Jackson Ip Week 13 part 1

Jackson Kyalo

8/26/2021

# Define the 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.

# The metric for success

This project will be successful if we are able to determine which individuals are most likely to click on the ads.

# The Outline context

The number of clicks an ad has helps understand how well the ad is being received by its audience. Ads that are targeted to the right audience receive the highest number of clicks. In our case determining the best audience for the ads will help company grow as well as increase the number of clicks and reach.

# Experimental design

1. Define the Questions.
2. Import, load and preview the data.
3. Data Cleaning.
4. Data Analysis.
5. Conclusion and Recommendation.

### Importing the libraries

#Import the data library  
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 3.1.3 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 2.0.1 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(caret)

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

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

library(caretEnsemble)

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

##   
## Attaching package: 'caretEnsemble'

## The following object is masked from 'package:ggplot2':  
##   
## autoplot

library(psych)

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

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library(Amelia)

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

## Loading required package: Rcpp

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

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

library(mice)

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

##   
## Attaching package: 'mice'

## The following object is masked from 'package:stats':  
##   
## filter

## The following objects are masked from 'package:base':  
##   
## cbind, rbind

library(GGally)

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

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

library(rpart)

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

library(randomForest)

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

## randomForest 4.6-14

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

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:psych':  
##   
## outlier

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

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

### Load the dataset

#Load our data  
dt=read.csv('C:/Users/Rino/Desktop/Remote/advertising.csv')

### Preview the data

# preview the head  
head(dt)

## 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

#Change the male column name to be gender  
names(dt)[names(dt)== 'Male']<-'Gender'

### Preview tail

tail(dt)

## Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage  
## 995 43.70 28 63126.96 173.01  
## 996 72.97 30 71384.57 208.58  
## 997 51.30 45 67782.17 134.42  
## 998 51.63 51 42415.72 120.37  
## 999 55.55 19 41920.79 187.95  
## 1000 45.01 26 29875.80 178.35  
## Ad.Topic.Line City Gender  
## 995 Front-line bifurcated ability Nicholasland 0  
## 996 Fundamental modular algorithm Duffystad 1  
## 997 Grass-roots cohesive monitoring New Darlene 1  
## 998 Expanded intangible solution South Jessica 1  
## 999 Proactive bandwidth-monitored policy West Steven 0  
## 1000 Virtual 5thgeneration emulation Ronniemouth 0  
## Country Timestamp Clicked.on.Ad  
## 995 Mayotte 2016-04-04 03:57:48 1  
## 996 Lebanon 2016-02-11 21:49:00 1  
## 997 Bosnia and Herzegovina 2016-04-22 02:07:01 1  
## 998 Mongolia 2016-02-01 17:24:57 1  
## 999 Guatemala 2016-03-24 02:35:54 0  
## 1000 Brazil 2016-06-03 21:43:21 1

### Check the info

str(dt)

## '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 standardization" "Organic bottom-line service-desk" "Triple-buffered reciprocal time-frame" ...  
## $ City : chr "Wrightburgh" "West Jodi" "Davidton" "West Terrifurt" ...  
## $ Gender : int 0 1 0 1 0 1 0 1 1 1 ...  
## $ Country : chr "Tunisia" "Nauru" "San Marino" "Italy" ...  
## $ Timestamp : chr "2016-03-27 00:53:11" "2016-04-04 01:39:02" "2016-03-13 20:35:42" "2016-01-10 02:31:19" ...  
## $ Clicked.on.Ad : int 0 0 0 0 0 0 0 1 0 0 ...

#dt$Date <- as.Date(df$Timestamp)  
#df$Time <- format(df$Timestamp,"%H:%M:%S")

### Check the shape

dim(dt)

## [1] 1000 10

#Our code has 1000 rows and 10 columns

# Data Cleaning

### Check for missing data(Null values)

sum(is.na(dt))

## [1] 0

Our data has no missing data

### Check for duplicates

#checking for duplicates  
duplicated <- dt[duplicated(dt),]  
duplicated

## [1] Daily.Time.Spent.on.Site Age Area.Income   
## [4] Daily.Internet.Usage Ad.Topic.Line City   
## [7] Gender Country Timestamp   
## [10] Clicked.on.Ad   
## <0 rows> (or 0-length row.names)

