week12moringacore

2022-07-25

```
ad_df <- read.csv("C:/moringa/GROUP WORK/advertising.csv")
View(ad_df)

# Preview dataset
head(ad_df)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    68.95  35    61833.90               256.09
## 2                    80.23  31    68441.85               193.77
## 3                    69.47  26    59785.94               236.50
## 4                    74.15  29    54806.18               245.89
## 5                    68.37  35    73889.99               225.58
## 6                    59.99  23    59761.56               226.74
##                          Ad.Topic.Line           City Male    Count
ry
## 1    Cloned 5thgeneration orchestration    Wrightburgh    0    Tunis
ia
## 2    Monitored national standardization      West Jodi    1      Nau
ru
## 3      Organic bottom-line service-desk       Davidton    0 San Mari
no
## 4 Triple-buffered reciprocal time-frame West Terrifurt    1      Ita
ly
## 5         Robust logistical utilization   South Manuel    0    Icela
nd
## 6       Sharable client-driven software      Jamieberg    1     Norw
ay
##             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
```

```
# Finding the Shape of the dataset
dim(ad_df)
```

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

```
# Finding the datatypes of the data
str(ad_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           : chr  "Cloned 5thgeneration orchestratio
n" "Monitored national standardization" "Organic bottom-line service-de
sk" "Triple-buffered reciprocal time-frame" ...
##  $ City                    : chr  "Wrightburgh" "West Jodi" "Davidto
n" "West Terrifurt" ...
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Country                 : chr  "Tunisia" "Nauru" "San Marino" "It
aly" ...
##  $ Timestamp               : chr  "2016-03-27 00:53:11" "2016-04-04
01:39:02" "2016-03-13 20:35:42" "2016-01-10 02:31:19" ...
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
```

Data cleaning

```
# checking for missing Data
colSums(is.na(ad_df))
```

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

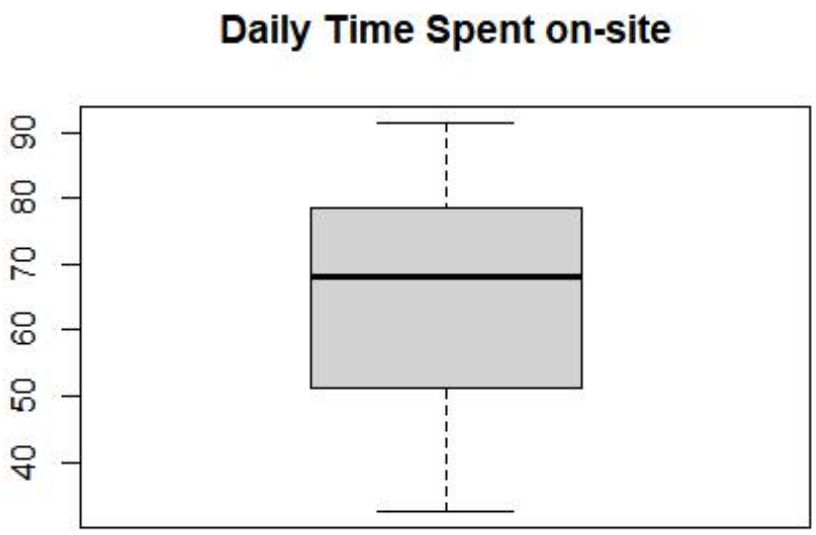
#There is no missing values in the dataset. # Check for duplicated data in the ad_Df

```
str(ad_df)
```

