OT_IP_Supervised_Learning_R_Part1.R

telly

2022-03-27

```r
# 1. Statement of the Problem
#A Kenyan entrepreneur has created an online cryptography course and would want to advertise it on her
#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.


#Metric for Success

#Build a model with at least 95% accuracy in classifying whether an advert will
#be clicked or not


#Experimental Design
#1. Data Cleaning
#2. Data Exploration
#3. Recommendations & Conclusions

#Downloading the relevant Packages

#install.packages("Hmisc")

#install.packages("ggthemes")

#install.packages("moments")

#install.packages("corrplot")

#install.packages("DataExplorer")

#install.packages("caTools")
#install.packages("caret")
#install.packages("rsample")
#install.packages("tidymodels")
#install.packages("kernlab")
#install.packages("GGally")

#Loading the relevant libraries

library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.0.5
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tibble' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'purrr' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

## Warning: package 'stringr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```
library(ggplot2)

library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.0.5

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'

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

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

```
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 4.0.5
```

```
library(moments)

library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(DataExplorer)
```

```
## Warning: package 'DataExplorer' was built under R version 4.0.5
```

```
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.0.5
```

```
library(tidymodels)
```

```
## Warning: package 'tidymodels' was built under R version 4.0.5

## -- Attaching packages -------------------------------------- tidymodels 0.2.0 --

## v broom        0.7.12     v rsample      0.1.1
## v dials        0.1.0      v tune         0.2.0
## v infer        1.0.0      v workflows    0.2.6
## v modeldata    0.1.1      v workflowsets 0.2.1
## v parsnip      0.2.1      v yardstick    0.0.9
## v recipes      0.2.0

## Warning: package 'broom' was built under R version 4.0.5

## Warning: package 'dials' was built under R version 4.0.5

## Warning: package 'scales' was built under R version 4.0.5
```

```
## Warning: package 'infer' was built under R version 4.0.5

## Warning: package 'modeldata' was built under R version 4.0.5

## Warning: package 'parsnip' was built under R version 4.0.5

## Warning: package 'recipes' was built under R version 4.0.5

## Warning: package 'rsample' was built under R version 4.0.5

## Warning: package 'tune' was built under R version 4.0.5

## Warning: package 'workflows' was built under R version 4.0.5

## Warning: package 'workflowsets' was built under R version 4.0.5

## Warning: package 'yardstick' was built under R version 4.0.5

## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x dplyr::between()     masks data.table::between()
## x scales::discard()    masks purrr::discard()
## x dplyr::filter()      masks stats::filter()
## x dplyr::first()       masks data.table::first()
## x recipes::fixed()     masks stringr::fixed()
## x dplyr::lag()         masks stats::lag()
## x dplyr::last()        masks data.table::last()
## x yardstick::spec()    masks readr::spec()
## x Hmisc::src()         masks dplyr::src()
## x recipes::step()      masks stats::step()
## x Hmisc::summarize()   masks dplyr::summarize()
## x parsnip::translate() masks Hmisc::translate()
## x purrr::transpose()   masks data.table::transpose()
## * Dig deeper into tidy modeling with R at https://www.tmwr.org
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.5

##
## Attaching package: 'caret'

## The following objects are masked from 'package:yardstick':
##
##     precision, recall, sensitivity, specificity

## The following object is masked from 'package:survival':
##
##     cluster

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

```r
library(class)
library(rsample)
library(kernlab)
```

```
##
## Attaching package: 'kernlab'

## The following object is masked from 'package:scales':
##
##     alpha

## The following object is masked from 'package:purrr':
##
##     cross

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

```r
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.0.5

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

```r
#Loading the Dataset

advert <- fread('http://bit.ly/IPAdvertisingData')

#Data Exploration

#Checking the first 6 rows
head(advert)
```

```
##    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    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
##               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
```

```
#Checking the last 6 rows

tail(advert)
```

```
##    Daily Time Spent on Site Age Area Income Daily Internet Usage
## 1:                    43.70  28    63126.96               173.01
## 2:                    72.97  30    71384.57               208.58
## 3:                    51.30  45    67782.17               134.42
## 4:                    51.63  51    42415.72               120.37
## 5:                    55.55  19    41920.79               187.95
## 6:                    45.01  26    29875.80               178.35
##                            Ad Topic Line          City Male
## 1:        Front-line bifurcated ability  Nicholasland    0
## 2:        Fundamental modular algorithm     Duffystad    1
## 3:      Grass-roots cohesive monitoring   New Darlene    1
## 4:         Expanded intangible solution South Jessica    1
## 5: Proactive bandwidth-monitored policy   West Steven    0
## 6:      Virtual 5thgeneration emulation   Ronniemouth    0
##                   Country           Timestamp Clicked on Ad
## 1:                Mayotte 2016-04-04 03:57:48             1
## 2:                Lebanon 2016-02-11 21:49:00             1
## 3: Bosnia and Herzegovina 2016-04-22 02:07:01             1
## 4:               Mongolia 2016-02-01 17:24:57             1
## 5:              Guatemala 2016-03-24 02:35:54             0
## 6:                 Brazil 2016-06-03 21:43:21             1
```

```
#Data Structure
str(advert)
```

```
## Classes 'data.table' and '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 orchestration" "Monitored national standardi
##  $ City                    : chr  "Wrightburgh" "West Jodi" "Davidton" "West Terrifurt" ...
##  $ Male                    : int  0 1 0 1 0 1 0 1 1 1 ...
##  $ Country                 : chr  "Tunisia" "Nauru" "San Marino" "Italy" ...
##  $ 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 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
#Dimension of Dataset
dim(advert)
```

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

```r
#We have 1000 rows and 10 columns in the dataset

#Checking the Data Types of the columns
sapply(advert, class)
```

```
## $`Daily Time Spent on Site`
## [1] "numeric"
##
## $Age
## [1] "integer"
##
## $`Area Income`
## [1] "numeric"
##
## $`Daily Internet Usage`
## [1] "numeric"
##
## $`Ad Topic Line`
## [1] "character"
##
## $City
## [1] "character"
##
## $Male
## [1] "integer"
##
## $Country
## [1] "character"
##
## $Timestamp
## [1] "POSIXct" "POSIXt"
##
## $`Clicked on Ad`
## [1] "integer"
```

```r
#3. Data Cleaning

# Standardize column names by using upper case and replacing the
#spaces with underscores using gsub() function

names(advert) <- gsub(" ","_", names(advert))

# lower the case of the column names using toupper() function
names(advert) <- toupper(names(advert))

# Confirming the changes
colnames(advert)
```

```
##  [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"
```

```r
#Checking for Missing Data in columns using the colSums & is.na

colSums(is.na(advert))
```

```
## 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
```

```r
#There are no missing entries in the dataset

#Checking for Duplicates in the Dataset
anyDuplicated((advert))
```

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

```r
#There are no duplicated records in the Dataset

#Renaming the Columns to make them precise
names(advert)[1] <- "BROWSE_TIME"

names(advert)[4] <- "NET_USAGE"

names(advert)[10] <- "CLICKS"

names(advert)[5]  <- "TOPIC"

names(advert)[3]  <- "INCOME"
names(advert)[7]  <- 'GENDER'

#Preview Dataset
head(advert, 3)
```

```
##    BROWSE_TIME AGE   INCOME NET_USAGE                              TOPIC
## 1:       68.95  35 61833.90    256.09 Cloned 5thgeneration orchestration
## 2:       80.23  31 68441.85    193.77 Monitored national standardization
## 3:       69.47  26 59785.94    236.50   Organic bottom-line service-desk
##           CITY GENDER    COUNTRY           TIMESTAMP CLICKS
## 1: Wrightburgh      0    Tunisia 2016-03-27 00:53:11      0
## 2:   West Jodi      1      Nauru 2016-04-04 01:39:02      0
## 3:    Davidton      0 San Marino 2016-03-13 20:35:42      0
```

```r
#Checking for Unique Values in the Gender Column to ensure
#alignment with expectations

distinct(select(advert, GENDER ))
```

```
##    GENDER
## 1:      0
## 2:      1
```

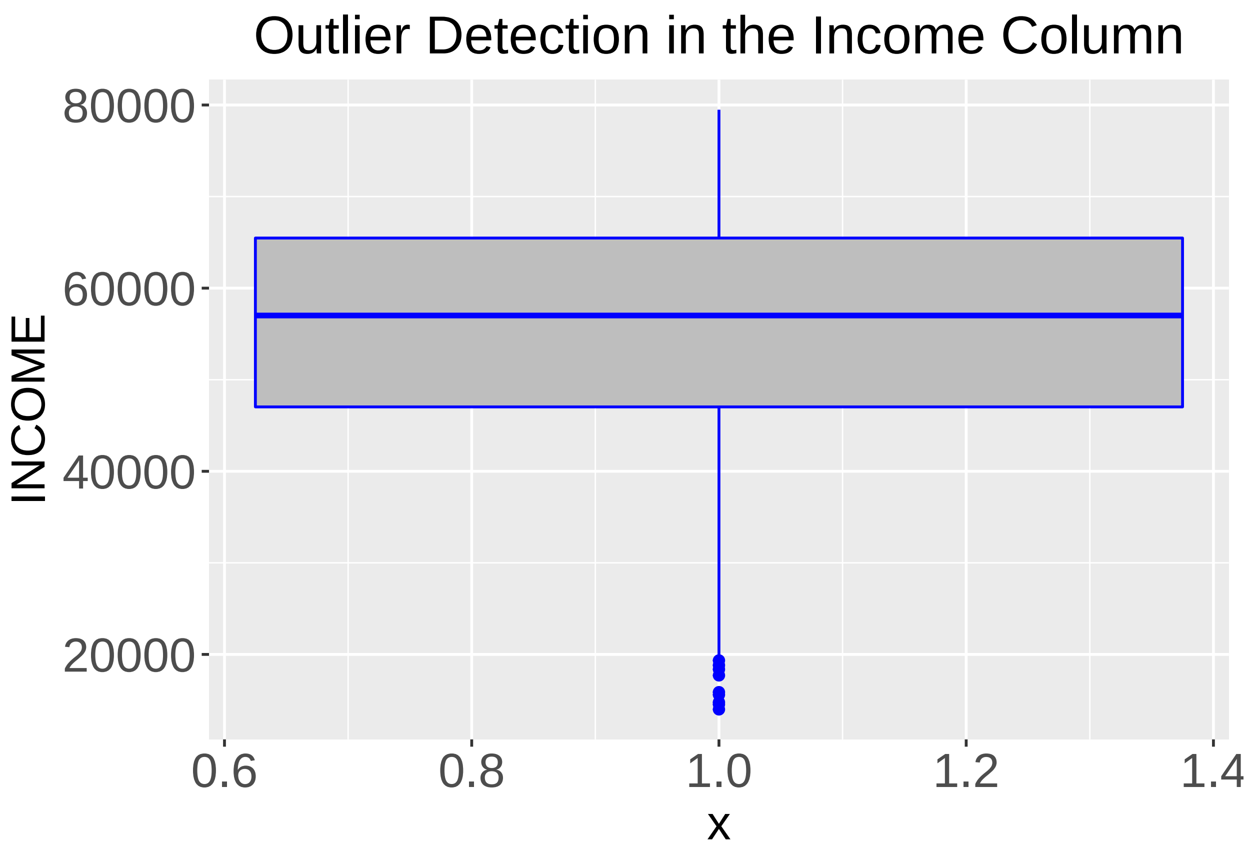
```r
#Gender column consists of expected values 0 & 1

#Checking for unique values in the Number of Clicks per Ad
distinct(select(advert, CLICKS))
```

```
##    CLICKS
## 1:      0
## 2:      1
```

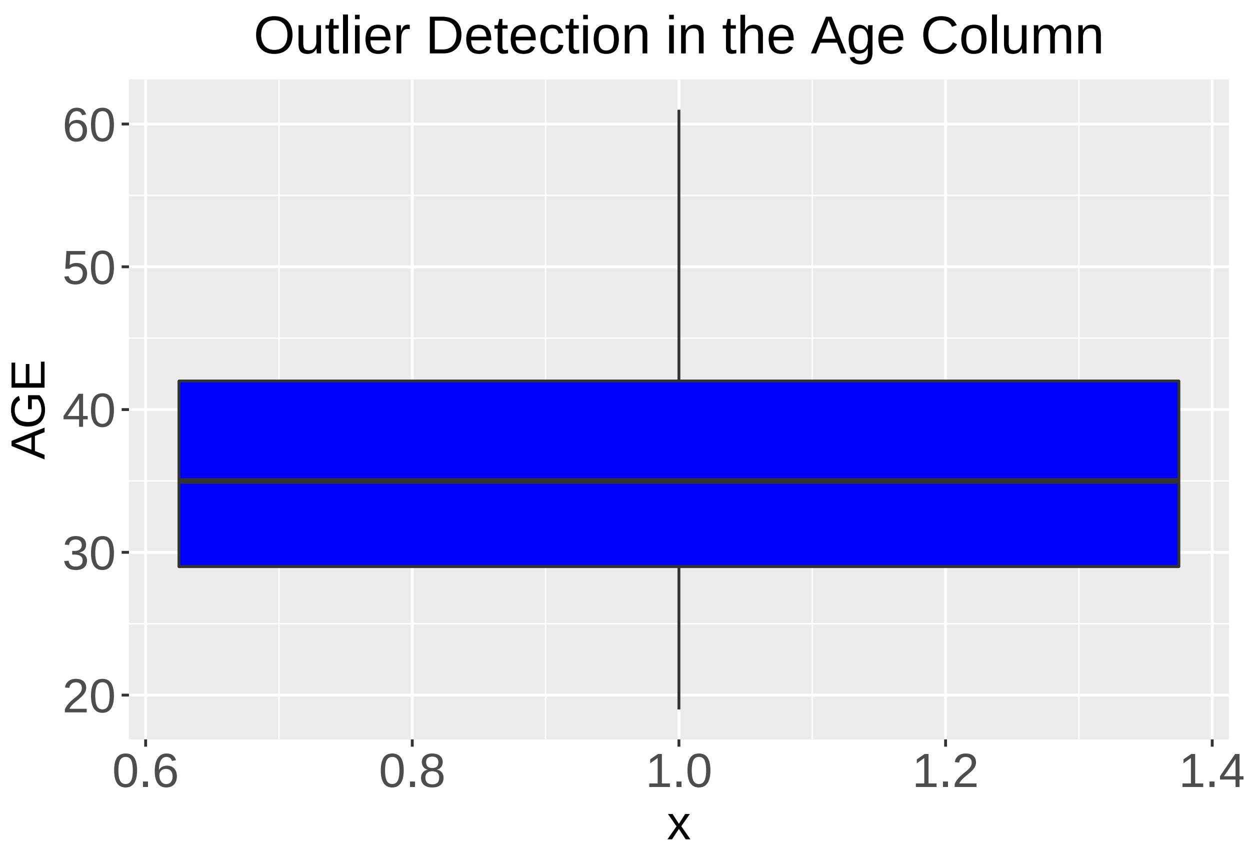
```r
#Clicks column has expected values of 0 for NO and 1 for Yes

#Gender and Clicks are erroneously classed as integers
#They are categorical features. Therefore we convert them
#to factors

advert$GENDER <- factor(advert$GENDER)

advert$CLICKS <- factor(advert$CLICKS)

#Checking Structure of Data
str(advert)
```

```
## Classes 'data.table' and 'data.frame':   1000 obs. of  10 variables:
##  $ BROWSE_TIME: num  69 80.2 69.5 74.2 68.4 ...
##  $ AGE        : int  35 31 26 29 35 23 33 48 30 20 ...
##  $ INCOME     : num  61834 68442 59786 54806 73890 ...
##  $ NET_USAGE  : num  256 194 236 246 226 ...
##  $ TOPIC      : chr  "Cloned 5thgeneration orchestration" "Monitored national standardization" "Orga
##  $ CITY       : chr  "Wrightburgh" "West Jodi" "Davidton" "West Terrifurt" ...
##  $ GENDER     : Factor w/ 2 levels "0","1": 1 2 1 2 1 2 1 2 2 2 ...
##  $ COUNTRY    : chr  "Tunisia" "Nauru" "San Marino" "Italy" ...
##  $ TIMESTAMP  : POSIXct, format: "2016-03-27 00:53:11" "2016-04-04 01:39:02" ...
##  $ CLICKS     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

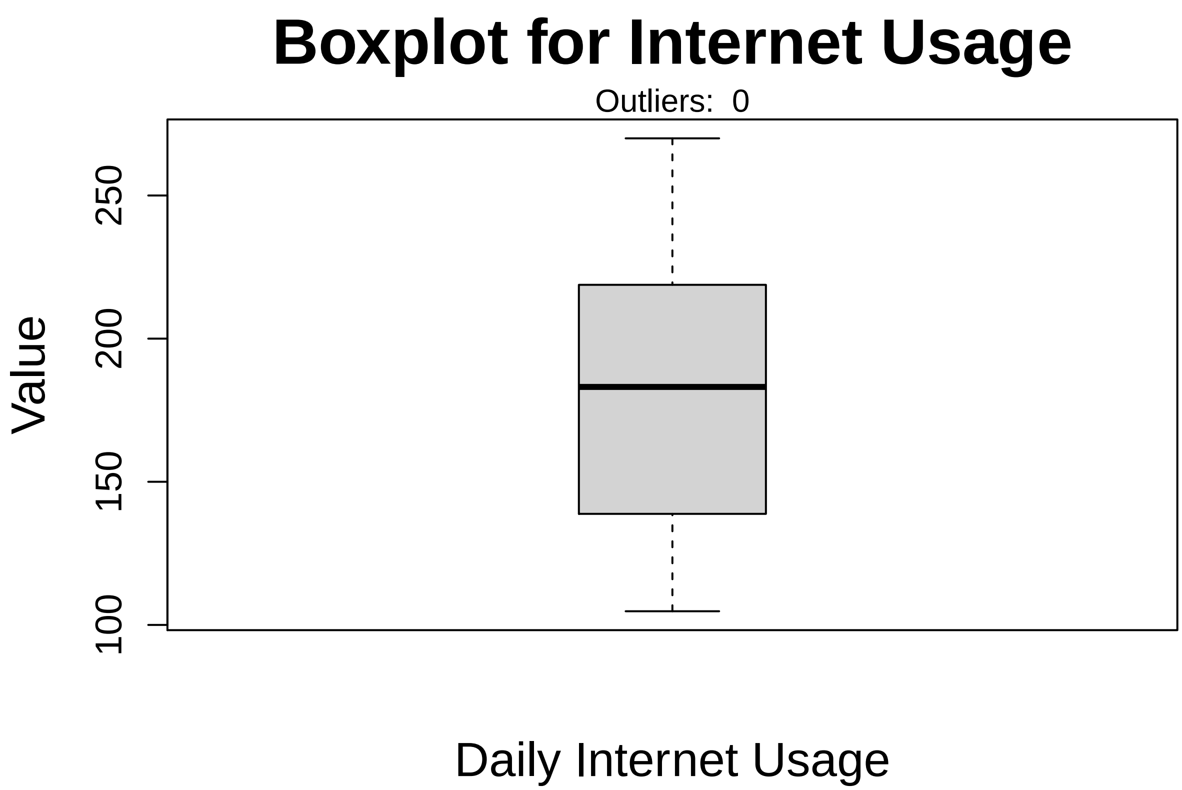
```r
#Outlier Detection
#Checking for Outliers in the Income Column