There are no duplicated rows/values in our data

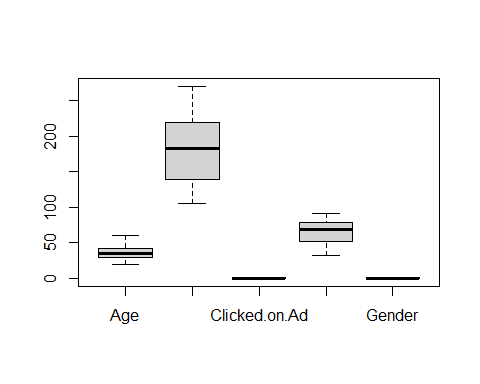
### Check for outliers

### Identify numeric cols  
nums <- unlist(lapply(dt, is.numeric))   
y<- colnames(dt[nums])  
y

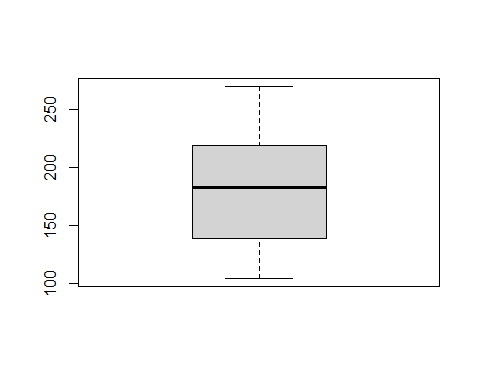
## [1] "Daily.Time.Spent.on.Site" "Age"   
## [3] "Area.Income" "Daily.Internet.Usage"   
## [5] "Gender" "Clicked.on.Ad"

### Check fo outliers

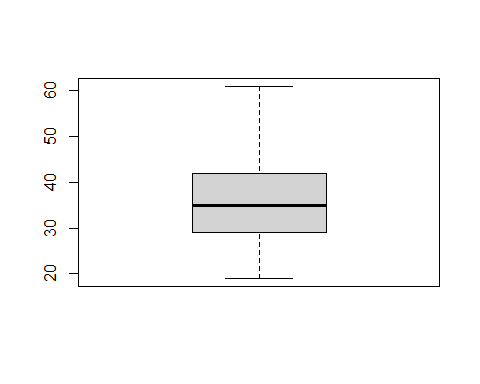
boxplot(dt[c('Age','Daily.Internet.Usage','Clicked.on.Ad','Daily.Time.Spent.on.Site','Gender')])



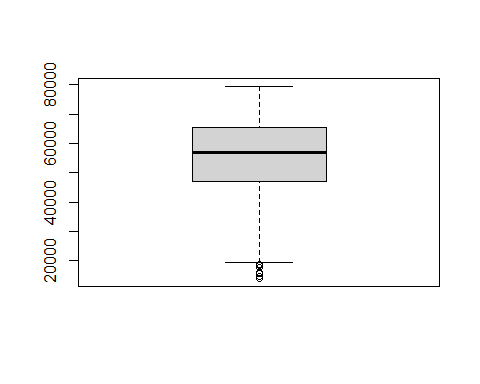
# checking for outliers on Daily Internet Usage  
boxplot(dt$Daily.Internet.Usage)



# checking for outliers on Age  
boxplot(dt$Age)



# checking for outliers on Area.Income  
boxplot(dt$Area.Income)

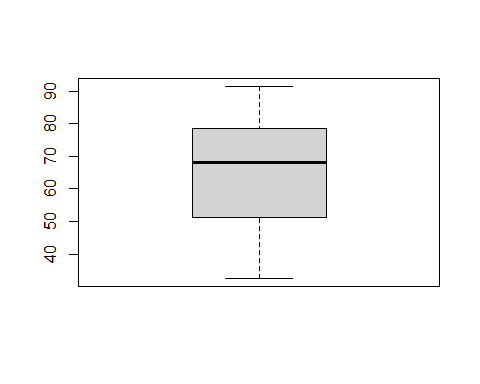
 There are outliers in area income column

boxplot.stats(dt$Area.Income)$out

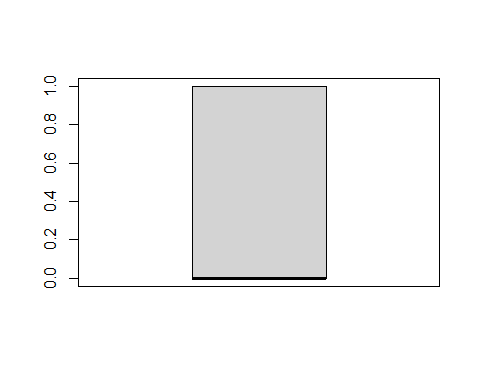
## [1] 17709.98 18819.34 15598.29 15879.10 14548.06 13996.50 14775.50 18368.57

