Cryptography Course Advertisement Ad Analysis

Brian Onyango

2022-05-27

Overview

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.

Specifying the Question

Which individuals are most likely to clink on course advertisement ads?

Defining the Metric for Success

This project will be successful if we will be able to determine factors that lead to a user to click an ad.

Understanding the Context

Ad Clicks, or simply Clicks, is a marketing metric that counts the number of times users have clicked on a digital advertisement to reach an online property.If you have a campaign running, you are probably able to access click data on each specific ad. You may see data like this: Ad 1: 4,686 clicks Ad 2: 1,248 clicks Ad 3: 984 clicks You can see that Ad 1 is the higher performing ad by clicks. You may want to evaluate this ad and figure out why audiences tend to click on it more. You may also want to review Ad 3 and try to determine why it is not receiving as many clicks.

Recording the Experimental Design

1. Data sourcing/loading
2. Data Understanding
3. Data Relevance
4. External Dataset Validation
5. Data Preparation
6. Univariate Analysis
7. Bivariate Analysis
8. Multivariate Analysis
9. Conclusion
10. Recommendations

Data Relevance

The dataset avialed the client can be downloaded from this link http://bit.ly/IPAdvertisingData

Loading the Dataset

```
df <- read.csv('http://bit.ly/IPAdvertisingData')
```

Previewing the top of our dataset

```
head(df)
```

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

Previewing the bottom of the dataset

```
tail(df)
```

```
##      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 Male
## 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
```

Checking the data types

```
# Data set structure.
str(df)
```

```
## 'data.frame':    1000 obs. of  10 variables:
##  $ Daily.Time.Spent.on.Site: num  69 80.2 69.5 74.2 68.4 ...
##  $ Age                     : int  35 31 26 29 35 23 33 48 30 20 ...
##  $ Area.Income             : num  61834 68442 59786 54806 73890 ...
##  $ Daily.Internet.Usage    : num  256 194 236 246 226 ...
##  $ Ad.Topic.Line           : chr  "Cloned 5thgeneration 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               : chr  "2016-03-27 00:53:11" "2016-04-04 01:39:02" "2016-03-13 20:35:42"
##  $ Clicked.on.Ad           : int  0 0 0 0 0 0 0 1 0 0 ...
```

Data Preparation

Validity

```
# checking for unnecesary columns
colnames(df)
```

```
##  [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"
```

columns seems necessary for this study

```
# Checking for anomalies
summary(df)
```

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

there are no anomalies in the data set

Consistency

```
# checking for missing values
colSums(is.na(df))
```

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

there are no missing values

Completeness

```
# checking for duplicates
sum(duplicated(df))
```

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

there are no duplicates in this dataset

Uniformity

```
# Checking column names uniformity
colnames(df)
```

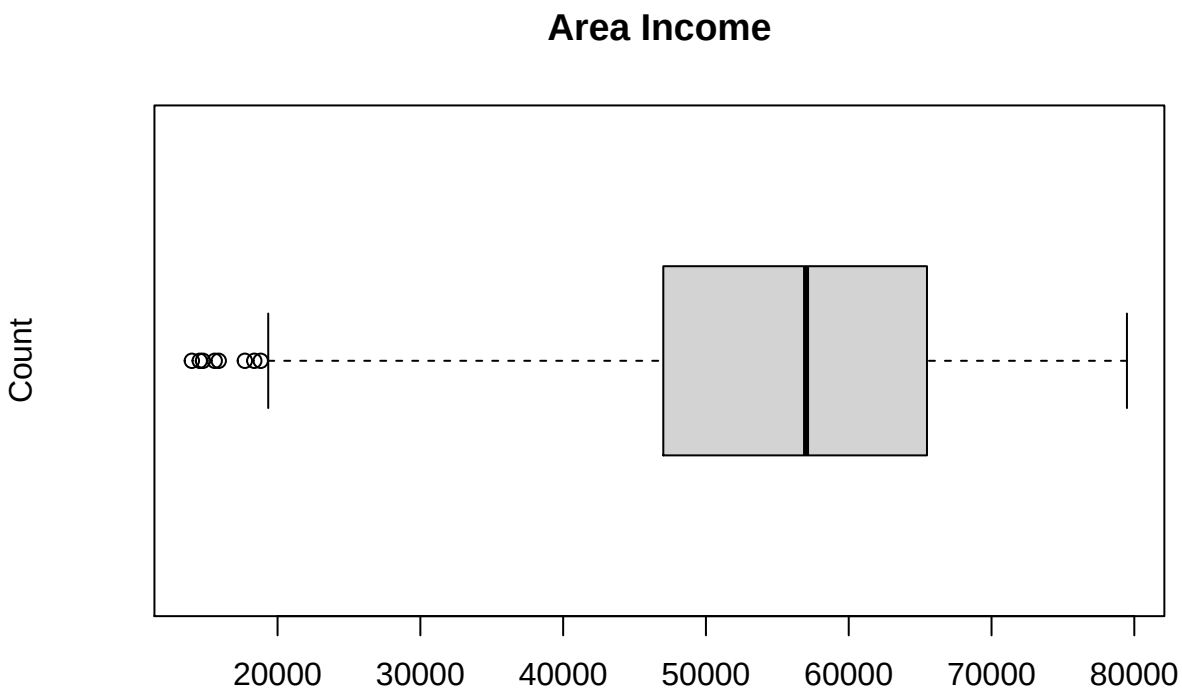
```
##  [1] "Daily.Time.Spent.on.Site" "Age"
##  [3] "Area.Income"              "Daily.Internet.Usage"
##  [5] "Ad.Topic.Line"            "City"
##  [7] "Male"                     "Country"
##  [9] "Timestamp"                "Clicked.on.Ad"
```

the column names are uniform

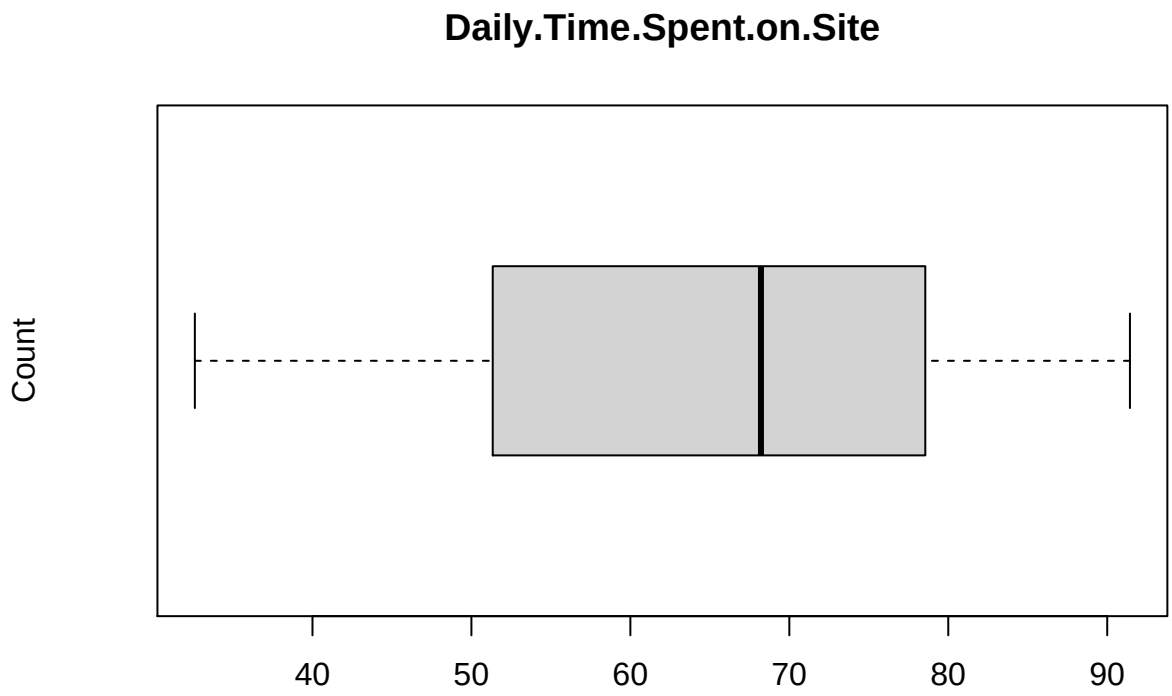
Outliers

Using boxplots to check for outliersin numerical columns

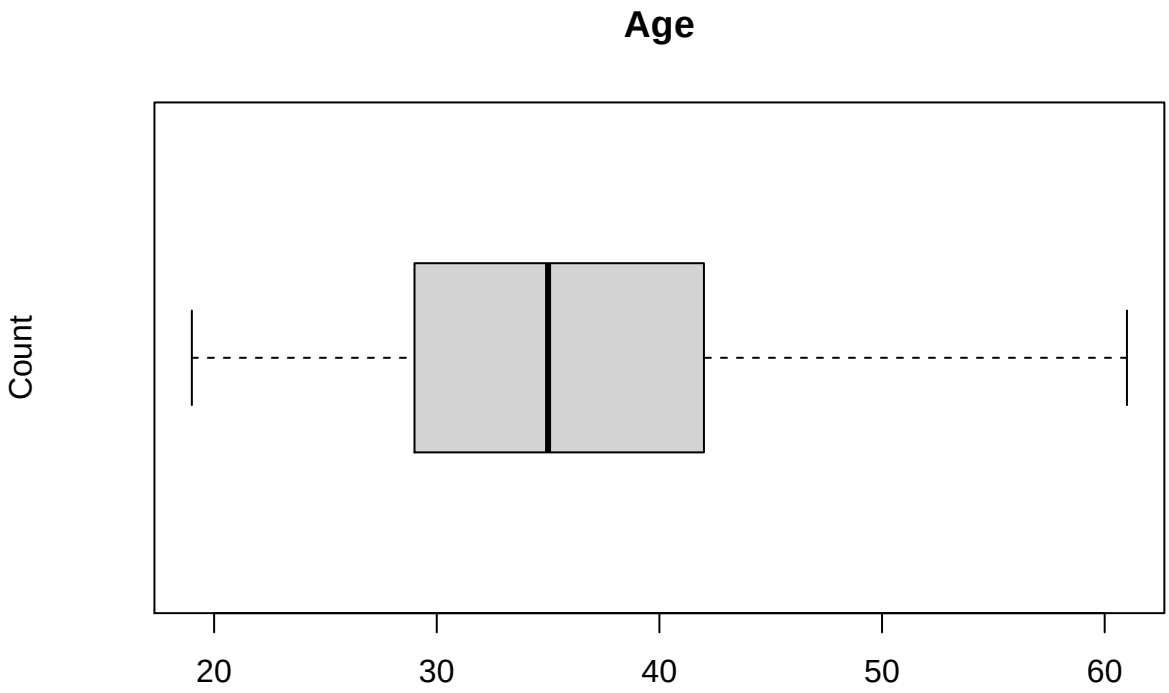
```
# Area.Income column
boxplot(df$Area.Income, data=df, main ="Area Income", ylab = 'Count', horizontal = TRUE)
```