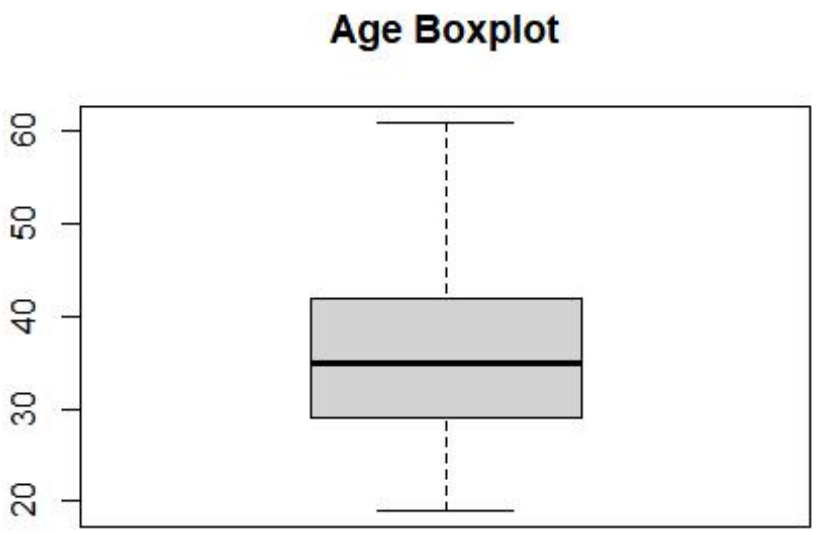
```
## 'data.frame':    1000 obs. of  10 variables:
##  $ Daily.Time.Spent.on.Site: num  69 80.2 69.5 74.2 68.4 ...
##  $ Age                     : int  35 31 26 29 35 23 33 48 30 20 ...
##  $ Area.Income             : num  61834 68442 59786 54806 73890 ...
##  $ Daily.Internet.Usage    : num  256 194 236 246 226 ...
##  $ Ad.Topic.Line           : chr  "Cloned 5thgeneration orchestratio
n" "Monitored national standardization" "Organic bottom-line service-de
sk" "Triple-buffered reciprocal time-frame" ...
##  $ City                    : chr  "Wrightburgh" "West Jodi" "Davidto
n" "West Terrifurt" ...
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Country                 : chr  "Tunisia" "Nauru" "San Marino" "It
aly" ...
```

```
##  $ Timestamp               : chr  "2016-03-27 00:53:11" "2016-04-04
01:39:02" "2016-03-13 20:35:42" "2016-01-10 02:31:19" ...
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...

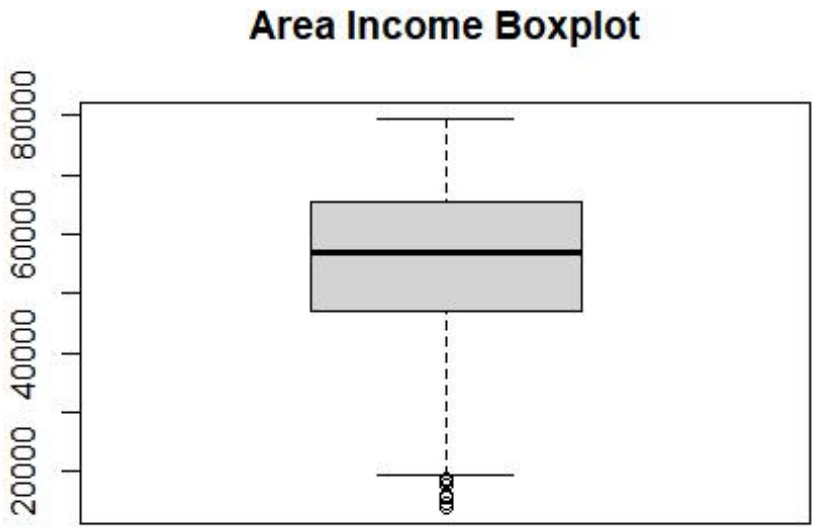
boxplot(ad_df$Daily.Time.Spent.on.Site, main = 'Daily Time Spent on-sit
e')
```



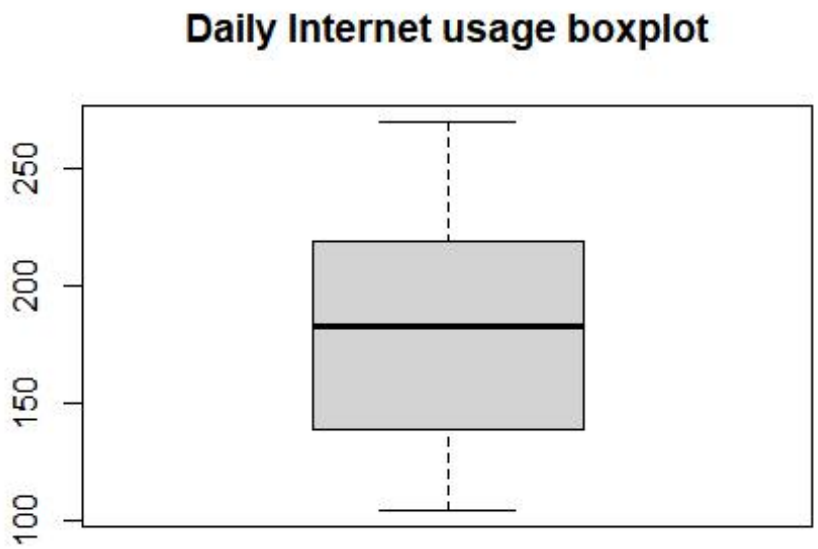

```
boxplot(ad_df$Age, main = 'Age Boxplot')
```




```
boxplot(ad_df$Area.Income, main = 'Area Income Boxplot')
```