#checking the values in area income that are outliers

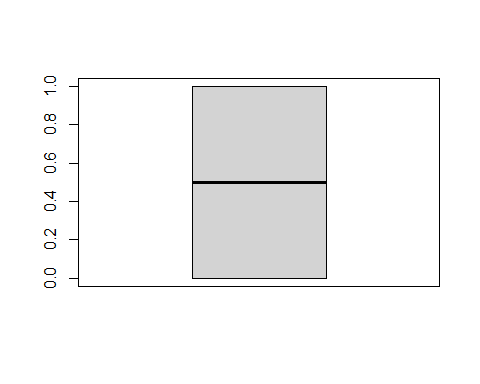
# checking for outliers on Daily.Time.Spent.on.Site  
boxplot(dt$Daily.Time.Spent.on.Site)



# checking for outliers on Male  
boxplot(dt$Gender)



# checking for outliers on Clicked.on.Ad  
boxplot(dt$Clicked.on.Ad)

 There are no outliers in our data except Area.Income.

# Data Analysis

## Univarient Analysis

### Measure of central tendacy

describe(dt)

## vars n mean sd median trimmed mad  
## Daily.Time.Spent.on.Site 1 1000 65.00 15.85 68.22 65.74 17.92  
## Age 2 1000 36.01 8.79 35.00 35.51 8.90  
## Area.Income 3 1000 55000.00 13414.63 57012.30 56038.94 13316.62  
## Daily.Internet.Usage 4 1000 180.00 43.90 183.13 179.99 58.61  
## Ad.Topic.Line\* 5 1000 500.50 288.82 500.50 500.50 370.65  
## City\* 6 1000 487.32 279.31 485.50 487.51 356.57  
## Gender 7 1000 0.48 0.50 0.00 0.48 0.00  
## Country\* 8 1000 116.41 69.94 114.50 115.82 89.70  
## Timestamp\* 9 1000 500.50 288.82 500.50 500.50 370.65  
## Clicked.on.Ad 10 1000 0.50 0.50 0.50 0.50 0.74  
## min max range skew kurtosis se  
## Daily.Time.Spent.on.Site 32.60 91.43 58.83 -0.37 -1.10 0.50  
## Age 19.00 61.00 42.00 0.48 -0.41 0.28  
## Area.Income 13996.50 79484.80 65488.30 -0.65 -0.11 424.21  
## Daily.Internet.Usage 104.78 269.96 165.18 -0.03 -1.28 1.39  
## Ad.Topic.Line\* 1.00 1000.00 999.00 0.00 -1.20 9.13  
## City\* 1.00 969.00 968.00 0.00 -1.19 8.83  
## Gender 0.00 1.00 1.00 0.08 -2.00 0.02  
## Country\* 1.00 237.00 236.00 0.08 -1.23 2.21  
## Timestamp\* 1.00 1000.00 999.00 0.00 -1.20 9.13  
## Clicked.on.Ad 0.00 1.00 1.00 0.00 -2.00 0.02

#Getting the statistical summaries of the data  
summary(dt)

## 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 Gender 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  
## Length:1000 Min. :0.0   
## Class :character 1st Qu.:0.0   
## Mode :character Median :0.5   
## Mean :0.5   
## 3rd Qu.:1.0   
## Max. :1.0

From the above we can see that maximum daily time spent on site is 91 mins while the minimum time spent is 32 mins. In average time spent on the blog is 65 minutes. The maximum age of the customers visiting the 61 years while the minimum age is 19 years. However the average age of viewers is 35 years. The average income earned by their viewers is 55,000 with the maximum amount earned being 79,000 and minimum amount is 13996.