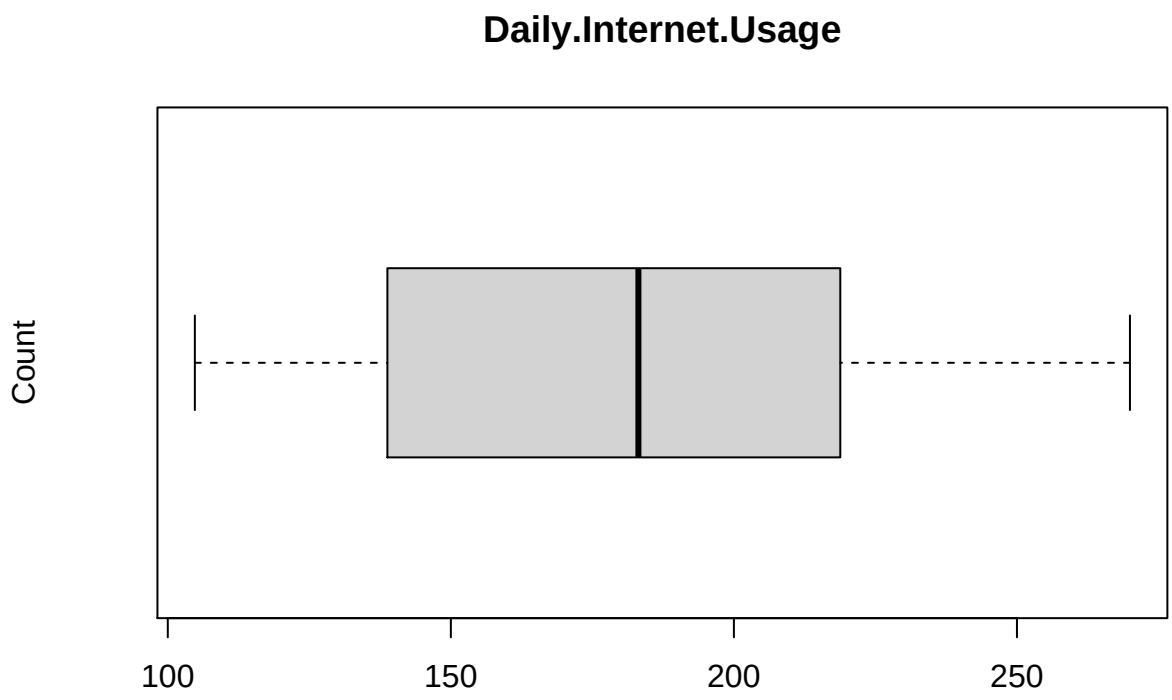

```
# Daily.Time.Spent.on.Site column
boxplot(df$Daily.Time.Spent.on.Site, data=df, main ="Daily.Time.Spent.on.Site", ylab = 'Count', horizon
```




```
# Age
boxplot(df$Age, data=df, main ="Age", ylab = 'Count', horizontal = TRUE)
```




```
# Daily.Internet.Usage
boxplot(df$Daily.Internet.Usage, data=df, main ="Daily.Internet.Usage", ylab = 'Count', horizontal = TRU
```



from the boxplots, Area income column has outliers but we will keep them for futher analysis

Exploratory Data Analysis

Univariate Analysis

Categorical Analysis Analysis using Countplots

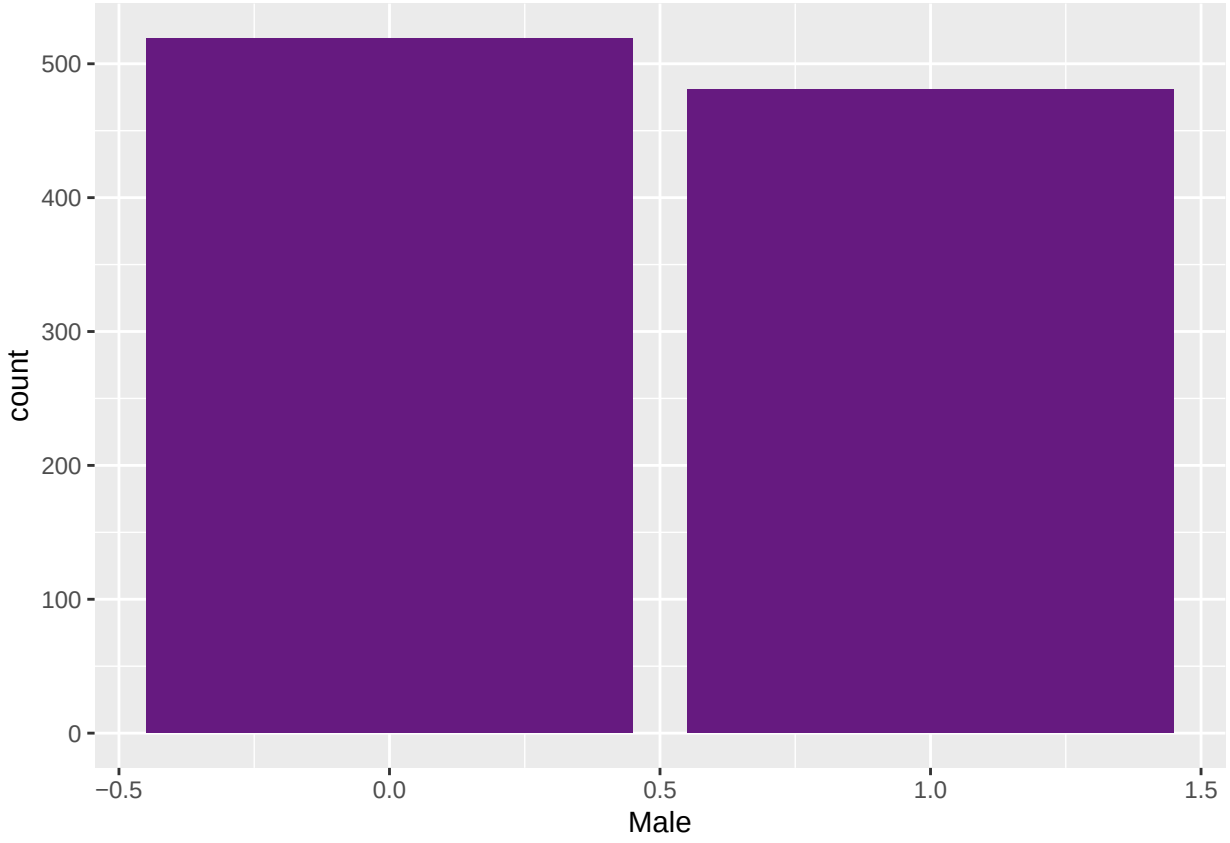
```
#
library(ggplot2)
```

```
## Warning: The packages `ellipsis` (>= 0.3.2) and `vctrs` (>= 0.3.8) are required
## as of rlang 1.0.0.

## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'tibble'

## Warning: replacing previous import 'ellipsis::check_dots_unnamed' by
## 'rlang::check_dots_unnamed' when loading 'tibble'

## Warning: replacing previous import 'ellipsis::check_dots_used' by
## 'rlang::check_dots_used' when loading 'tibble'

## Warning: replacing previous import 'ellipsis::check_dots_empty' by
## 'rlang::check_dots_empty' when loading 'tibble'

## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'pillar'

## Warning: replacing previous import 'ellipsis::check_dots_unnamed' by
## 'rlang::check_dots_unnamed' when loading 'pillar'
```

```
## Warning: replacing previous import 'ellipsis::check_dots_used' by
## 'rlang::check_dots_used' when loading 'pillar'
```

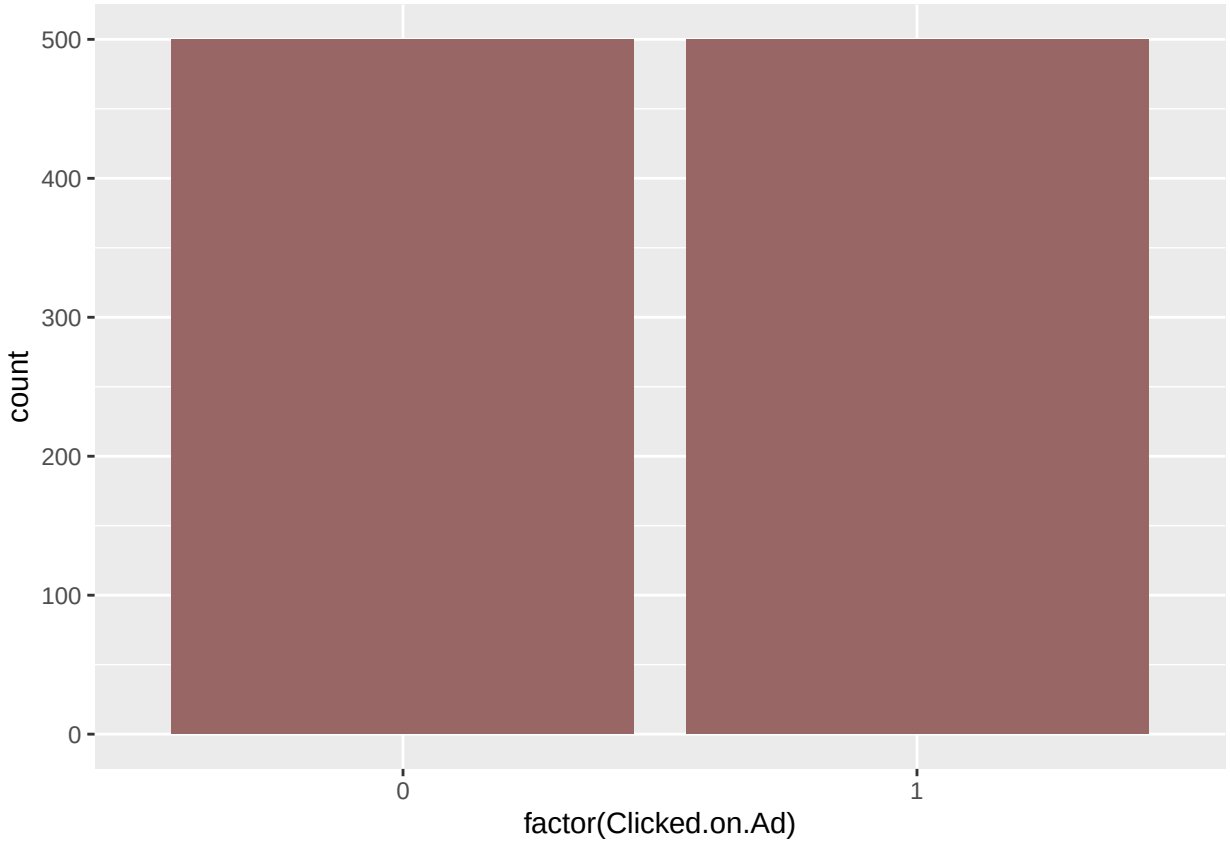
```
## Warning: replacing previous import 'ellipsis::check_dots_empty' by
## 'rlang::check_dots_empty' when loading 'pillar'
```

```
ggplot(df, aes(x=Male)) + geom_bar(fill=rgb(0.4,0.1,0.5))
```



ther are more females than males

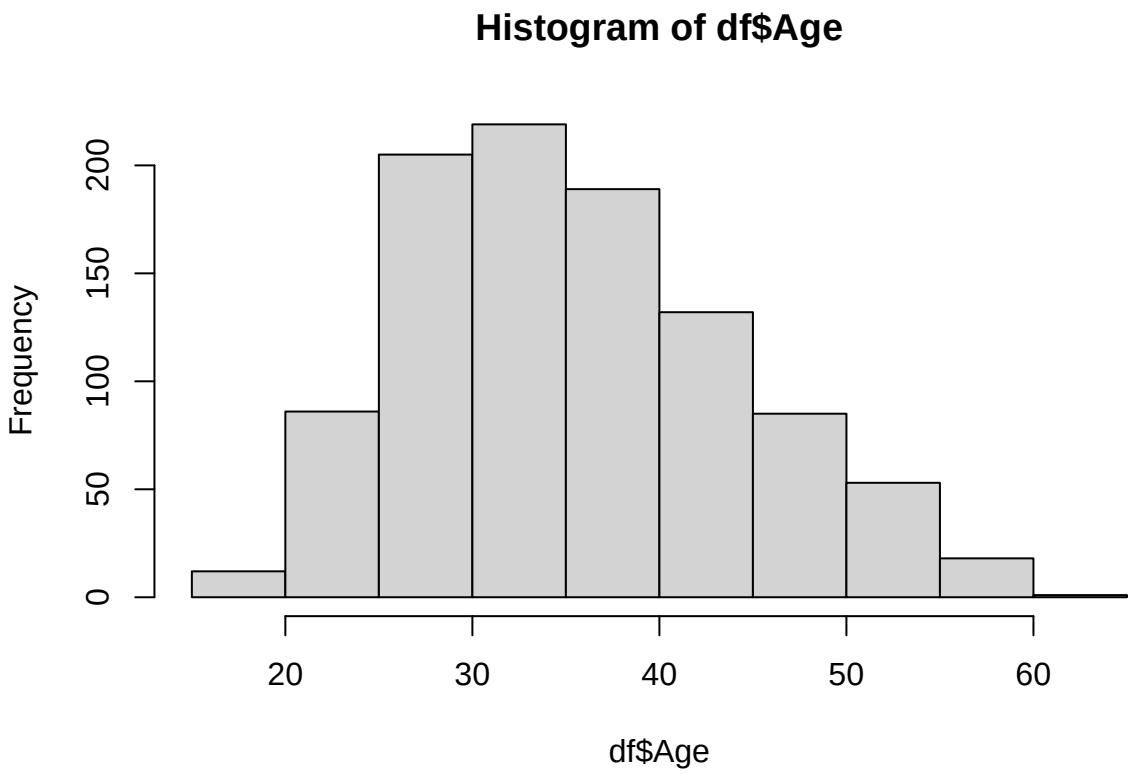
```
ggplot(df, aes(x=factor(`Clicked.on.Ad`))) + geom_bar( fill=rgb(0.6,0.4,0.4))
```



