Independent Project: Data Cleaning and EDA using R

Cynthia Mwadime

May 27, 2022

Question

A Kenyan enterpreneur who created an online cryptocurrency course would like to advertise on their blog. They need help identifying the most likely users to click on ads.

Success Metric

- finding the individual who are likely to click the ads

Context

knowing your clients who are likely to click on the ads are the most potential audience and customers and thus being able to know what they need and meeting their demands increases they satisfaction and are more likely to get the world out about the blog hence greating a larger audience

Experimental design

- Load the Libraries
- Read the data
- Cleaning
- EDA

```
#("devtools"); devtools::install_github("username/packagename")
```

```
library(tidyverse)
```

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

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

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

```
library(readr)
library(ROCR)
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 objects are masked from 'package:dplyr':
##
##     first, last
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

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

```
library(e1071)
```

```
##
## Attaching package: 'e1071'
```

```
## The following objects are masked from 'package:PerformanceAnalytics':
##
##     kurtosis, skewness
```

```
library(caret)
```

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

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(ggcorrplot)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(rpart)
library(caTools)
library(naivebayes)
```

```
## naivebayes 0.9.7 loaded
```

```
library(class)
library(ISLR)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-4
```

```
library(Hmisc)
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following object is masked from 'package:e1071':
##
##     impute

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(funModeling)
```

```
## funModeling v.1.9.4 :)
## Examples and tutorials at livebook.datascienceheroes.com
##  / Now in Spanish: librovivodecienciadedatos.ai
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(klaR)
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(DataExplorer)
library(ClustOfVar)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:funModeling':
##
##     range01
```

```
tinytex::install_tinytex(force =TRUE)
```

Loading the dataset

```
library(csvread)
url <- "http://bit.ly/IPAdvertisingData"
destfile <- "IPAdvertisingData.xls"
curl::curl_download(url, destfile)
IPAdvertisingData <- read_csv(destfile)
```

```
## Rows: 1000 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): Ad Topic Line, City, Country
## dbl  (6): Daily Time Spent on Site, Age, Area Income, Daily Internet Usage, ...
## dttm (1): Timestamp
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

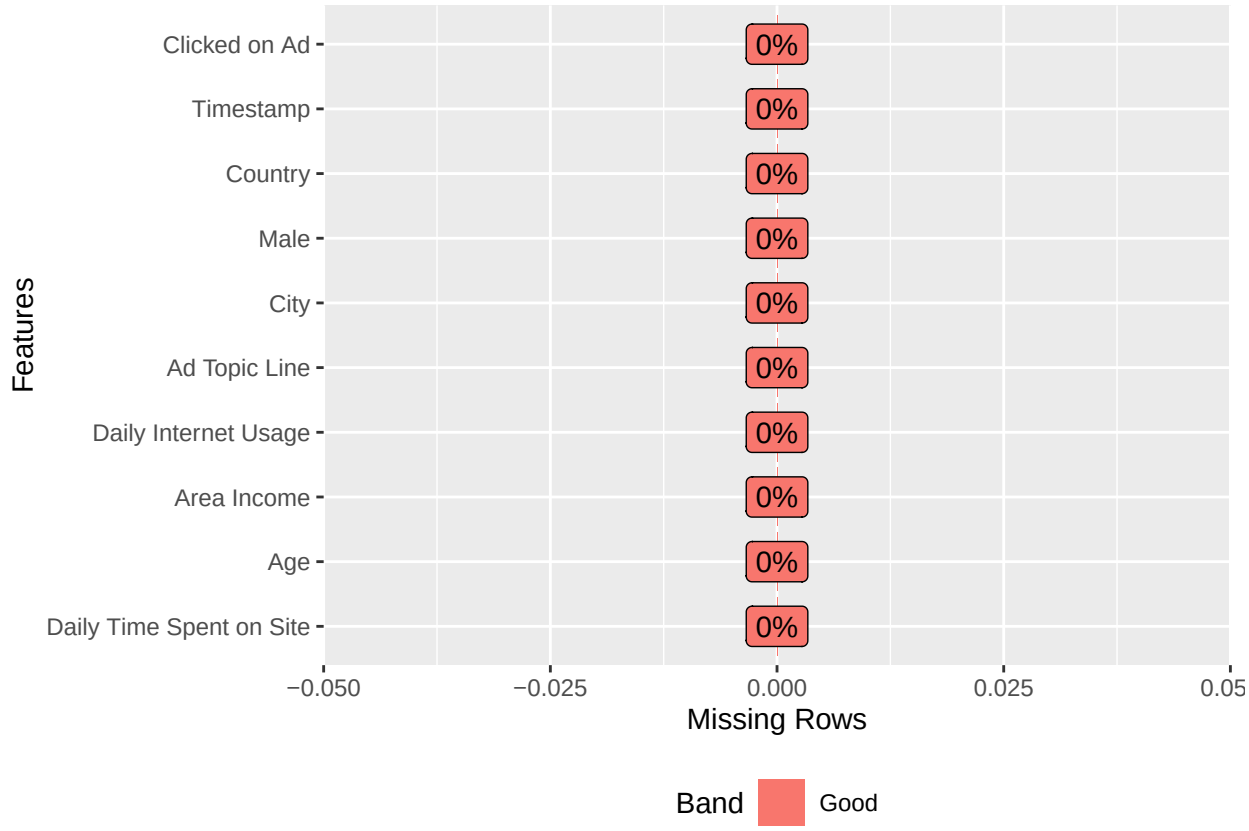
```
View(IPAdvertisingData)
```

```
# Removing duplicates from all columns
IPAdvertisingData = IPAdvertisingData[!duplicated(IPAdvertisingData), ]
```

```
# previewing the dataset
head(IPAdvertisingData)
```

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

```
# checking the percentage of missing values for all variables
plot_missing(IPAdvertisingData)
```



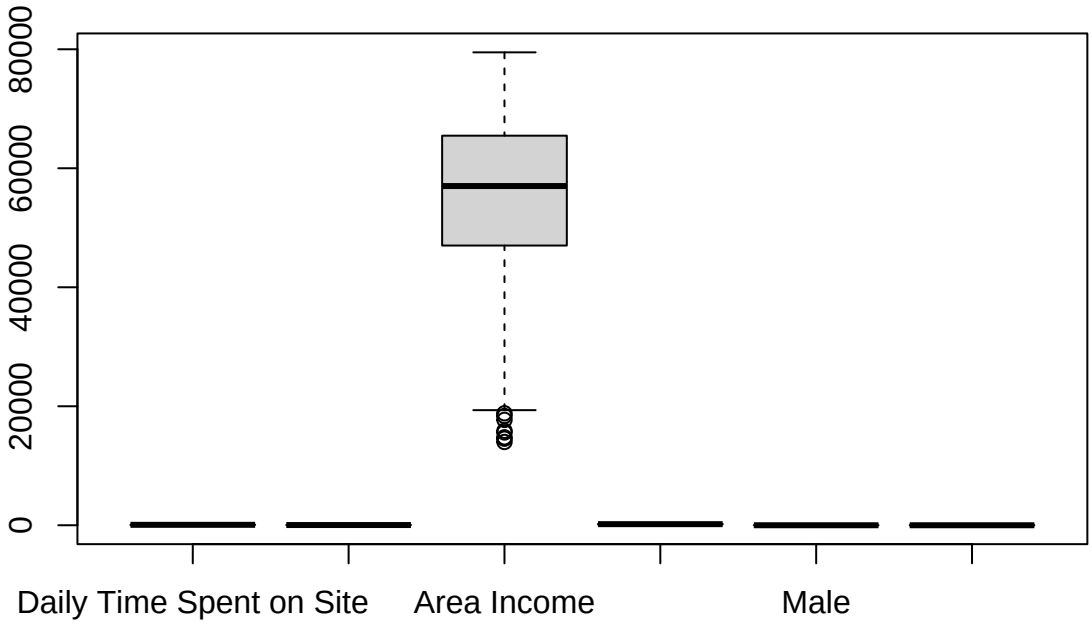
It seems there are no missing values