### Measure of dispersion

#create a function  
library(moments)  
summary.list = function(x)list(  
 Mean=mean(x, na.rm=TRUE),  
 Median=median(x, na.rm=TRUE),  
 Skewness=skewness(x, na.rm=TRUE),  
 Kurtosis=kurtosi(x, na.rm=TRUE),  
 Variance=var(x, na.rm=TRUE),  
 Std.Dev=sd(x, na.rm=TRUE),  
 Coeff.Variation.Prcnt=sd(x, na.rm=TRUE)/mean(x, na.rm=TRUE)\*100,  
 Std.Error=sd(x, na.rm=TRUE)/sqrt(length(x[!is.na(x)]))  
)

Calling the function for each column

#For Daily.Time.Spent.on.Site  
summary.list(dt$Daily.Time.Spent.on.Site)

## $Mean  
## [1] 65.0002  
##   
## $Median  
## [1] 68.215  
##   
## $Skewness  
## [1] -0.3712026  
##   
## $Kurtosis  
## [1] -1.099864  
##   
## $Variance  
## [1] 251.3371  
##   
## $Std.Dev  
## [1] 15.85361  
##   
## $Coeff.Variation.Prcnt  
## [1] 24.3901  
##   
## $Std.Error  
## [1] 0.5013353

#For Age  
summary.list(dt$Age)

## $Mean  
## [1] 36.009  
##   
## $Median  
## [1] 35  
##   
## $Skewness  
## [1] 0.4784227  
##   
## $Kurtosis  
## [1] -0.4097066  
##   
## $Variance  
## [1] 77.18611  
##   
## $Std.Dev  
## [1] 8.785562  
##   
## $Coeff.Variation.Prcnt  
## [1] 24.39824  
##   
## $Std.Error  
## [1] 0.2778239

#For Daily.Time.Spent.on.Site  
summary.list(dt$Area.Income)

## $Mean  
## [1] 55000  
##   
## $Median  
## [1] 57012.3  
##   
## $Skewness  
## [1] -0.6493967  
##   
## $Kurtosis  
## [1] -0.1110924  
##   
## $Variance  
## [1] 179952406  
##   
## $Std.Dev  
## [1] 13414.63  
##   
## $Coeff.Variation.Prcnt  
## [1] 24.39024  
##   
## $Std.Error  
## [1] 424.208

#For Daily.Internet.Usage  
summary.list(dt$Daily.Internet.Usage)

## $Mean  
## [1] 180.0001  
##   
## $Median  
## [1] 183.13  
##   
## $Skewness  
## [1] -0.03348703  
##   
## $Kurtosis  
## [1] -1.275752  
##   
## $Variance  
## [1] 1927.415  
##   
## $Std.Dev  
## [1] 43.90234  
##   
## $Coeff.Variation.Prcnt  
## [1] 24.39017  
##   
## $Std.Error  
## [1] 1.388314

#### Summaries when ad is cliecked

#Get the summaries when there is a click  
dt.sub <- subset(dt, Clicked.on.Ad == 1)

Summaries

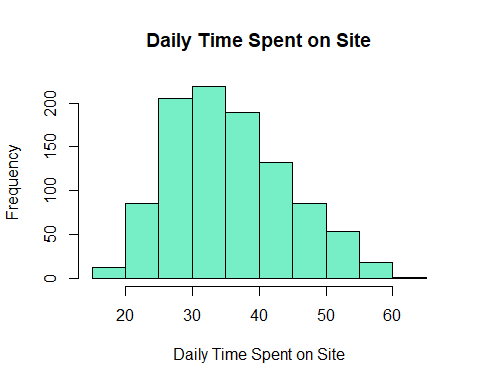
summary(dt.sub)

## Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage  
## Min. :32.60 Min. :19.00 Min. :13996 Min. :104.8   
## 1st Qu.:42.84 1st Qu.:34.00 1st Qu.:39107 1st Qu.:123.6   
## Median :51.53 Median :40.00 Median :49417 Median :138.8   
## Mean :53.15 Mean :40.33 Mean :48614 Mean :145.5   
## 3rd Qu.:62.08 3rd Qu.:47.00 3rd Qu.:59241 3rd Qu.:161.2   
## Max. :91.37 Max. :61.00 Max. :78521 Max. :270.0   
## Ad.Topic.Line City Gender Country   
## Length:500 Length:500 Min. :0.000 Length:500   
## Class :character Class :character 1st Qu.:0.000 Class :character   
## Mode :character Mode :character Median :0.000 Mode :character   
## Mean :0.462   
## 3rd Qu.:1.000   
## Max. :1.000   
## Timestamp Clicked.on.Ad  
## Length:500 Min. :1   
## Class :character 1st Qu.:1   
## Mode :character Median :1   
## Mean :1   
## 3rd Qu.:1   
## Max. :1

When there was a click on the ad, the average time spent was 53 mins, with the average age of the viewers being 40 years. The average income of the viewers who viewed the ads was 48,000 and they spent in an average 145 minutes on the internet.

