.) The ratio of the elderly who may click on the ad is higher than that of the younger group. This age ranges from 45 and above.
iv.) Client should target people from Australia, Ethopia and Turkey countries who seem most interested in the cryptography course. v.) Client should also narrow her target market to the following cities: - Lake David - Lake James - Lisamouth - Michelleside - Millerbury - Robertfurt - South Lisa - West Amanda - West Shannon - Williamsport vi.) In order to attract more customers, the client should consider discounts for her course in February and June in order to attract more customers. This way, she is likely to attract more people to her blog and course.