```r
# getting the summary statistics
summary(IPAdvertisingData)
```

```
##  Daily Time Spent on Site      Age          Area Income    Daily Internet Usage
##  Min.   :32.60            Min.   :19.00   Min.   :13996   Min.   :104.8
##  1st Qu.:51.36            1st Qu.:29.00   1st Qu.:47032   1st Qu.:138.8
##  Median :68.22            Median :35.00   Median :57012   Median :183.1
##  Mean   :65.00            Mean   :36.01   Mean   :55000   Mean   :180.0
##  3rd Qu.:78.55            3rd Qu.:42.00   3rd Qu.:65471   3rd Qu.:218.8
##  Max.   :91.43            Max.   :61.00   Max.   :79485   Max.   :270.0
##  Ad Topic Line          City                Male         Country
##  Length:1000        Length:1000        Min.   :0.000   Length:1000
##  Class :character   Class :character   1st Qu.:0.000   Class :character
##  Mode  :character   Mode  :character   Median :0.000   Mode  :character
##                                        Mean   :0.481
##                                        3rd Qu.:1.000
##                                        Max.   :1.000
##    Timestamp                       Clicked on Ad
##  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
```

The mean and medians of each feature are not far appart suggesting the data is normaly distributed and outliers are not altering

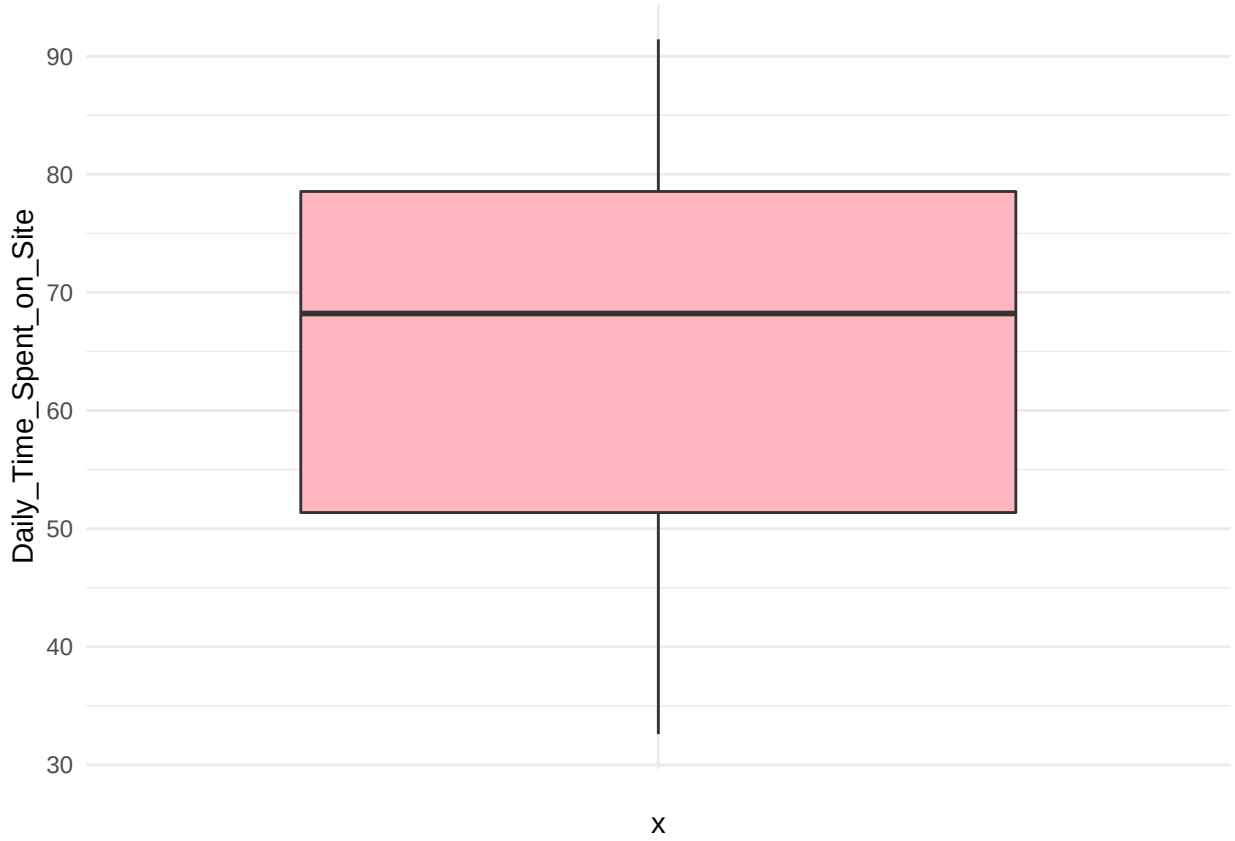
```r
# first attemplt at displaying outliers in lal
boxplot(IPAdvertisingData%>% select_if(is.numeric))
```




```
# remnaming the column names so it's easier to refference
names(IPAdvertisingData) <- gsub(" ", "_", names(IPAdvertisingData))
colnames(IPAdvertisingData)
```

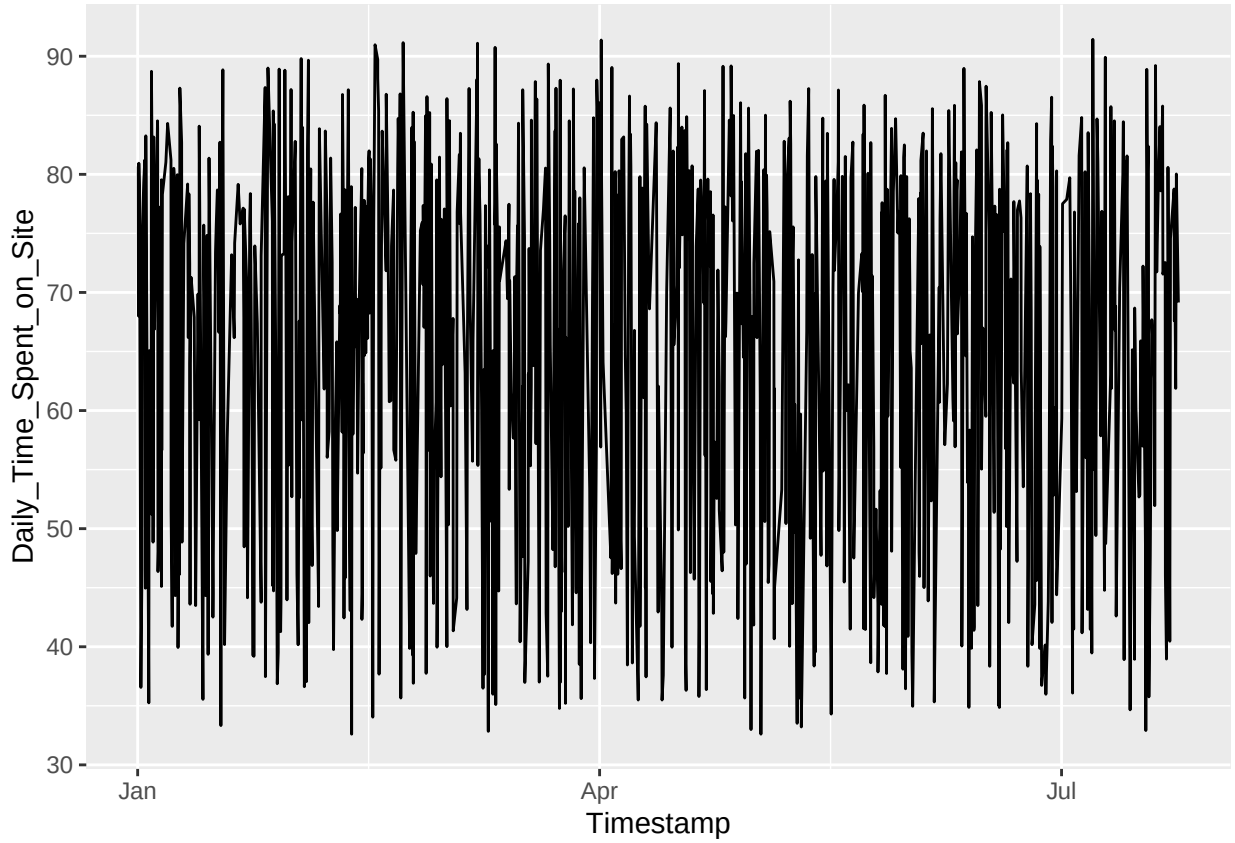
```
##  [1] "Daily_Time_Spent_on_Site" "Age"
##  [3] "Area_Income"              "Daily_Internet_Usage"
##  [5] "Ad_Topic_Line"            "City"
##  [7] "Male"                     "Country"
##  [9] "Timestamp"                "Clicked_on_Ad"
```

