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

d.) Experimental design taken

e.) Data appropriateness to answer the given question.

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 st
##  $ 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

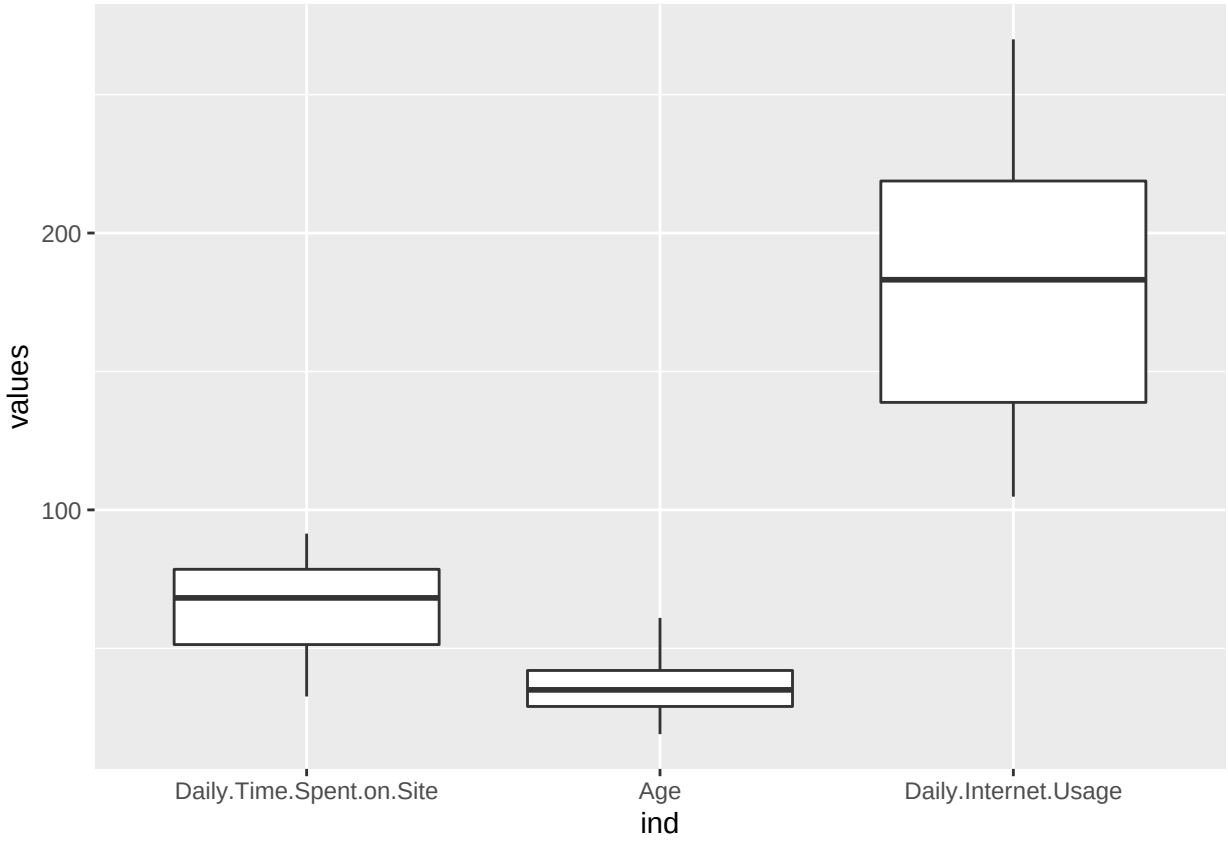
```
# check for outliers

# 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
```