advert %>%
  ggplot(aes(x= 1, y=INCOME)) +
  geom_boxplot(fill = "grey", color= 'blue') +
  ggtitle("Outlier Detection in the Income Column") +
  theme(axis.text = element_text(size=18),
        axis.title = element_text(size = 18),
        plot.title = element_text(hjust = 0.5, size = 20))
```




```
#We have about 8 outliers in the dataset that represent actual
#income levels of individuals. We will not drop them from the
#dataset as they are actual datapoints.
#Checking for Outliers in the Age Column
advert %>%
  ggplot(aes(x= 1, y=AGE)) +
  geom_boxplot(fill= 'blue') +
  ggtitle("Outlier Detection in the Age Column") +
  theme(axis.text = element_text(size=18),
        axis.title = element_text(size = 18),
        plot.title = element_text(hjust = 0.5, size = 20))
```




```
# plot a boxplot to check for outliers in the 'Net_Usage' column
boxplot(advert$NET_USAGE, main="Boxplot for Internet Usage",
        xlab = "Daily Internet Usage", ylab = "Value", boxwex=0.4, cex.main=2,
        cex.lab=1.5, cex.axis=1.2)

# display number of outlier values in the column
outlier_NetUSage <- boxplot.stats(advert$NET_USAGE)$out
mtext(paste("Outliers: ", paste(length(outlier_NetUSage), collapse=", ")),
      cex=1)
```




```r
#With the exception of the Individual Income Level which had circa eight
#outliers on the higher side, the rest of the columns had no outliers.Given that
#the outlier values are valid data points, we make the decision to retain them
#in the dataset.

#Leveraging power of Regular Expressions to check for non-charnumeric values
sum(grepl(':', advert))
```

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

```r
#There are no non-charnumeric values


#FEATURE ENGINEERING

#Additional Feature Engineering to get the Gender factors to easily comprehensible
#types

# replace the ones and zeros in 'gender' column with 'male' and 'female' using
#the ifelse() function

advert$GENDER <- ifelse(advert$GENDER == 1,"Male", "Female")

advert$CLICKS <- ifelse(advert$CLICKS == 1, "Yes", "No")
```

```
#Grouping Countries by Continent

AFRICA <- advert %>%
  mutate(AFRICA = COUNTRY %in% c("Lesotho", "Mozambique", "Namibia", "Cape Verde",
                                  "Comoros", "Ethiopia", "Mali", "Djibouti", "Sudan",
                                  "Cameroon","Egypt", "Burundi", "Ghana", "Tunisia"))

EUROPE <- advert %>%
  mutate(EUROPE = COUNTRY %in% c("Slovakia (Slovak Republic)", "Andorra",
                                  "Denmark", "Slovenia", "Romania", "Isle of Man",
                                  "Greece", "Monaco", "Russian Federation", "Spain",
                                  "Bosnia and Herzegovina", "Norway", "Iceland",
                                  "Italy", "San Marino"))
ASIA <- advert %>%
  mutate(ASIA = COUNTRY %in% c("Armenia", "Kiribati", "Marshall Islands",
                                "India", "Nepal", "Vanuatu", "Macao", "Tuvalu" ,
                                "Tokelau" , "Korea",
                                "British Indian Ocean Territory (Chagos Archipelago)",
                                "Australia", "Myanmar","Nauru"))
AMERICA <- advert %>%
  mutate(AMERICA = COUNTRY %in% c("South Georgia and the South Sandwich Islands",
                                   "Uruguay", "Cayman Islands", "United States Virgin Islands",
                                   "Aruba", "Peru", "British Virgin Islands",
                                   "Bouvet Island (Bouvetoya)" , "Barbados", "Grenada" ))
MID_EAST <- advert %>%
  mutate(MID_EAST = COUNTRY %in% c("Syrian Arab Republic","Yemen", "Afghanistan",
                                    "Palestinian Territory" , "Qatar"  ))

#Creating Region Column in Our Dataset
advert <- mutate (advert, REGION = ifelse(COUNTRY %in% c("Congo", "Uganda", "Sierra Leone", "Angola", "
                                          ifelse(COUNTRY %in% c("Saint Barthelemy", "Germany", "Pitcair
                                                 ifelse(COUNTRY %in% c("Saint Martin", "Panama", "Guam"
                                                        ifelse(COUNTRY %in% c("Niue", "Mauritius", "Fij
                                                               ifelse(COUNTRY %in% c("Kuwait", "Jordan"


#Subsetting the Other Region Sub-classification to ensure we have all the countries
#in the Region Column
OTHER <- subset(advert, advert$REGION == "OTHER_REGION")

OTHER
```

```
## Empty data.table (0 rows and 11 cols): BROWSE_TIME,AGE,INCOME,NET_USAGE,TOPIC,CITY...
```

```
#Previewing the dataset
tail(advert)
```

```
##    BROWSE_TIME AGE   INCOME NET_USAGE                               TOPIC
## 1:       43.70  28 63126.96    173.01        Front-line bifurcated ability
## 2:       72.97  30 71384.57    208.58        Fundamental modular algorithm
## 3:       51.30  45 67782.17    134.42      Grass-roots cohesive monitoring
## 4:       51.63  51 42415.72    120.37         Expanded intangible solution
```

```
## 5:       55.55  19 41920.79    187.95 Proactive bandwidth-monitored policy
## 6:       45.01  26 29875.80    178.35      Virtual 5thgeneration emulation
##             CITY GENDER                COUNTRY           TIMESTAMP CLICKS
## 1:  Nicholasland Female                Mayotte 2016-04-04 03:57:48    Yes
## 2:     Duffystad   Male                Lebanon 2016-02-11 21:49:00    Yes
## 3:   New Darlene   Male Bosnia and Herzegovina 2016-04-22 02:07:01    Yes
## 4: South Jessica   Male               Mongolia 2016-02-01 17:24:57    Yes
## 5:   West Steven Female              Guatemala 2016-03-24 02:35:54     No
## 6:   Ronniemouth Female                 Brazil 2016-06-03 21:43:21    Yes
##      REGION
## 1:   AFRICA
## 2: MID_EAST
## 3:   EUROPE
## 4:     ASIA
## 5:  AMERICA
## 6:  AMERICA
```

```
#We will Split Date and Time from Timestamp in order to carry out further analysis
advert$DATE <- as.Date(advert$TIMESTAMP)
advert$TIME <- format(as.POSIXct(advert$TIMESTAMP), format = "%H:%M:%S")

#Extracting time from the date/time stamp

advert <- advert %>% separate(TIME, c("HOUR", "MINUTE", "SECONDS"))

#Apportioning the Hour Column into features that can be analyzed
advert$HOUR = ifelse(advert$HOUR >= "00" & advert$HOUR <= "06", "Wee Hours",
                     ifelse(advert$HOUR >= "07" & advert$HOUR <= "12", "Morning Hours",
                            ifelse(advert$HOUR >= "13" & advert$HOUR <= "18",
                                   "Afternoon Hours", "Night")))


#Previewing the dataset
head(advert)
```

```
##    BROWSE_TIME AGE   INCOME NET_USAGE                                TOPIC
## 1:       68.95  35 61833.90    256.09    Cloned 5thgeneration orchestration
## 2:       80.23  31 68441.85    193.77    Monitored national standardization
## 3:       69.47  26 59785.94    236.50      Organic bottom-line service-desk
## 4:       74.15  29 54806.18    245.89 Triple-buffered reciprocal time-frame
## 5:       68.37  35 73889.99    225.58         Robust logistical utilization
## 6:       59.99  23 59761.56    226.74       Sharable client-driven software
##              CITY GENDER    COUNTRY           TIMESTAMP CLICKS REGION
## 1:    Wrightburgh Female    Tunisia 2016-03-27 00:53:11     No AFRICA
## 2:      West Jodi   Male      Nauru 2016-04-04 01:39:02     No   ASIA
## 3:       Davidton Female San Marino 2016-03-13 20:35:42     No EUROPE
## 4: West Terrifurt   Male      Italy 2016-01-10 02:31:19     No EUROPE
## 5:   South Manuel Female    Iceland 2016-06-03 03:36:18     No EUROPE
## 6:      Jamieberg   Male     Norway 2016-05-19 14:30:17     No EUROPE
##          DATE         HOUR MINUTE SECONDS
## 1: 2016-03-27    Wee Hours     53      11
## 2: 2016-04-04    Wee Hours     39      02
## 3: 2016-03-13        Night     35      42
## 4: 2016-01-10    Wee Hours     31      19
```

```
## 5: 2016-06-03       Wee Hours     36      18
## 6: 2016-05-19 Afternoon Hours     30      17
```

```
#Dropping Columns we don't need for analysis

advert <- select(advert, -c(TOPIC, CITY, TIMESTAMP, MINUTE, DATE, SECONDS))


numeric <- select(advert, c(BROWSE_TIME, AGE, INCOME, NET_USAGE) )

non.numeric <- select(advert, c(GENDER, COUNTRY, CLICKS, REGION, HOUR))

#EXPLORATORY DATA ANALYSIS

#UNIVARIATE ANALYSIS


top.countries <-advert %>%
  select(COUNTRY, CLICKS) %>%
  filter(CLICKS== "Yes") %>%
  count(COUNTRY) %>%
  arrange(desc(n)) %>%
  head(5)

#Australia, Turkey and Ethiopia are the countries with the most clicks

top.countries <-advert %>%
  select(COUNTRY, CLICKS) %>%
  filter(CLICKS== "Yes") %>%
  count(COUNTRY) %>%
  arrange(-desc(n)) %>%
  head(5)
#THe list number of clicks per country was 1 click
nop.countries <-advert %>%
  select(COUNTRY, CLICKS) %>%
  filter(CLICKS== "No") %>%
  count(COUNTRY) %>%
  arrange(desc(n)) %>%
  head(5)
#Bolivia, Croatia & Gabon registered the highest number of Zero CLicks on an advert
nop.countries <-advert %>%
  select(COUNTRY, CLICKS) %>%
  filter(CLICKS== "No") %>%
  count(COUNTRY) %>%
  arrange(-desc(n)) %>%
  head(5)


#Measures of Central Tendency
#Summary of the numeric values using the function summary
summary(numeric)
```

```
##   BROWSE_TIME         AGE             INCOME         NET_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
```

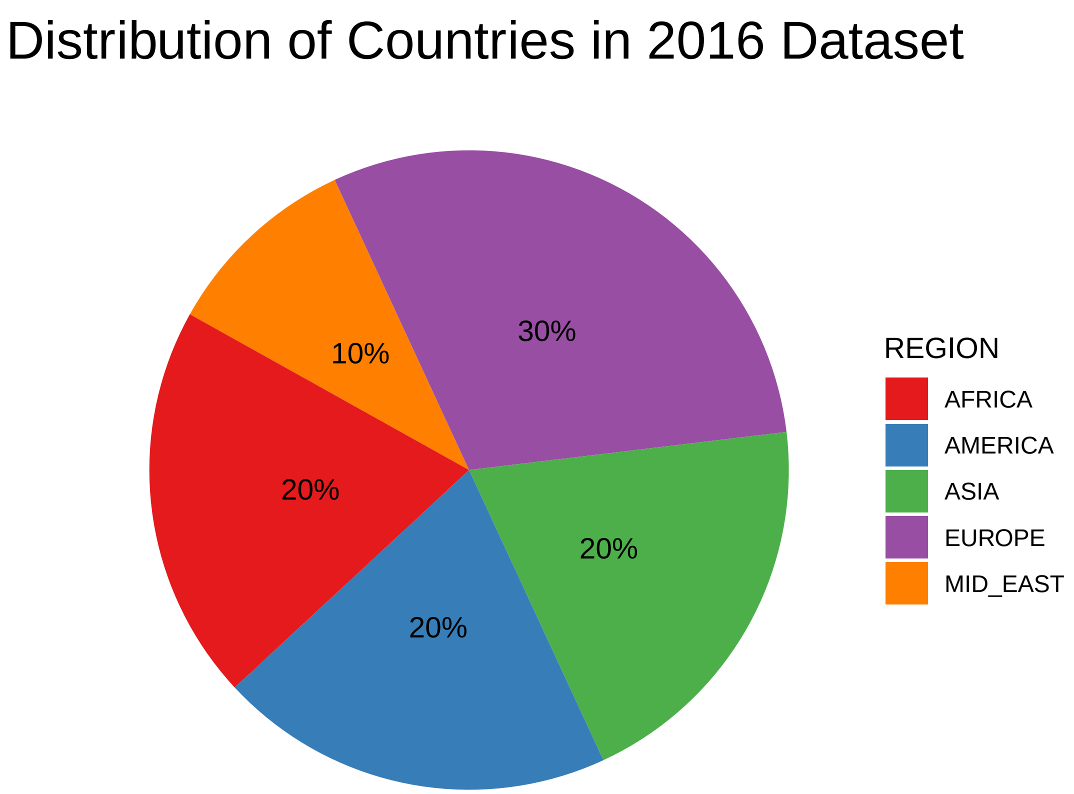
```
#The average Browse time was 65, average age of users 36 years, average region income
#being 55000 and the average network usage 180.