```
ggplot(IPAdvertisingData) +
  aes(x = "", y = Daily_Time_Spent_on_Site) +
  geom_boxplot(fill = "#FFB6C1") +
  theme_minimal()
```

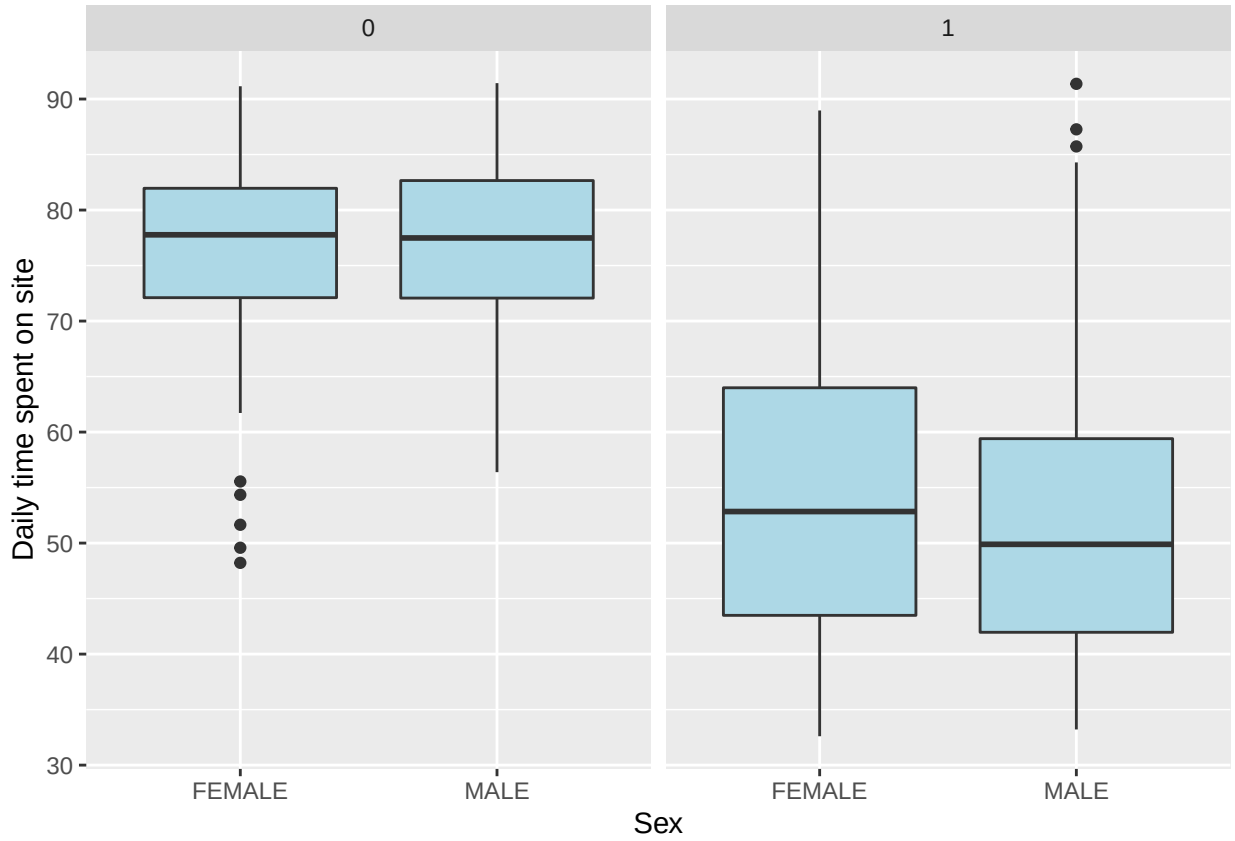



```
# Converting 0,1 to Female, Male so visualization's better
IPAdvertisingData <- IPAdvertisingData %>%
  mutate(Male = if_else(Male == 1, "MALE", "FEMALE"))
```

```
#creating a time series plot
p <- ggplot(IPAdvertisingData, aes(x=Timestamp, y=Daily_Time_Spent_on_Site)) +
       geom_line()

#display time series plot
p
```

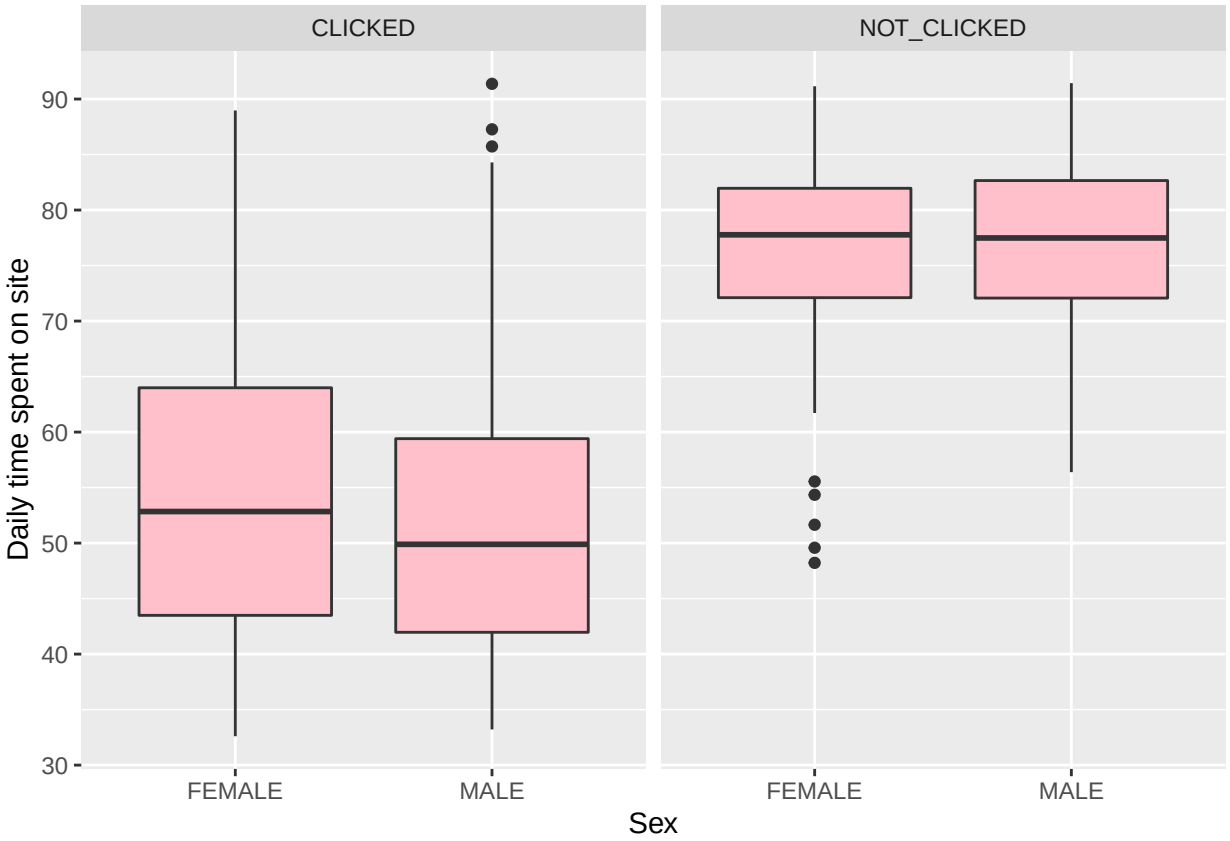



```
# Daily time pent on the site comparison by gender
IPAdvertisingData %>%
  ggplot(aes(x=Male,y=Daily_Time_Spent_on_Site))+
  geom_boxplot(fill='lightblue')+
  xlab("Sex")+
  ylab("Daily time spent on site")+
  facet_grid(~Clicked_on_Ad)
```