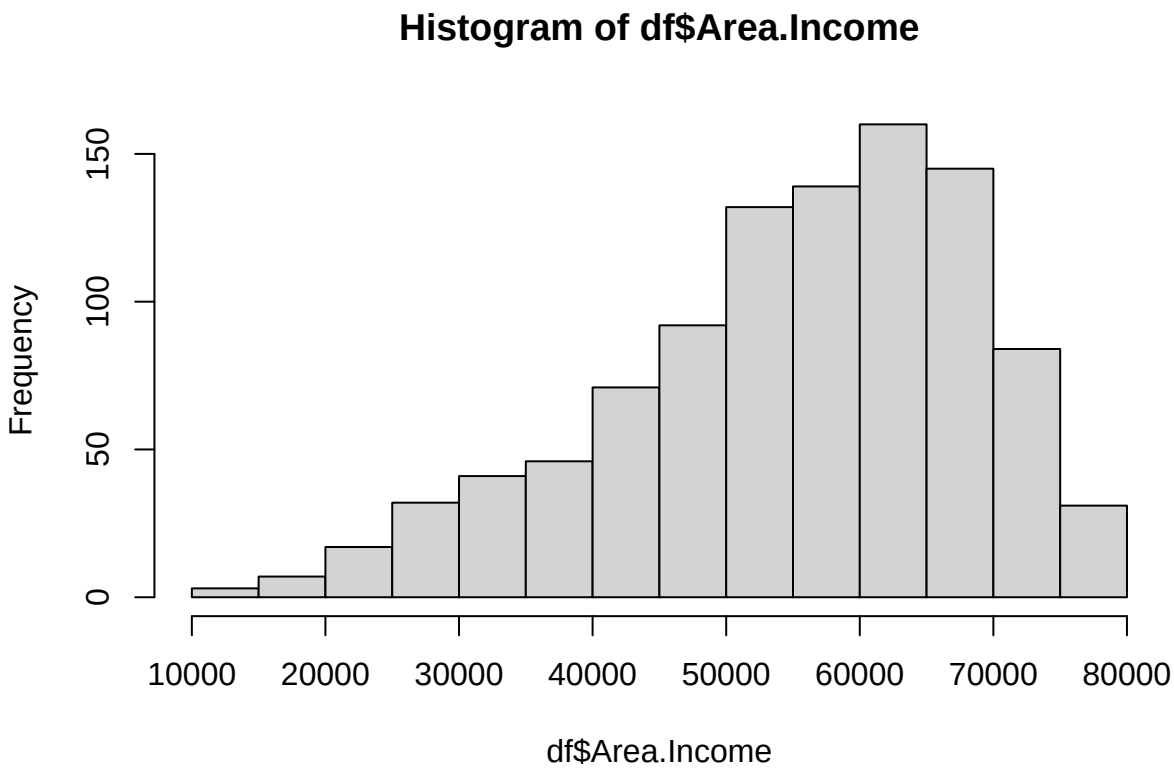
the number of people who clicked the ad is equal to those who did not click

Analysis using Histograms

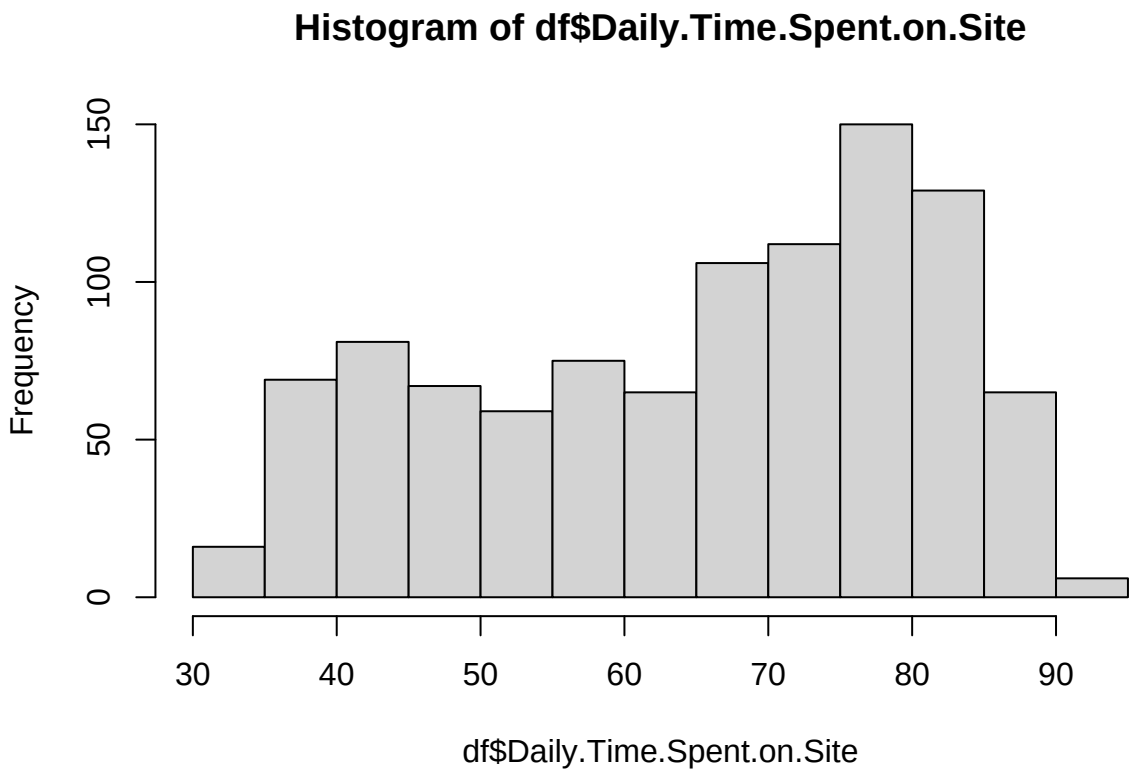
```
# Age column
hist(df$Age)
```



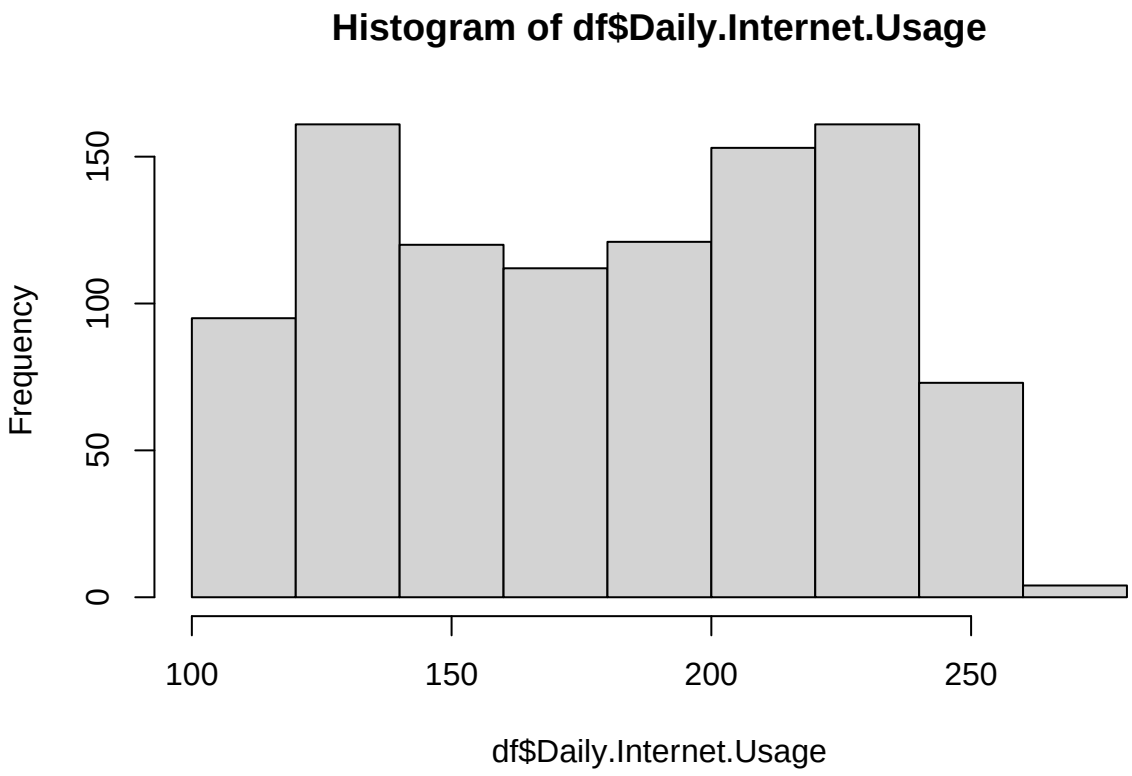

```
# Area.Income column
hist(df$Area.Income)
```




```
# Daily.Time.Spent.on.Site column
hist(df$Daily.Time.Spent.on.Site)
```




```
# Daily.Internet.Usage column
hist(df$Daily.Internet.Usage)
```



Numerical Analysis Measures of Central Tendency Mean

```
# Mean of all numeric columns
colMeans(df[sapply(df,is.numeric)])
```

```
## Daily.Time.Spent.on.Site                      Age             Area.Income
##                  65.0002                  36.0090              55000.0001
##     Daily.Internet.Usage                     Male            Clicked.on.Ad
##                 180.0001                   0.4810                   0.5000
```

Median

```
# Median of Daily.Time.Spent
median <- median(df$Daily.Time.Spent.on.Site)
print(median)
```

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

```
# Median of Age
median <- median(df$Age)
print(median)
```

```
## [1] 35
```

```
# Median of Area.Income
median <- median(df$Area.Income)
print(median)
```

```
## [1] 57012.3
```

```r
# Median of Area.Income
median <- median(df$Daily.Internet.Usage)
print(median)
```

```
## [1] 183.13
```

Mode

```r
# Creating the mode function
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]}
```

```r
# Age Mode
getmode(df$Age)
```

```
## [1] 31
```

```r
# Daily.Time.Spent.on.Site Mode
getmode(df$Daily.Time.Spent.on.Site)
```

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

```r
# Area.Income Mode
getmode(df$Area.Income)
```

```
## [1] 61833.9
```

```r
# Daily.Internet.Usage Mode
getmode(df$Daily.Internet.Usage)
```

```
## [1] 167.22
```

```r
# City Mode
getmode(df$City)
```

```
## [1] "Lisamouth"
```

```r
# Ad.Topic.Line Mode
getmode(df$Ad.Topic.Line)
```

```
## [1] "Cloned 5thgeneration orchestration"
```

```r
# Country Mode
getmode(df$Country)
```

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

```r
# Timestamp Mode
getmode(df$Timestamp)
```

```
## [1] "2016-03-27 00:53:11"
```

