Shrikanth_Mahale_HW3_Part2

Shrikanth Mahale

4/11/2020

Loading Libaries DataExplorer package for exploratory data analysis Useful Documentation- https://cran.r-project.org/web/packages/DataExplorer/vignettes/dataexplorer-intro.html (https://cran.r-project.org/web/packages/DataExplorer/vignettes/dataexplorer-intro.html) dplyr - Data Wrangling Package. Check R Learning Guide for resources to quickly learn dplyr

```
if (!require(dplyr)) install.packages("dplyr")
```

```
## Loading required package: dplyr
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(dplyr)
if (!require(tidyverse)) install.packages("tidyverse")
```

```
## Loading required package: tidyverse
```

```
## Warning: package 'tidyverse' was built under R version 3.5.3
```

```
## -- Attaching packages ---------------------------------------------------
---------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  3.0.0     v stringr 1.4.0
## v tidyr   1.0.2     v forcats 0.4.0
## v readr   1.3.1
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
## Warning: package 'tibble' was built under R version 3.5.3
```

```
## Warning: package 'tidyr' was built under R version 3.5.3
```

```
## Warning: package 'readr' was built under R version 3.5.2
```

```
## Warning: package 'purrr' was built under R version 3.5.3
```

```
## Warning: package 'stringr' was built under R version 3.5.3
```

```
## Warning: package 'forcats' was built under R version 3.5.3
```

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

```
library(tidyverse)
if (!require(DataExplorer)) install.packages("DataExplorer")
```

```
## Loading required package: DataExplorer
```

```
## Warning: package 'DataExplorer' was built under R version 3.5.3
```

```
library(DataExplorer)
if (!require(ggplot2)) install.packages("ggplot2")
library(ggplot2)
```

Importing Data

```
data <- read.csv("KAG_data.csv", stringsAsFactors = FALSE)

data <- data %>% mutate(CTR = round(((Clicks / Impressions) * 100),4),
                        CPC = ifelse(Clicks != 0, round(Spent / Clicks,4), Spen
t),
                        CostPerConv_Total = ifelse(Total_Conversion !=0,round(S
pent/Total_Conversion,4),Spent),
                        CostPerConv_Approved = ifelse(Approved_Conversion !=0,r
ound(Spent/Approved_Conversion,4),Spent),
                        CPM = round((Spent / Impressions) * 1000, 2) )
```

Q.1 Which ad (provide ad_id as the answer) among the ads that have the least CPC led to the most impressions?

```
q1 <- dplyr::arrange(data,CPC,desc(Impressions))
head(q1$ad_id,1)
```

```
## [1] 1121094
```

Q.2 What campaign (provide compaign_id as the answer) had spent least efficiently on brand awareness on an average (i.e. most Cost per mille or CPM: use total cost for the campaign / total impressions in thousands)?

```
q2 <- dplyr::arrange(data,desc(CPM))
head(q2$campaign_id,1)
```