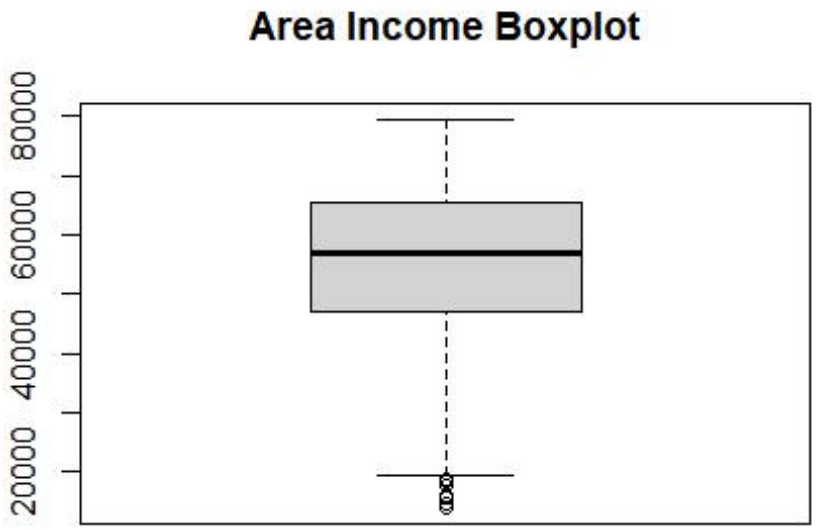


```
boxplot(ad_df$Daily.Internet.Usage, main = 'Daily Internet usage boxplo
t')
```



From the boxplots, only the Area_income column has outliers.

```
#Print out the outliers
boxplot(ad_df$Area.Income, main = 'Area Income Boxplot')$out
```




```
## [1] 17709.98 18819.34 15598.29 15879.10 14548.06 13996.50 14775.50 1
8368.57
```

There are outliers that do not look like they are in the extreme. There are areas where poverty is prevelant in such areas the total income could be that small.

str (ad_df)

```
```r
ad_df[['Timestamp']] <- as.POSIXct(ad_df[['Timestamp']],
                                   format = "%Y-%m-%d %H:%M:%S")
str(ad_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           : chr  "Cloned 5thgeneration orchestratio
n" "Monitored national standardization" "Organic bottom-line service-de
sk" "Triple-buffered reciprocal time-frame" ...
##  $ City                    : chr  "Wrightburgh" "West Jodi" "Davidto
n" "West Terrifurt" ...
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Country                 : chr  "Tunisia" "Nauru" "San Marino" "It
```
```

```
aly" ...
##  $ Timestamp               : POSIXct, format: "2016-03-27 00:53:11"
"2016-04-04 01:39:02" ...
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
```

The timestamp column is now in the correct dtype

Univariate Data Analysis

###Numerical Columns

```
summary(ad_df)
```

```
##  Daily.Time.Spent.on.Site      Age         Area.Income    Daily.Inte
rnet.Usage
##  Min.   :32.60            Min.   :19.00   Min.   :13996   Min.   :10
4.8
##  1st Qu.:51.36            1st Qu.:29.00   1st Qu.:47032   1st Qu.:13
8.8
##  Median :68.22            Median :35.00   Median :57012   Median :18
3.1
##  Mean   :65.00            Mean   :36.01   Mean   :55000   Mean   :18
0.0
##  3rd Qu.:78.55            3rd Qu.:42.00   3rd Qu.:65471   3rd Qu.:21
8.8
##  Max.   :91.43            Max.   :61.00   Max.   :79485   Max.   :27
0.0
##  Ad.Topic.Line          City                Male         Country

##  Length:1000        Length:1000        Min.   :0.000   Length:1000

##  Class :character   Class :character   1st Qu.:0.000   Class :charac
ter
##  Mode  :character   Mode  :character   Median :0.000   Mode  :charac
ter
##                                        Mean   :0.481

##                                        3rd Qu.:1.000

##                                        Max.   :1.000

##    Timestamp                       Clicked.on.Ad
##  Min.   :2016-01-01 02:52:10.00   Min.   :0.0
##  1st Qu.:2016-02-18 02:55:42.00   1st Qu.:0.0
##  Median :2016-04-07 17:27:29.50   Median :0.5
##  Mean   :2016-04-10 10:34:06.64   Mean   :0.5
##  3rd Qu.:2016-05-31 03:18:14.00   3rd Qu.:1.0
##  Max.   :2016-07-24 00:22:16.00   Max.   :1.0
```

```
Age
# Mean
mean.age <- mean(ad_df$Age)
mean.age

## [1] 36.009

#median
median.age <- median (ad_df$Age)
median.age

## [1] 35

# Function to get the mode.
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

mode.age <- getmode(ad_df$age)
mode.age

## NULL
```