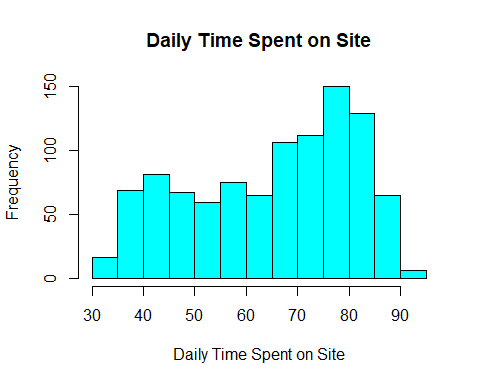
### Distribution of Numeric columns

#For Age  
hist(dt$Age,   
 main = "Daily Time Spent on Site",  
 xlab = "Daily Time Spent on Site",  
 col = "aquamarine2")

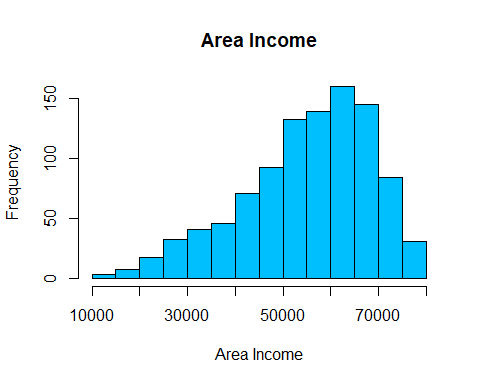


Most respondents fall in the age bracket 25-40 years.

# Histograms for Daily.Time.Spent.on.Site  
hist(dt$Daily.Time.Spent.on.Site,  
 main = "Daily Time Spent on Site",  
 xlab = "Daily Time Spent on Site",  
 col = "cyan1")

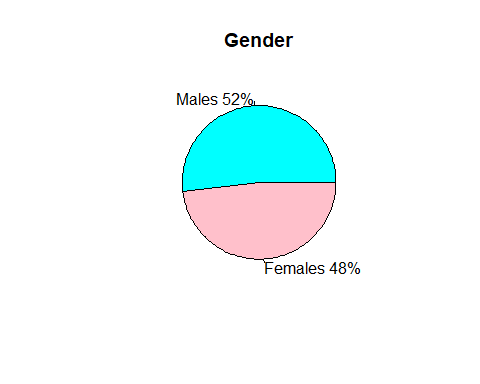
 Daily time speant on site is skewed to the left.Most time spent is between 75 mins to 85 mins.

# Histograms for Area Income  
hist(dt$Area.Income,  
 main = "Area Income",  
 xlab = "Area Income",  
 col = "deepskyblue")

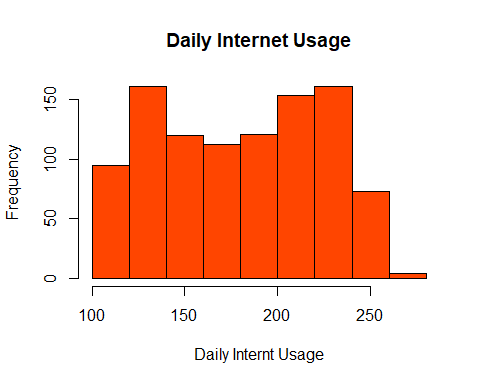
 The area income columns is skewed to the left.Most respondent spend between 55,000 to 7,0000.