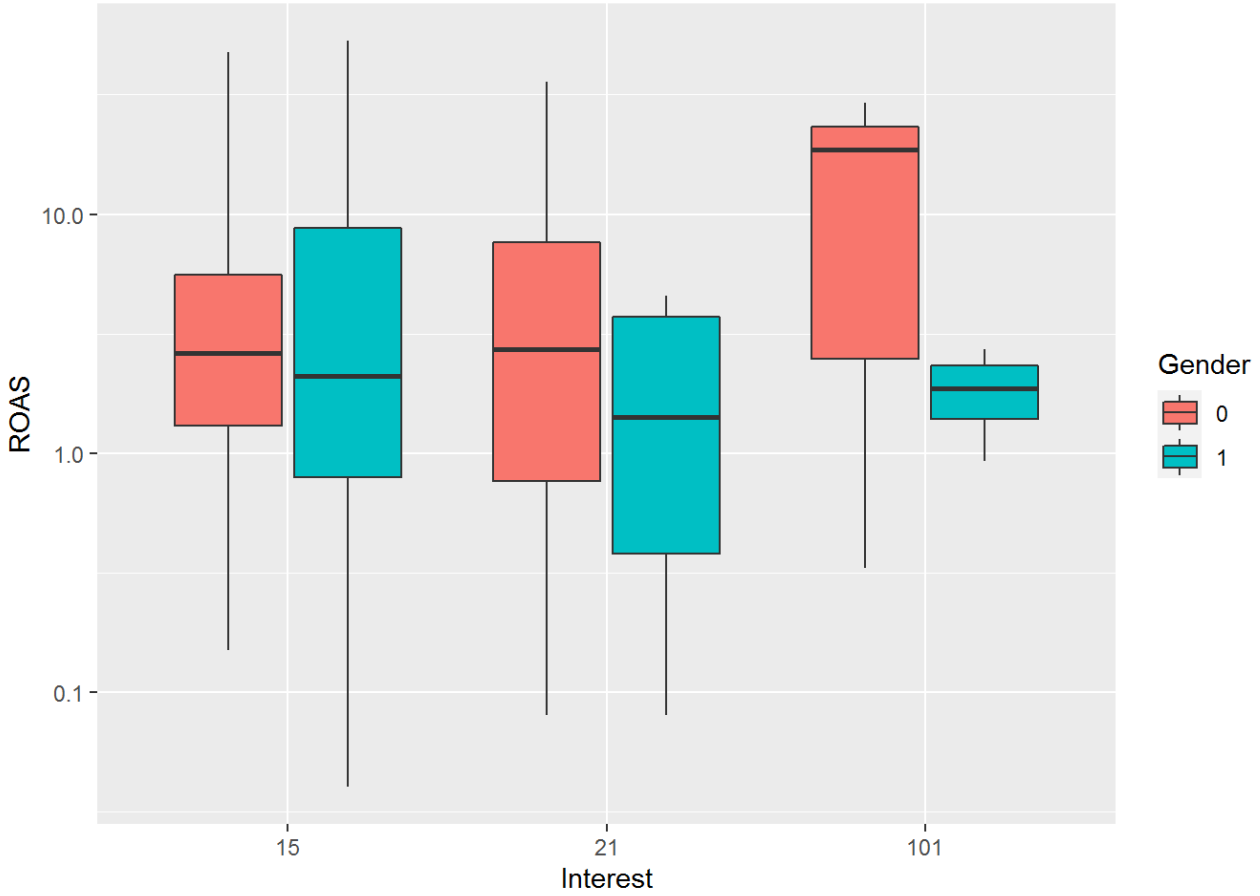
```
## [1] 936
```

Q.3 Assume each conversion ('Total_Conversion') is worth $5, each approved conversion ('Approved_Conversion') is worth $50. ROAS (return on advertising spent) is revenue as a percentage of the advertising spent . Calculate ROAS and round it to two decimals. Make a boxplot of the ROAS grouped by gender for interest = 15, 21, 101 (or interest_id = 15, 21, 101) in one graph. Also try to use the function '+ scale_y_log10()' in ggplot to make the visualization look better (to do so, you just need to add '+ scale_y_log10()' after your ggplot function). The x-axis label should be 'Interest ID' while the y-axis label should be ROAS. [8 points]