####Area income

```
mean.areaincome <- mean(ad_df$Area.Income)
mean.areaincome

## [1] 55000

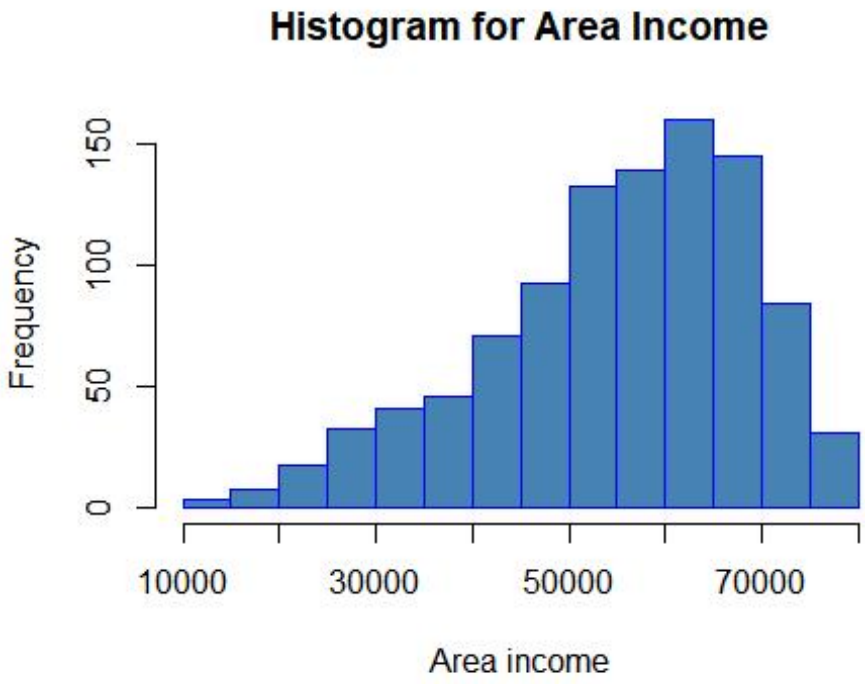
median.areaincome <- median(ad_df$Area.Income)
median.areaincome

## [1] 57012.3

mode.areaincome <- getmode(ad_df$Area.Income)
mode.areaincome

## [1] 61833.9

hist(ad_df$Area.Income,
     main="Histogram for Area Income",
     xlab="Area income",
     border="blue",
     col="steelblue",)
```



Daily.Internet.Usage

```
mean.daily.internet <- mean(ad_df$Daily.Internet.Usage)
mean.daily.internet

## [1] 180.0001

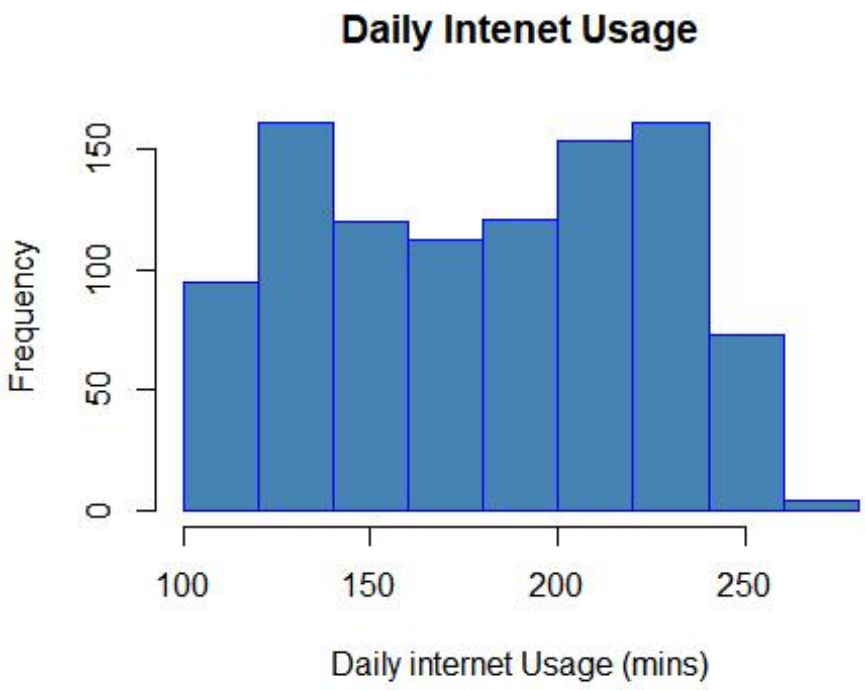
median.daily.internet <- median(ad_df$Daily.Internet.Usage)
median.daily.internet

## [1] 183.13

mode.daily.internet <- getmode(ad_df$Daily.Internet.Usage)
mode.daily.internet

## [1] 167.22

hist(ad_df$Daily.Internet.Usage,
     main = 'Daily Intenet Usage',
     xlab="Daily internet Usage (mins)",
     border="blue",
     col="steelblue")
```



Daily time spent on site

```
mean.dtsos <- mean(ad_df$Daily.Time.Spent.on.Site)
mean.dtsos
```

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

```
median.dtsos <- median(ad_df$Daily.Time.Spent.on.Site)
median.dtsos
```

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

```
mode.dtsos <- getmode(ad_df$Daily.Time.Spent.on.Site)
mode.dtsos
```