# Histograms for Area Income  
df<-table(dt$Gender)

# Create a vector of labels  
lbls<- c("Males", "Females")  
pct <- round(df/sum(df)\*100)  
lbls <- paste(lbls, pct) # add percents to labels  
lbls <- paste(lbls,"%",sep="") # ad % to labels  
pie(df,   
 labels <- lbls,  
 col = c("cyan", "pink"),  
 main="Gender")



# Histograms for Daily.Time.Spent.on.Site  
hist(dt$Daily.Internet.Usage,  
 main = "Daily Internet Usage",  
 xlab = "Daily Internt Usage",  
 col = "orangered")

 ## Bivarient Analysis

### Correlation matrix

cor(dt[,unlist(lapply(dt, is.numeric))])

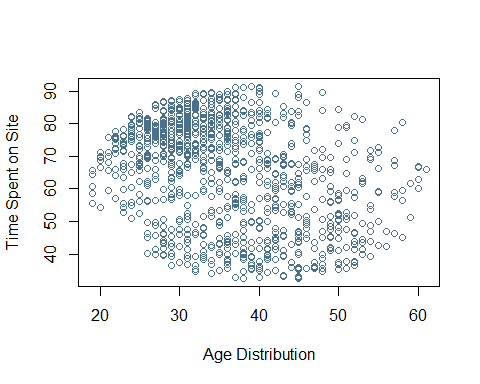
## Daily.Time.Spent.on.Site Age Area.Income  
## Daily.Time.Spent.on.Site 1.00000000 -0.33151334 0.310954413  
## Age -0.33151334 1.00000000 -0.182604955  
## Area.Income 0.31095441 -0.18260496 1.000000000  
## Daily.Internet.Usage 0.51865848 -0.36720856 0.337495533  
## Gender -0.01895085 -0.02104406 0.001322359  
## Clicked.on.Ad -0.74811656 0.49253127 -0.476254628  
## Daily.Internet.Usage Gender Clicked.on.Ad  
## Daily.Time.Spent.on.Site 0.51865848 -0.018950855 -0.74811656  
## Age -0.36720856 -0.021044064 0.49253127  
## Area.Income 0.33749553 0.001322359 -0.47625463  
## Daily.Internet.Usage 1.00000000 0.028012326 -0.78653918  
## Gender 0.02801233 1.000000000 -0.03802747  
## Clicked.on.Ad -0.78653918 -0.038027466 1.00000000

The Table shows the correlations between each columns. The most correlated features are daily internet usage and daily time spent on the site while the least correlated items are clicks on ad and daily internet usage. There is positive correlation between age an clicks on ads.

## Scatter plots

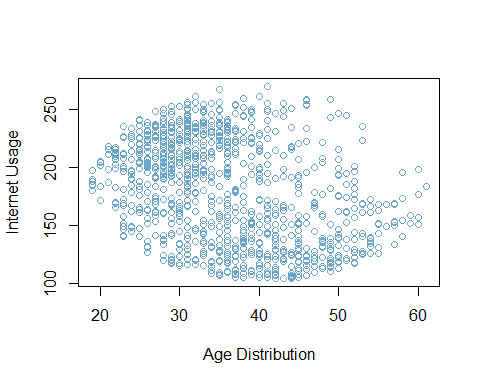
Let’s plot a scatter plot for age and daily time spent on site.

plot(dt$Age,dt$Daily.Time.Spent.on.Site,   
 xlab = "Age Distribution",  
 ylab = "Time Spent on Site",  
 col="skyblue4")

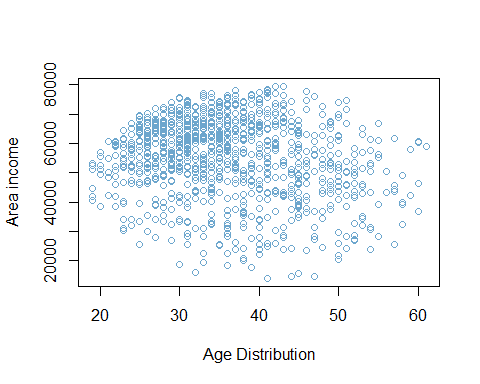
 Most customers spending the largest amount of time in the sites are between 37yrs and 45 years

Let’s plot a scatter plot for age and daily internet usage.

plot(dt$Age,dt$Daily.Internet.Usage,   
 xlab = "Age Distribution",  
 ylab = "Internet Usage",  
 col="skyblue3")

 Let’s plot a scatter plot for age and Area Income.

plot(dt$Age,dt$Area.Income,   
 xlab = "Age Distribution",  
 ylab = "Area income",  
 col="skyblue3")

 Most of the customers with the highest area income are between 40 and 45 years.