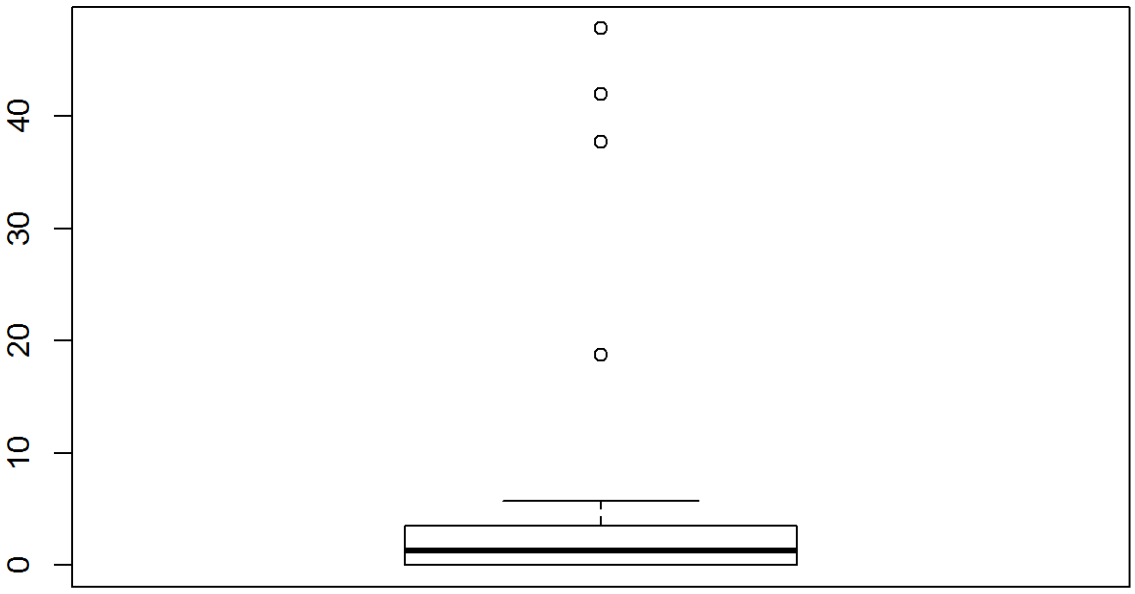
```
data <- data %>% mutate(ROAS = ifelse(Spent !=0.00,round((5.00*Total_Conversion
+50.00*Approved_Conversion)/Spent*1.00,2),0.00))
data1 <- data %>%
  filter(ROAS!=Inf, interest %in% c('15','21','101'))%>%
  select(interest,gender,ROAS)#%>%
ggplot(data = data1, aes(x=as.factor(interest), y=ROAS)) +geom_boxplot(aes(fill
=as.factor(gender)))+ scale_y_log10()+ xlab("Interest") + ylab("ROAS")+guides(f
ill=guide_legend(title="Gender"))
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

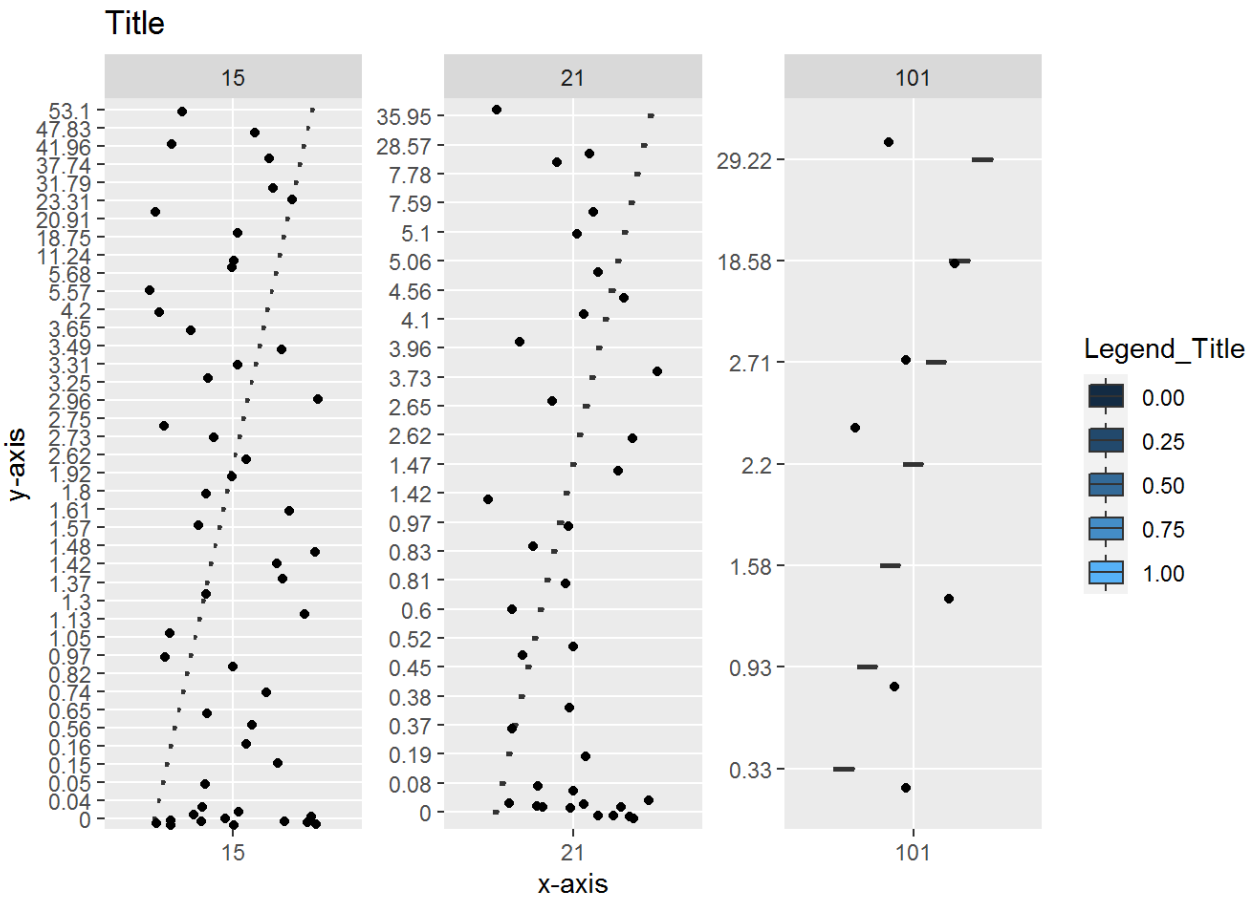
```
## Warning: Removed 23 rows containing non-finite values (stat_boxplot).
```