```
# Converting 0,1 to Female, Male so visualization's better
IPAdvertisingData <- IPAdvertisingData %>%
  mutate(Clicked_on_Ad = if_else(Clicked_on_Ad == 1, "CLICKED", "NOT_CLICKED"))
```

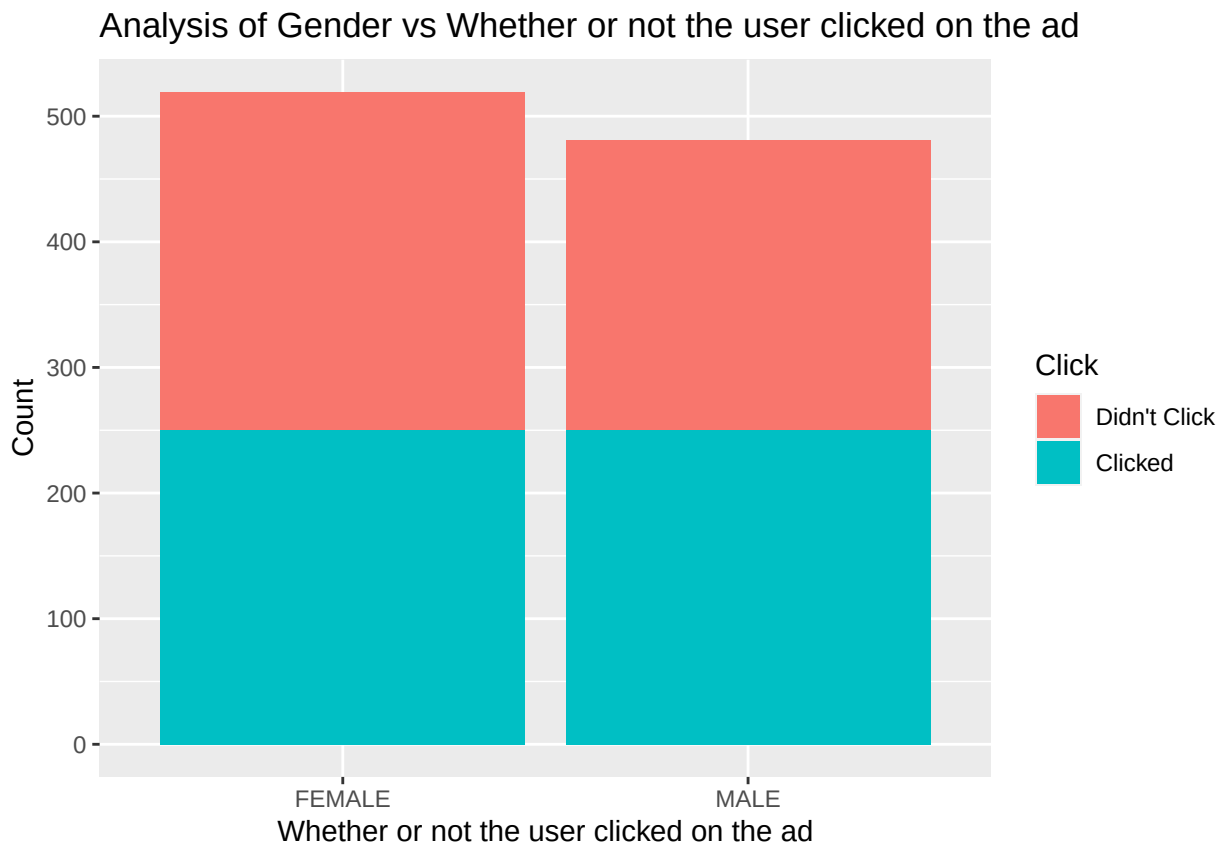
```
# Daily time pent on the site comparison by gender and age
IPAdvertisingData %>%

  ggplot(aes(x=Male,y=Daily_Time_Spent_on_Site, group=Male))+
  geom_boxplot(fill='pink')+
  xlab("Sex")+
  ylab("Daily time spent on site")+
  facet_grid(~Clicked_on_Ad)
```




```
# Bar plot for target (Whether or not the user clicked on the ad)
ggplot(IPAdvertisingData, aes(x=IPAdvertisingData$Male, fill=IPAdvertisingData$Clicked_on_Ad)) +
  geom_bar() +
  xlab("Whether or not the user clicked on the ad") +
  ylab("Count") +
  ggtitle("Analysis of Gender vs Whether or not the user clicked on the ad") +
  scale_fill_discrete(name = "Click", labels = c("Didn't Click", "Clicked"))
```

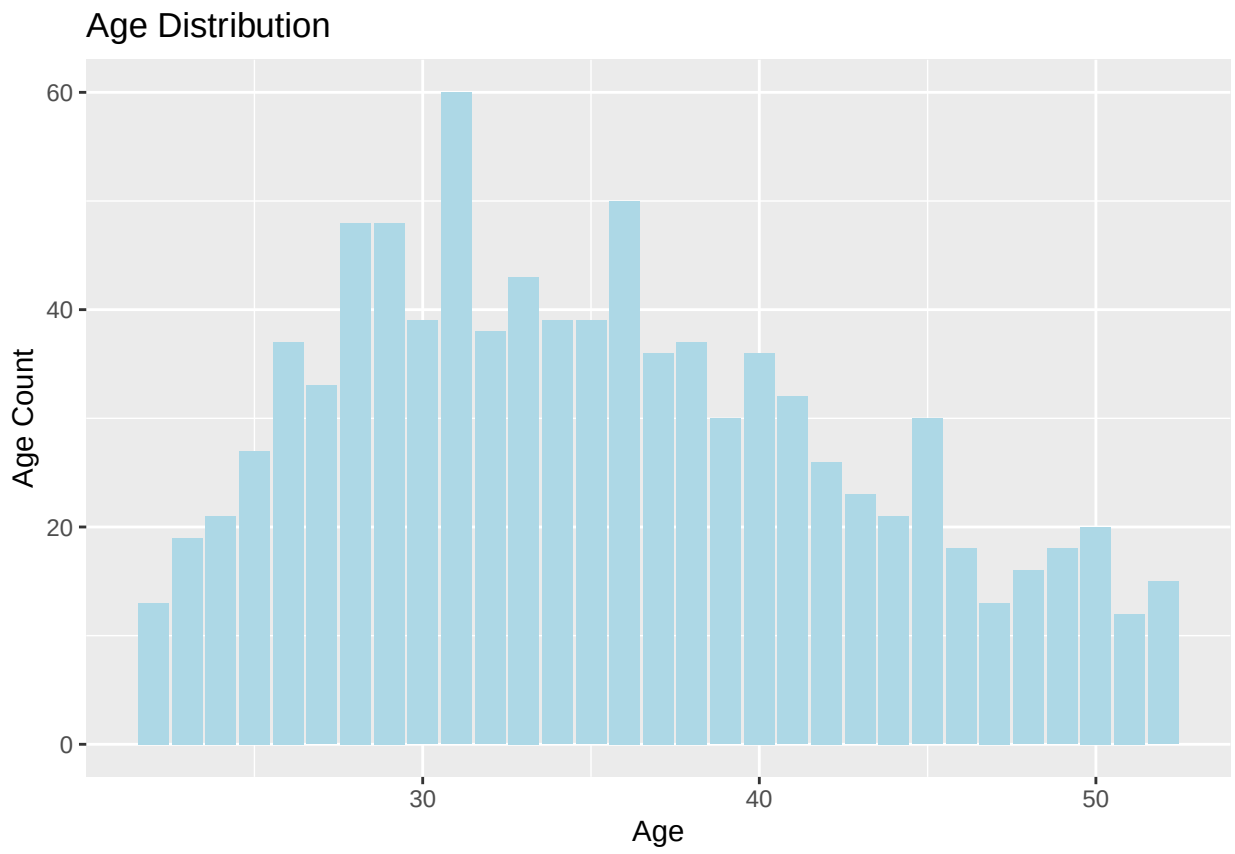
```
## Warning: Use of ‘IPAdvertisingData$Male‘ is discouraged. Use ‘Male‘ instead.
```