#The maximum time spent online was 91.43 while the least was 32.60
#The oldest person online was age 61 whilst the youngest was only 19

#The highest area income was around 79000 whilst the least was around 14000

#The highest internet usage per day was 270 whilst the least was 105


#Description of the entire Dataset using the Describe function

describe(advert)
```

```
## advert
##
##  9  Variables      1000  Observations
## --------------------------------------------------------------------------------
## BROWSE_TIME
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1000        0      900        1       65    18.11    37.58    41.34
##      .25      .50      .75      .90      .95
##    51.36    68.22    78.55    83.89    86.20
##
## lowest : 32.60 32.84 32.91 32.99 33.21, highest: 90.97 91.10 91.15 91.37 91.43
## --------------------------------------------------------------------------------
## AGE
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1000        0       43    0.999    36.01    9.943    23.95    26.00
##      .25      .50      .75      .90      .95
##    29.00    35.00    42.00    49.00    52.00
##
## lowest : 19 20 21 22 23, highest: 57 58 59 60 61
## --------------------------------------------------------------------------------
## INCOME
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1000        0     1000        1    55000    15037    28275    35223
##      .25      .50      .75      .90      .95
##    47032    57012    65471    70506    73601
##
## lowest : 13996.50 14548.06 14775.50 15598.29 15879.10
## highest: 78092.95 78119.50 78520.99 79332.33 79484.80
## --------------------------------------------------------------------------------
## NET_USAGE
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     1000        0      966        1      180    50.63    113.5    120.5
##      .25      .50      .75      .90      .95
```

```
##    138.8    183.1    218.8    236.2    246.7
##
## lowest : 104.78 105.00 105.04 105.15 105.22, highest: 259.76 261.02 261.52 267.01 269.96
## ---------------------------------------------------------------------------
## GENDER
##        n  missing distinct
##     1000        0        2
##
## Value      Female   Male
## Frequency     519    481
## Proportion  0.519  0.481
## ---------------------------------------------------------------------------
## COUNTRY
##        n  missing distinct
##     1000        0      237
##
## lowest : Afghanistan       Albania           Algeria           American Samoa    Andorra
## highest: Wallis and Futuna Western Sahara    Yemen             Zambia            Zimbabwe
## ---------------------------------------------------------------------------
## CLICKS
##        n  missing distinct
##     1000        0        2
##
## Value       No Yes
## Frequency  500 500
## Proportion 0.5 0.5
## ---------------------------------------------------------------------------
## REGION
##        n  missing distinct
##     1000        0        5
##
## lowest : AFRICA   AMERICA  ASIA     EUROPE   MID_EAST
## highest: AFRICA   AMERICA  ASIA     EUROPE   MID_EAST
##
## Value        AFRICA  AMERICA     ASIA   EUROPE MID_EAST
## Frequency       205      224      236      273       62
## Proportion    0.205    0.224    0.236    0.273    0.062
## ---------------------------------------------------------------------------
## HOUR
##        n  missing distinct
##     1000        0        4
##
## Value      Afternoon Hours   Morning Hours           Night       Wee Hours
## Frequency              241             255             224             280
## Proportion           0.241           0.255           0.224           0.280
## ---------------------------------------------------------------------------
```

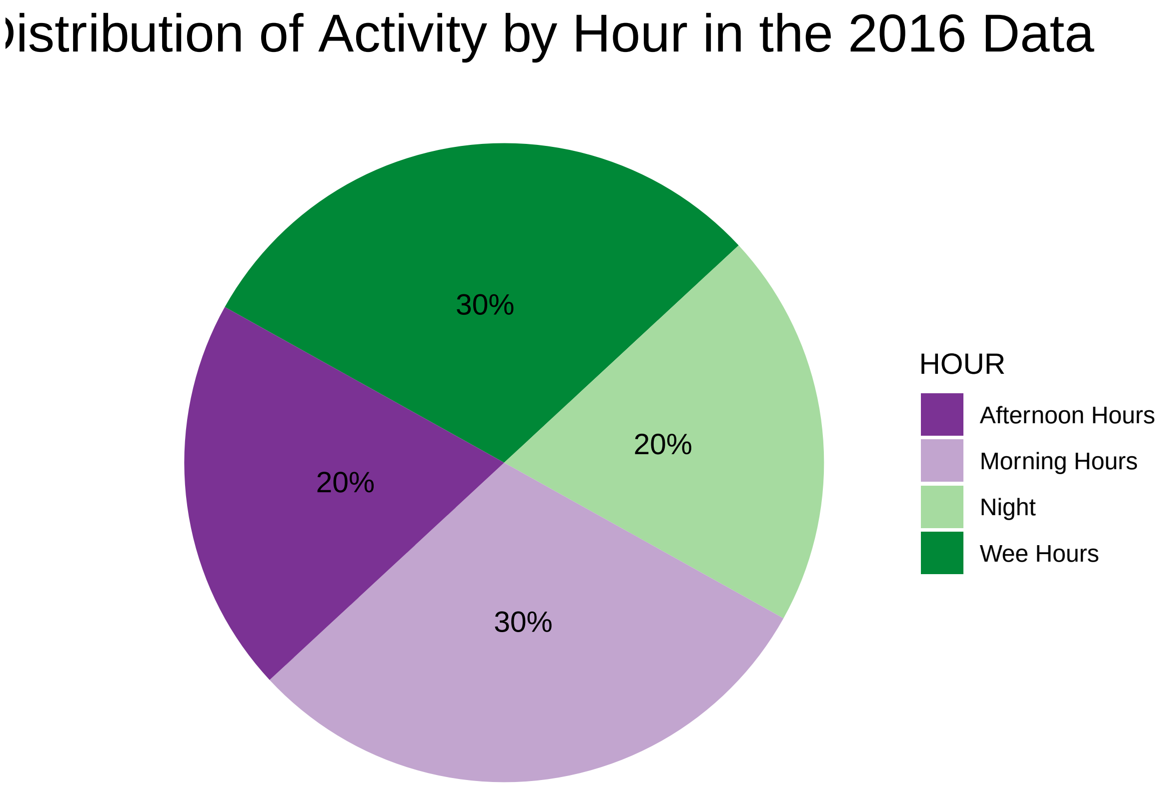
```
# Pie-chart displaying the distribution of the countries in the Dataset

region_perc <- advert %>%
  filter(REGION != "NA") %>%
  group_by(REGION) %>%
  count() %>%
  ungroup() %>%
```

```
  arrange(desc(REGION)) %>%
  mutate( percentage = round(n/sum(n), 1)*100, lab.pos = cumsum(percentage)- 0.5 * percentage)
ggplot(region_perc, aes(x = "", y= percentage, fill = REGION)) +
  geom_bar(stat = "identity")+
  coord_polar("y", start = 200) +
  geom_text(aes(y = lab.pos, label = paste(percentage,"%", sep = "")), col = "black") +
  theme_void() + scale_fill_brewer(palette = "Set1") + labs(title= "Distribution of Countries in 2016 Da
  theme(plot.title = element_text(hjust = 0.4, size = 20))
```

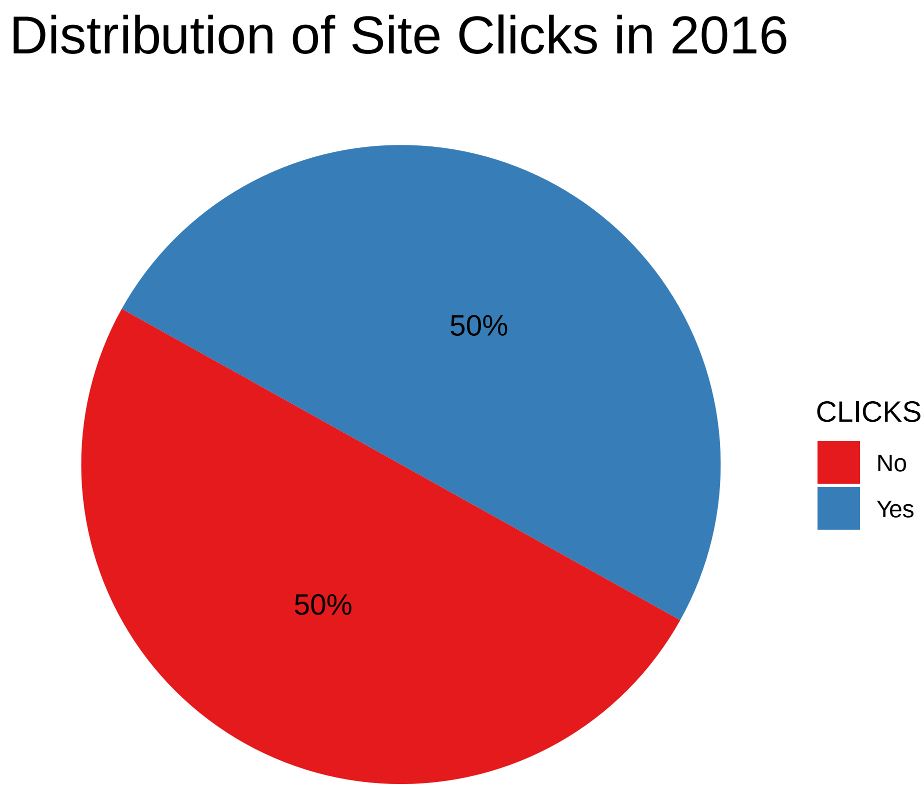



```
#Europe was the most represented region in the dataset whilst the Mid_East was
#the least represented


#Display of the most active hours
hour_perc <- advert %>%
  filter(HOUR != "NA") %>%
  group_by(HOUR) %>%
  count() %>%
  ungroup() %>%
  arrange(desc(HOUR)) %>%
  mutate( percentage = round(n/sum(n), 1)*100, lab.pos = cumsum(percentage)- 0.5 * percentage)
ggplot(hour_perc, aes(x = "", y= percentage, fill = HOUR)) +
  geom_bar(stat = "identity")+
  coord_polar("y", start = 200) +
  geom_text(aes(y = lab.pos, label = paste(percentage,"%", sep = "")), col = "black") +
```

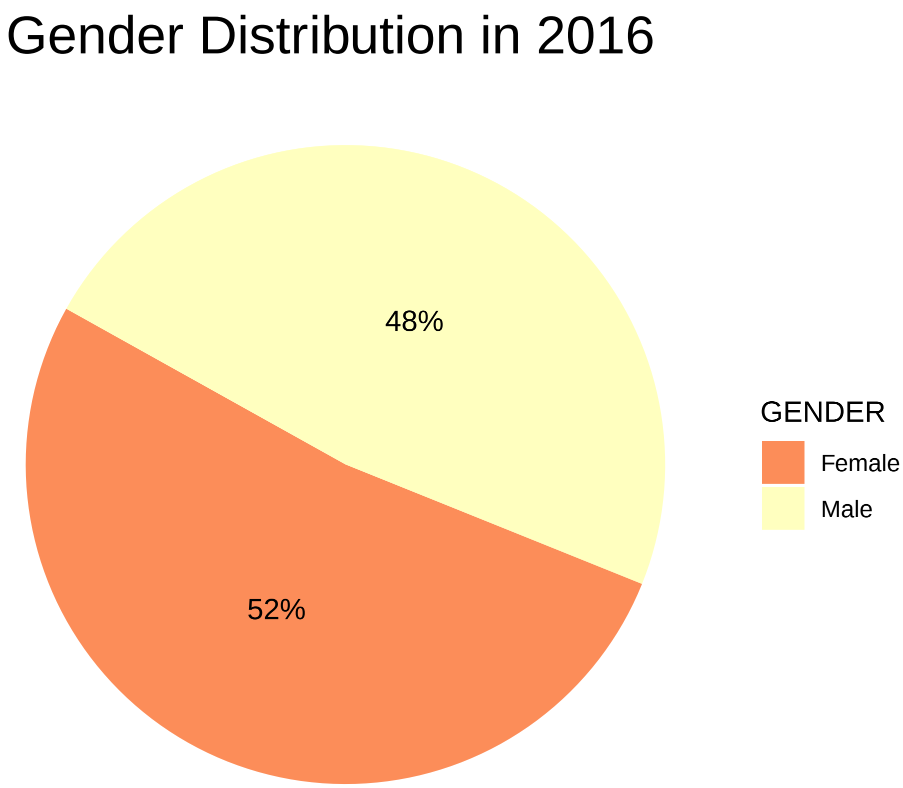
```
  theme_void() + scale_fill_brewer(palette = "PRGn") + labs(title= "Distribution of Activity by Hour in
  theme(plot.title = element_text(hjust = 0.4, size = 20))
```



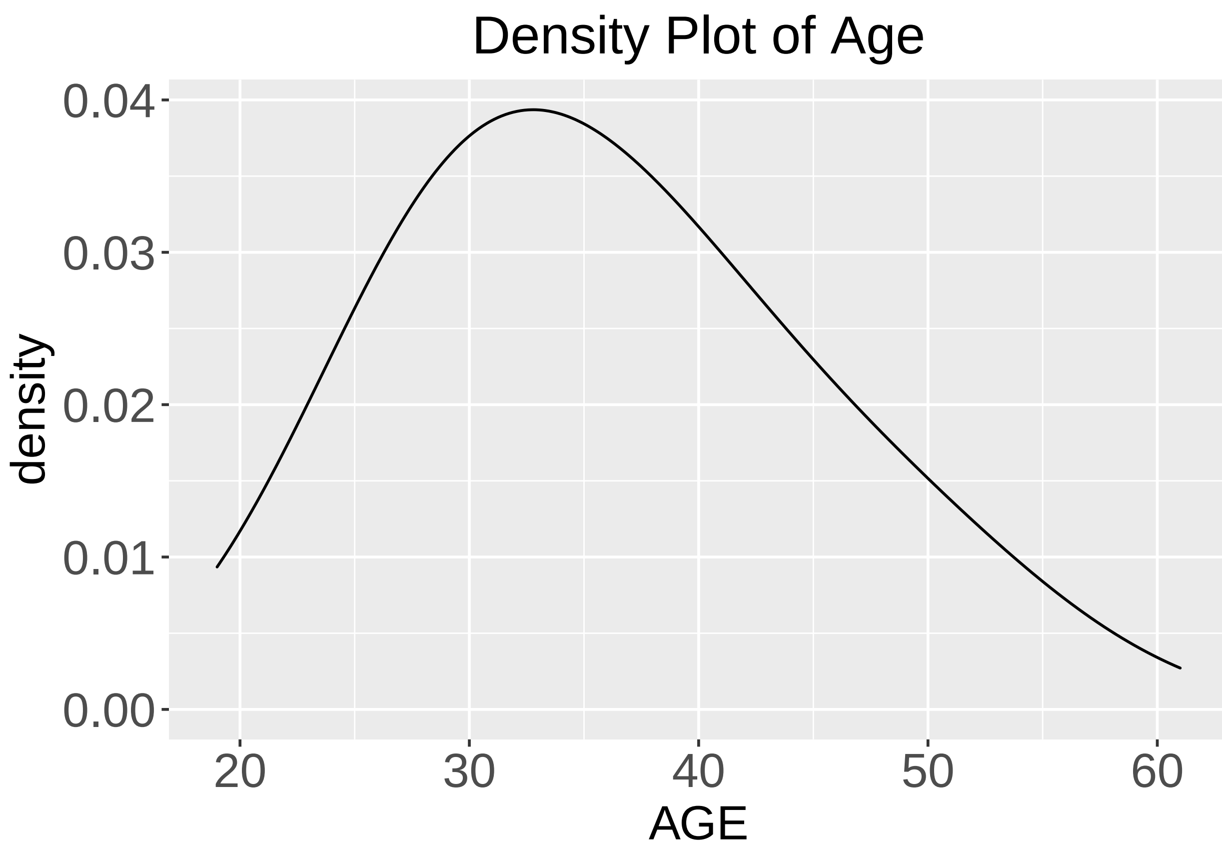

```
#Most browsing activity took place in the wee Hours of the night and the morning
#hours


#Display of whether an advert was clicked or not

click_perc <- advert %>%
  filter(CLICKS != "NA") %>%
  group_by(CLICKS) %>%
  count() %>%
  ungroup() %>%
  arrange(desc(CLICKS)) %>%
  mutate( percentage = round(n/sum(n), 1)*100, lab.pos = cumsum(percentage)- 0.5 * percentage)
ggplot(click_perc, aes(x = "", y= percentage, fill = CLICKS)) +
  geom_bar(stat = "identity")+
  coord_polar("y", start = 200) +
  geom_text(aes(y = lab.pos, label = paste(percentage,"%", sep = "")), col = "black") +
  theme_void() + scale_fill_brewer(palette = "Set1") + labs(title= "Distribution of Site Clicks in 2016
  theme(plot.title = element_text(hjust = 0.4, size = 20))
```



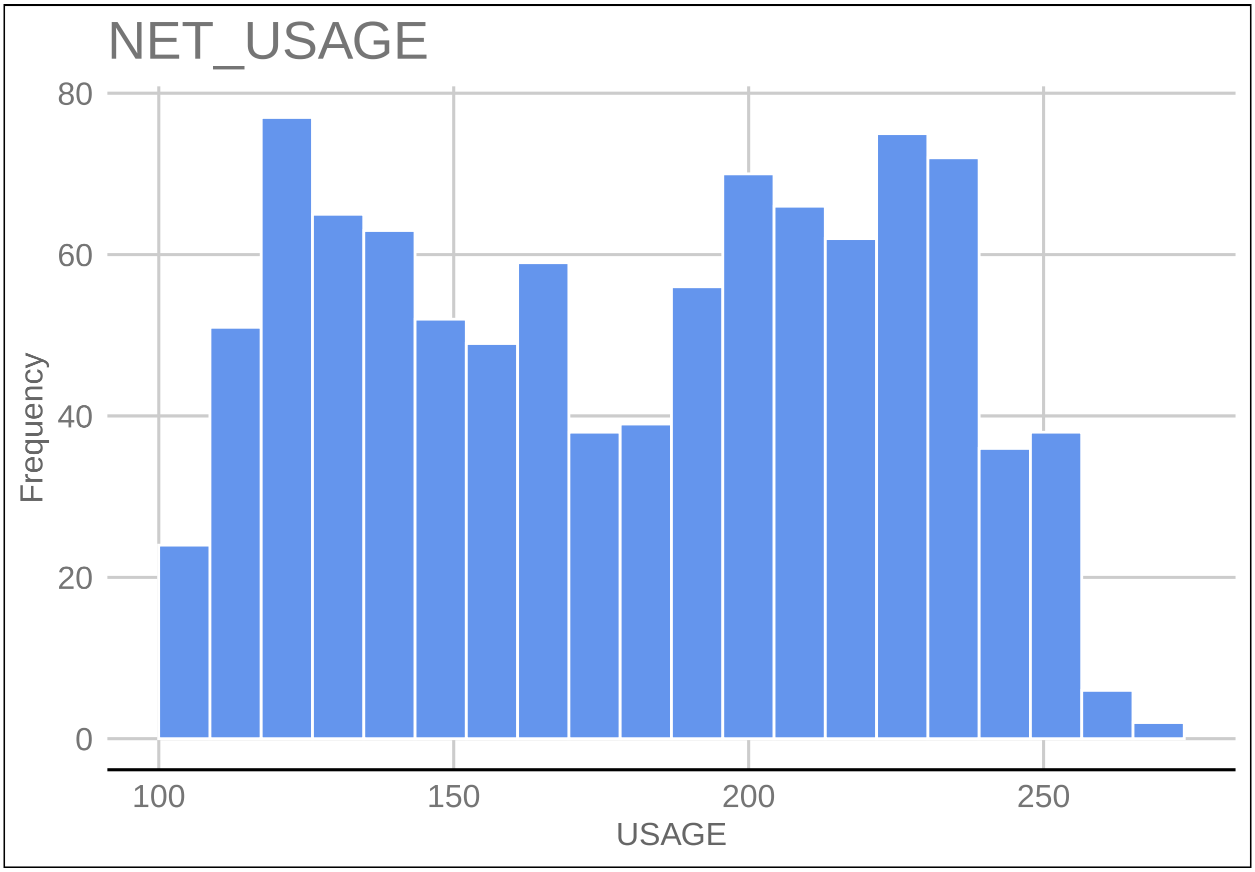