```
data2 <- data1 %>%
  filter(interest=='15', gender=='0')
boxplot(data2$ROAS)
```




```
dataplot <- ggplot(data = data1, aes(x=as.factor(interest), y=as.factor(ROAS)))
+ geom_boxplot(aes(fill=gender))
dataplot <- dataplot + geom_jitter()
dataplot <- dataplot + facet_wrap( ~ interest, scales="free")
dataplot <- dataplot + xlab("x-axis") + ylab("y-axis") + ggtitle("Title")
dataplot <- dataplot + guides(fill=guide_legend(title="Legend_Title"))
dataplot
```



Q.4 Summarize the median and mean of ROAS by genders when campign_id == 1178.

```
q4 <- data %>% filter(campaign_id == 1178 ) %>% group_by(gender) %>% summarise
(ROAS_Mean = mean(ROAS), ROAS_Median = median(ROAS))
q4
```

```
## # A tibble: 2 x 3
##   gender ROAS_Mean ROAS_Median
##    <int>     <dbl>       <dbl>
## 1      0      2.49        1.13
## 2      1      1.56        0.7
```

Loading Libaries

```
if (!require(readr)) install.packages("readr")
library(readr)
if (!require(correlationfunnel)) install.packages("correlationfunnel")
```

```
## Loading required package: correlationfunnel
```

```
## Warning: package 'correlationfunnel' was built under R version 3.5.3
```

```
## == correlationfunnel Tip #2 ====================================================
==========================================================
## Clean your NA's prior to using `binarize()`.
## Missing values and cleaning data are critical to getting great correlation
s. :)
```

```
library(correlationfunnel)
if (!require(DataExplorer)) install.packages("DataExplorer")
library(DataExplorer)
if (!require(WVPlots)) install.packages("WVPlots")
```

```
## Loading required package: WVPlots
```

```
## Warning: package 'WVPlots' was built under R version 3.5.3
```

```
library(WVPlots)
if (!require(ggthemes)) install.packages("ggthemes")
```

```
## Loading required package: ggthemes
```

```
## Warning: package 'ggthemes' was built under R version 3.5.3
```

```
library(ggthemes)
if (!require(ROCR)) install.packages("ROCR")
```

```
## Loading required package: ROCR
```

```
## Warning: package 'ROCR' was built under R version 3.5.3
```

```
## Loading required package: gplots
```

```
## Warning: package 'gplots' was built under R version 3.5.3
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(ROCR)
if (!require(caret)) install.packages("caret")
```

```
## Loading required package: caret
```

```
## Warning: package 'caret' was built under R version 3.5.2
```

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

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

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

```
library(caret)
if (!require(e1071)) install.packages("e1071")
```

```
## Loading required package: e1071
```

```
## Warning: package 'e1071' was built under R version 3.5.3
```

```
library(e1071)
if (!require(corrplot)) install.packages("corrplot")
```

```
## Loading required package: corrplot
```

```
## Warning: package 'corrplot' was built under R version 3.5.3
```

```
## corrplot 0.84 loaded
```

```
library(corrplot)
```

Importing Advertising Data

```
advertising <- read.csv("advertising1.csv", stringsAsFactors = FALSE)
head(advertising,1)
```