Varience

```r
# varience in Age
var(df$Age)
```

```
## [1] 77.18611
```

```r
# Daily.Time.Spent.on.Site varience
var(df$Daily.Time.Spent.on.Site)
```

```
## [1] 251.3371
```

```
# Area.Income varience
var(df$Area.Income)
```

```
## [1] 179952406
```

```
# Daily.Internet.Usage varience
var(df$Daily.Internet.Usage)
```

```
## [1] 1927.415
```

Standard Deviation

```
# Age SD
sd(df$Age)
```

```
## [1] 8.785562
```

```
# Daily.Time.Spent.on.Site SD
sd(df$Daily.Time.Spent.on.Site)
```

```
## [1] 15.85361
```

```
# Area.Income SD
sd(df$Area.Income)
```

```
## [1] 13414.63
```

```
# Daily.Internet.Usage SD
sd(df$Daily.Internet.Usage)
```

```
## [1] 43.90234
```

Quantiles

```
# Age quantiles
quantile(df$Age)
```

```
##   0%  25%  50%  75% 100%
##   19   29   35   42   61
```

```
# Daily.Time.Spent.on.Site quantiles
quantile(df$Daily.Time.Spent.on.Site)
```

```
##      0%     25%     50%     75%    100%
## 32.6000 51.3600 68.2150 78.5475 91.4300
```

```
# Area.Income quantiles
quantile(df$Area.Income)
```

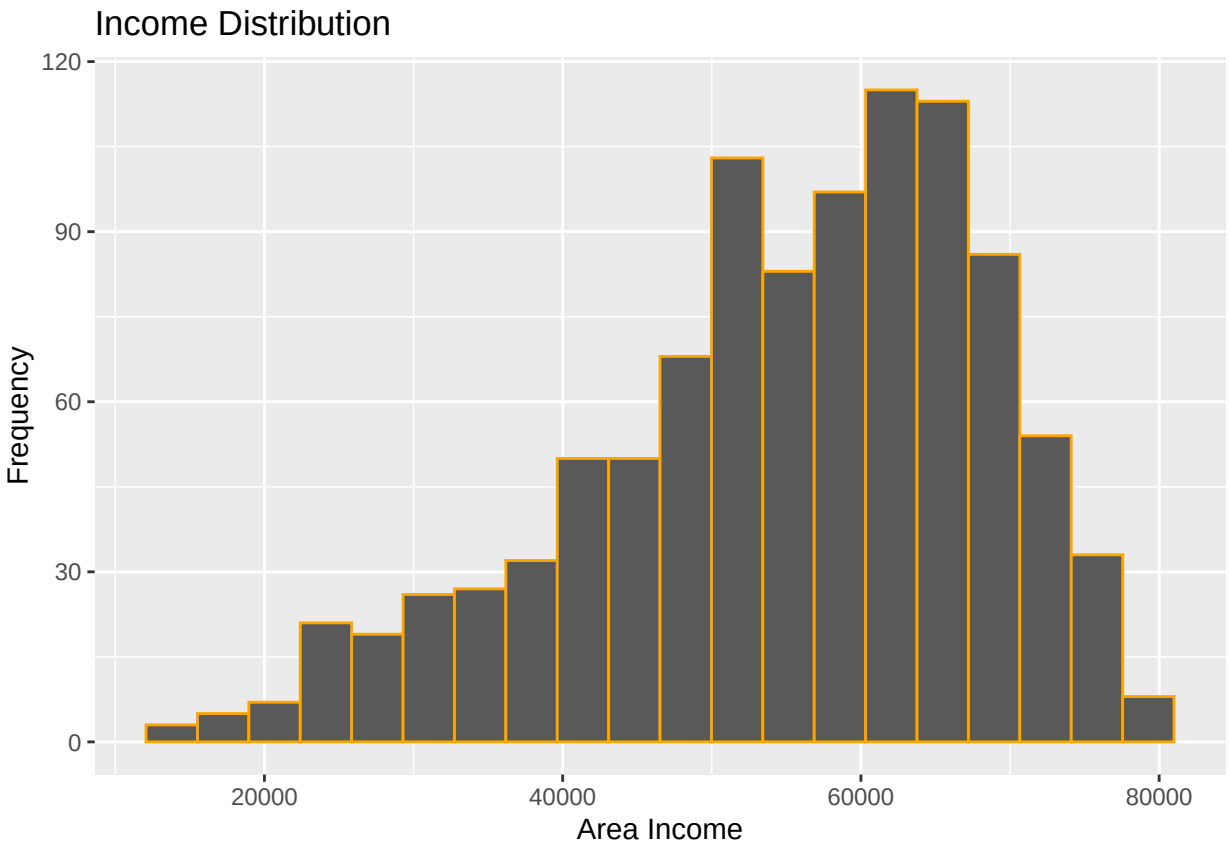
```
##       0%      25%      50%      75%     100%
## 13996.50 47031.80 57012.30 65470.64 79484.80
```

```
# Daily.Internet.Usage quantiles
quantile(df$Daily.Internet.Usage)
```

```
##       0%      25%      50%      75%     100%
## 104.7800 138.8300 183.1300 218.7925 269.9600
```

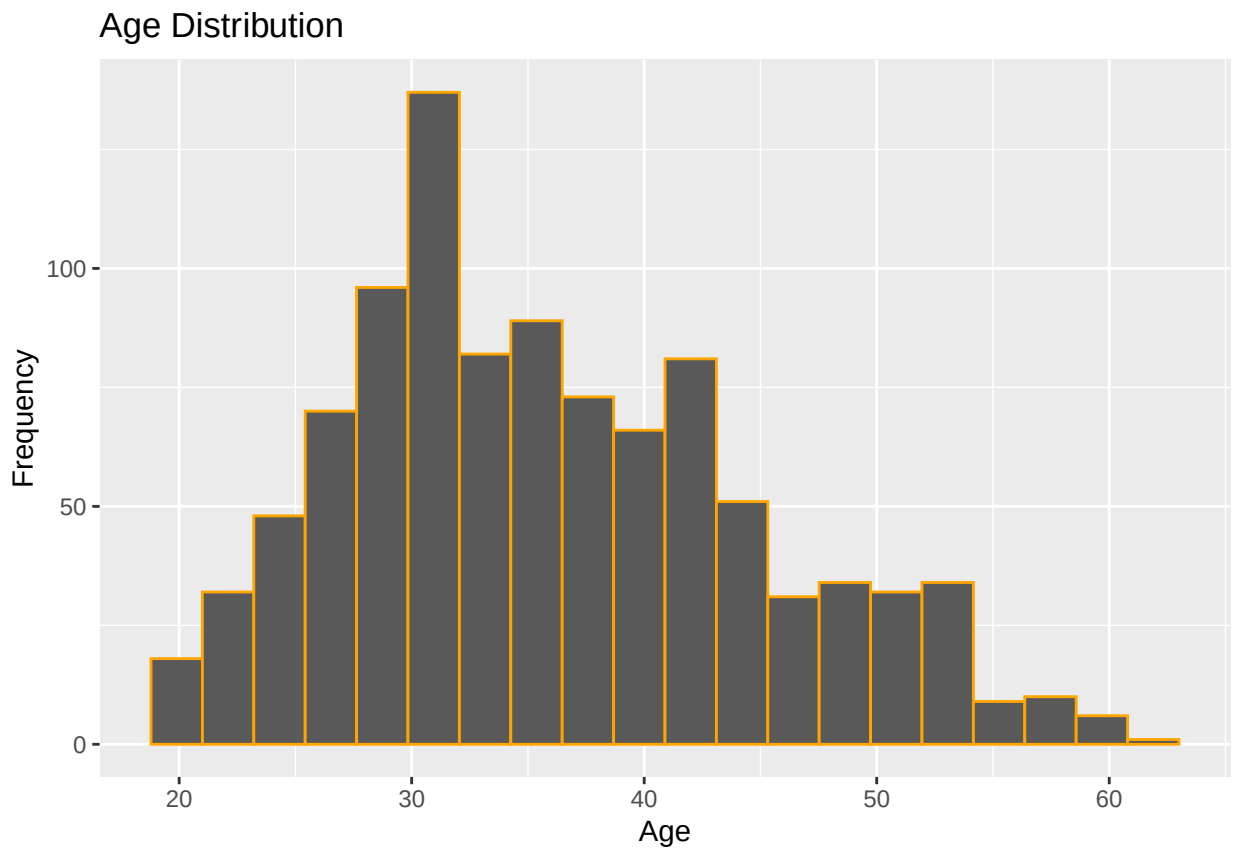
Bivariate Analysis

```
# Clicked.on.Ad vs Area.Income
ggplot(data = df, aes(x = Area.Income, fill = Clicked.on.Ad))+
        geom_histogram(bins  =20,col = "orange")+
        labs(title = "Income Distribution", x = "Area Income", y= "Frequency", fill = "Clicked on Ad")+
                palette = "Set1"
        )
```

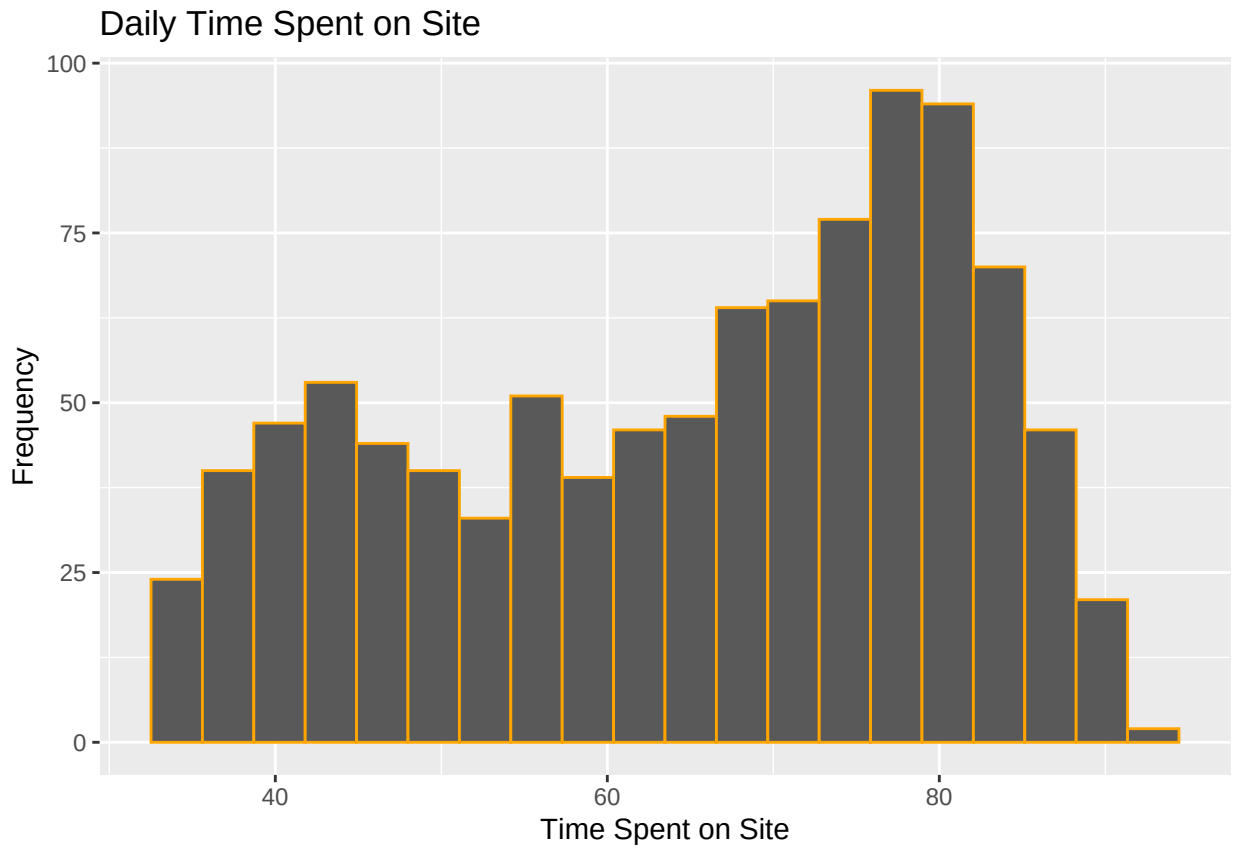


Pair Ggplots

```
# Clicked.on.Ad vs Age
ggplot(data = df, aes(x = Age, fill = Clicked.on.Ad))+
        geom_histogram(bins  =20,col = "orange")+
        labs(title = "Age Distribution", x = "Age", y= "Frequency", fill = "Clicked on Ad")+ scale_colo
                palette = "Set1"
        )
```




```
# Clicked.on.Ad vs Daily.Time.Spent.on.Site
ggplot(data = df, aes(x =Daily.Time.Spent.on.Site, fill = Clicked.on.Ad))+
        geom_histogram(bins  =20,col = "orange")+
        labs(title = "Daily Time Spent on Site", x = "Time Spent on Site", y= "Frequency", fill = "Click
                palette = "Set1"
        )
```