```
# There was no split on whether an advert was clicked or not. There was always
#a 50% chance that a user would click on an advert


#Plotting Pie Chart for Gender Distribution

#Filtering the gender df
pie_gender <- advert %>%
  filter(GENDER != "NA") %>%
  group_by(GENDER) %>%
  count() %>%
  ungroup() %>%
  arrange(desc(GENDER)) %>%
  mutate( percentage = round(n/sum(n), 2)*100, lab.pos = cumsum(percentage)- 0.5 * percentage)
ggplot(pie_gender, aes(x = "", y= percentage, fill = GENDER)) +
  geom_bar(stat = "identity")+
  coord_polar("y", start = 200) +
  geom_text(aes(y = lab.pos, label = paste(percentage,"%", sep = "")), col = "black") +
  theme_void() + scale_fill_brewer(palette = "Spectral") + labs(title= "Gender Distribution in 2016") +
  theme(plot.title = element_text(hjust = 0.4, size = 20))
```




```
#It appears that more women in the dataset browsed on the internet

#Density Plot Distribution of the Age Column
ggplot(advert, aes(x= AGE)) +
  geom_density(bw = 5) +
  ggtitle("Density Plot of Age") +
  theme(axis.text = element_text(size=18),
        axis.title = element_text(size = 18),
        plot.title = element_text(hjust = 0.5, size = 20))
```




```r
#The individuals in the dataset were between ages 19 and 61 with the median age
#being around 35 years.

# Histogram for Daily time spent on site


ggplot(advert, aes(x = `NET_USAGE` )) +
  geom_histogram(fill = "cornflowerblue",
                 color = "white",bins = 20) +
  theme_gdocs() +
  labs(title="NET_USAGE",
       x = "USAGE", y = "Frequency")
```




```r
# Skewness and kurtosis of Daily Browsing
cat('The skewness and kurtosis of daily browsing', '\n')
```

```
## The skewness and kurtosis of daily browsing
```

```r
cat("Skewness: ", skewness(advert$BROWSE_TIME), '\n')
```

```
## Skewness:  -0.3712026
```

```r
cat("Kurtosis: ", kurtosis(advert$BROWSE_TIME), '\n')
```

```
## Kurtosis:  1.903942
```

```r
cat("Variance: ", var(advert$BROWSE_TIME),  '\n')
```

```
## Variance:  251.3371
```

```r
cat("Standard Deviation: ", sd(advert$BROWSE_TIME),  '\n')
```

```
## Standard Deviation:  15.85361
```

```r
#Skewness, variance, standard deviation and Kurtosis of Income

cat('The skewness and kurtosis of Area Income', '\n')
```

```
## The skewness and kurtosis of Area Income
```

```r
cat("Skewness: ", skewness(advert$INCOME), '\n')
```

```
## Skewness:  -0.6493967
```

```r
cat("Kurtosis: ", kurtosis(advert$INCOME), '\n')
```

```
## Kurtosis:  2.894694
```

```r
cat("Variance: ", var(advert$INCOME),  '\n')
```

```
## Variance:  179952406
```

```r
cat("Standard Deviation: ", sd(advert$INCOME),  '\n')
```

```
## Standard Deviation:  13414.63
```

```r
#Skewness and Kurtosis of Age

cat('The skewness and kurtosis of Age', '\n')
```

```
## The skewness and kurtosis of Age
```

```r
cat("Skewness: ", skewness(advert$AGE), '\n')
```

```
## Skewness:  0.4784227
```

```r
cat("Kurtosis: ", kurtosis(advert$AGE), '\n')
```

```
## Kurtosis:  2.595482
```

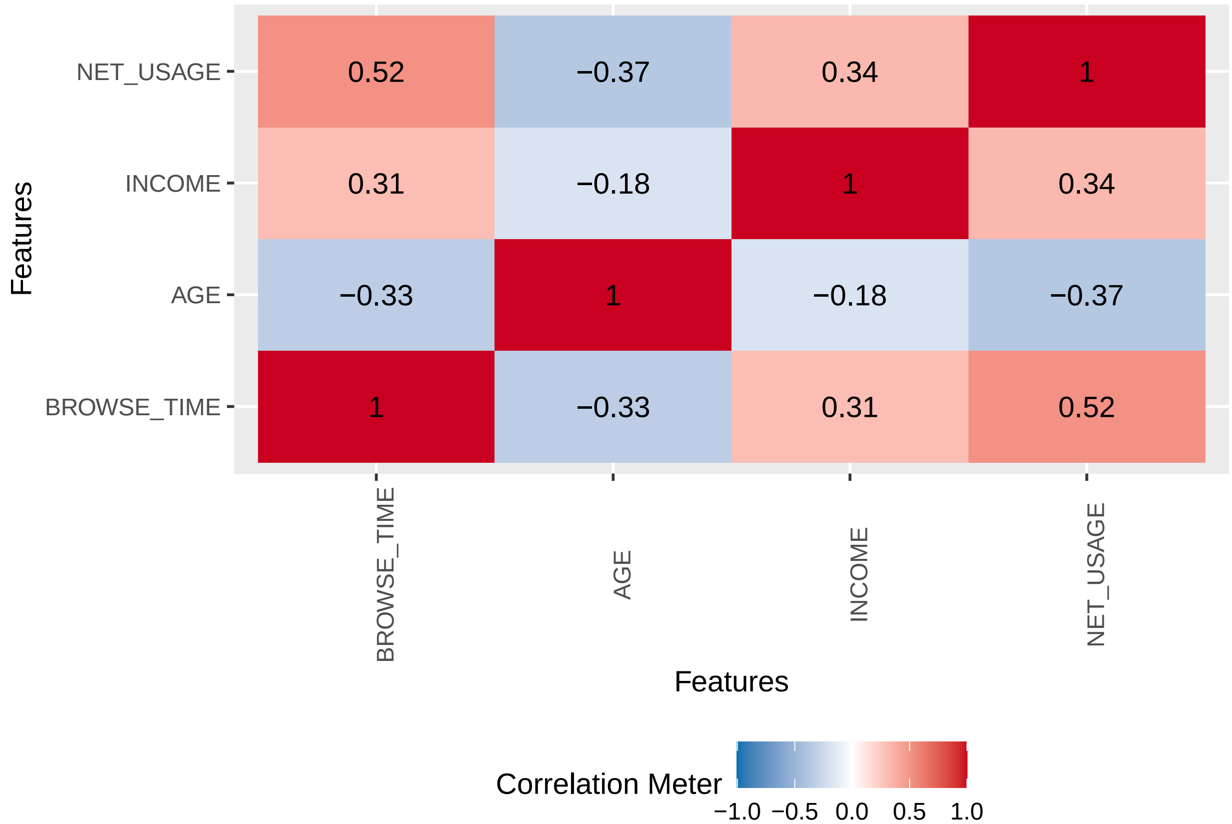
```r
cat("Variance: ", var(advert$AGE),  '\n')
```

```
## Variance:  77.18611
```

```r
cat("Standard Deviation: ", sd(advert$AGE),  '\n')
```

```
## Standard Deviation:  8.785562
```

```
#The values are fairly symmetrical, very slightly skewed to the right and platykurtic


#Bivariate Analysis

#Correlation Plot

options(repr.plot.width = 18, repr.plot.height = 18)

plot_correlation(advert, type = 'c',cor_args = list( 'use' = 'complete.obs'))
```




```
#Pairwise Scatterplot
head(advert)
```

```
##    BROWSE_TIME AGE   INCOME NET_USAGE GENDER    COUNTRY CLICKS REGION
## 1:       68.95  35 61833.90    256.09 Female    Tunisia     No AFRICA
## 2:       80.23  31 68441.85    193.77   Male      Nauru     No   ASIA
## 3:       69.47  26 59785.94    236.50 Female San Marino     No EUROPE
## 4:       74.15  29 54806.18    245.89   Male      Italy     No EUROPE
## 5:       68.37  35 73889.99    225.58 Female    Iceland     No EUROPE
## 6:       59.99  23 59761.56    226.74   Male     Norway     No EUROPE
##          HOUR
## 1:  Wee Hours
## 2:  Wee Hours
## 3:      Night
```

```
## 4:       Wee Hours
## 5:       Wee Hours
## 6: Afternoon Hours
```

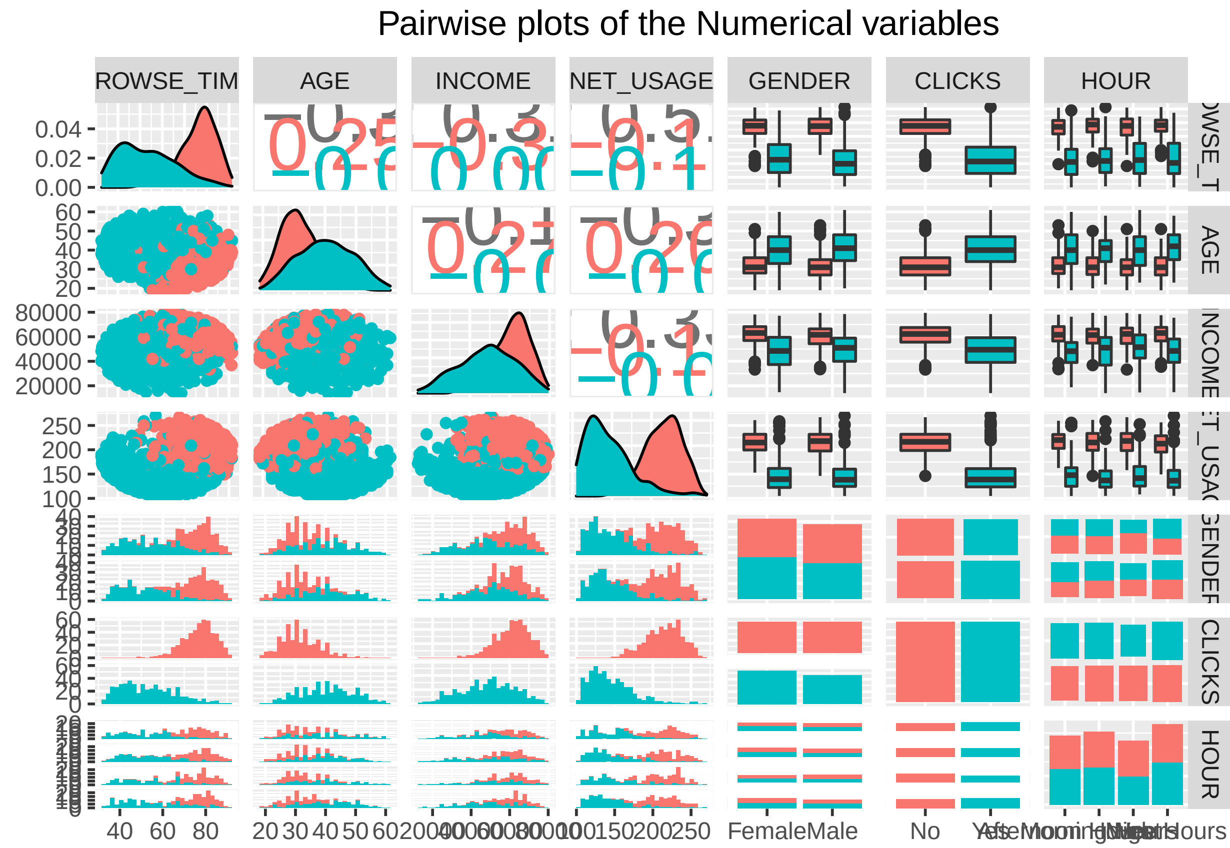
```r
df.pw <- select(advert, -c(COUNTRY, REGION))
head(df.pw)
```

```
##    BROWSE_TIME AGE   INCOME NET_USAGE GENDER CLICKS            HOUR
## 1:       68.95  35 61833.90    256.09 Female     No       Wee Hours
## 2:       80.23  31 68441.85    193.77   Male     No       Wee Hours
## 3:       69.47  26 59785.94    236.50 Female     No           Night
## 4:       74.15  29 54806.18    245.89   Male     No       Wee Hours
## 5:       68.37  35 73889.99    225.58 Female     No       Wee Hours
## 6:       59.99  23 59761.56    226.74   Male     No Afternoon Hours
```

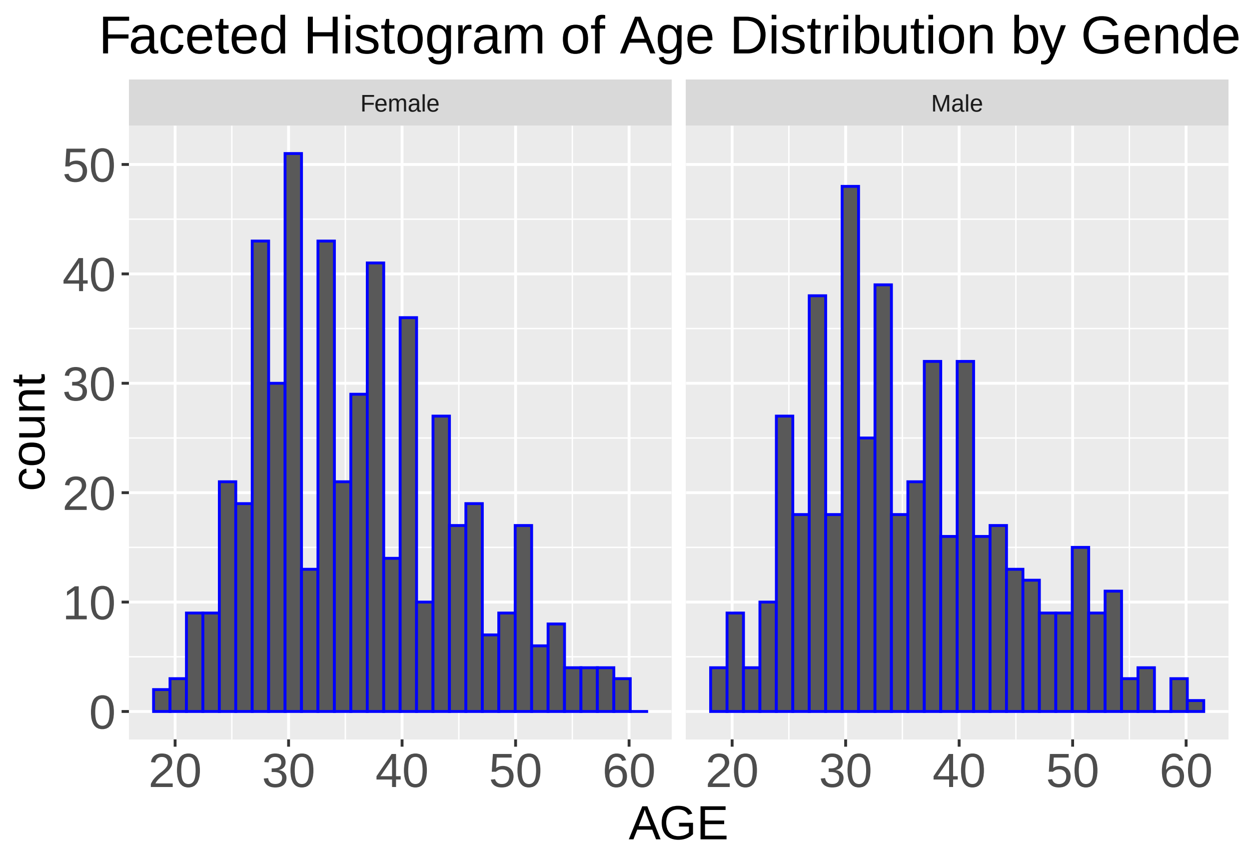
```r
options(repr.plot.width = 18, repr.plot.height = 18)
ggpairs(df.pw, mapping= aes(color = CLICKS),upper = list(continuous = wrap("cor", size = 10))) +
  labs(title = "Pairwise plots of the Numerical variables") +
  theme(plot.title = element_text(hjust = 0.6))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