```
##   Daily.Time.Spent.on.Site Age Area.Income Daily.Internet.Usage
## 1                    68.95  35     61833.9               256.09
##                       Ad.Topic.Line        City Male Country
## 1 Cloned 5thgeneration orchestration Wrightburgh    0 Tunisia
##             Timestamp Clicked.on.Ad
## 1 2016-03-27 00:53:11             0
```

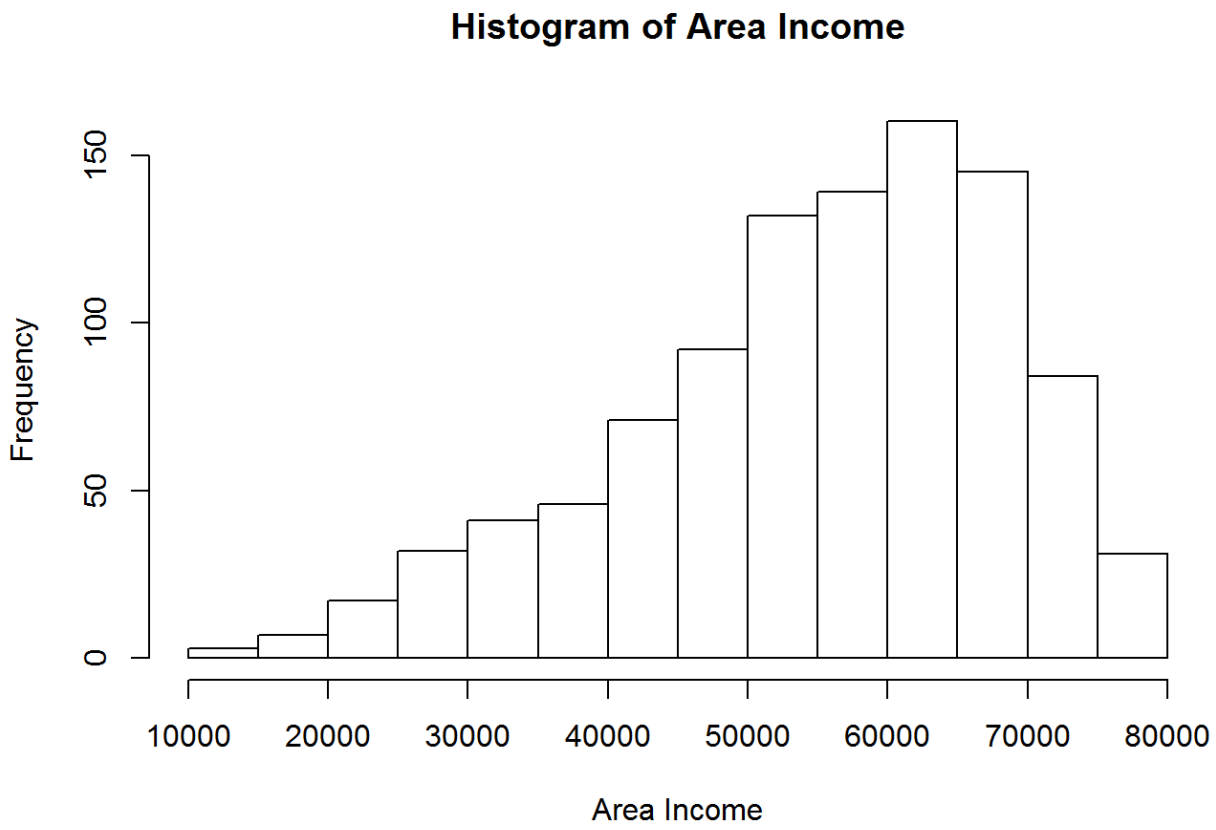
```
#changing datatype to factor
advertising$Clicked.on.Ad <- as.factor(advertising$Clicked.on.Ad)
glimpse(advertising)
```