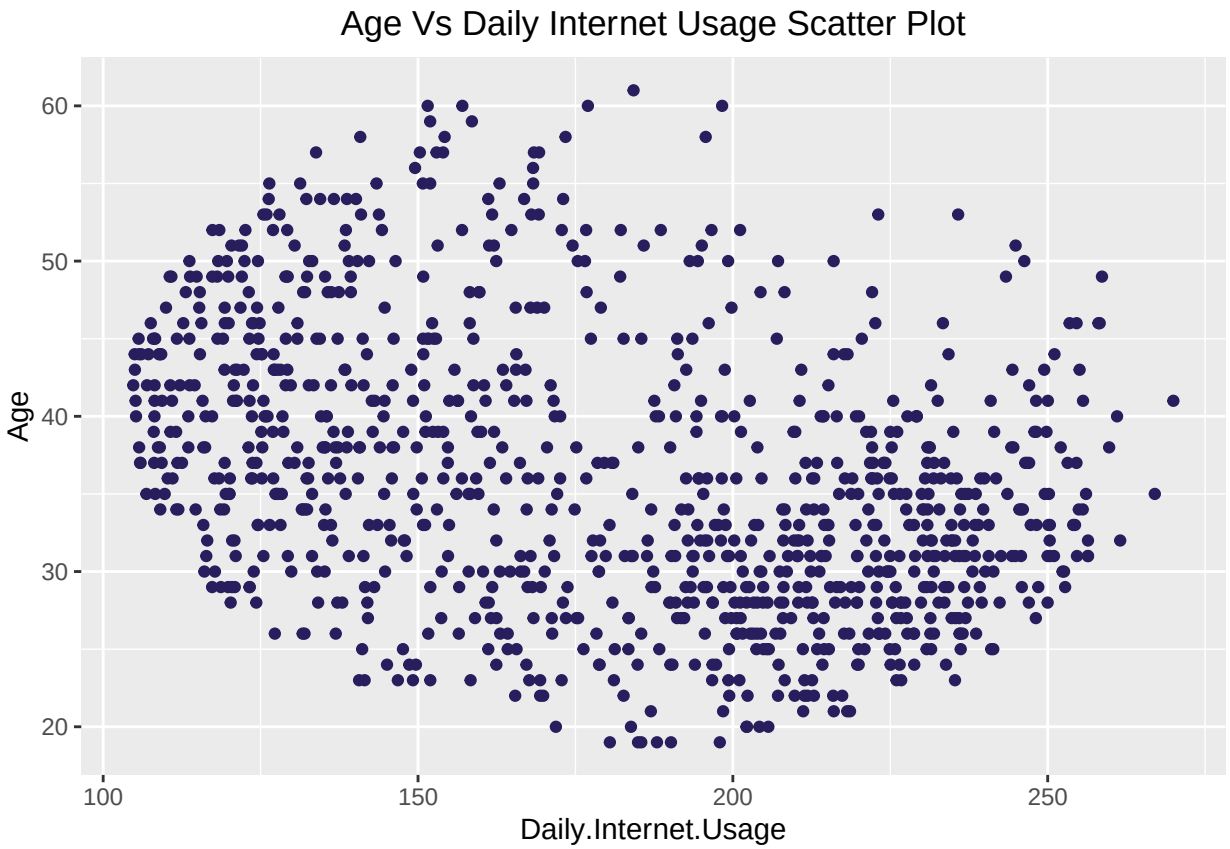


Scatter Plots

```
# Scatter plot and correlation function
scatter.plt <- function(col1, col2, corr1, corr2, data, title){
data <- ggplot(data, aes(x = {{col1}}, y= {{col2}})) + geom_point(color = '#281E5D') + ggtitle(paste(ti
correlation <- cor(df[ , c(corr1)], df[, c(corr2)])
plot(data)
print(paste0('Correlation = ', correlation, '.'))
}
```

Age Vs Daily Internet Usage

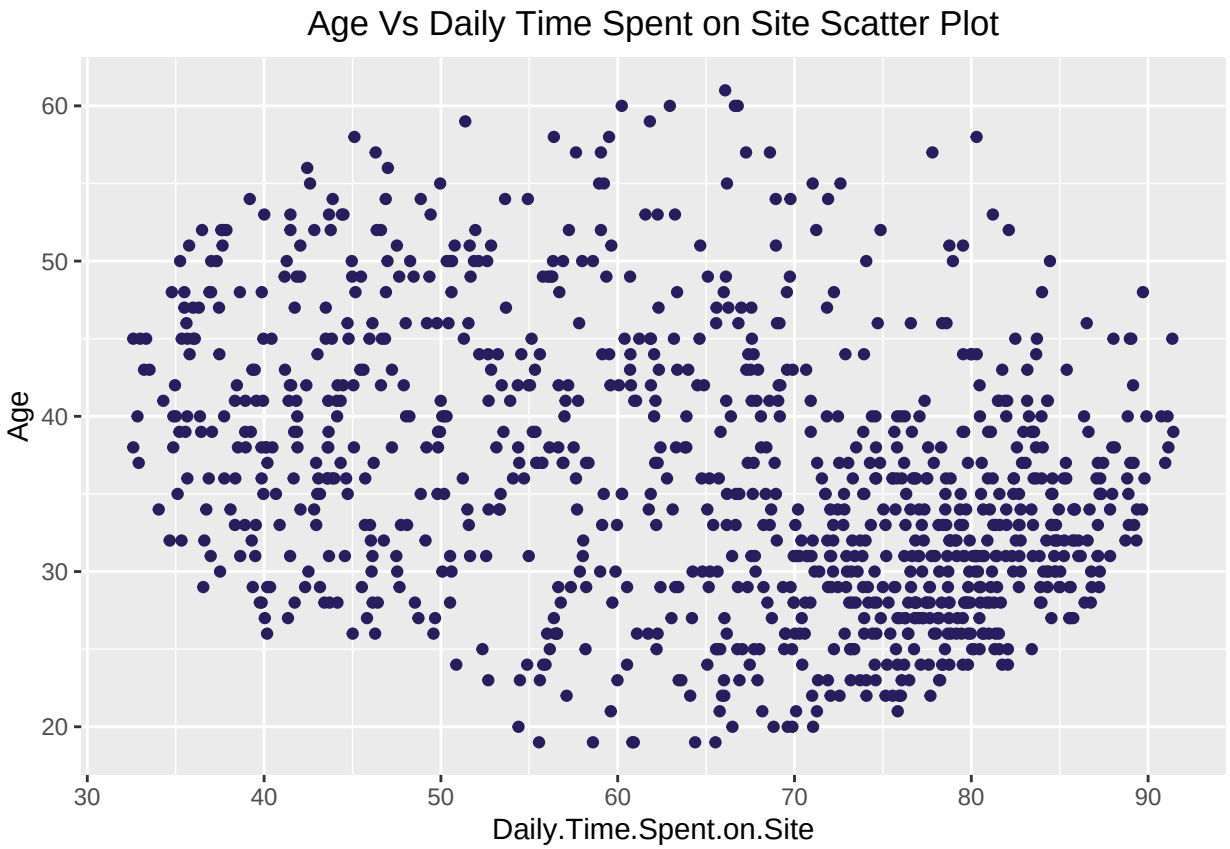
```
scatter.plt(Daily.Internet.Usage, Age, data = df, corr1 = 'Daily.Internet.Usage', corr2 = 'Age', 'Age V
```




```
## [1] "Correlation = -0.367208560147359."
```

Age Vs Daily Time Spent on Site

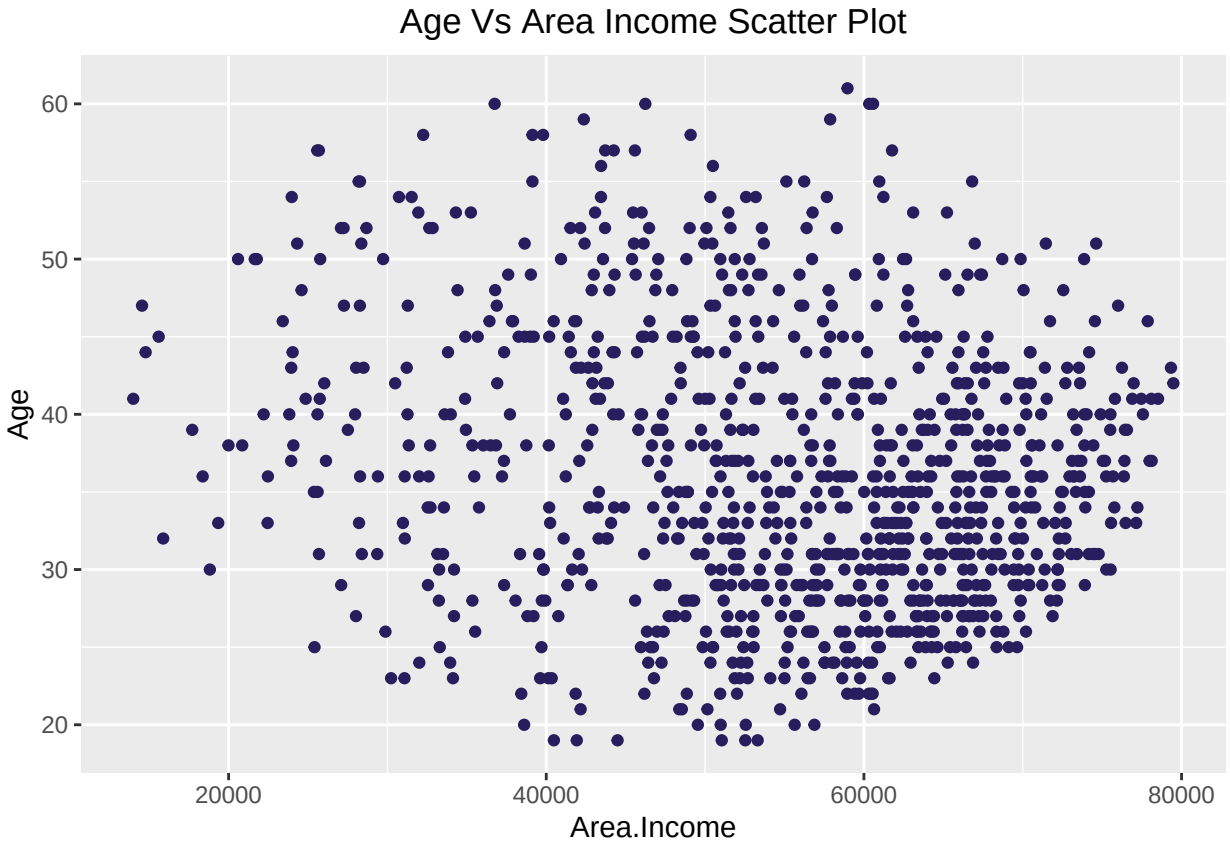
```
scatter.plt(Daily.Time.Spent.on.Site, Age, data = df, corr1 = 'Daily.Time.Spent.on.Site', corr2 = 'Age'
```




```
## [1] "Correlation = -0.331513342786584."
```