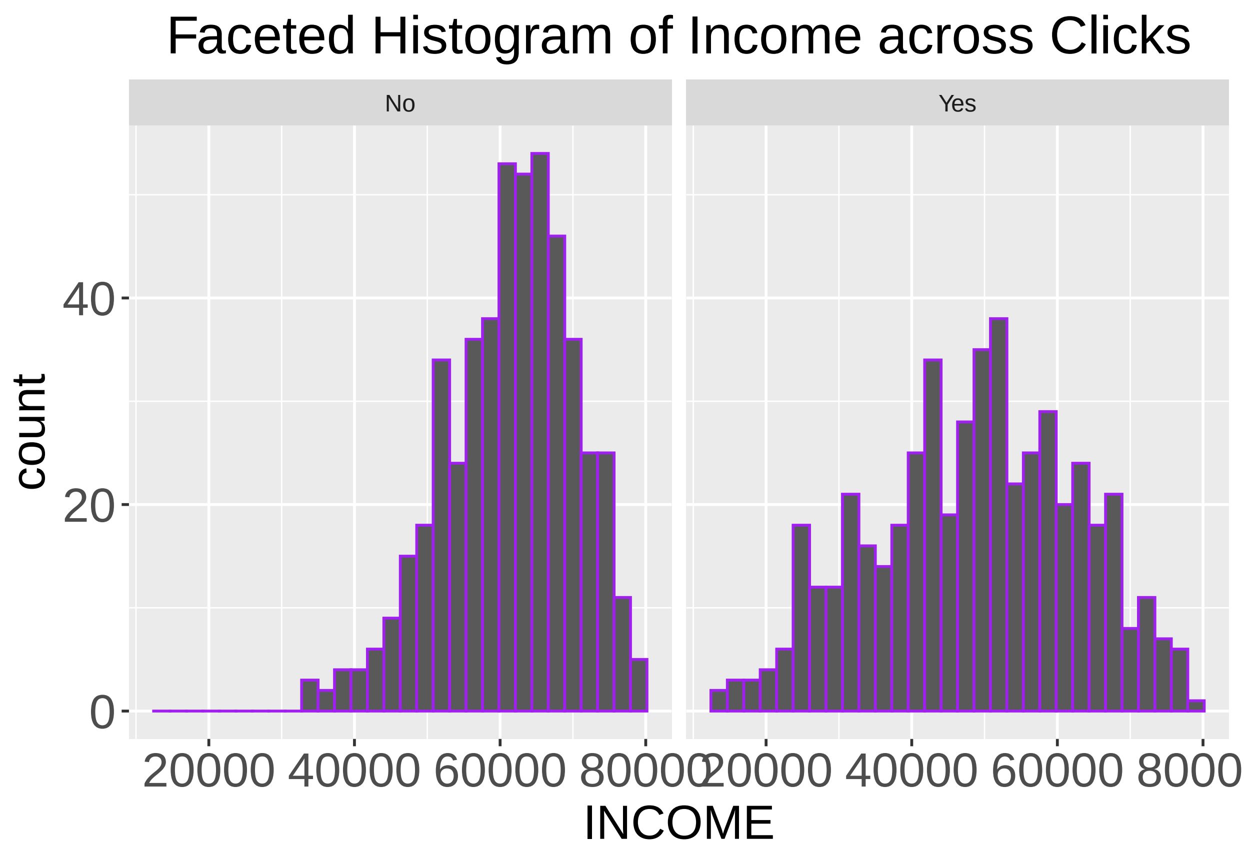
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



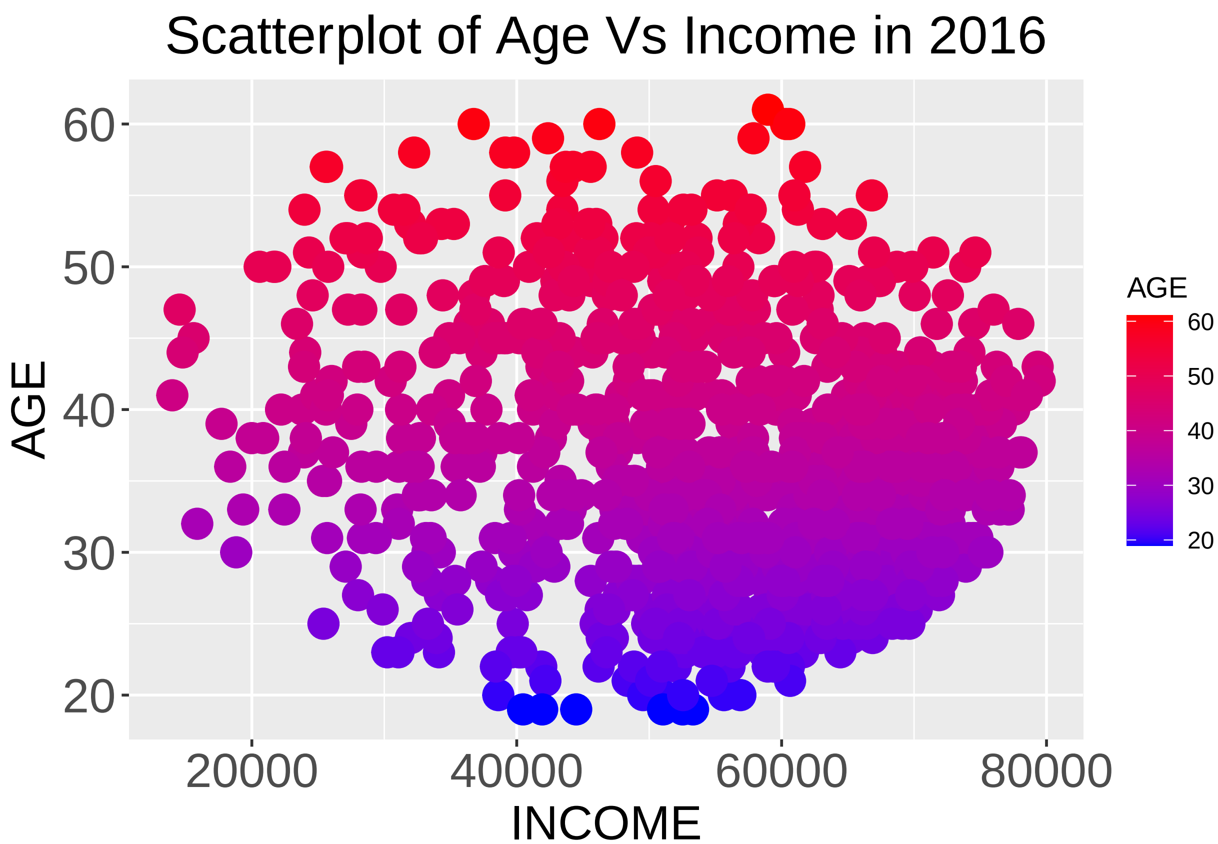

```
#Using Faceted Histograms, we investigate the distribution of Age along
#Gender Lines  theme_grey(base_size = 22) +

ggplot(advert, aes(x= AGE)) +
  geom_histogram(bins = 30, color = "blue") +
  facet_wrap(~GENDER) +
  ggtitle("Faceted Histogram of Age Distribution by Gender") +  theme(axis.text = element_text(size=18)
                                                                      axis.title = element_text(size = 
                                                                      plot.title = element_text(hjust = 
```



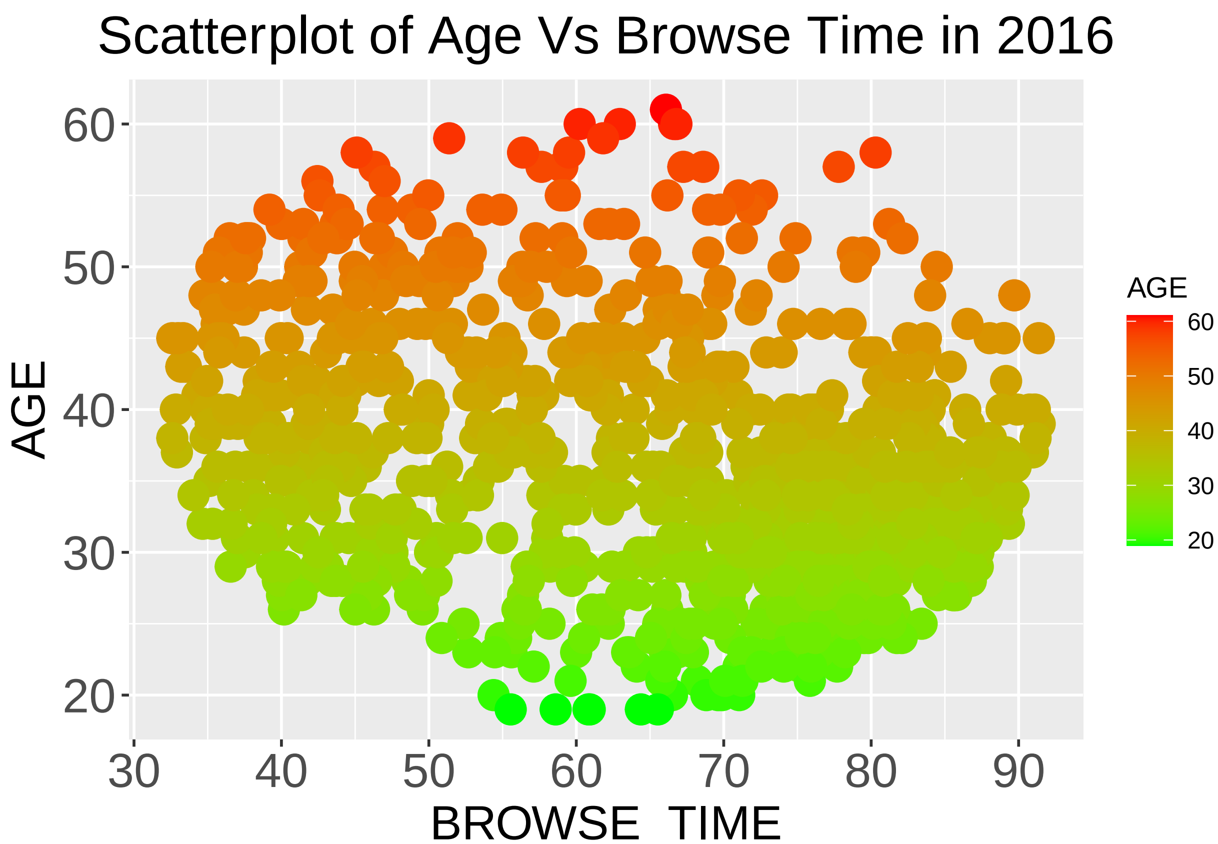

```r
#Distibution of Income along Click Lens
ggplot(advert, aes(x= INCOME)) +
  geom_histogram(bins = 30, color = "purple") +
  facet_wrap(~CLICKS) +
  ggtitle("Faceted Histogram of Income across Clicks") +
  theme(axis.text = element_text(size=18),
        axis.title = element_text(size = 18),
        plot.title = element_text(hjust = 0.5, size = 20))
```



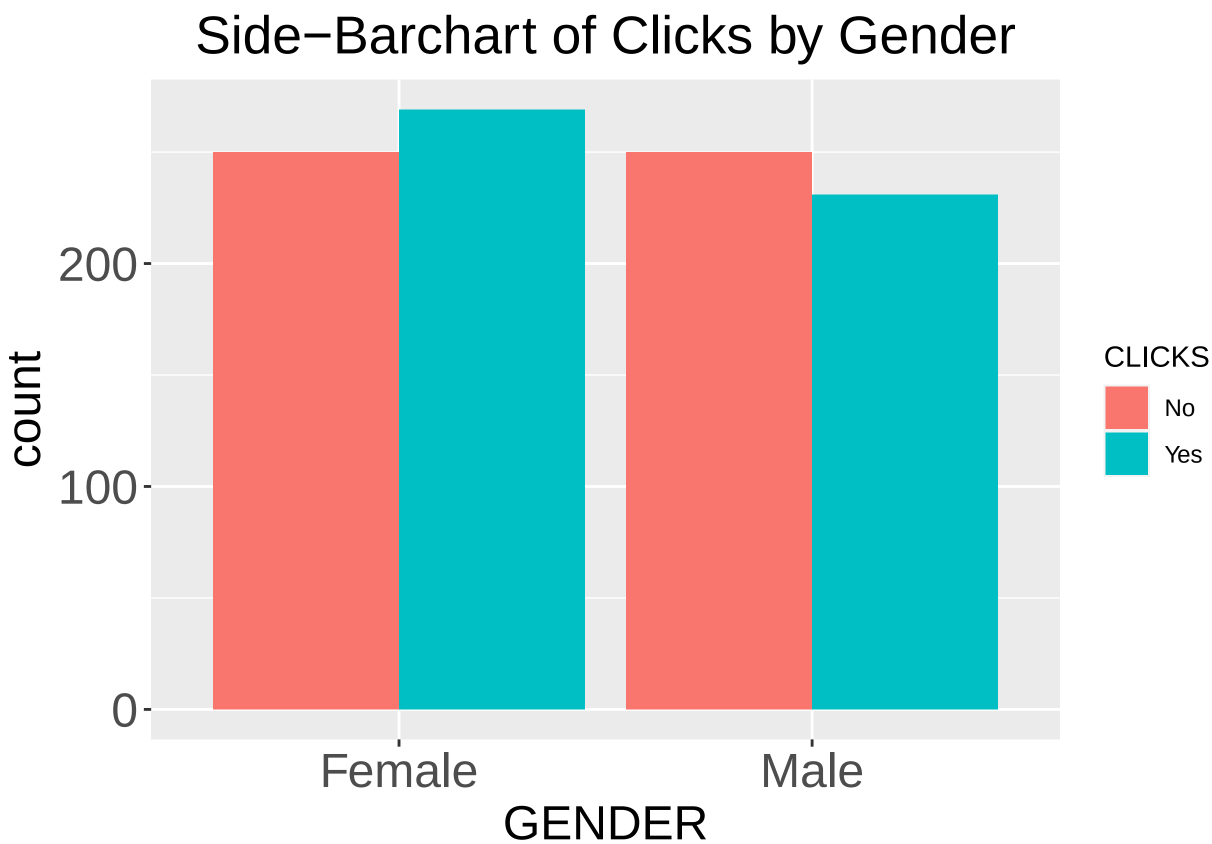

```
#Scatterplot of Age VS Income

#print(b1 + geom_point())
b1 <- ggplot(advert, aes(x=INCOME, y=AGE))

b2 <- b1 + geom_point(aes(color=AGE), size=5) + scale_color_gradient(low='blue', high = 'red')
print(b2 + ggtitle("Scatterplot of Age Vs Income in 2016") +  theme(axis.text = element_text(size=18),
                                                                     axis.title = element_text(size = 18
                                                                     plot.title = element_text(hjust = 0
```



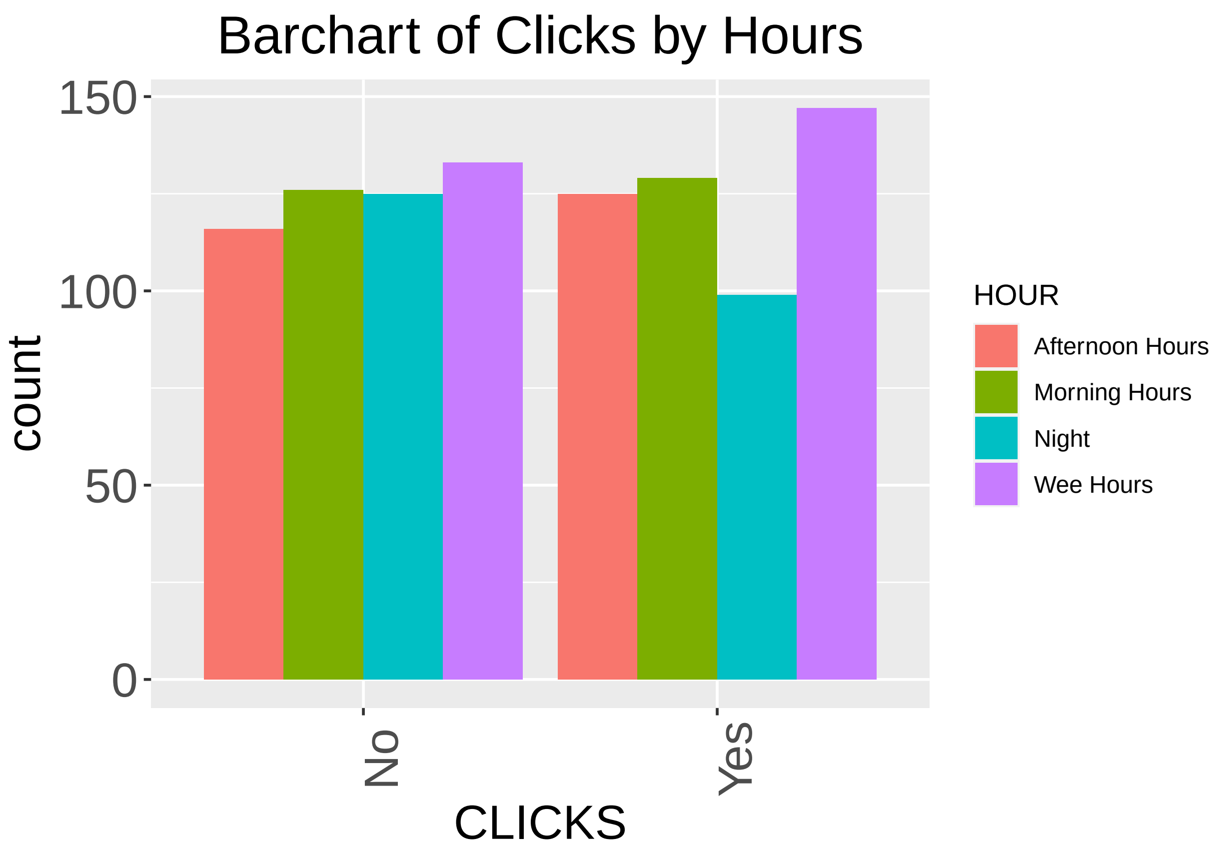

```
#Highest income levels registered by people under the age of 40 but greater than 20.



#Scatterplot of Age Vs Daily time on the Internet
b2 <- ggplot(advert, aes(x=BROWSE_TIME, y=AGE))

b3 <- b2 + geom_point(aes(color=AGE), size=5) + scale_color_gradient(low='green', high = 'red')
print(b3 + ggtitle("Scatterplot of Age Vs Browse Time in 2016") +  theme(axis.text = element_text(size=
                                                                         axis.title = element_text(size
                                                                         plot.title = element_text(hjus
```