```
## Rows: 1,000
## Columns: 10
## $ Daily.Time.Spent.on.Site <dbl> 68.95, 80.23, 69.47, 74.15, 68.37, 59...
## $ Age                      <int> 35, 31, 26, 29, 35, 23, 33, 48, 30, 2...
## $ Area.Income              <dbl> 61833.90, 68441.85, 59785.94, 54806.1...
## $ Daily.Internet.Usage     <dbl> 256.09, 193.77, 236.50, 245.89, 225.5...
## $ Ad.Topic.Line            <chr> "Cloned 5thgeneration orchestration",...
## $ City                     <chr> "Wrightburgh", "West Jodi", "Davidton...
## $ Male                     <int> 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1...
## $ Country                  <chr> "Tunisia", "Nauru", "San Marino", "It...
## $ Timestamp                <chr> "2016-03-27 00:53:11", "2016-04-04 01...
## $ Clicked.on.Ad            <fct> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1...
```

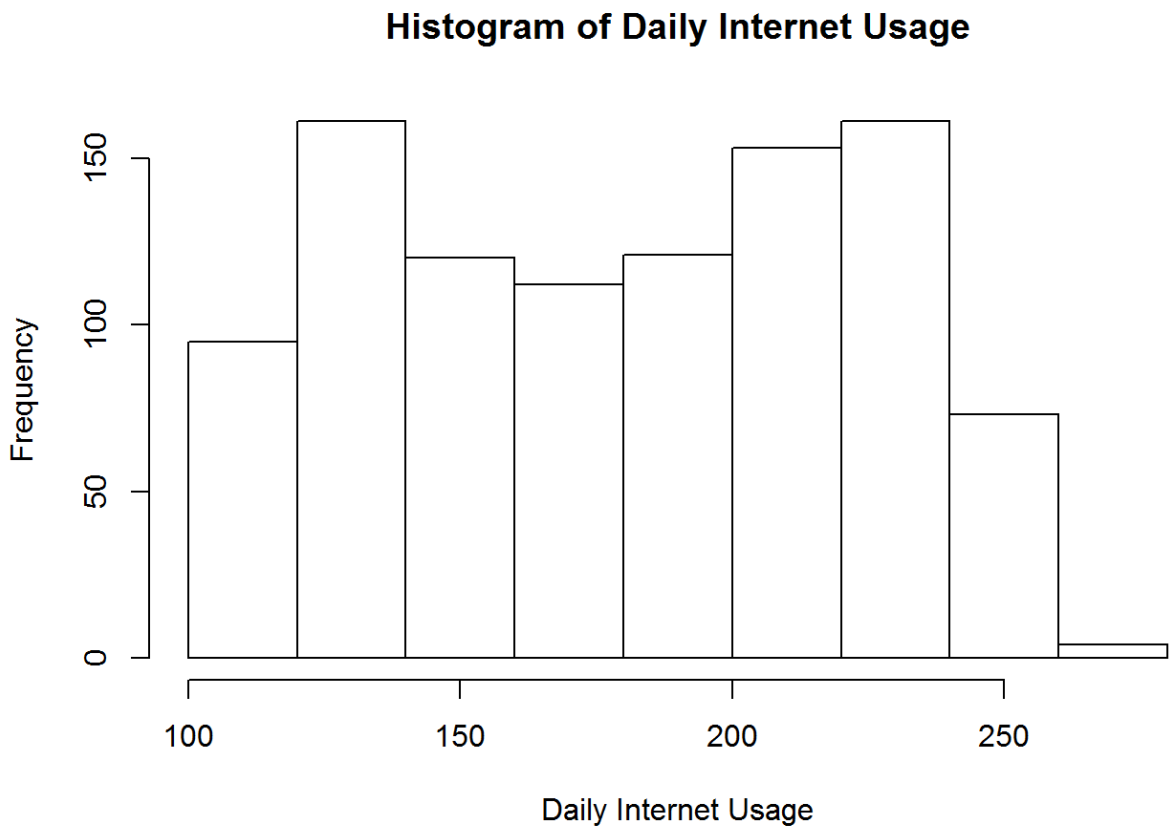
Q.5

a. We aim to explore the dataset so that we can better choose a model to implement. Plot histograms for at least 2 of the continuous variables in the dataset. Note it is acceptable to plot more than 2. [1 point]

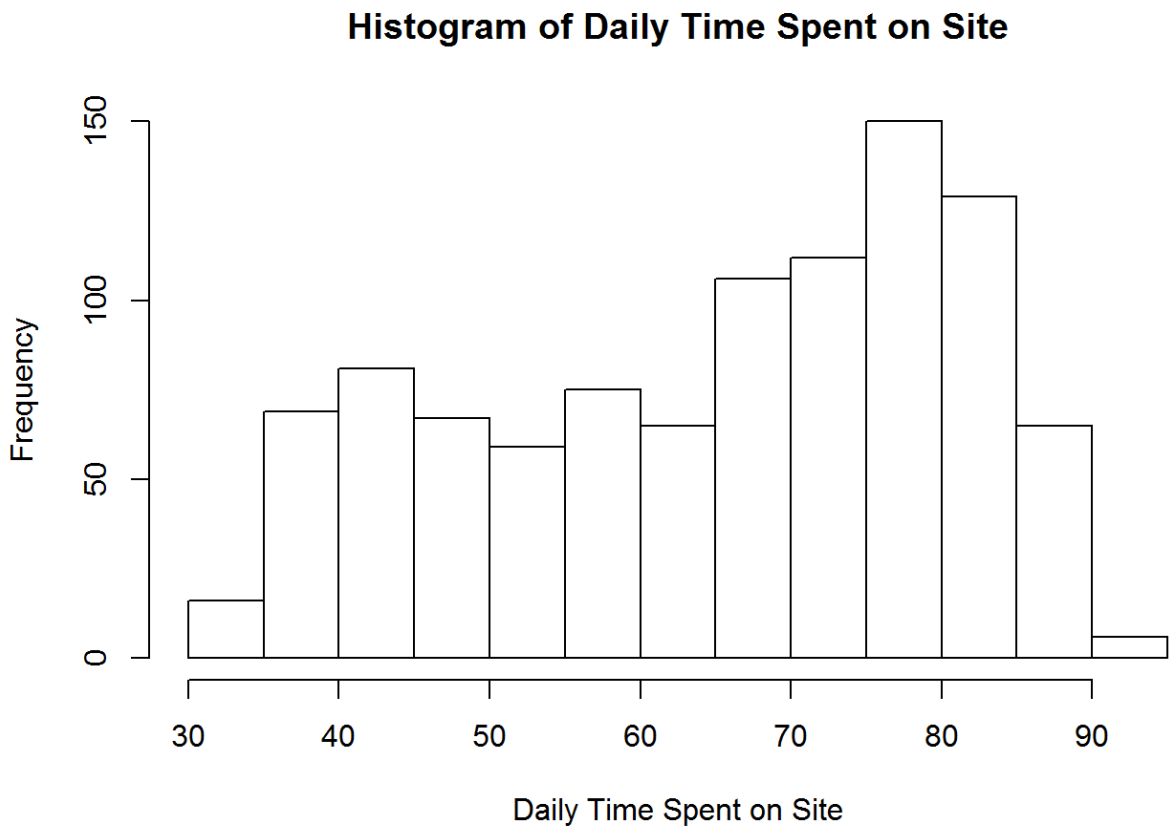
```
hist(advertising$Area.Income, xlab = "Area Income", main = "Histogram of Area I
ncome")
```