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

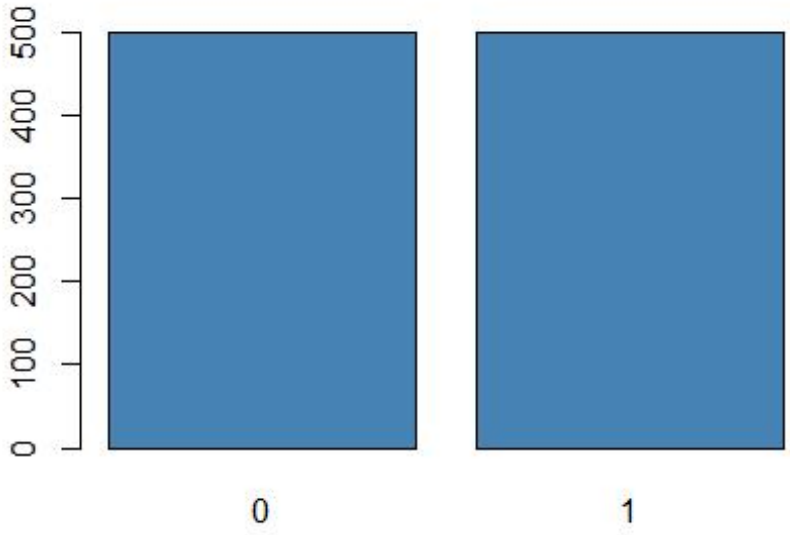
Clicked.on.Ad

```
uniq_clickers <- unique(ad_df$Clicked.on.Ad, )
length(uniq_clickers)
```

```
## [1] 2
```

There are two categories of the people who clicked on ads Let us plot the frequency of each

```
clickers <- ad_df$Clicked.on.Ad
clickers_frequency <- table (clickers)
barplot(clickers_frequency, col = "steelblue")
```



There are 500 people who clicked on ads and another 500 did not click on the ads.

Categorical Columns

####Ad.Topic.line

```
uniq_topic <- unique(ad_df$Ad.Topic.Line, )
length(uniq_topic)
```

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

There are 1000 unique topic lines meaning it would be impossible to get a good visualization.

City

```
uniq_city <- unique(ad_df$City, )
length(uniq_city)
```

```
## [1] 969
```

There are 969 unique cities hence it would also be impossible to get a good visualization

Country

```
uniq_country <- unique(ad_df$Country)
length(uniq_country)
```

```
## [1] 237
```

There are 237 unique countries.

```
library(sf)

## Linking to GEOS 3.9.1, GDAL 3.4.3, PROJ 7.2.1; sf_use_s2() is TRUE

library(raster)

## Loading required package: sp

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:raster':
##
##     intersect, select, union

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

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

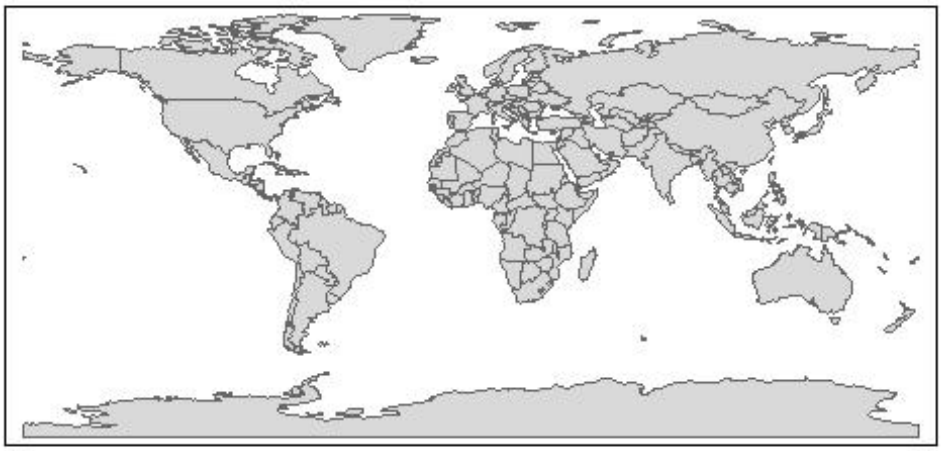
library(spData)

## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`

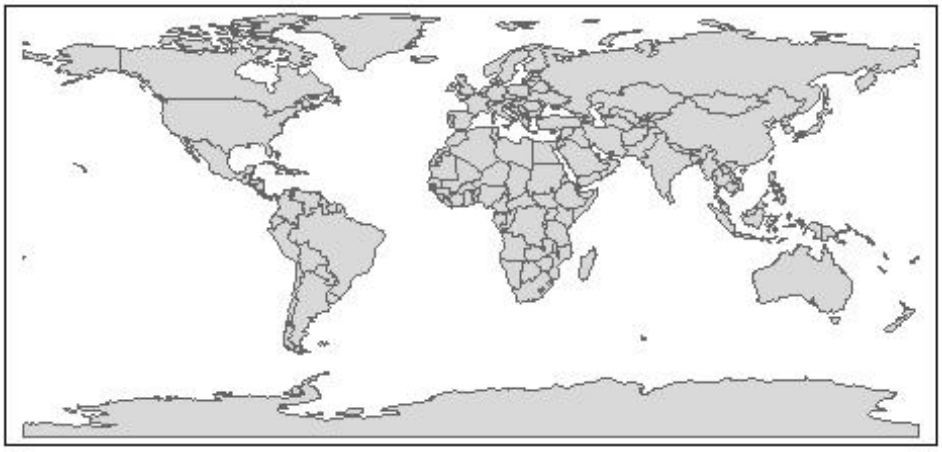
#library(spDataLarge)
library(tmap)    # for static and interactive maps
library(leaflet) # for interactive maps
library(ggplot2)

Country <- ad_df$Country
countyfreq <- table(Country)

tm_shape(world) +
  tm_polygons()
```