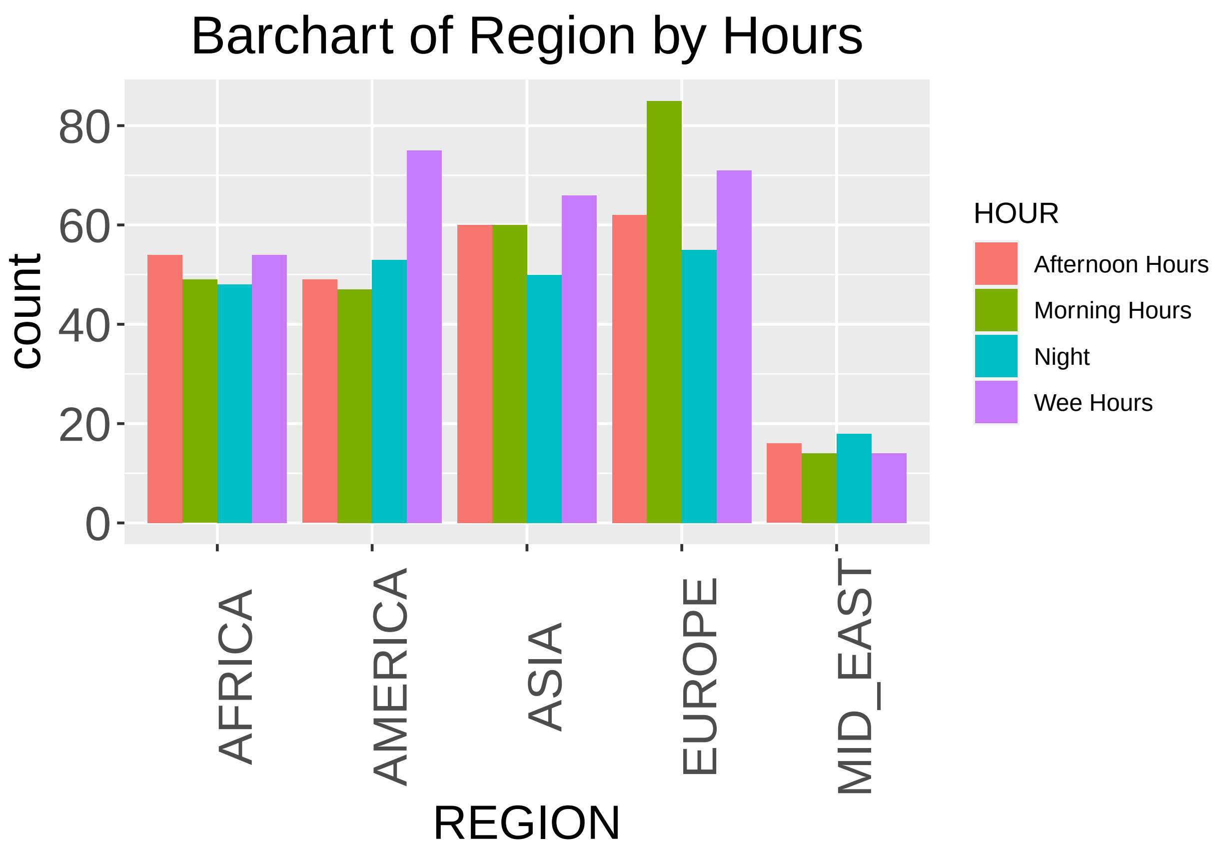

```
#Individuals between ages 25 and 45 spend the most amount of time online.

# Creating a side-by-side barchart of Gender by Clicks
ggplot(advert, aes(x = GENDER, fill = CLICKS)) +
  geom_bar(position = "dodge") +
  ggtitle("Side-Barchart of Clicks by Gender") +
  theme(axis.text = element_text(size=18),
        axis.title = element_text(size = 18),
        plot.title = element_text(hjust = 0.5, size = 20))
```



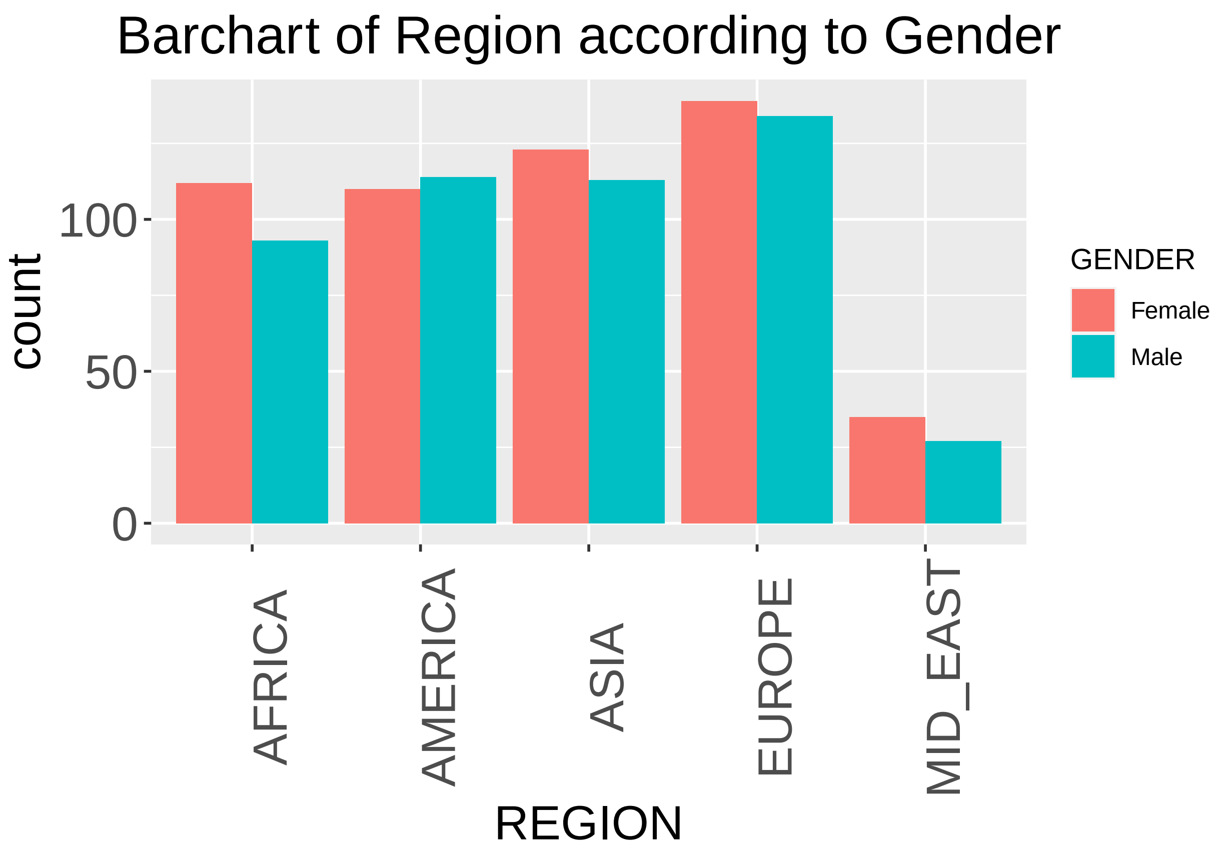

```
#More males clicked sites than females.

# Creating a side-by-side barchart of Clicks by Hour
ggplot(advert, aes(x = CLICKS, fill = HOUR)) +
  geom_bar(position= "dodge") +
  theme(axis.text.x = element_text(angle = 90)) +
  ggtitle("Barchart of Clicks by Hours") +  theme(axis.text = element_text(size=18),
                                                   axis.title = element_text(size = 18),
                                                   plot.title = element_text(hjust = 0.5, size = 20))
```



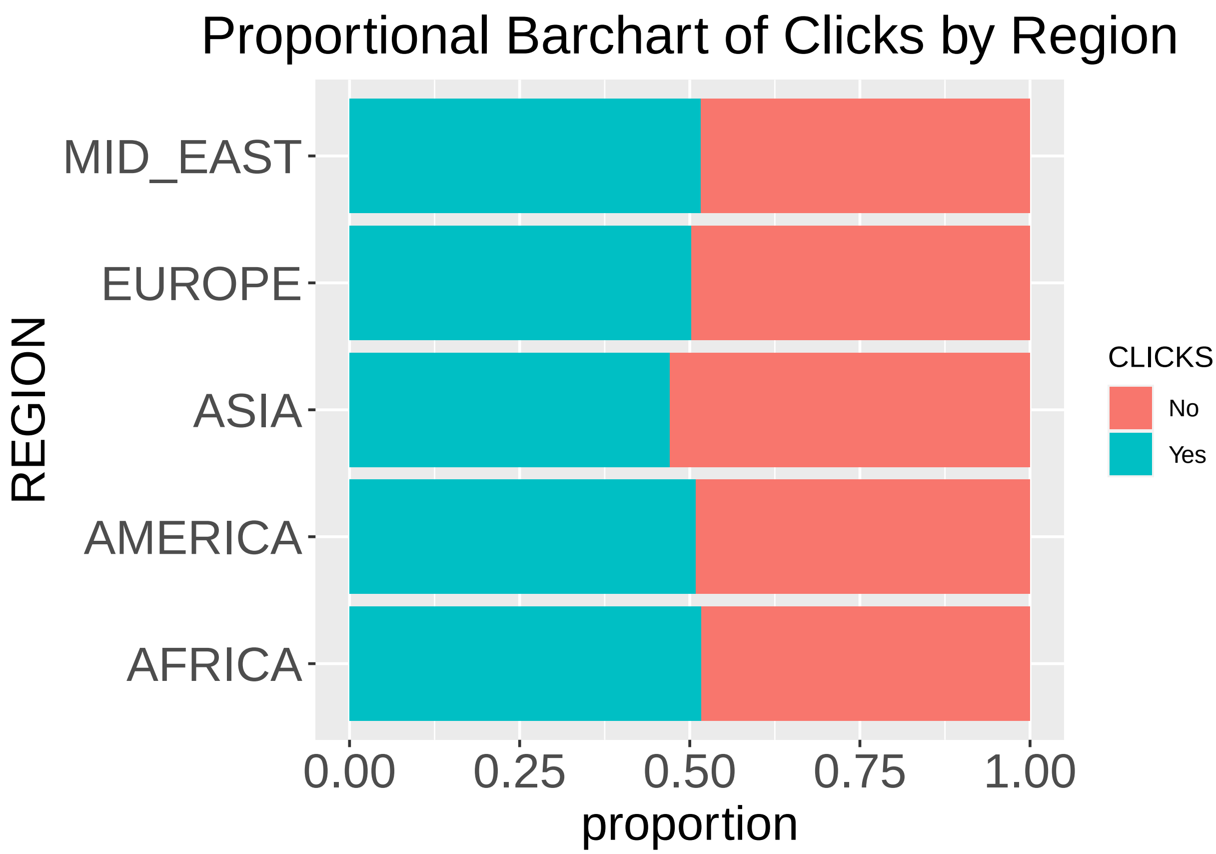

```
#There are more clicks in the Wee Hours of the Night than Morning, Night and afternoon.
#The least number of clicks were registered at night.
#Still, the wee hours also registered the highest number of no clicks.
#Night hours offered the least number of zero activities.


#Bar chart showing how the regions compared by the Hour
ggplot(advert, aes(x = REGION, fill = HOUR)) +
  geom_bar(position= "dodge") +
  theme(axis.text.x = element_text(angle = 90)) +
  ggtitle("Barchart of Region by Hours") +  theme(axis.text = element_text(size=18),
                                                  axis.title = element_text(size = 18),
                                                  plot.title = element_text(hjust = 0.5, size = 20))
```



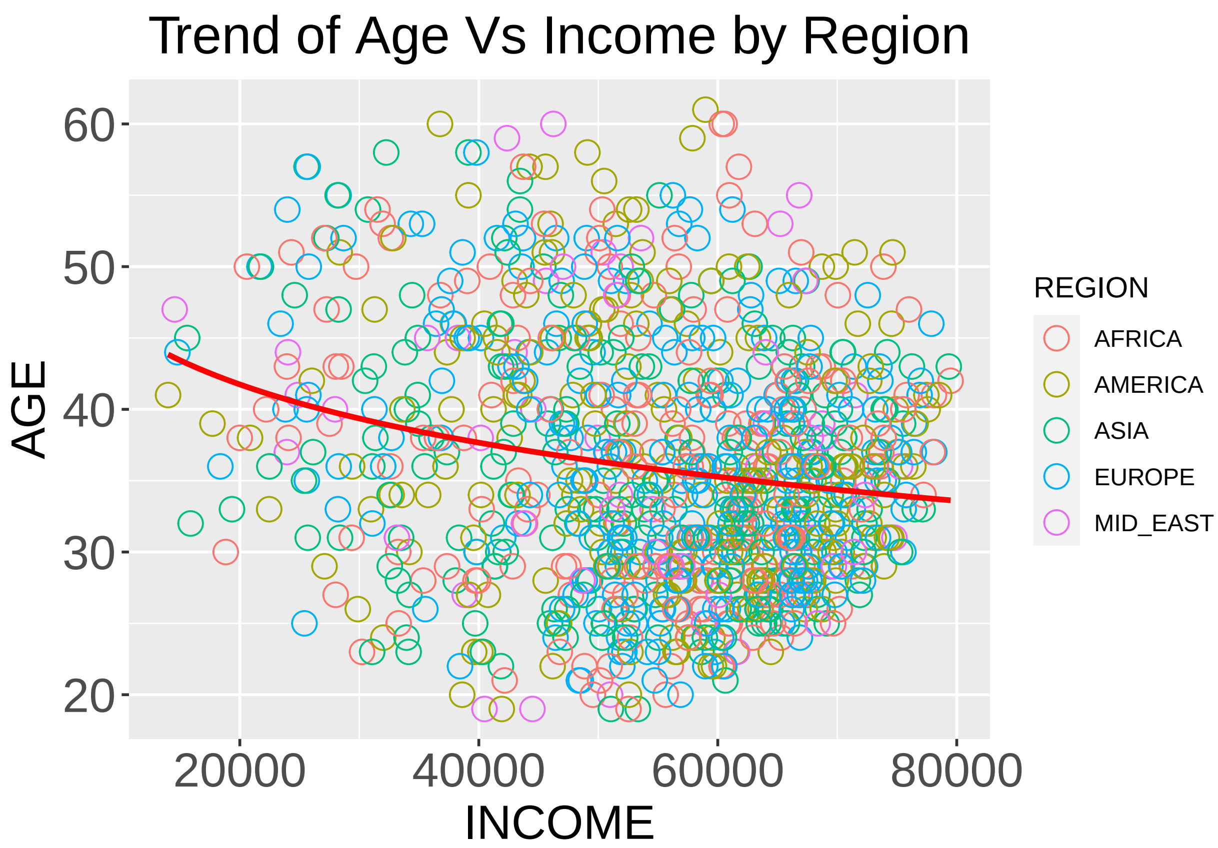

```
#In the African region, the afternoon and wee hours were the most active
#In the European region, morning hours were the most active whilst the Night was
#quieter
#Generally, the wee hours were the busiest in the regions. Only the Mid_East had
#Night time as the busiest
#Comparatively, it was less busy in the Mid_East at any point in time than in any
#other region

# Creating a side-by-side barchart of Region by gender
ggplot(advert, aes(x = REGION, fill = GENDER)) +
  geom_bar(position= "dodge") +
  theme(axis.text.x = element_text(angle = 90)) +
  ggtitle("Barchart of Region according to Gender") +  theme(axis.text = element_text(size=18),
                                                              axis.title = element_text(size = 18),
                                                              plot.title = element_text(hjust = 0.5, siz
```