```
hist(advertising$Daily.Internet.Usage, xlab = "Daily Internet Usage", main = "H
istogram of Daily Internet Usage")
```

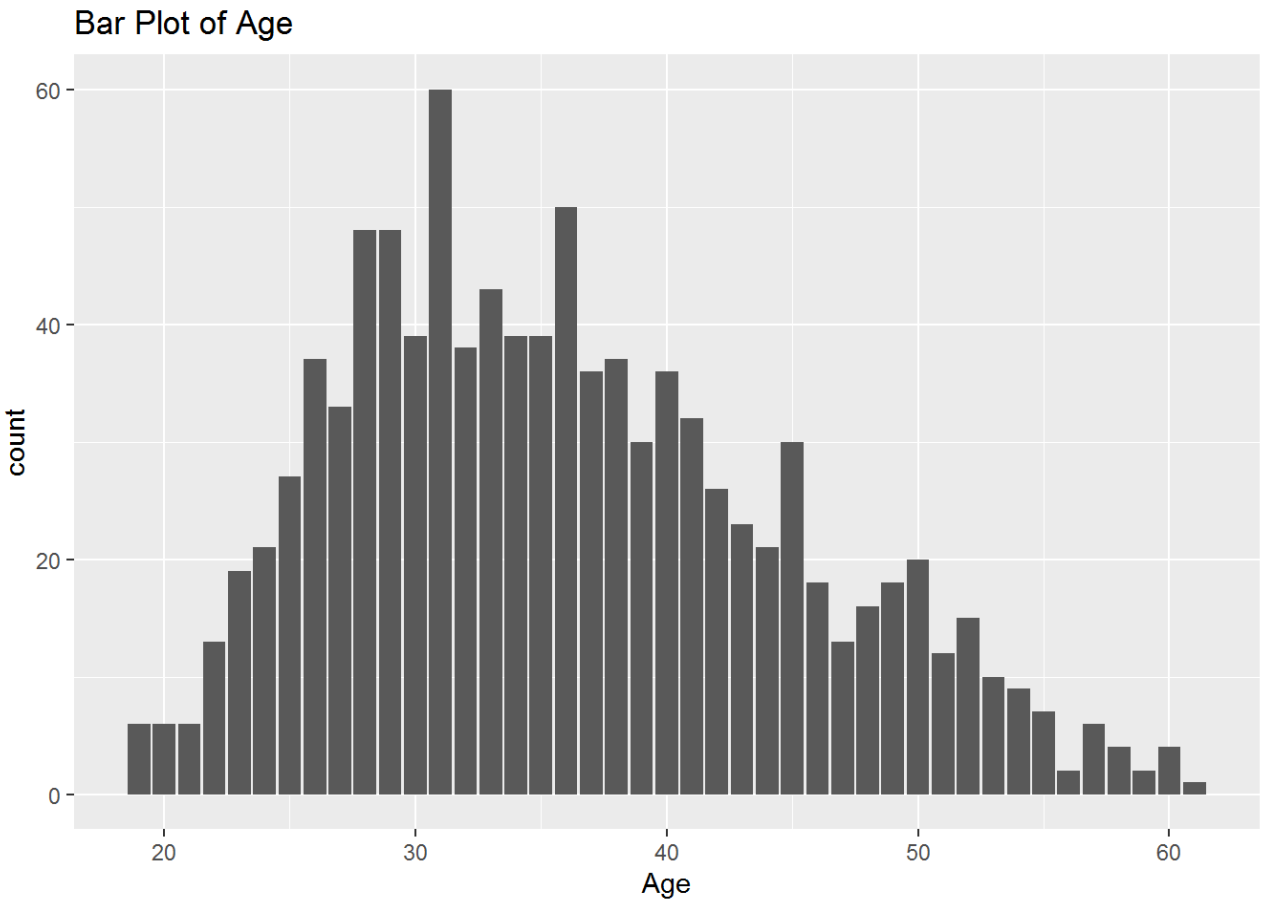