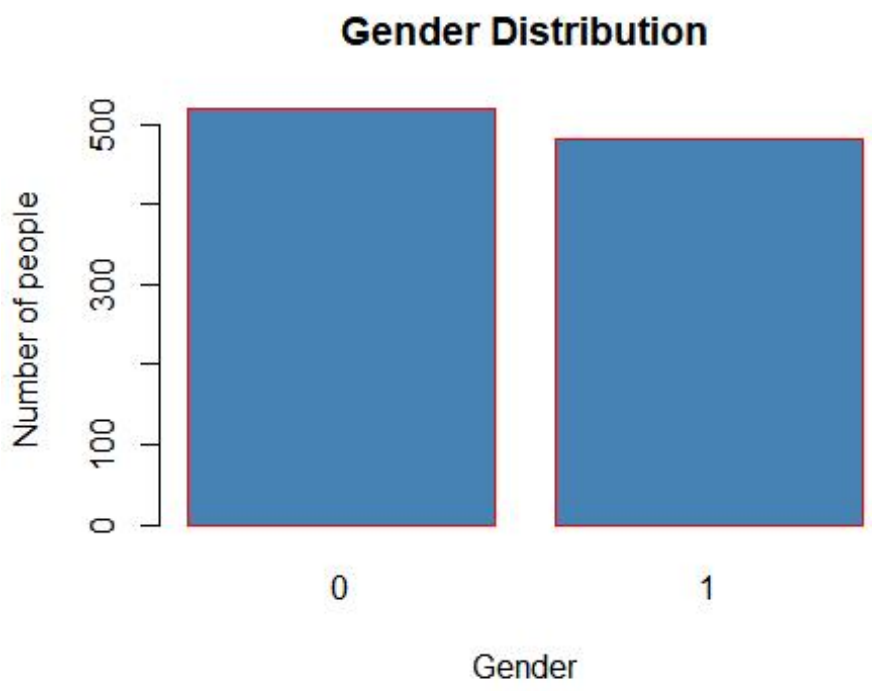


```
tm_shape(world) +
  tm_fill() +
  tm_borders()
```



Gender

```
male <- ad_df$Male
male_freq <- table(male)
barplot(male_freq, main= 'Gender Distribution', xlab="Gender",
        ylab="Number of people",
        border="red",
        col="steelblue")
```



###Overall Summary

```
summary(ad_df)
```

```
##  Daily.Time.Spent.on.Site      Age         Area.Income    Daily.Inte
rnet.Usage
##  Min.   :32.60           Min.   :19.00   Min.   :13996   Min.   :10
4.8
##  1st Qu.:51.36           1st Qu.:29.00   1st Qu.:47032   1st Qu.:13
8.8
##  Median :68.22           Median :35.00   Median :57012   Median :18
3.1
##  Mean   :65.00           Mean   :36.01   Mean   :55000   Mean   :18
0.0
##  3rd Qu.:78.55           3rd Qu.:42.00   3rd Qu.:65471   3rd Qu.:21
8.8
##  Max.   :91.43           Max.   :61.00   Max.   :79485   Max.   :27
0.0
##  Ad.Topic.Line          City                Male         Country
```

```
##  Length:1000        Length:1000        Min.   :0.000   Length:1000

##  Class :character   Class :character   1st Qu.:0.000   Class :charac
ter
##  Mode  :character   Mode  :character   Median :0.000   Mode  :charac
ter
##                                        Mean   :0.481

##                                        3rd Qu.:1.000

##                                        Max.   :1.000

##    Timestamp                       Clicked.on.Ad
##  Min.   :2016-01-01 02:52:10.00   Min.   :0.0
##  1st Qu.:2016-02-18 02:55:42.00   1st Qu.:0.0
##  Median :2016-04-07 17:27:29.50   Median :0.5
##  Mean   :2016-04-10 10:34:06.64   Mean   :0.5
##  3rd Qu.:2016-05-31 03:18:14.00   3rd Qu.:1.0
##  Max.   :2016-07-24 00:22:16.00   Max.   :1.0
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:raster':
##
##     intersect, union
```