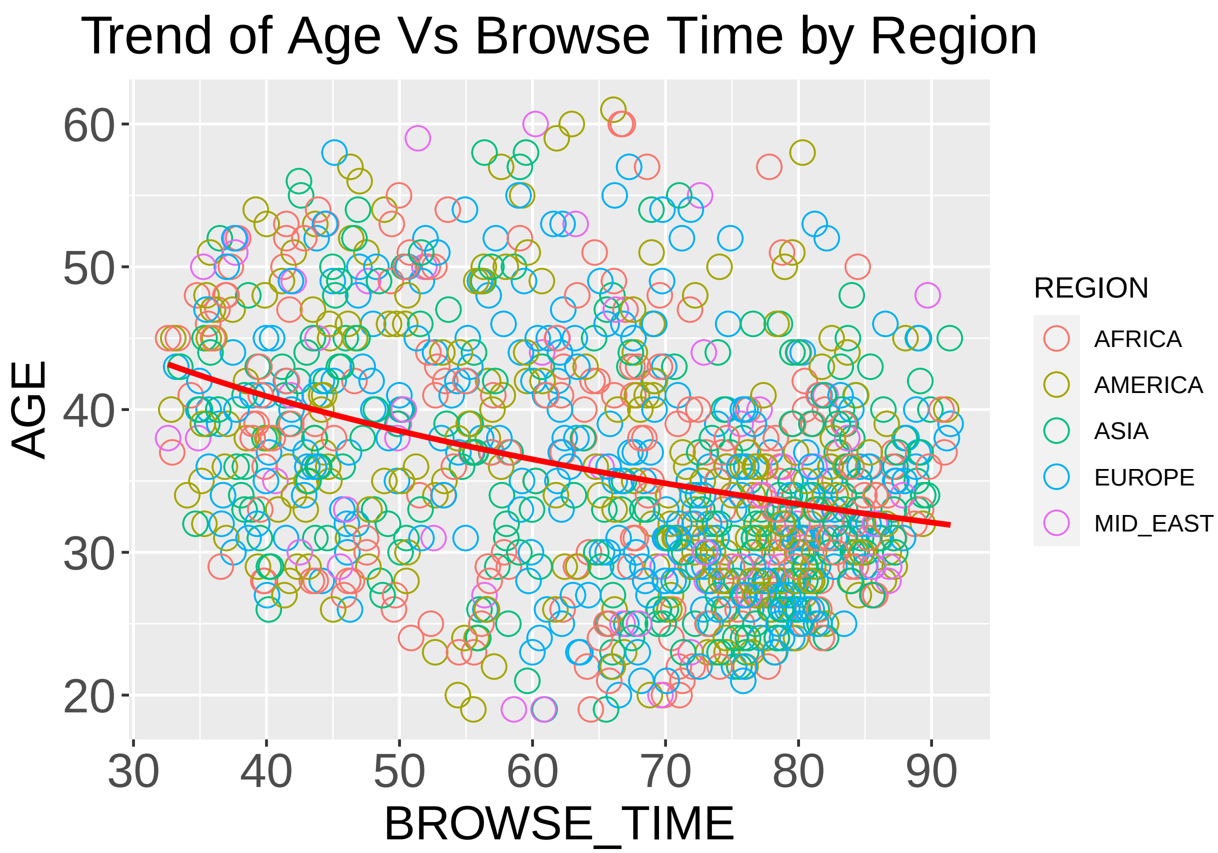

```
#With the exception of America's region, women were the majority across all other regions.

#Europe was highly represented compared to other regions while the middle East
#was the least represented.

#Plot proportion of Clicks, conditional on Region
ggplot(advert, aes(x = REGION, fill = CLICKS)) +
  geom_bar(position = "fill") + coord_flip() +
  ylab("proportion") +
  ggtitle("Proportional Barchart of Clicks by Region") +  theme(axis.text = element_text(size=18),
                                                                axis.title = element_text(size = 18),
                                                                plot.title = element_text(hjust = 0.5, s
```



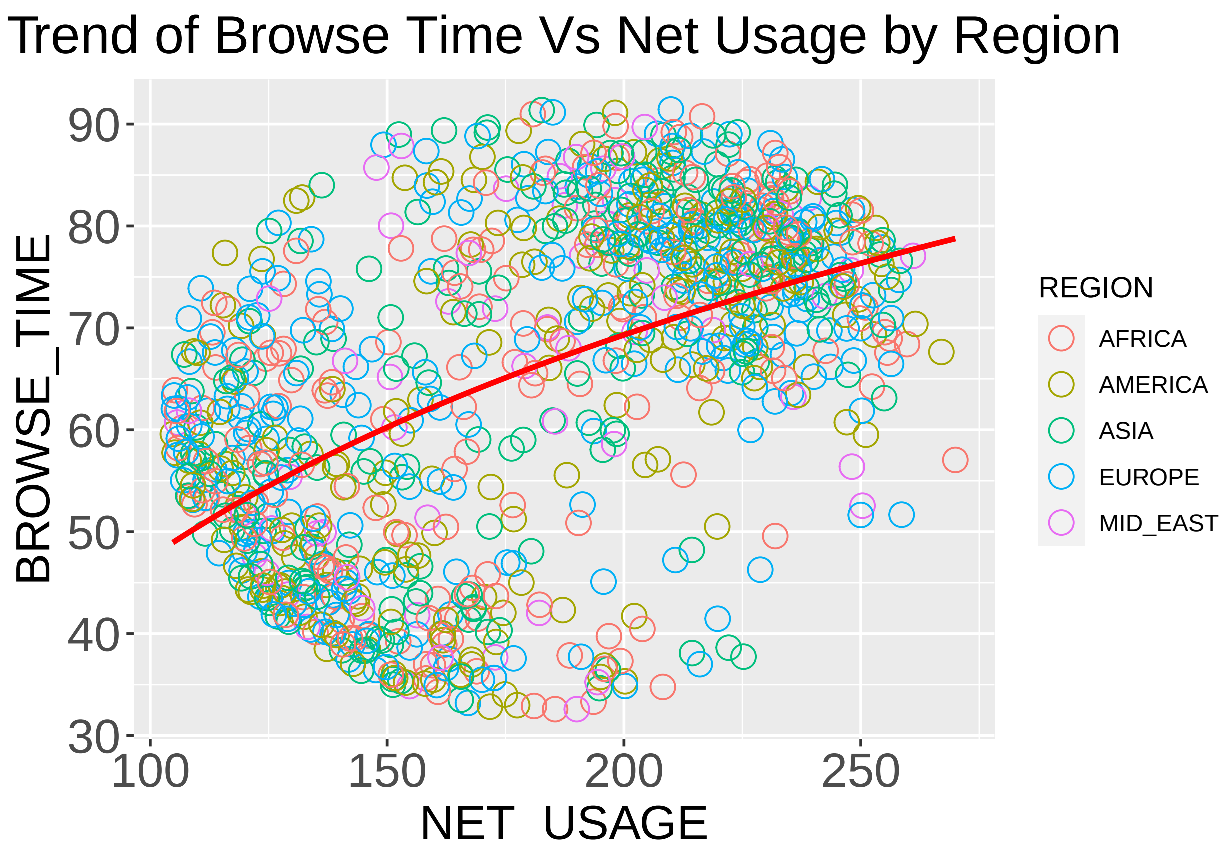

```
#Of all the regions, Asia had the most clicks.

#The Middle East, Europe, Africa and the America's had pretty much the same proportion of Clicks.

#The Regions aforementioned oscillated around 50% clicks and no clicks

ggplot(advert, aes(x = INCOME, y=AGE, color = REGION)) +
  geom_point(size = 4, shape=1) +
  geom_smooth(aes(group=1), method= 'lm', formula = y~log(x), se=F, color ='red') +
  ggtitle("Trend of Age Vs Income by Region") +  theme(axis.text = element_text(size=18),
                                                       axis.title = element_text(size = 18),
                                                       plot.title = element_text(hjust = 0.5, size = 20)
```




```
#Although the trend line suggests a drop in age as the income increases,
#we cannot tell with certainty whether there is any relationships with the fall by region.
#However, we can see a huge concentration of Europe, Asia and America around age 35 and income levels o

#Trend of Age Vs Browse Time by Region
ggplot(advert, aes(x = BROWSE_TIME, y=AGE, color = REGION)) +
  geom_point(size = 4, shape=1) +
  geom_smooth(aes(group=1), method= 'lm', formula = y~log(x), se=F, color ='red') +
  ggtitle("Trend of Age Vs Browse Time by Region") +  theme(axis.text = element_text(size=18),
                                                            axis.title = element_text(size = 18),
                                                            plot.title = element_text(hjust = 0.5, size
```




```
#Generally, the Browse_time tends to increase as the age reduces. We still see great concentration in A


#Trend of Browse TIme vs Net Usage by Region
ggplot(advert, aes(x = NET_USAGE, y=BROWSE_TIME, color = REGION)) +
  geom_point(size = 4, shape=1) +
  geom_smooth(aes(group=1), method= 'lm', formula = y~log(x), se=F, color ='red') +
  ggtitle("Trend of Browse Time Vs Net Usage by Region") +  theme(axis.text = element_text(size=18),
                                                                  axis.title = element_text(size = 18),
                                                                  plot.title = element_text(hjust = 0.5
```




```
#Net usage increases as the Browse time increases.



#FOLLOW UP QUESTIONS

#Reflecting on whether we have achieved the objectives we set out

#1. Did we have the right data? Yes, we did

#2. Do we need other Data top answer our question? Yes, it would go along way in
#explaining and validating certain observations in the current dataset e.g
#why they is a 50% chance of CLicking or not clicking an add & a fair representation
#of countries in the Mid_East


#3. Did we have the right Question? Yes, we did.


#COnclusions & Recommendations

#In conclusion, women are the least likely to click on a link.
#Perhaps focus should be placed on items or topics likely to get women interested in clicking a link.

#Men are most likely to click a link. We recommend that the be targeted the most.
#A lot of traffic be directed to men.
```

```r
#Clearly the afternoons are the worst possible times to advertise online.
#It appears the wee hours of the night are the best times to advertise Crypto topics.

#Asia is clearly a key focus area as most of the clicks were registered there
#


#MODEL DEVELOPMENT

#In this section we will create a supervised learning model that would
#help identify the individuals most likely to click on the ads in the blog

#1. Data Preprocessing

#Checking & Confirming that the Dataset is machine learning worthy

#Renaming the Dataset to avoid confusion as we manipulate it further

df <- advert
#Previewing the columns


str(df)
```

```
## Classes 'data.table' and 'data.frame':   1000 obs. of  9 variables:
##  $ BROWSE_TIME: num  69 80.2 69.5 74.2 68.4 ...
##  $ AGE        : int  35 31 26 29 35 23 33 48 30 20 ...
##  $ INCOME     : num  61834 68442 59786 54806 73890 ...
##  $ NET_USAGE  : num  256 194 236 246 226 ...
##  $ GENDER     : chr  "Female" "Male" "Female" "Male" ...
##  $ COUNTRY    : chr  "Tunisia" "Nauru" "San Marino" "Italy" ...
##  $ CLICKS     : chr  "No" "No" "No" "No" ...
##  $ REGION     : chr  "AFRICA" "ASIA" "EUROPE" "EUROPE" ...
##  $ HOUR       : chr  "Wee Hours" "Wee Hours" "Night" "Wee Hours" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
#We have categorical variables in the dataset i.e'., Gender, Country
#Clicks, Region & Hour

#Dropping the Country column as we have the Regions column in the dataset
#which contains the countries anyway

df <- select(df, -COUNTRY)

#Checking the Dimensions

dim(df)
```

```
## [1] 1000    8
```

```r
#We need to convert the four remaining categorical variables into a format
#our machine learning model can comprehend

#Encoding the Categorical columns

#CLICKS
df$CLICKS <- ifelse(df$CLICKS== "No", 1, 2)

#GENDER
df$GENDER <- ifelse(df$GENDER== "Female", 0, 1)

#REGION
df$REGION <- ifelse(df$REGION == "AFRICA", 1,
                    ifelse(df$REGION == "ASIA", 2,
                           ifelse(df$REGION == "EUROPE", 3,
                                  ifelse(df$REGION == "MID_EAST", 4, 5))
                    ))

#Hour

df$HOUR <- ifelse(df$HOUR == "Wee Hours", 1,
                  ifelse(df$HOUR == "Morning", 2,
                         ifelse(df$HOUR == "Afternoon", 3, 4)))




#Confirming the changes
head(df)
```

```
##    BROWSE_TIME AGE   INCOME NET_USAGE GENDER CLICKS REGION HOUR
## 1:       68.95  35 61833.90    256.09      0      1      1    1
## 2:       80.23  31 68441.85    193.77      1      1      2    1
## 3:       69.47  26 59785.94    236.50      0      1      3    4
## 4:       74.15  29 54806.18    245.89      1      1      3    1
## 5:       68.37  35 73889.99    225.58      0      1      3    1
## 6:       59.99  23 59761.56    226.74      1      1      3    4
```

```r
#Factorizing the independent variable
df$CLICKS <- as.factor(df$CLICKS)




#K-Nearest Neighbours (KNN)
#KNN is a supervised learning algorithm that can perform both regression
#and classification. It predicts the correct class for the test data by calculating
#the distance between the test data and all the training points.

#Given our problem, KNN could help create an algorithm to predict whether a user
#with the various features in our dataset will click an advert.
#Some of the Merits of KNN include: training is not complex, it works with any number
#of classes, it is easy to add more data and it requires very few parameters
```

```
#Splitting the Dataset into Training & Test set

#Setting a seed

set.seed(32)

# splitting dataset into train and test

split = initial_split(df, prop = 0.65)
train = training(split)
test = testing(split)

#Confirming the dataset is split as anticipated
dim(train)
```

```
## [1] 650   8
```

```
dim(test)
```

```
## [1] 350   8
```

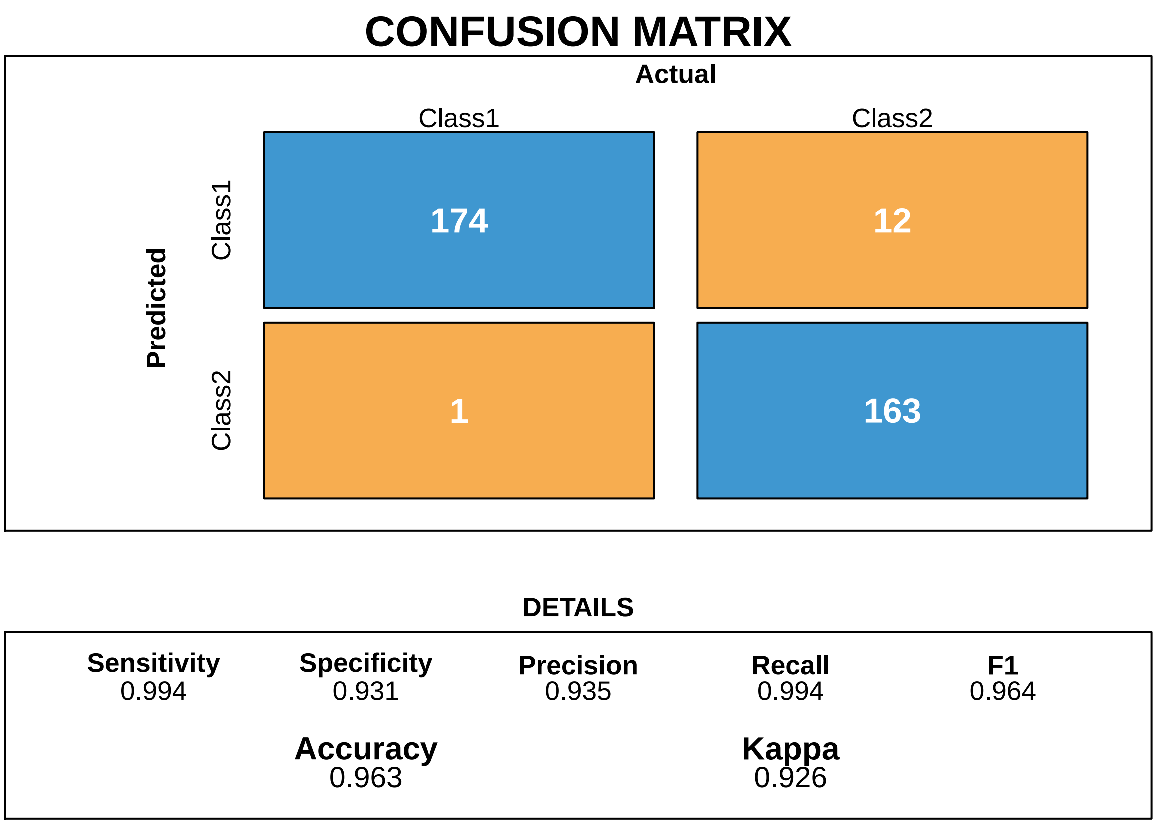
```
#Creating the KNN model

#First using the traincontrol method from Caret, we specify the parameters.
# We start with 10 resampling iterations and 3 complete sets of folds to
#recompute the cross-validation
trctrl = trainControl(method = "repeatedcv", number = 10, repeats = 3 )


# Training

#The Browse time, Age, Income and net usage are not scaled and this could
#be problematic. Most machine learning algorithms work with euclidean distances
#and not scaling the data assures of wrong results. Thus we include preProcess a
#step to ensure our data is scaled and centered.
knn.mod <- train(CLICKS ~., train, method = "knn", trControl = trctrl,
                 preProcess = c("center", "scale"), tuneLength = 10 )

knn.mod
```

```
## k-Nearest Neighbors
##
## 650 samples
##   7 predictor
##   2 classes: '1', '2'
##
## Pre-processing: centered (7), scaled (7)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 585, 586, 584, 585, 584, 585, ...
## Resampling results across tuning parameters:
##
```

```
##    k   Accuracy   Kappa
##    5  0.9518347  0.9036557
##    7  0.9548963  0.9097804
##    9  0.9538551  0.9077019
##   11  0.9564275  0.9128423
##   13  0.9538865  0.9077660
##   15  0.9538789  0.9077488
##   17  0.9564435  0.9128826
##   19  0.9585031  0.9169975
##   21  0.9564591  0.9129122
##   23  0.9549204  0.9098302
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 19.
```