```
## Warning: Use of ‘IPAdvertisingData$Clicked_on_Ad‘ is discouraged. Use
## ‘Clicked_on_Ad‘ instead.
```

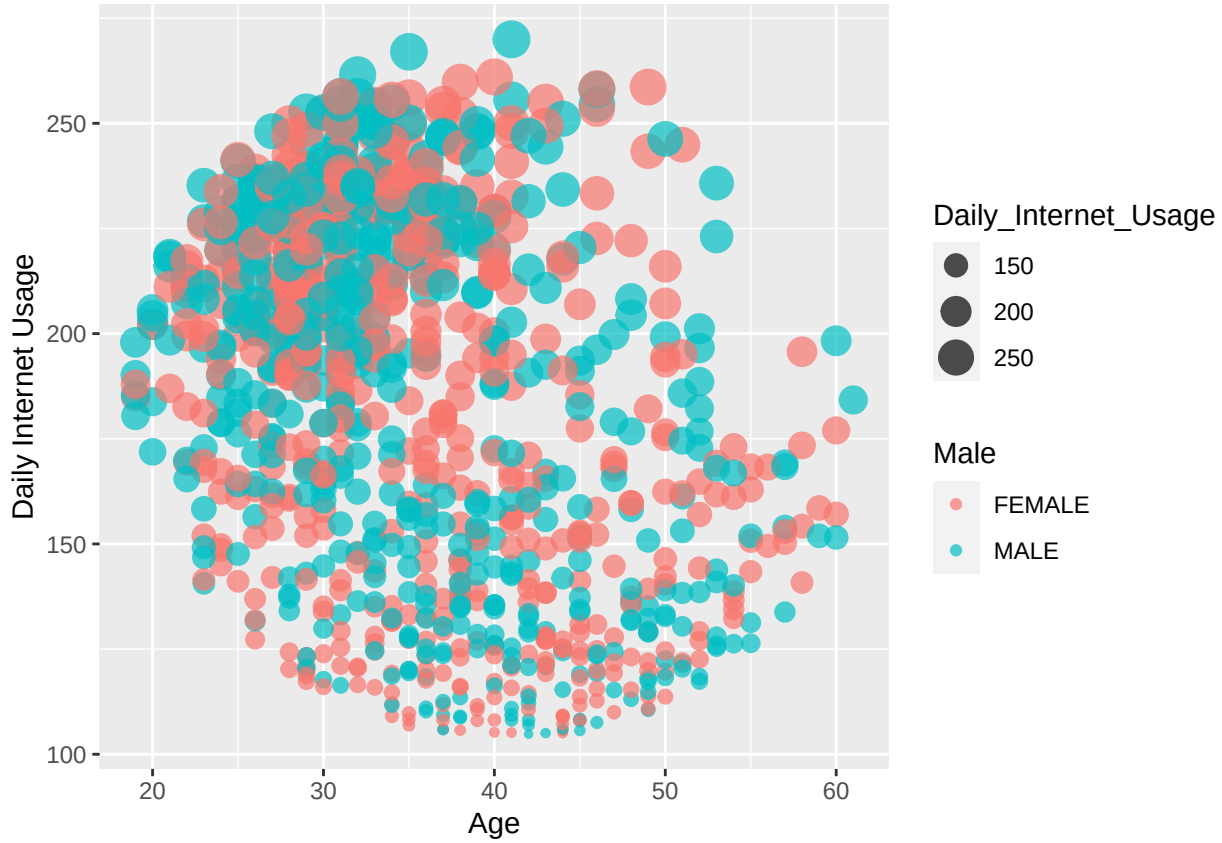


here is no imbalance issue in the target variable.

```
# Counting the age distribution
IPAdvertisingData %>%
  group_by(Age) %>%
  count() %>%
  filter(n > 10) %>%
  ggplot()+
  geom_col(aes(Age, n), fill = "lightblue")+
  ggtitle("Age Distribution") +
  xlab("Age")  +
  ylab("Age Count")
```




```
# bivariate analsis on Age, Gender and Daily internet Usage
IPAdvertisingData %>%
  ggplot(aes(x=Age,y=Daily_Internet_Usage,color=Male, size=Daily_Internet_Usage))+
  geom_point(alpha=0.7)+xlab("Age") +
  ylab("Daily Internet Usage")+
  guides(fill = guide_legend(title = "Gender"))
```

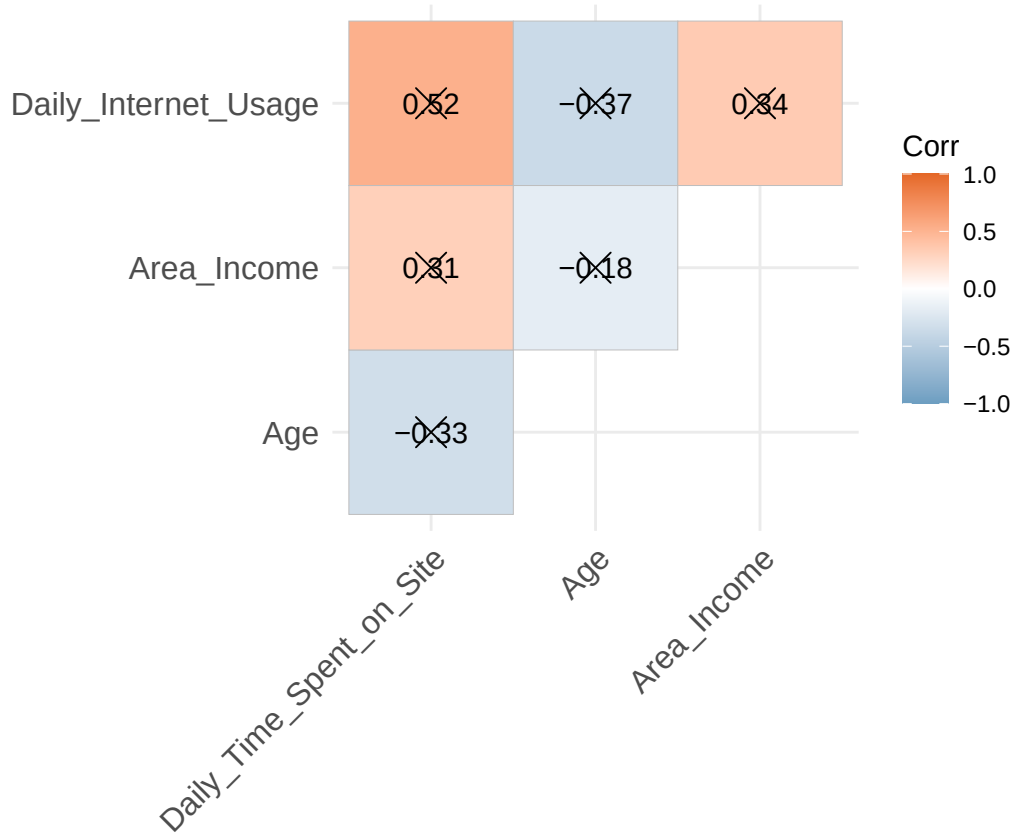