### Covariance

#Covariance between age and daily time spent  
cov(dt$Age, dt$Daily.Time.Spent.on.Site)

## [1] -46.17415

The covariance of Age and Daily.Time.Usage variable is about -46.17415, It indicates a negative linear relationship between the two variables

# Covariance between age and daily internet usage   
cov(dt$Age, dt$Daily.Internet.Usage)

## [1] -141.6348

The covariance of Age and Daily.Internet.Usage variable is about -141.6348, It indicates a negative linear relationship between the two variables

#Covariance between age and area income  
cov(dt$Age, dt$Area.Income)

## [1] -21520.93

The covariance of Age and area income variable is about -21520.93, It indicates a negative linear relationship between the two features.

#Covariance between age and clicks  
cov(dt$Age, dt$Clicked.on.Ad)

## [1] 2.164665

The covariance of Age and clicks on ad variable is about 2.164665, It indicates a positive linear relationship between the two features.

#Covariance between age and gender  
cov(dt$Age, dt$Gender)

## [1] -0.09242142

The covariance of Age and gender variable is about -0.09242142, It indicates a negative linear relationship between the two features.

# EDA Conclusion

1. From the above we can see that maximum daily time spent on site is 91 mins while the minimum time spent is 32 mins. In average time spent on the blog is 65 minutes.
2. The maximum age of the customers visiting the 61 years while the minimum age is 19 years. However the average age of viewers is 35 years.
3. The average income earned by their viewers is 55,000 with the maximum amount earned being 79,000 and minimum amount is 13996.
4. When there was a click on the ad, the average time spent was 53 mins, with the average age of the viewers being 40 years. The average income of the viewers who viewed the ads was 48,000 and they spent in an average 145 minutes on the internet.
5. Most respondents fall in the age bracket 25-40 years.
6. Daily time speant on site is skewed to the left.Most time spent is between 75 mins to 85 mins.
7. The area income columns is skewed to the left.Most respondent spend between 55,000 to 7,0000.
8. The Table shows the correlations between each columns. The most correlated features are daily internet usage and daily time spent on the site while the least correlated items are clicks on ad and daily internet usage. There is positive correlation between age an clicks on ads.
9. Most customers spending the largest amount of time in the sites are between 37yrs and 45 years

# EDA Recommendation

1. The ads should target people with an income between 50,000 and 70,000 since they are the people most interested with the ad.
2. We recommend that ads to be tailor to suit viewers of the age group between 25 years and 40 years.
3. Our client should tailor the course to be less than 85 mins or between 75 mins and 85 mins.

# Modelling

## KNN

#preview the data  
head(dt)

## 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 Gender 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

#Drop irrelevant columns  
dt\_new<-dt[-c(5,6,8,9)]  
head(dt\_new)

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

### Normalizing the data and scaling our data

library(caret)  
#we shall use range method as it suppress the effect of outliers  
preproc1 <- preProcess(dt\_new, method=c("range"))  
   
norm1 <- predict(preproc1, dt\_new)  
   
summary(norm1)

## Daily.Time.Spent.on.Site Age Area.Income   
## Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.3189 1st Qu.:0.2381 1st Qu.:0.5044   
## Median :0.6054 Median :0.3810 Median :0.6568   
## Mean :0.5507 Mean :0.4050 Mean :0.6261   
## 3rd Qu.:0.7810 3rd Qu.:0.5476 3rd Qu.:0.7860   
## Max. :1.0000 Max. :1.0000 Max. :1.0000   
## Daily.Internet.Usage Gender Clicked.on.Ad  
## Min. :0.0000 Min. :0.000 Min. :0.0   
## 1st Qu.:0.2061 1st Qu.:0.000 1st Qu.:0.0   
## Median :0.4743 Median :0.000 Median :0.5   
## Mean :0.4554 Mean :0.481 Mean :0.5   
## 3rd Qu.:0.6902 3rd Qu.:1.000 3rd Qu.:1.0   
## Max. :1.0000 Max. :1.000 Max. :1.0

## Split the data; train and test dataset.seed(101) # Set Seed so that same sample can be reproduced in future also

set.seed(123) # Set Seed so that same sample can be reproduced in future also  
# Now Selecting 80% of data as sample from total 'n' rows of the data   
sample <- sample.int(n = nrow(norm1), size = floor(.80\*nrow(norm1)), replace = F)  
train <- norm1[sample, ]  
test <- norm1[-sample, ]  
dim(test)

## [1] 200 6

dim(train)

## [1] 800 6

The test dataset has 200 rows with the train dataset has 800 rows.