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

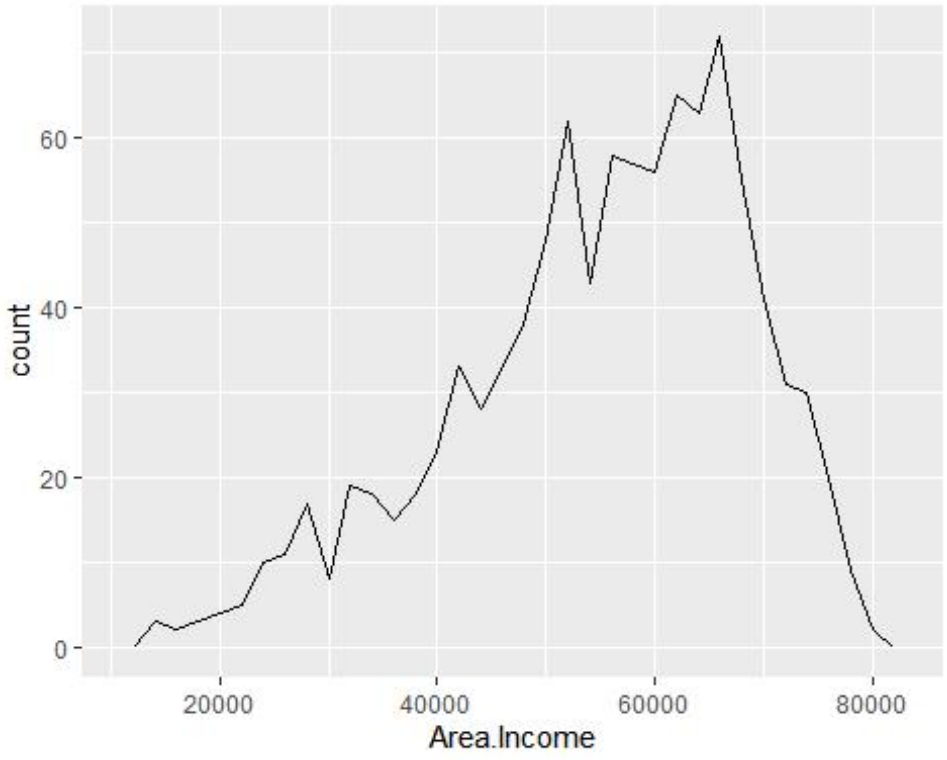
```
ad_df$Month_Yr <- format(as.Date(ad_df$Timestamp), "%Y-%m")
head(ad_df)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    68.95  35    61833.90               256.09
## 2                    80.23  31    68441.85               193.77
## 3                    69.47  26    59785.94               236.50
## 4                    74.15  29    54806.18               245.89
## 5                    68.37  35    73889.99               225.58
## 6                    59.99  23    59761.56               226.74
##                          Ad.Topic.Line           City Male    Count
ry
## 1    Cloned 5thgeneration orchestration    Wrightburgh    0    Tunis
ia
## 2    Monitored national standardization      West Jodi    1      Nau
ru
## 3      Organic bottom-line service-desk       Davidton    0 San Mari
```

```
no
## 4 Triple-buffered reciprocal time-frame West Terrifurt    1      Ita
ly
## 5         Robust logistical utilization   South Manuel    0    Icela
nd
## 6       Sharable client-driven software      Jamieberg    1     Norw
ay
##             Timestamp Clicked.on.Ad Month_Yr
## 1 2016-03-27 00:53:11             0  2016-03
## 2 2016-04-04 01:39:02             0  2016-04
## 3 2016-03-13 20:35:42             0  2016-03
## 4 2016-01-10 02:31:19             0  2016-01
## 5 2016-06-03 03:36:18             0  2016-06
## 6 2016-05-19 14:30:17             0  2016-05
```

Bivariate Analysis

```
ggplot(data = ad_df, mapping = aes(x = Area.Income)) +
  geom_freqpoly(mapping = aes(colour = Clicked.on.Ad), binwidth = 2000)
```



In areas where the income lies between 60,000 and & 70,000 there is a higher number of people clicking the ads #### Correlation

```
#creating with only interger columns
numerical_df = ad_df[c("Daily.Time.Spent.on.Site", "Age", "Area.Income",
"Daily.Internet.Usage" ,"Male", "Clicked.on.Ad" )]
head(numerical_df)
```