Age Vs Area Income

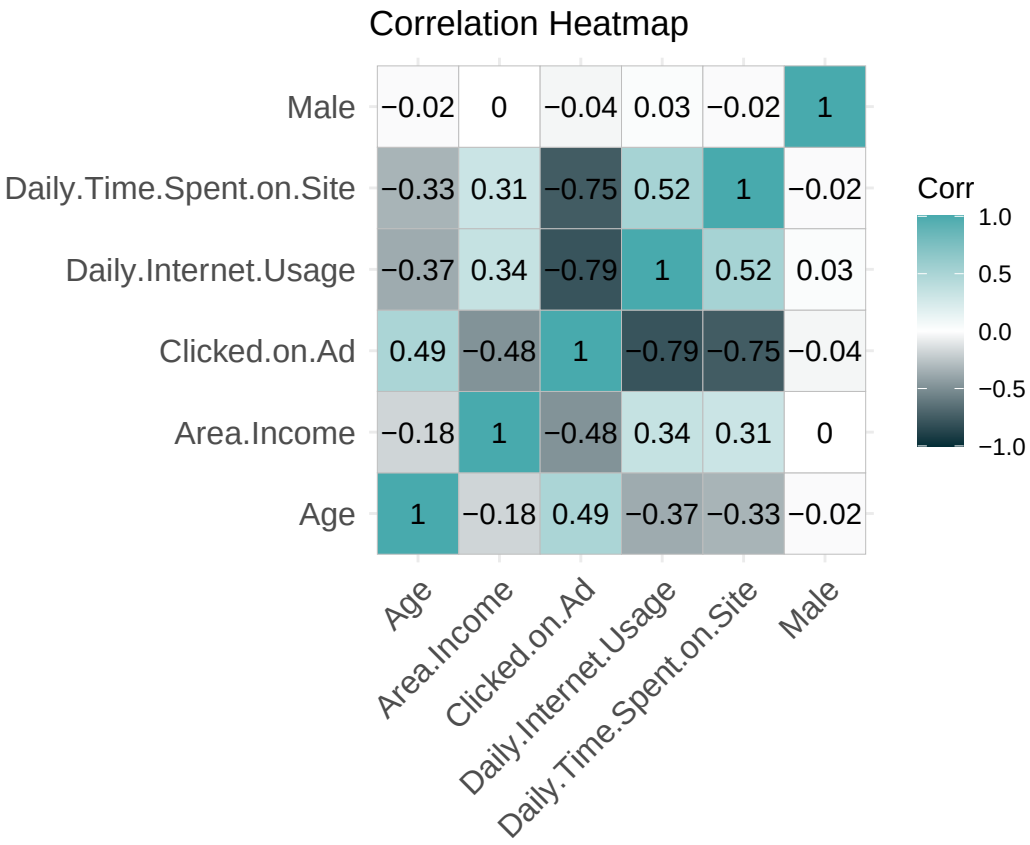
```
scatter.plt(Area.Income, Age, data = df, corr1 = 'Area.Income', corr2 = 'Age', 'Age Vs Area Income')
```




```
## [1] "Correlation = -0.182604955032622."
```

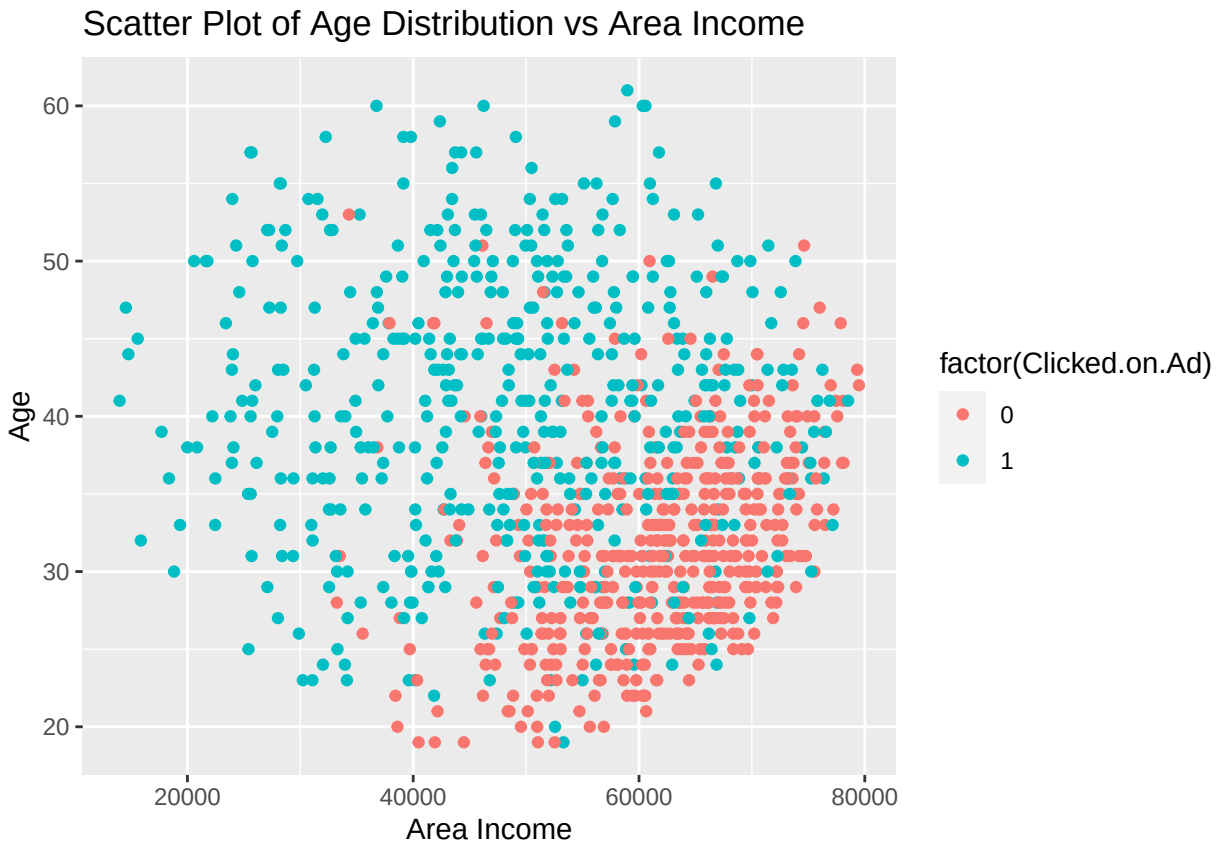
Multivariate Analysis

```
library(ggcorrplot)
corr <- dplyr::select(df,Age,Area.Income,Clicked.on.Ad,Daily.Internet.Usage,Daily.Time.Spent.on.Site,Ma
ggcorrplot(cor(corr), lab = TRUE, title = 'Correlation Heatmap', colors = c('#022D36', 'white', '#48AAA
```



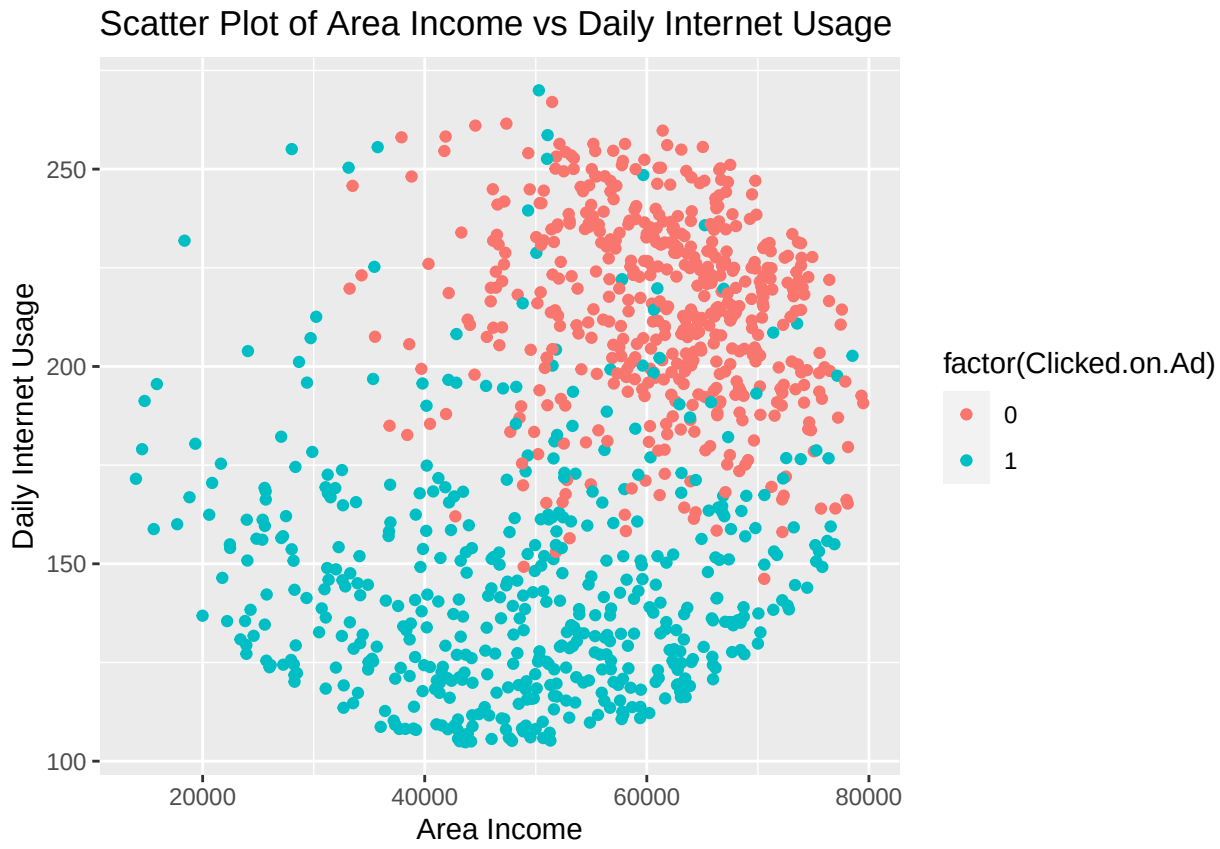
Correlation matrix

```
ggplot(df, aes(Area.Income,Age))+geom_point(aes(colour= factor(`Clicked.on.Ad`)))+
  labs(title = "Scatter Plot of Age Distribution vs Area Income",
       x = "Area Income",
       y = "Age")
```