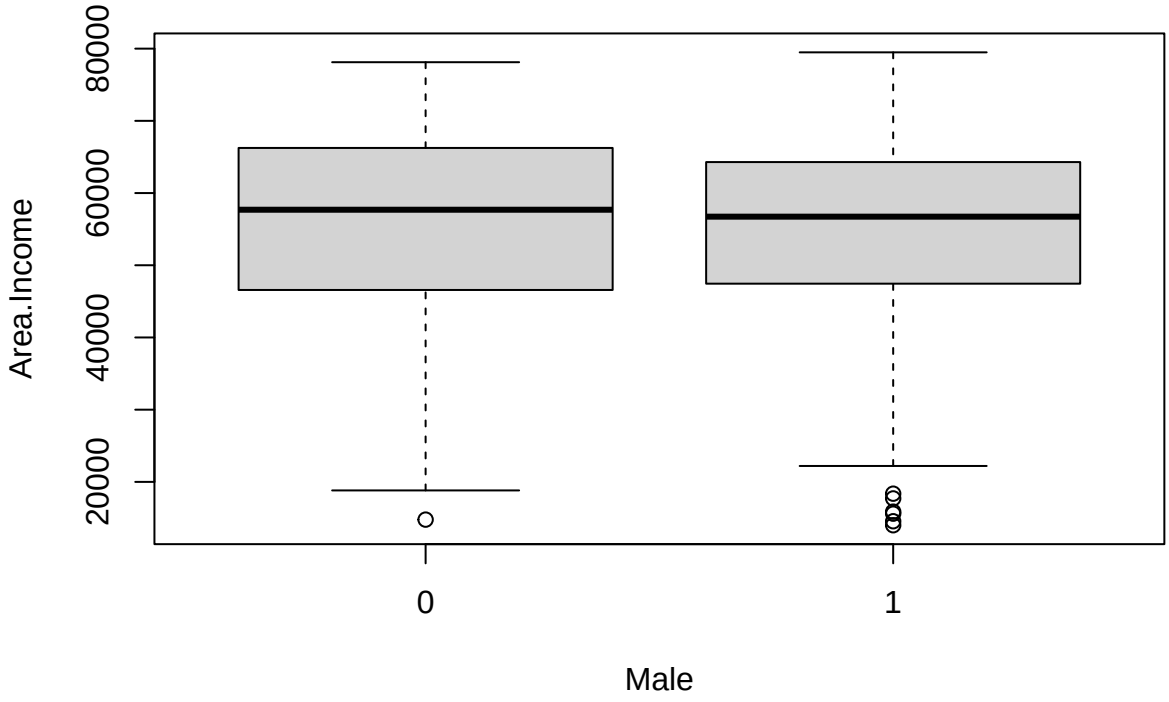
```
## # A tibble: 1,000 x 4
##    Daily.Time.Spent.on.Site   Age Area.Income Daily.Internet.Usage
##                       <dbl> <int>       <dbl>                <dbl>
##  1                     69.0    35      61834.                256.
##  2                     80.2    31      68442.                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
```

```
# 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.