```
corr <- cor(IPAdvertisingData%>% select_if(is.numeric))
corr
```

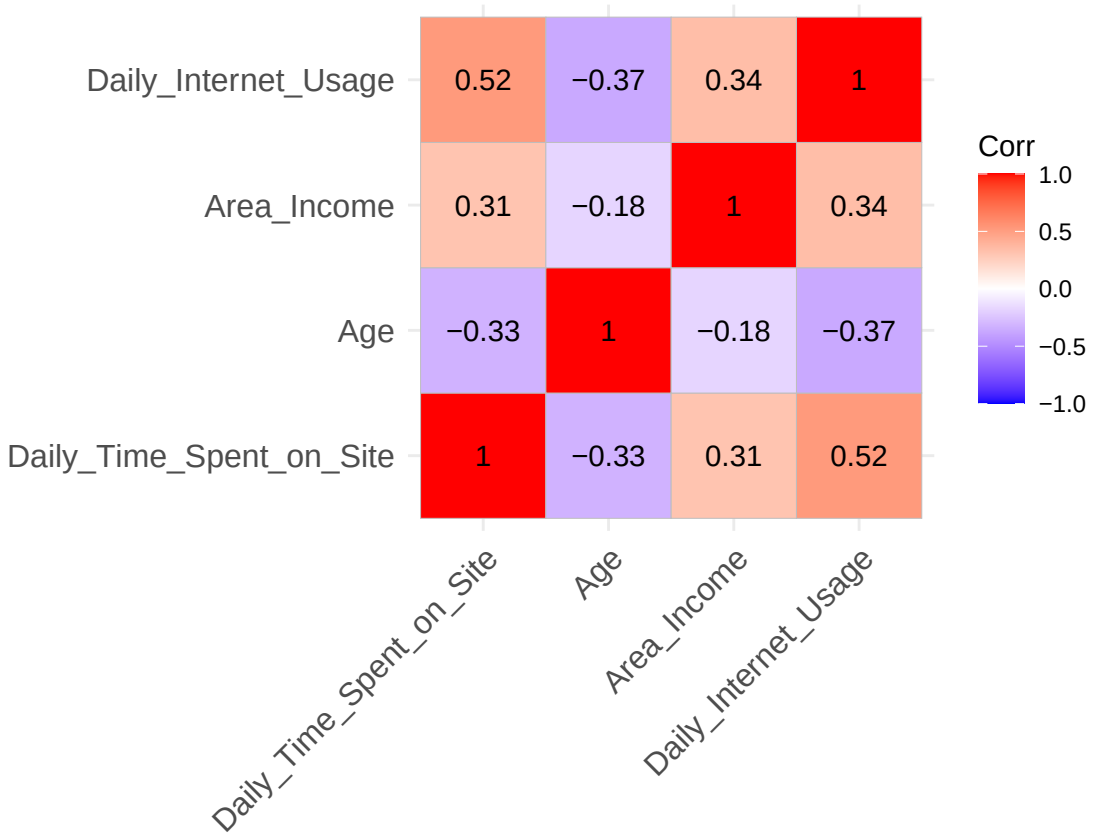
```
##                           Daily_Time_Spent_on_Site        Age Area_Income
## Daily_Time_Spent_on_Site                1.0000000 -0.3315133   0.3109544
## Age                                    -0.3315133  1.0000000  -0.1826050
## Area_Income                             0.3109544 -0.1826050   1.0000000
## Daily_Internet_Usage                    0.5186585 -0.3672086   0.3374955
##                           Daily_Internet_Usage
## Daily_Time_Spent_on_Site            0.5186585
## Age                                -0.3672086
## Area_Income                         0.3374955
## Daily_Internet_Usage                1.0000000
```

```
#corrplot(corr, method = "ellipse", type="upper",)
```

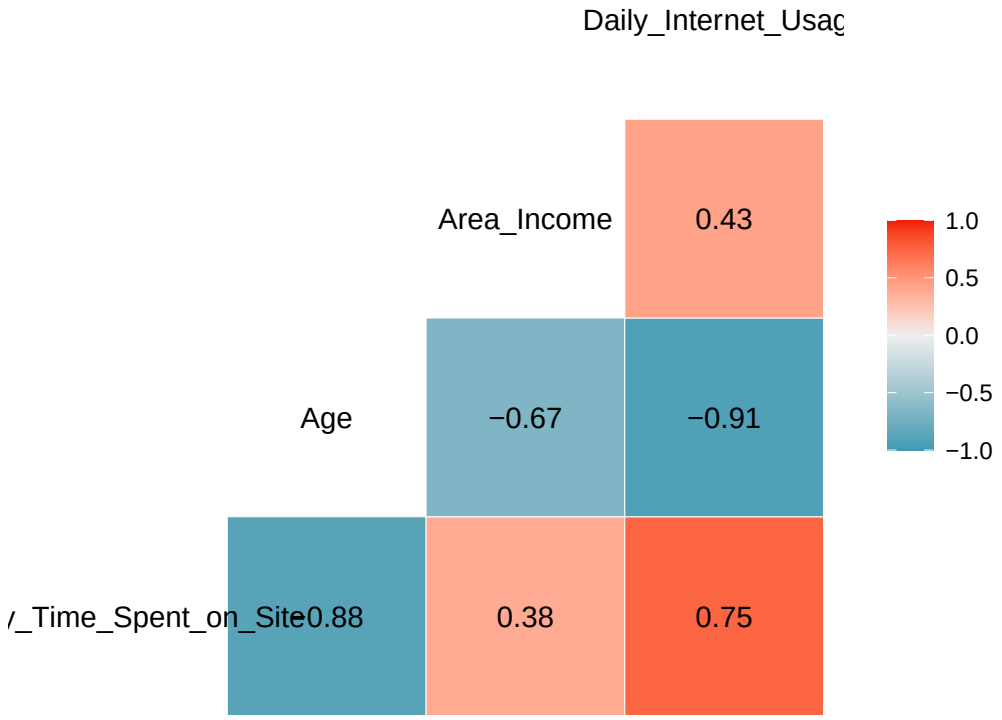
```
p.mat <- cor_pmat(corr, method = "spearman")

ggcorrplot(corr, method = "square", type = "upper",
           colors = c("#6D9EC1", "white", "#E46726"),
           lab = TRUE, p.mat=p.mat, sig.level = .05)
```