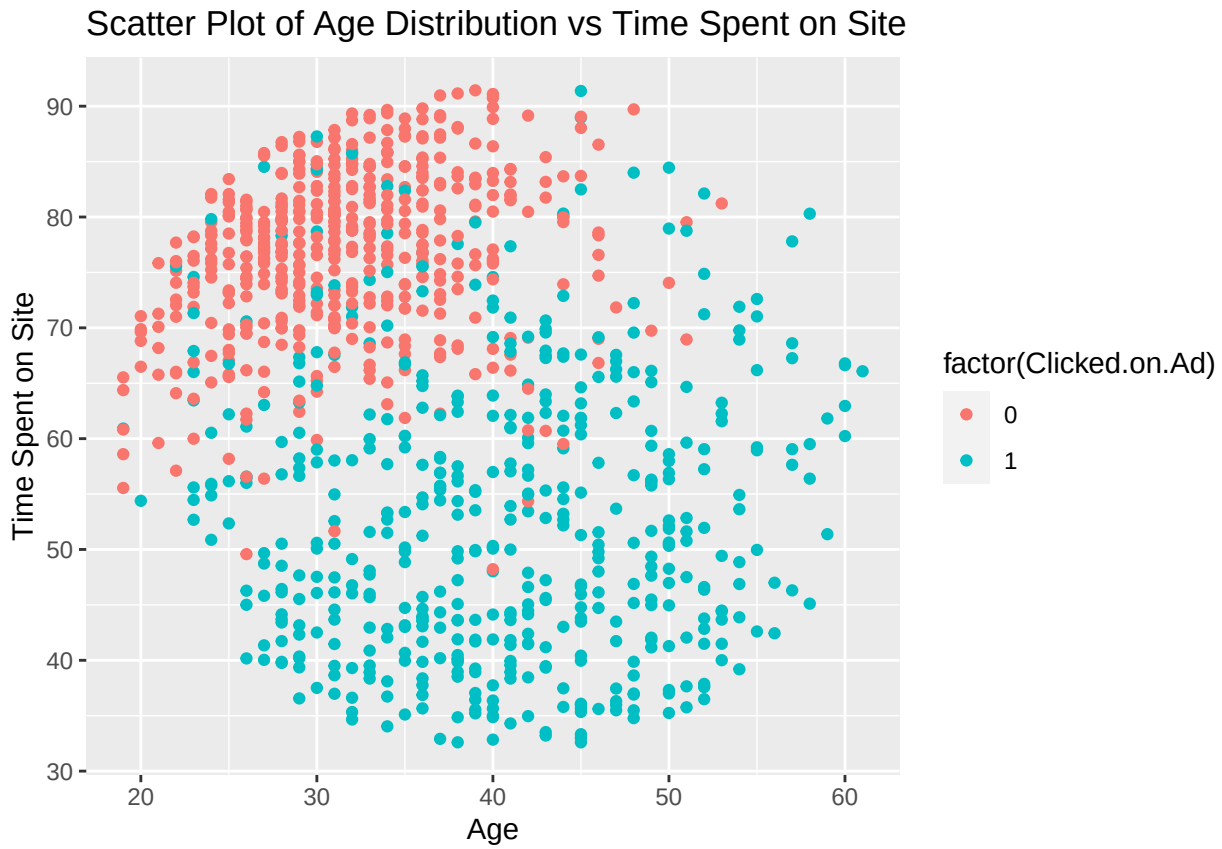


Scatter Plots

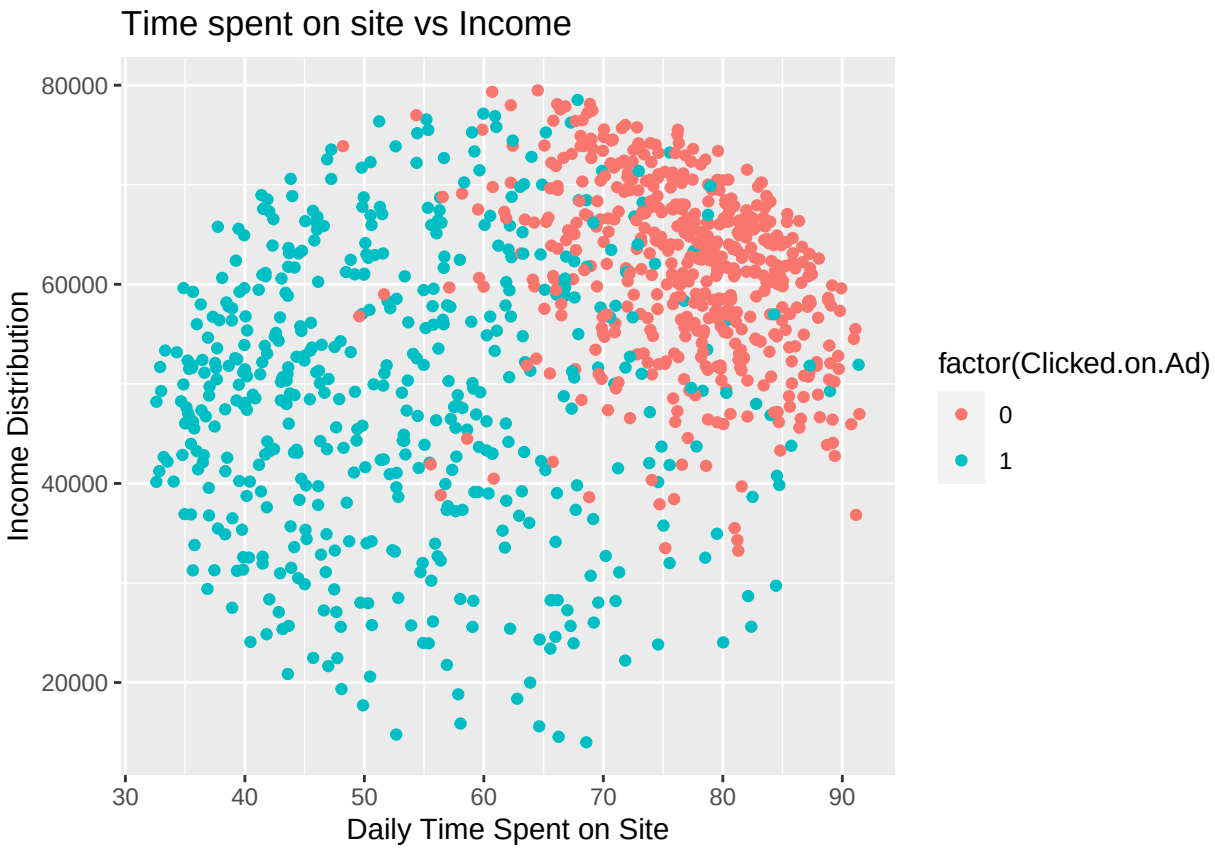
```
ggplot(df, aes(Area.Income, Daily.Internet.Usage))+
  geom_point(aes(colour= factor(`Clicked.on.Ad`)))+
  labs(title = "Scatter Plot of Area Income vs Daily Internet Usage",
       x = "Area Income",
       y = "Daily Internet Usage")
```



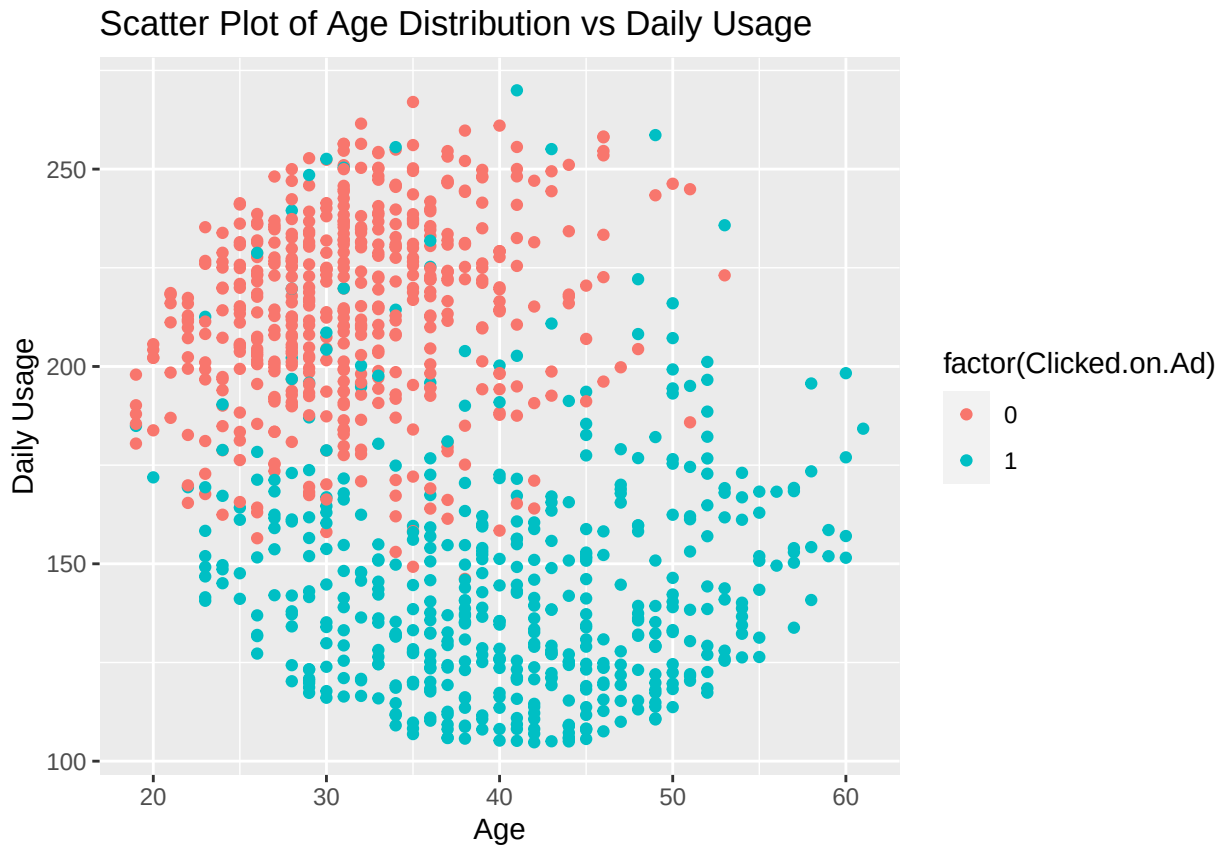

```
ggplot(df, aes(Age, Daily.Time.Spent.on.Site))+
  geom_point(aes(colour= factor(`Clicked.on.Ad`)))+
  labs(title = "Scatter Plot of Age Distribution vs Time Spent on Site",
       x = "Age",
       y = "Time Spent on Site")
```




```
ggplot(df, aes(Daily.Time.Spent.on.Site, Area.Income))+
  geom_point(aes(colour= factor(`Clicked.on.Ad`)))+
  labs(title = "Time spent on site vs Income",
       x = "Daily Time Spent on Site",
       y = "Income Distribution")
```




```
ggplot(df, aes(Age, Daily.Internet.Usage))+
  geom_point(aes(colour= factor(`Clicked.on.Ad`)))+
  labs(title = "Scatter Plot of Age Distribution vs Daily Usage",
       x = "Age",
       y = "Daily Usage")
```



Modeling

Importing Libraries

```
library(caret)
```

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

```
library(rpart)
library(rpart.plot)
```

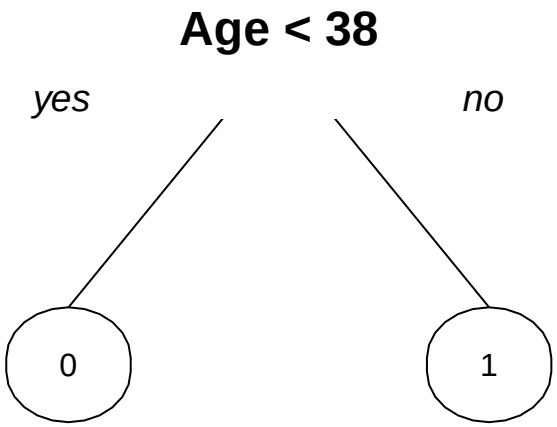
Previewing our dataset

```
head(df)
```

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

```
m <- rpart(Clicked.on.Ad ~ Age, data = df, method = "class")


prp(m,      # Plot the decision tree
    space=4,          # (Formatting options chosen for notebook)
    split.cex = 1.5,
    nn.border.col=0)
```