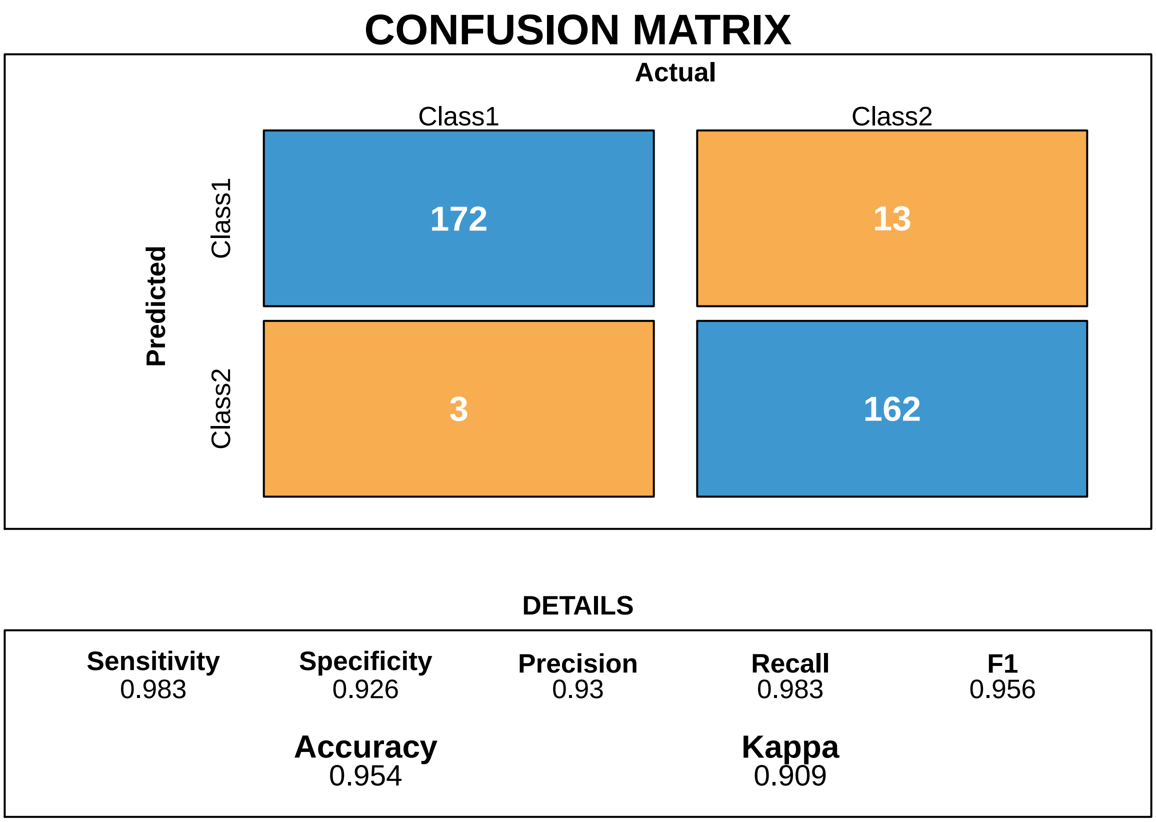
```r
#Using predict() method to predict the results

pred = predict(knn.mod, newdata = test)

# Obtaining the confusion matrix

plot.knn <- confusionMatrix(pred, test$CLICKS)

#Function to Visualize Our results
draw_confusion_matrix <- function(cm) {

  layout(matrix(c(1,1,2)))
  par(mar=c(2,2,2,2))
  plot(c(100, 345), c(300, 450), type = "n", xlab="", ylab="", xaxt='n', yaxt='n')
  title('CONFUSION MATRIX', cex.main=2)

  # create the matrix
  rect(150, 430, 240, 370, col='#3F97D0')
  text(195, 435, 'Class1', cex=1.2)
  rect(250, 430, 340, 370, col='#F7AD50')
  text(295, 435, 'Class2', cex=1.2)
  text(125, 370, 'Predicted', cex=1.3, srt=90, font=2)
  text(245, 450, 'Actual', cex=1.3, font=2)
  rect(150, 305, 240, 365, col='#F7AD50')
  rect(250, 305, 340, 365, col='#3F97D0')
  text(140, 400, 'Class1', cex=1.2, srt=90)
  text(140, 335, 'Class2', cex=1.2, srt=90)

  # add in the cm results
  res <- as.numeric(cm$table)
  text(195, 400, res[1], cex=1.6, font=2, col='white')
  text(195, 335, res[2], cex=1.6, font=2, col='white')
  text(295, 400, res[3], cex=1.6, font=2, col='white')
  text(295, 335, res[4], cex=1.6, font=2, col='white')

  # add in the specifics
  plot(c(100, 0), c(100, 0), type = "n", xlab="", ylab="", main = "DETAILS", xaxt='n', yaxt='n')
  text(10, 85, names(cm$byClass[1]), cex=1.2, font=2)
  text(10, 70, round(as.numeric(cm$byClass[1]), 3), cex=1.2)
```

```
  text(30, 85, names(cm$byClass[2]), cex=1.2, font=2)
  text(30, 70, round(as.numeric(cm$byClass[2]), 3), cex=1.2)
  text(50, 85, names(cm$byClass[5]), cex=1.2, font=2)
  text(50, 70, round(as.numeric(cm$byClass[5]), 3), cex=1.2)
  text(70, 85, names(cm$byClass[6]), cex=1.2, font=2)
  text(70, 70, round(as.numeric(cm$byClass[6]), 3), cex=1.2)
  text(90, 85, names(cm$byClass[7]), cex=1.2, font=2)
  text(90, 70, round(as.numeric(cm$byClass[7]), 3), cex=1.2)

  # add in the accuracy information
  text(30, 35, names(cm$overall[1]), cex=1.5, font=2)
  text(30, 20, round(as.numeric(cm$overall[1]), 3), cex=1.4)
  text(70, 35, names(cm$overall[2]), cex=1.5, font=2)
  text(70, 20, round(as.numeric(cm$overall[2]), 3), cex=1.4)
}


draw_confusion_matrix(plot.knn)
```




```
#The model is 96.3% accurate


#Parameter Tuning to further improve performance of the model

search_grid <- expand.grid( k = seq(1, 25, by=2))
```

```r
#Checking our results with the search grid included

knn.ad <- train(CLICKS ~., train, method ="knn", trControl = trctrl, preProcess= c("center", "scale"),
                tuneGrid = search_grid)


# Checking how the model performs
pred = predict(knn.ad, newdata = test)


plot.knn.ad <- confusionMatrix(pred, test$CLICKS)

draw_confusion_matrix(plot.knn.ad)
```




```r
#Our Model performance does not improve with the further tuning of parameters
#the optimal k= 7


#Challenging the Solution

#Random Forest

#Leveraging the Carets Library we only change the method from knn to random forest using ranger

#Creating Random Forest model and training data
```

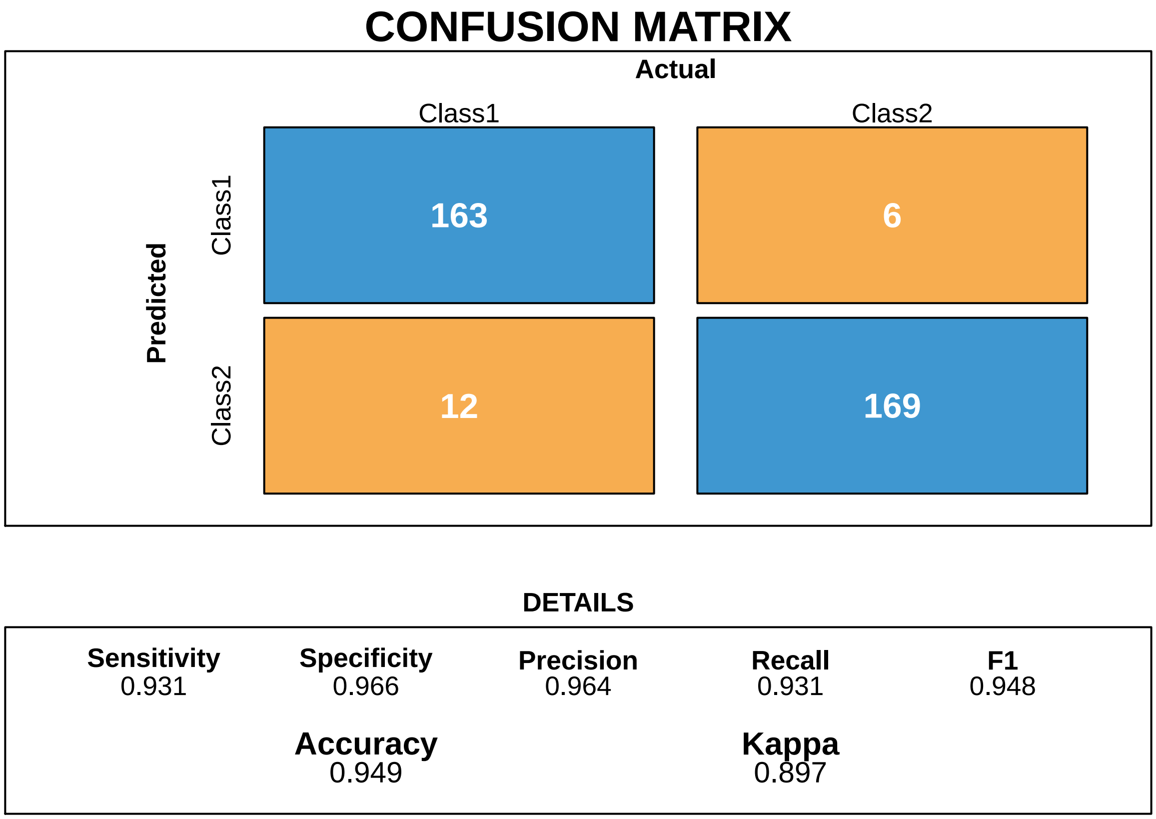
```r
trctrl = trainControl(method = "repeatedcv", number = 10, repeats = 3 )

random.forest <- train(CLICKS ~., train, method = "ranger", trControl = trctrl,
                 preProcess = c("center", "scale"), tuneLength = 10 )
```

```
## note: only 6 unique complexity parameters in default grid. Truncating the grid to 6 .
```

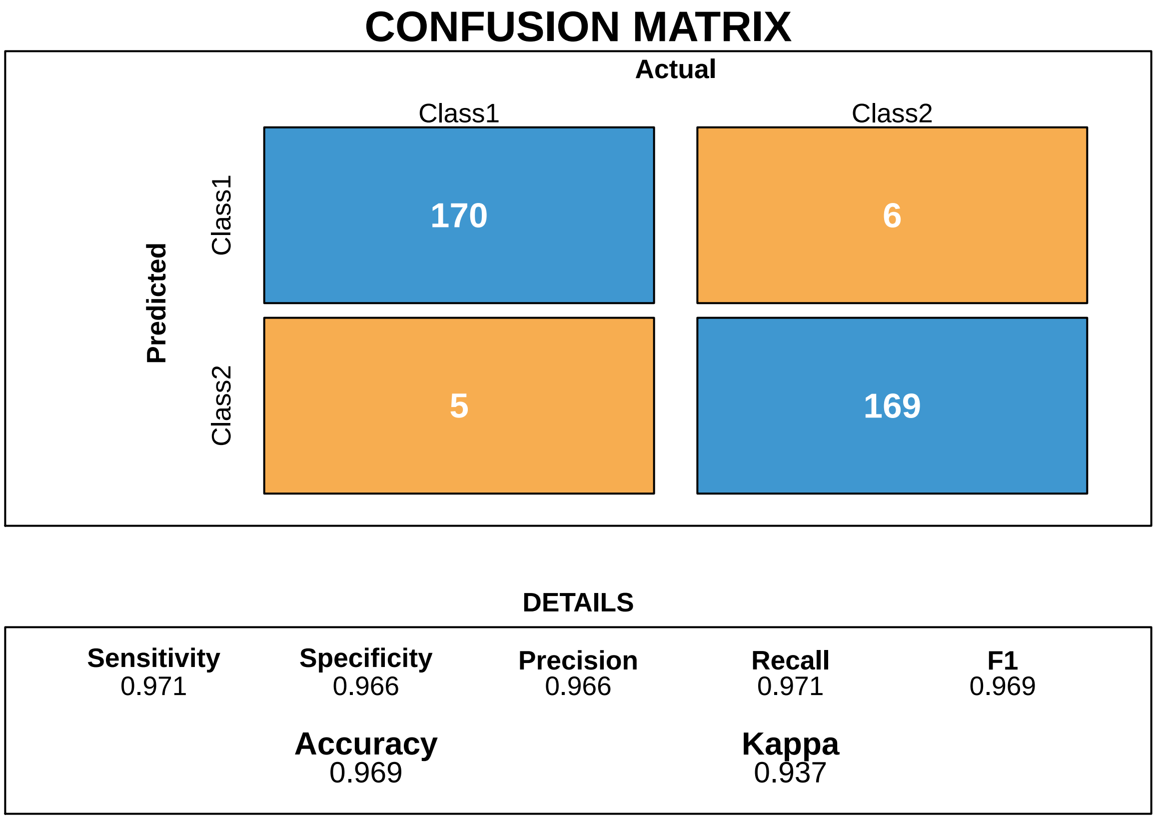
```r
random.forest
```

```
## Random Forest
##
## 650 samples
##   7 predictor
##   2 classes: '1', '2'
##
## Pre-processing: centered (7), scaled (7)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 584, 584, 586, 586, 585, 586, ...
## Resampling results across tuning parameters:
##
##   mtry  splitrule   Accuracy   Kappa
##   2     gini        0.9723422  0.9446846
##   2     extratrees  0.9636932  0.9273827
##   3     gini        0.9723735  0.9447477
##   3     extratrees  0.9626362  0.9252668
##   4     gini        0.9698169  0.9396351
##   4     extratrees  0.9646797  0.9293541
##   5     gini        0.9693354  0.9386708
##   5     extratrees  0.9651926  0.9303815
##   6     gini        0.9662660  0.9325279
##   6     extratrees  0.9657139  0.9314251
##   7     gini        0.9606320  0.9212561
##   7     extratrees  0.9651930  0.9303853
##
## Tuning parameter 'min.node.size' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were mtry = 3, splitrule = gini
##  and min.node.size = 1.
```

```r
#Fitting model to the test data

pred <- predict(random.forest, newdata=test)


#Evaluating the model performance
plot.randomforest <- confusionMatrix(pred, test$CLICKS)

draw_confusion_matrix(plot.randomforest)
```




```r
#With Random Forest the accuracy for classification drops by 2%



#Support Vector Machine(SVM)


trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)

svm.mod <- train(CLICKS ~., data = train, method = "svmRadial",
                 trControl=trainControl(method = "repeatedcv", number = 10, repeats =3),
                 preProcess = c("center", "scale"),
                 tuneLength = 10)

#Fitting the model to the test data


pred <- predict(svm.mod, newdata= test)

plot.svm <- confusionMatrix(pred, test$CLICKS)

draw_confusion_matrix(plot.svm)
```




```
#The SVM model has the most accurate prediction of 97%


#Model Development Conclusion
#The K Nearest Neighbour Model has a decent accuracy of 96% and is not improved by the
#parameter tuning endevours

#The Linear Support Vector Machine Model we build to challenge the results of the KNN
#Model performs much better than the KNN Model. We achieve a classification accuracy of
#circa 97%. It also has the highest precision and a high specificity output. This means
#it better responds to the question of how many people who didn't click on the advert were
#correctly classified as such and an even higher Sensitivity (more people who clicked ad add were corre

#With Random Forest, there was a significant drop in accuracy and it underperformed on Sensitivity,
#Specificity and precision compared to KNN and the SVM models


#We therefore implement our classification prediction using the Linear Support Vector Machine.


#We have also achieved our objective of building a model with at least 95% accuracy
```