```
# 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)
```

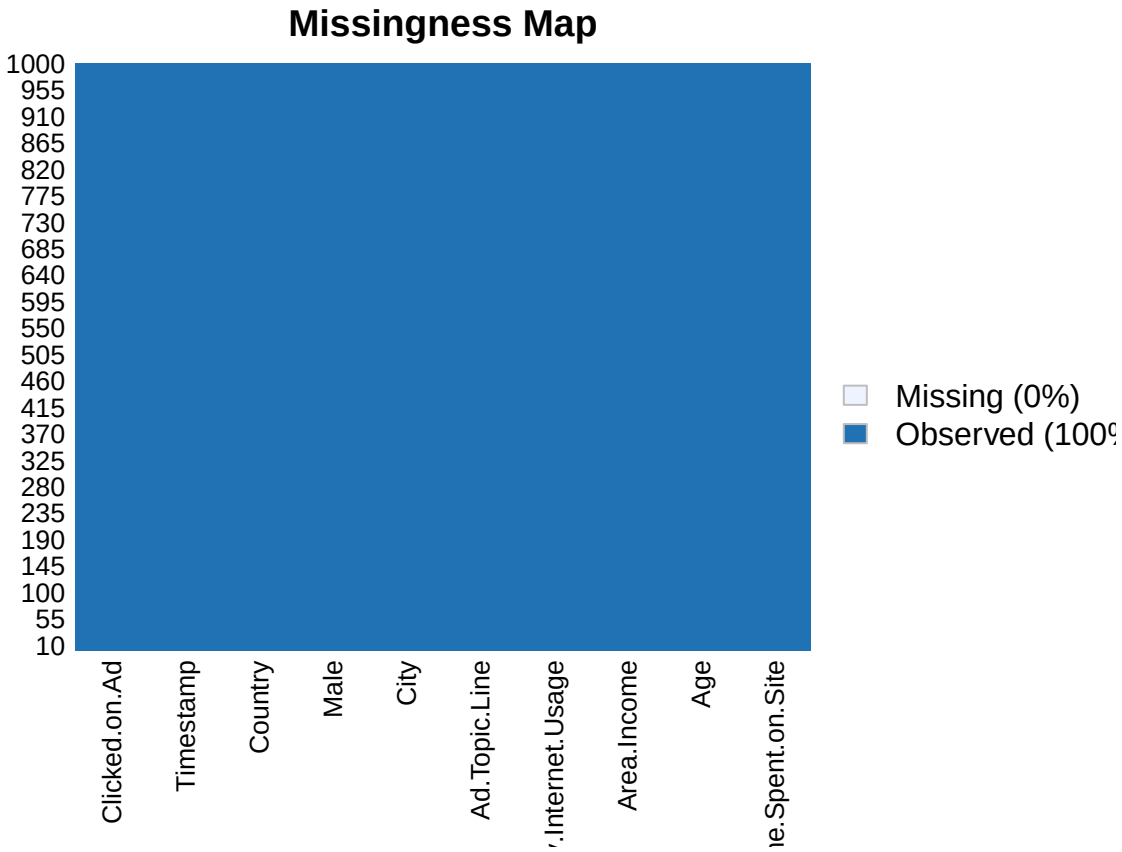
```
## Loading required package: Rcpp
```

```
## ##
## ## 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 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 on people who clicked on the ad and those who did not.

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



print(paste("Male: ", getmode(df$Male)))
```

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

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

```
## [1] "Daily.Time.Spent.on.Site:  62.26"
```

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

```
## [1] "Daily.Internet.Usage:  167.22"
```

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

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

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

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

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

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

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

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

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

```
## [1] "Area.Income:  61833.9"
```

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

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

```
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.

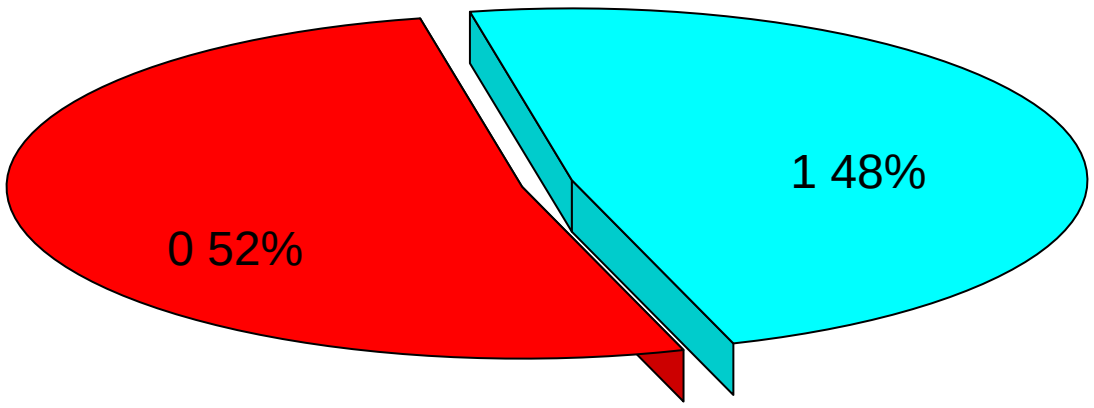
```
tabulate(df$Male)
```

```
## [1] 519 481
```

```
c(tabulate(df$Male))
```

```
## [1] 519 481
```

```
# Gender representation with pie chart

# import plotrix
library(plotrix)

# Pie Chart with Percentages
slices <- c(tabulate(df$Male))
lbls <- c(unique(df$Male))
pct <- round(slices/sum(slices)*100)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie3D(slices,labels = lbls, col=rainbow(length(lbls)), explode=0.1, radius =2, start = 140,
   main="Pie Chart for Gender representation")
```



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