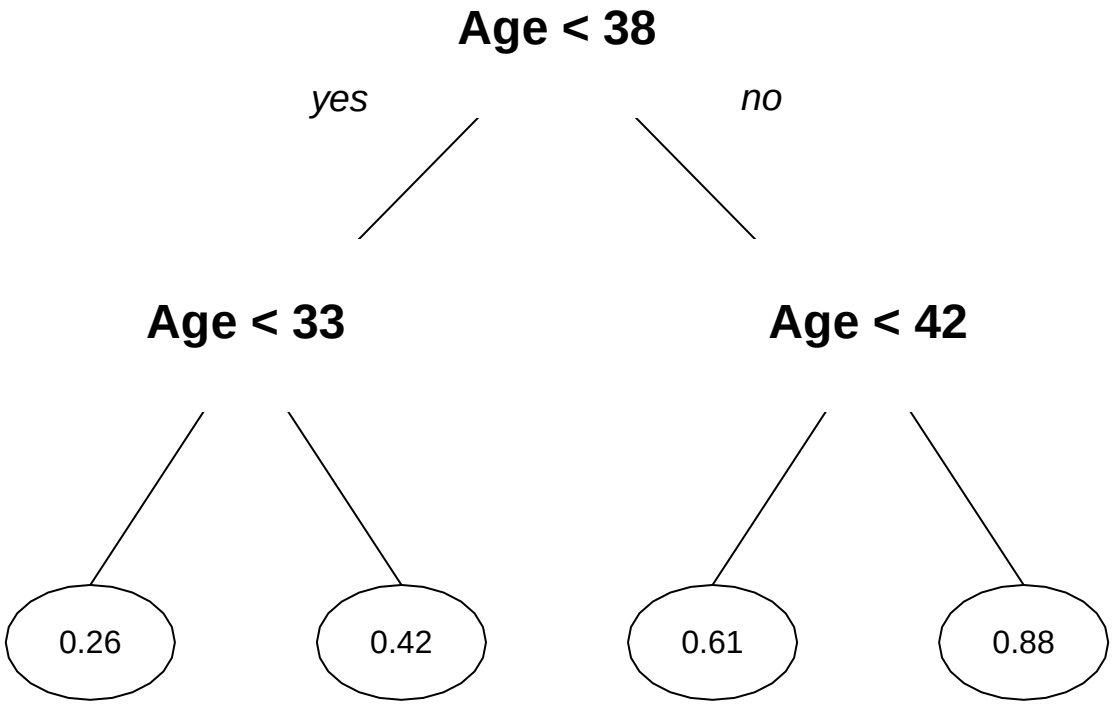


Simple visualization Model
Key: 0 - Did not click the Add 1 - Clicked the Ad

From this simple tree, we can see that those with the age <38 did not click the ad while those with ages >38 clicked the Ad. Will not use the method =“class” argument to visualize the probabilities.

```
m <- rpart(Clicked.on.Ad ~ Age, data = df)


prp(m,      # Plot the decision tree
    space=4,          # (Formatting options chosen for notebook)
```

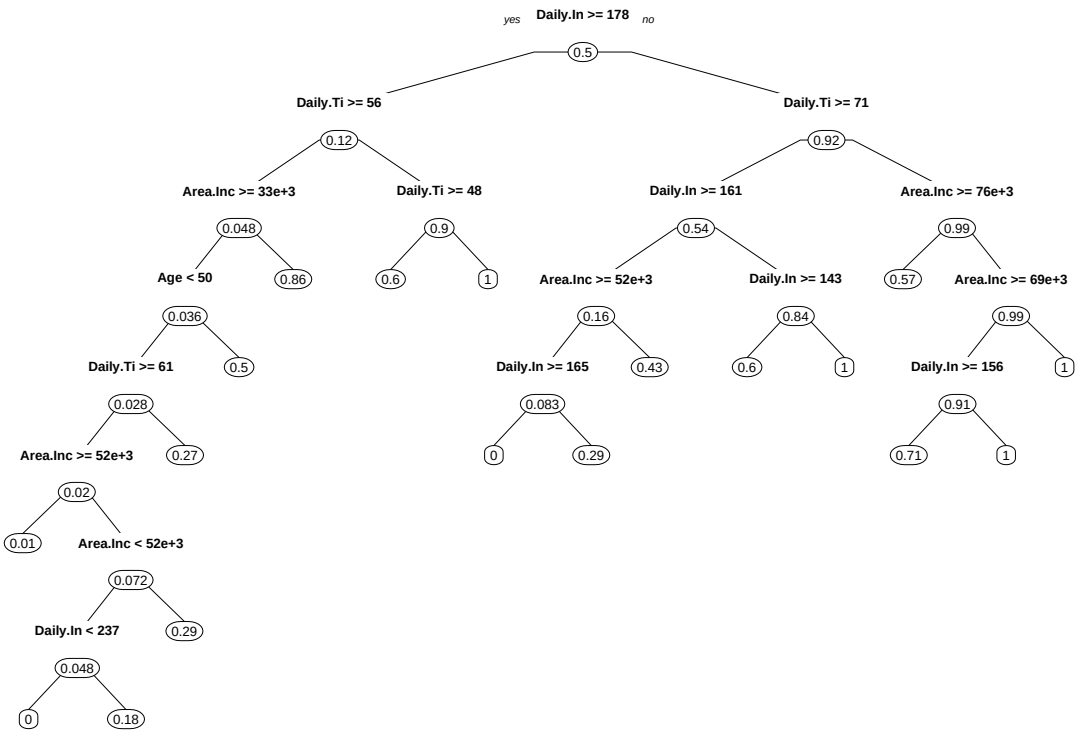
```
    split.cex = 1.5,
    nn.border.col=0)
```



Complex Model

Now we will add more features to our Model

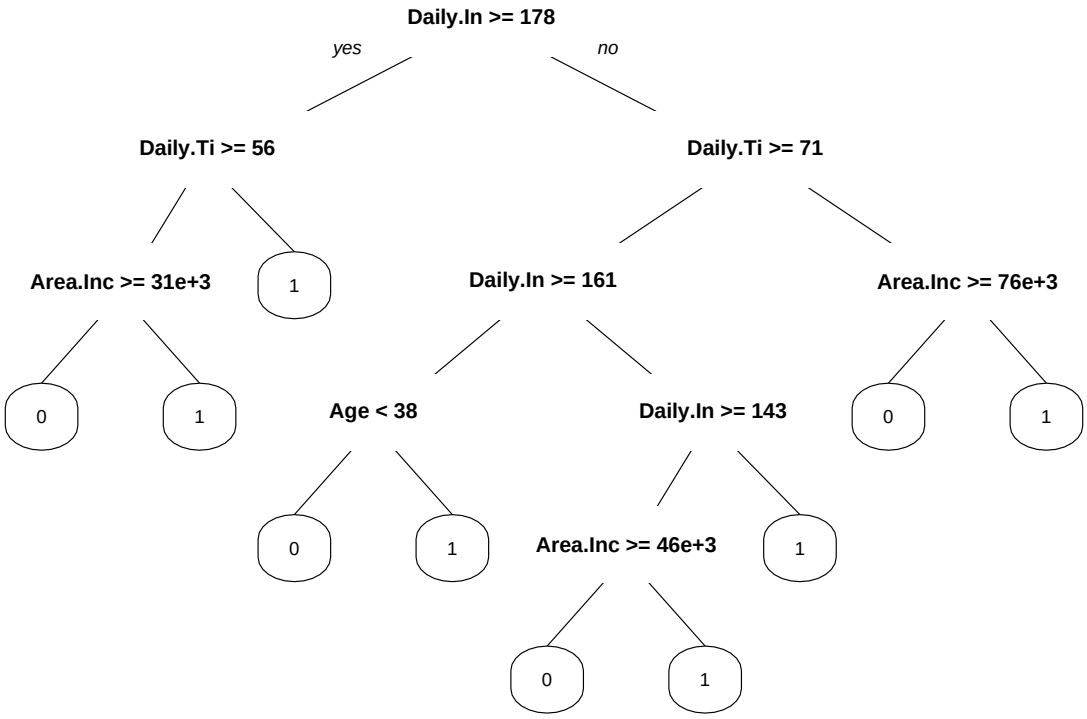
```
complex_tree <- rpart(Clicked.on.Ad ~ Age + Daily.Time.Spent.on.Site + Area.Income +Daily.Internet.Usag
                      cp = 0.001)                  # Set complexity parameter*


prp(complex_tree,
    type = 1,
    nn.border.col=0,
    border.col=1,
    cex=0.4)
```



Note: the complexity parameter governs model complexity. A smaller complexity parameter will allow for more complex models.

The plot above illustrates how complex (and difficult to visualize) decision trees can become when we start adding several explanatory variables. A model that is too complex is prone to overfitting the training data, which can lead to poor generalization to unseen data. The rpart() function includes several optional parameters that we can set to control model complexity. As noted above, the "cp" parameter lets us adjust model complexity (cp adjusts the improvement of the model fit necessary for it to create a new branch.). Apart from the complexity parameter we can also adjust the maximum depth of the tree and the minimum number of observations at each leaf node to limit model complexity:

```
limited_complexity_tree <- rpart(Clicked.on.Ad ~ Age + Daily.Time.Spent.on.Site + Area.Income +Daily.In
                      cp = 0.001,              # Set complexity parameter
                      maxdepth = 5,            # Set maximum tree depth
                      minbucket = 5,           # Set min number of obs in leaf nodes
                      method = "class")        # Return classifications instead of probs

options(repr.plot.width = 6, repr.plot.height = 6)

prp(limited_complexity_tree,
    space=4,
    split.cex = 1.2,
    nn.border.col=0)
```



This model seems a little more reasonable, we will use it to predict and test accuracy

```
train_preds <- predict(limited_complexity_tree,
                       newdata = df,
                       type = "class")               # Return class predictions

confusionMatrix(factor(train_preds), factor(df$Clicked.on.Ad))
```

Accuracy

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 492  24
##          1   8 476
##
##                Accuracy : 0.968
##                  95% CI : (0.9551, 0.978)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.936
##
##  Mcnemar's Test P-Value : 0.00801
##
```

```
##             Sensitivity : 0.9840
##             Specificity : 0.9520
##          Pos Pred Value : 0.9535
##          Neg Pred Value : 0.9835
##              Prevalence : 0.5000
##          Detection Rate : 0.4920
##    Detection Prevalence : 0.5160
##       Balanced Accuracy : 0.9680
##
##        'Positive' Class : 0
##
```

from this model, we get 96.8% accuracy. This probably indicates that our model is overfitting. We will use Holdout Validation and Cross Validation to understand how our model will generalize to unseen data.

```
set.seed(12)
split_model <- createDataPartition(y=df$Clicked.on.Ad,    # Split on survived
                                   list = FALSE,      # Return indexes as a vector
                                   p=0.8,            # 75% of data in the training set
                                   times=1)           # Make 1 split

training_set <- df[split_model,]     # Get the new training set
validation_set <- df[-split_model,]  # Get the validation set

nrow(training_set)/nrow(df)      # Check proportion in each partition
```

Holdout Validation and Cross Validation

```
## [1] 0.8
```

```
nrow(validation_set)/nrow(df)
```

```
## [1] 0.2
```

We have splitted our dataset into 80% train set and 20% validation test

```
set.seed(12)

df$Clicked.on.Ad <- as.factor(df$Clicked.on.Ad) # Convert target to factor*

# Create a trainControl object to control how the train function creates the model
train_control <- trainControl(method = "repeatedcv",    # Use cross validation
                              number = 10,              # Use 10 partitions
                              repeats = 2)              # Repeat 2 times

# Set required parameters for the model type we are using**
tune_grid = expand.grid(cp=c(0.001))

# Use the train() function to create the model
validated_tree <- train(Clicked.on.Ad ~ Age + Daily.Time.Spent.on.Site + Area.Income +Daily.Internet.Usa
                        data=df,                  # Data set
                        method="rpart",                       # Model type(decision tree)
                        trControl= train_control,             # Model control options
                        tuneGrid = tune_grid,                 # Required model parameters
                        maxdepth = 5,                         # Additional parameters***
                        minbucket=5)
```

```
validated_tree          # View a summary of the model
```

```
## CART
##
## 1000 samples
##    5 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 2 times)
## Summary of sample sizes: 900, 900, 900, 900, 900, 900, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.9405    0.881
##
## Tuning parameter 'cp' was held constant at a value of 0.001
```

With cross validation and hold out(splitting the dataset), we get 94% accuracy. This shows that our model can clearly predict unseen data with 94% accuracy.

Conclusion

In conclusion, from the analysis, the major factors that determine if a user will click an ad are the:

1. Gender
2. Daily time spent on the site
3. Area Income
4. Time of day and month

From the analysis, we can conclude that:

1. The Older people (above 35),were more likely to click on the course advert.
2. The higher the person earns the less likely he/she will click the add.
3. There is an equal chance for someone to either click on the advert or not
4. The amount of time someone spent on the blog was inversely proportional to the probability of him/her clicking the add.

Recommendations

From the analysis, I will recommend that:

1. More adverts to be targeted to older people above 35
2. The adverts to target low income (less than 60,000) population.
3. The adverts to be designed in such a way that a user clearly sees it when he/she access the blog