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

```
correlation = cor(numerical_df)
correlation
```

```
##                           Daily.Time.Spent.on.Site         Age  Area.
Income
## Daily.Time.Spent.on.Site               1.00000000 -0.33151334  0.310
954413
## Age                                   -0.33151334  1.00000000 -0.182
604955
## Area.Income                            0.31095441 -0.18260496  1.000
000000
## Daily.Internet.Usage                   0.51865848 -0.36720856  0.337
495533
## Male                                  -0.01895085 -0.02104406  0.001
322359
## Clicked.on.Ad                         -0.74811656  0.49253127 -0.476
254628
##                           Daily.Internet.Usage         Male Clicked.o
n.Ad
## Daily.Time.Spent.on.Site           0.51865848 -0.018950855   -0.7481
1656
## Age                               -0.36720856 -0.021044064    0.4925
3127
## Area.Income                        0.33749553  0.001322359   -0.4762
5463
## Daily.Internet.Usage               1.00000000  0.028012326   -0.7865
3918
## Male                               0.02801233  1.000000000   -0.0380
2747
## Clicked.on.Ad                     -0.78653918 -0.038027466    1.0000
0000
```

```
library("PerformanceAnalytics")
```

```
## Loading required package: xts
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'xts'

## The following object is masked from 'package:leaflet':
##
##     addLegend

## The following objects are masked from 'package:dplyr':
##
##     first, last

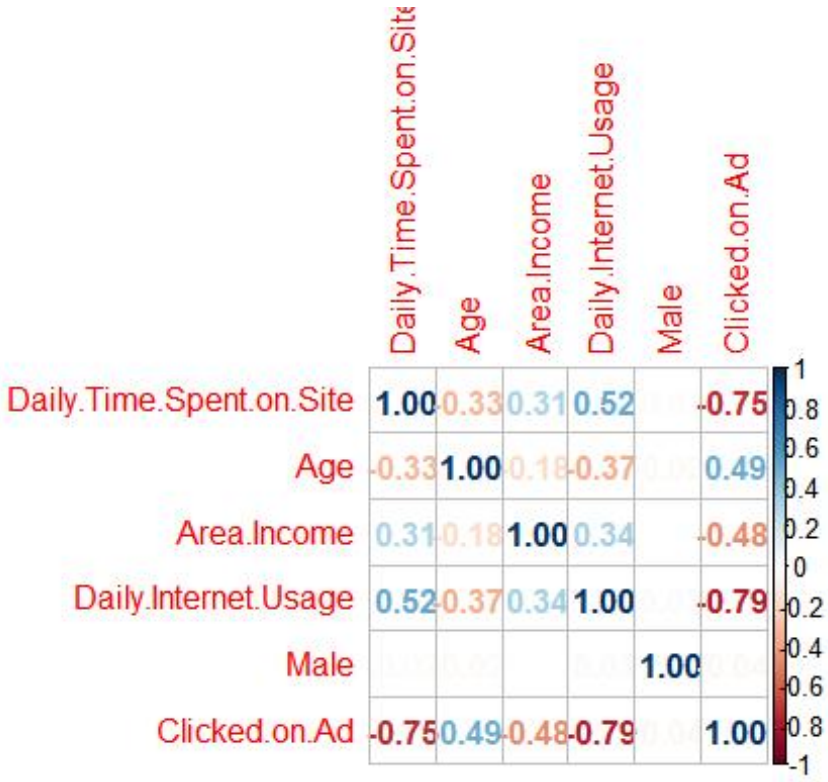
##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend

library(corrplot)

## corrplot 0.92 loaded

# Correlation Matrix
corrplot(correlation, method = 'number')
```




```
chart.Correlation(numerical_df, histogram = TRUE, pch = 19, )
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

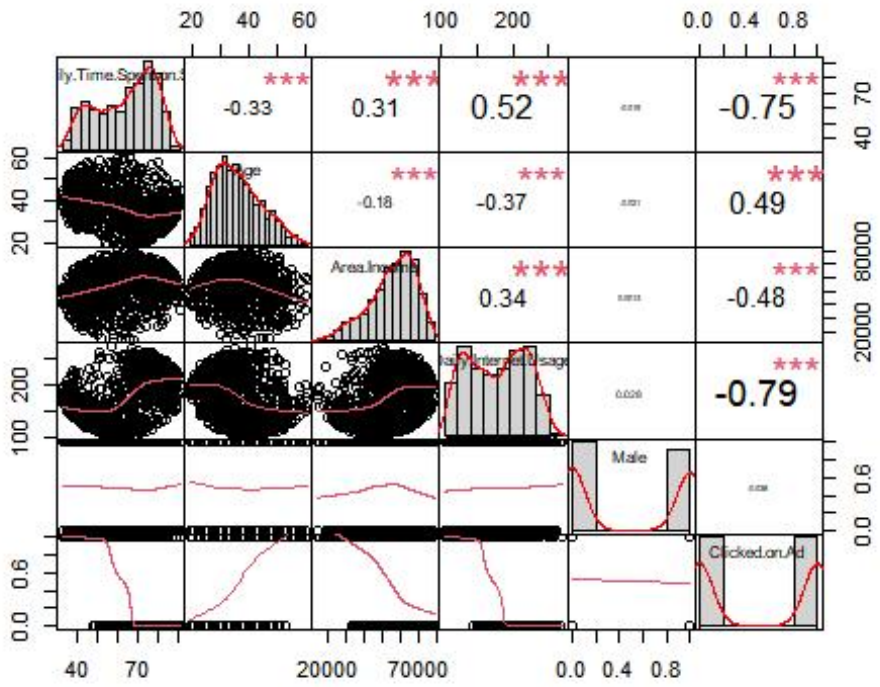
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter

## Warning in par(usr): argument 1 does not name a graphical parameter
```



The chart correlations gives a clear summary on the Bivariate analysis of the dataframe.