```
ggcorrplot(corr,lab = T)
```

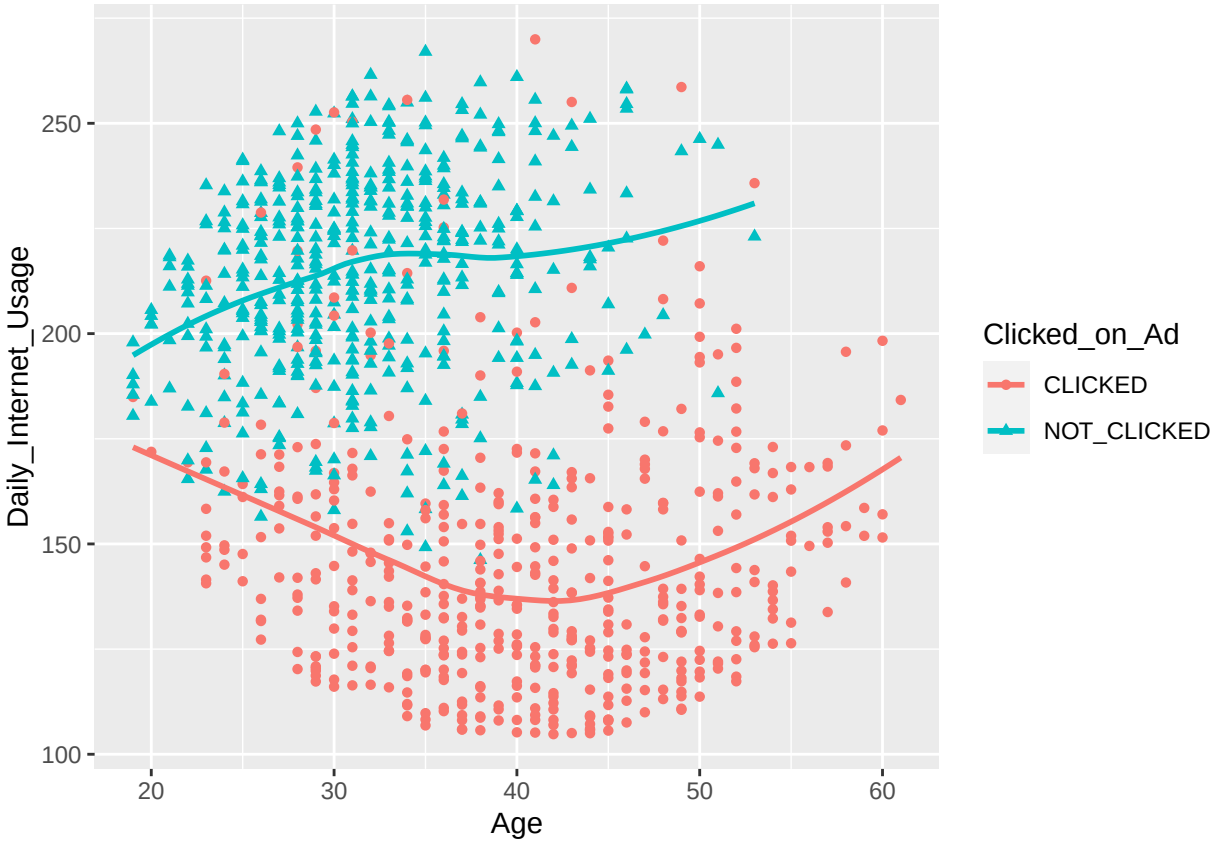



```
ggcorr(corr, label = T, label_round = 2)
```



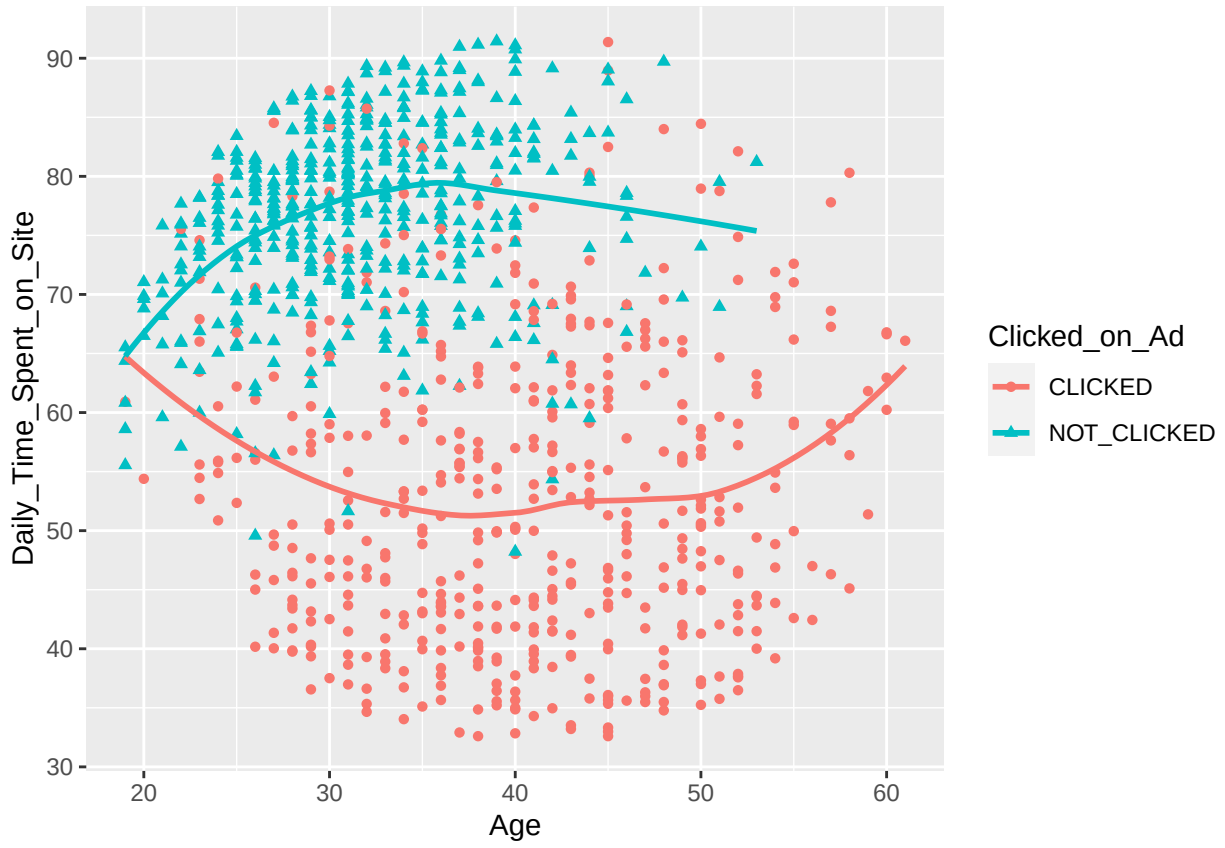

```
ggplot(IPAdvertisingData, aes(x = Age, y = Daily_Internet_Usage, color = Clicked_on_Ad, shape = Clicked
  geom_point()+
  geom_smooth(se = FALSE);
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(IPAdvertisingData, aes(x = Age, y = Daily_Time_Spent_on_Site, color = Clicked_on_Ad, shape = Cli
  geom_point()+
  geom_smooth(se = FALSE);
```

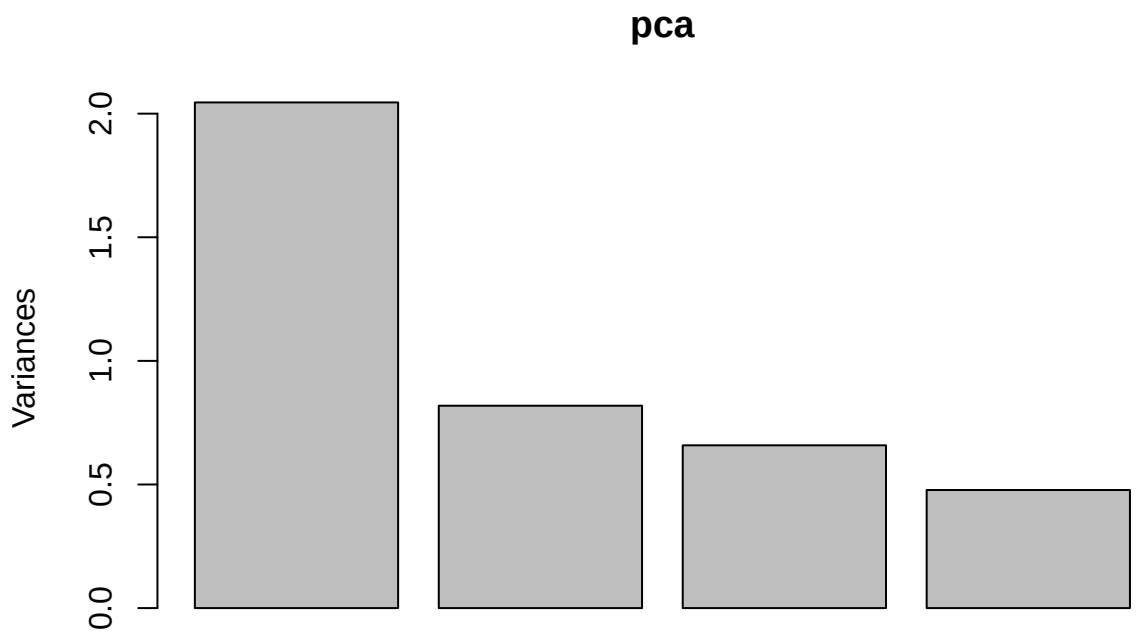
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



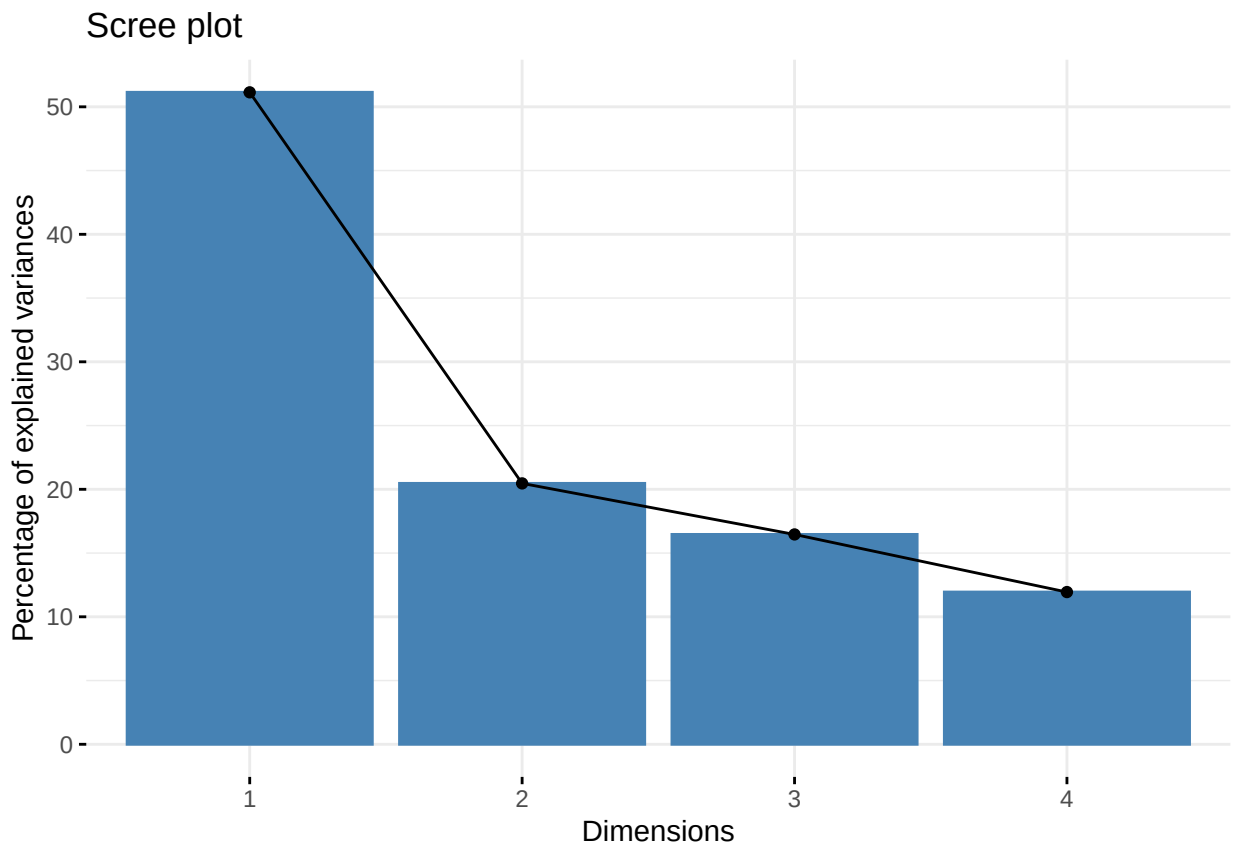