### KNN Aligorithm

library(class) #The library contains the aligorithm  
#The total number of rows are 1000. To get the best value of k we shall get the sqrt of the 1000  
sqrt(1000)

## [1] 31.62278

Our value of K = 32 ###Fit the model and evaluate the model

# fitting KNN classifier to the training set and predicting the test set results  
  
y\_pred = knn(train = train[,-6],  
 test = test[,-6],  
 cl = train[,6],  
 k = 32)  
# Creating the confusion matrix  
tb <- table(y\_pred,test[,6])  
tb

##   
## y\_pred 0 1  
## 0 111 4  
## 1 0 85

# Checking the accuracy  
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) \* 100}  
accuracy(tb)

## [1] 98

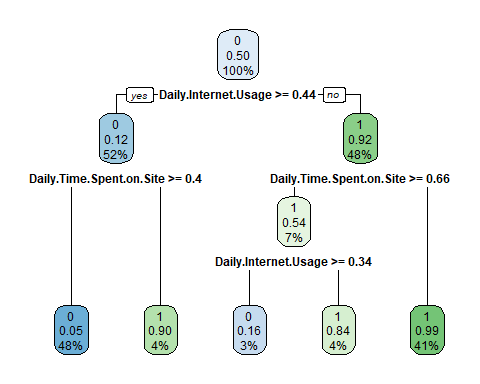
The model has been corrected identified 111 true positive and 85 true negatives with 4 being identified as false positive and 0 as false negatives.The model has achieved an accuracy of 98%

## Decision Trees

library(rpart)  
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.0.5

model <- rpart(formula = Clicked.on.Ad~ ., data = norm1,  
 method = "class")  
  
rpart.plot(model)



#Predicting   
pred <- predict(model, norm1, type = "class")  
#Classification report  
cl\_table<-table(pred, norm1$Clicked.on.Ad)  
cl\_table

##   
## pred 0 1  
## 0 485 28  
## 1 15 472

#Get accuracy  
accuracy(cl\_table)

## [1] 95.7

The model has been corrected identified 485 true positive and 472 true negatives with 28 being identified as false positive and 15 as false negatives.The model has achieved an accuracy of 95.7%

## SVN

#### fit the model and evaluate it

library(e1071)

## Warning: package 'e1071' was built under R version 4.0.5

##   
## Attaching package: 'e1071'

## The following objects are masked from 'package:moments':  
##   
## kurtosis, moment, skewness

model\_svn = svm(formula =Clicked.on.Ad~.,  
 data = train,  
 type = 'C-classification',  
 kernel = 'linear')  
# prediction  
pred\_svn<- predict(model\_svn, newdata = test[-6])  
#Evaluate the model  
#confusion matrix  
clm <- table(test[,6],pred\_svn)  
clm

## pred\_svn  
## 0 1  
## 0 110 1  
## 1 3 86

#accuracy  
accuracy(clm)

## [1] 98

The model has been able to identify 110 true positive and 86 true negatives with 1 being identified as false positive and 3 as false negatives. The accuracy achieved was 98%.

## Naives Bayes

#### Fit the model and evaluate the model

model\_naives = naiveBayes(x = train[-6],  
 y = train$Clicked.on.Ad)  
# Predicting   
pred\_naives = predict(model\_naives, newdata = test[-6])  
#Evaluate the model  
#confusion matrix  
clm\_naives <- table(test[,6],pred\_naives)  
clm\_naives

## pred\_naives  
## 0 1  
## 0 109 2  
## 1 3 86

#accuracy  
accuracy(clm\_naives)

## [1] 97.5

The model has been able to identify 109 true positive and 86 true negatives with 2 being identified as false positive and 3 as false negatives. The accuracy achieved was 97.5%.

# Conclusion

SVN model performed the best with an accuracy score of 98%.