```
# performing principle component analysis
pca <- prcomp(IPAdvertisingData%>% select_if(is.numeric), scale = TRUE) # prcomp temel bileşen fonksiyo
pca
```

```
## Standard deviations (1, .., p=4):
## [1] 1.4301799 0.9047446 0.8114198 0.6911009
##
## Rotation (n x k) = (4 x 4):
##                                 PC1         PC2        PC3         PC4
## Daily_Time_Spent_on_Site -0.5484092  0.02789664 -0.5290308 -0.64698960
## Age                       0.4466724 -0.64437870 -0.6133591  0.09513391
## Area_Income              -0.4238013 -0.76334793  0.4827160 -0.06839365
## Daily_Internet_Usage     -0.5657947  0.03602522 -0.3330199  0.75344297
```

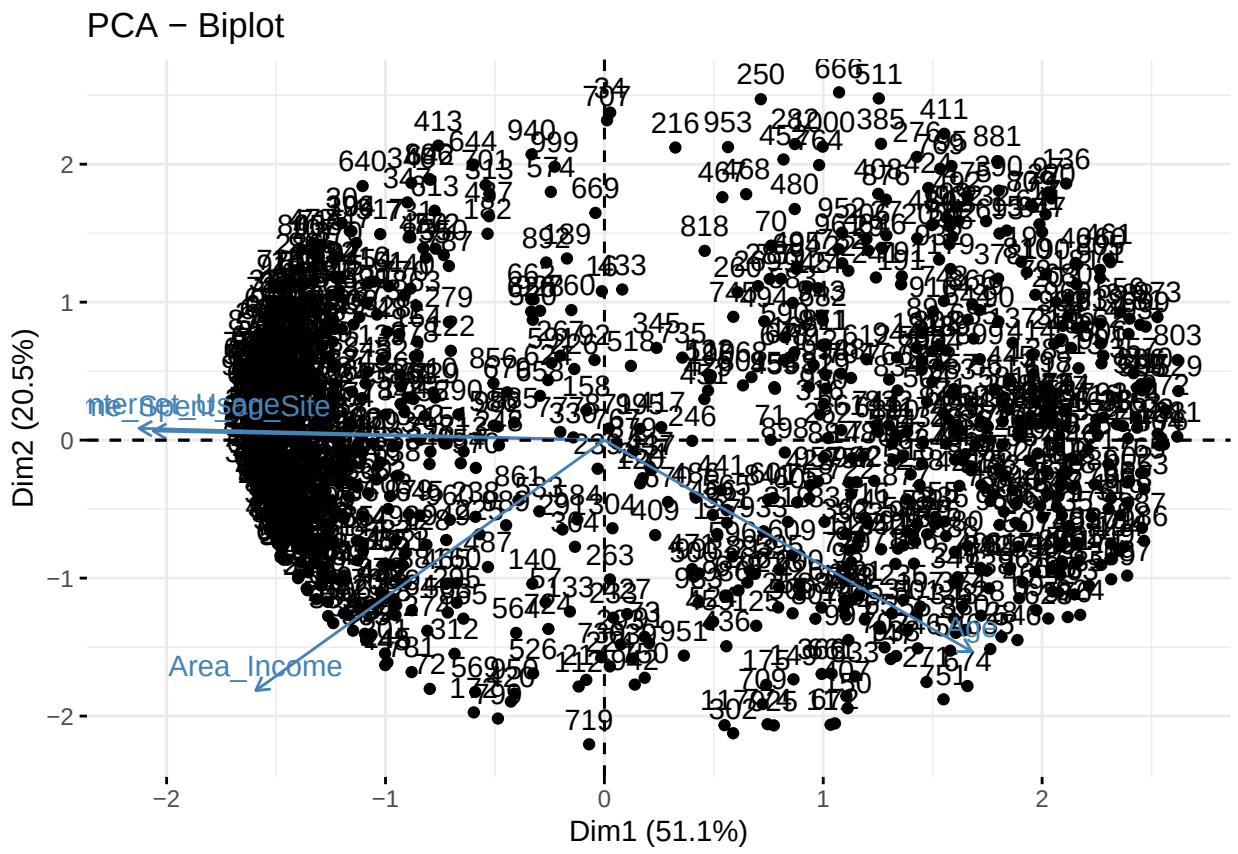
```
screeplot(pca)
```




```
fviz_screeplot(pca)
```


Scree plot
Percentage of explained variances
50
40
30
20
10
0
1
2
3
4
Dimensions

```
fviz_pca(pca)
```




```
# component variance
pca$sdev^2
```

```
## [1] 2.0454146 0.8185628 0.6584021 0.4776205
```

```
# component variance
pca$rotation <- -pca$rotation

pca
```

```
## Standard deviations (1, .., p=4):
## [1] 1.4301799 0.9047446 0.8114198 0.6911009
##
## Rotation (n x k) = (4 x 4):
##                                PC1         PC2        PC3         PC4
## Daily_Time_Spent_on_Site  0.5484092 -0.02789664  0.5290308  0.64698960
## Age                      -0.4466724  0.64437870  0.6133591 -0.09513391
## Area_Income               0.4238013  0.76334793 -0.4827160  0.06839365
## Daily_Internet_Usage      0.5657947 -0.03602522  0.3330199 -0.75344297
```

```
# component variance
pca$x <- -pca$x

head(pca$x)
```

```
##           PC1        PC2        PC3        PC4
## [1,] 1.384445  0.2454840  0.3926256 -1.0988790
## [2,] 1.383618  0.3594124 -0.2207304  0.5079874
## [3,] 1.542840 -0.5160000 -0.2932571 -0.6544435
## [4,] 1.515897 -0.5952736  0.3227773 -0.6824752
## [5,] 1.352063  0.9575815 -0.2919913 -0.5374753
## [6,] 1.240879 -0.7127327 -0.8922023 -0.8414644
```

```
# component variance
fviz_pca(pca)
```