```
hist(advertising$Daily.Time.Spent.on.Site, xlab = "Daily Time Spent on Site", m
ain = "Histogram of Daily Time Spent on Site")
```

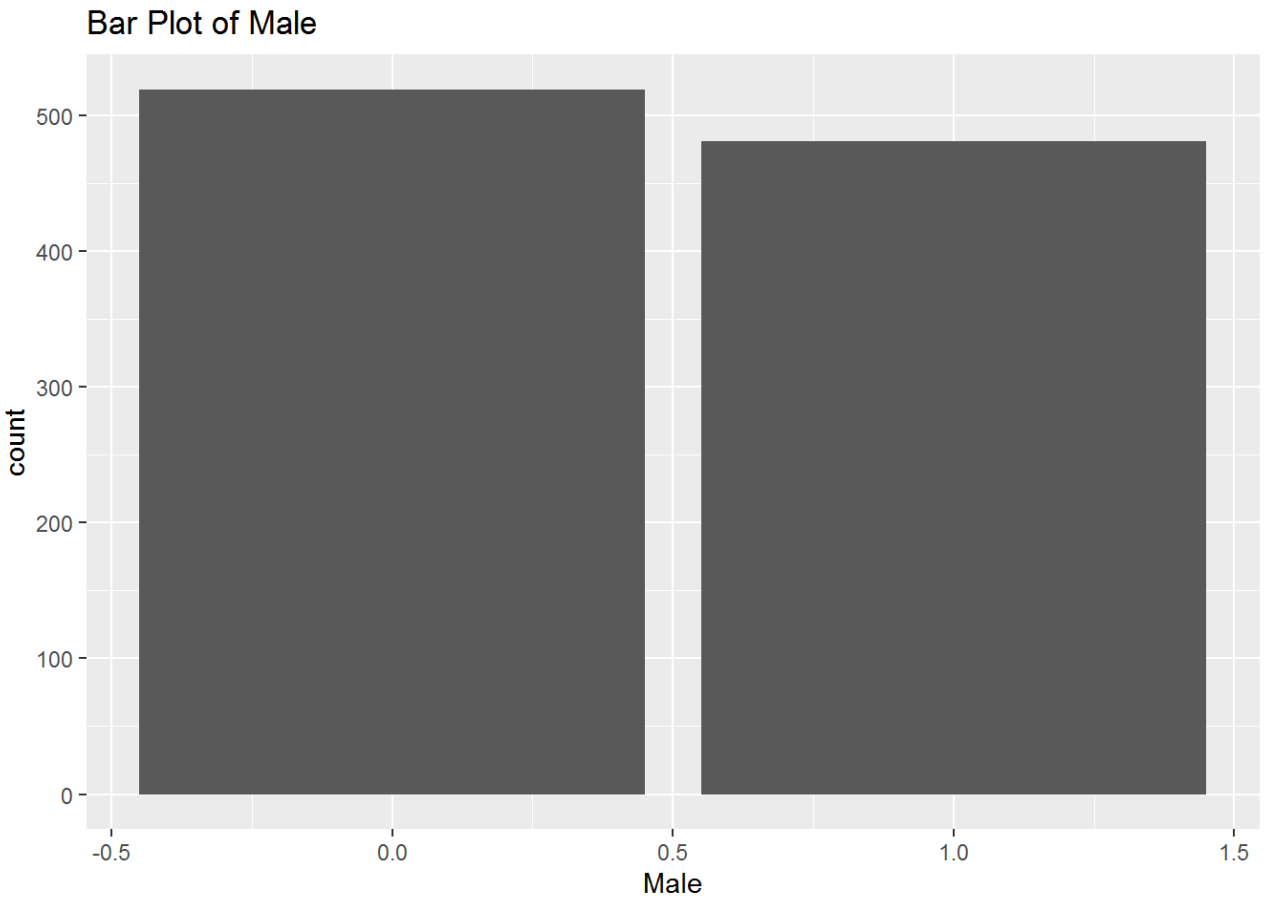


b. Again on the track of exploring the dataset, plot at least 2 bar charts reflecting the counts of different values for different variables. Note it is acceptable to plot more than 2. [1 point]

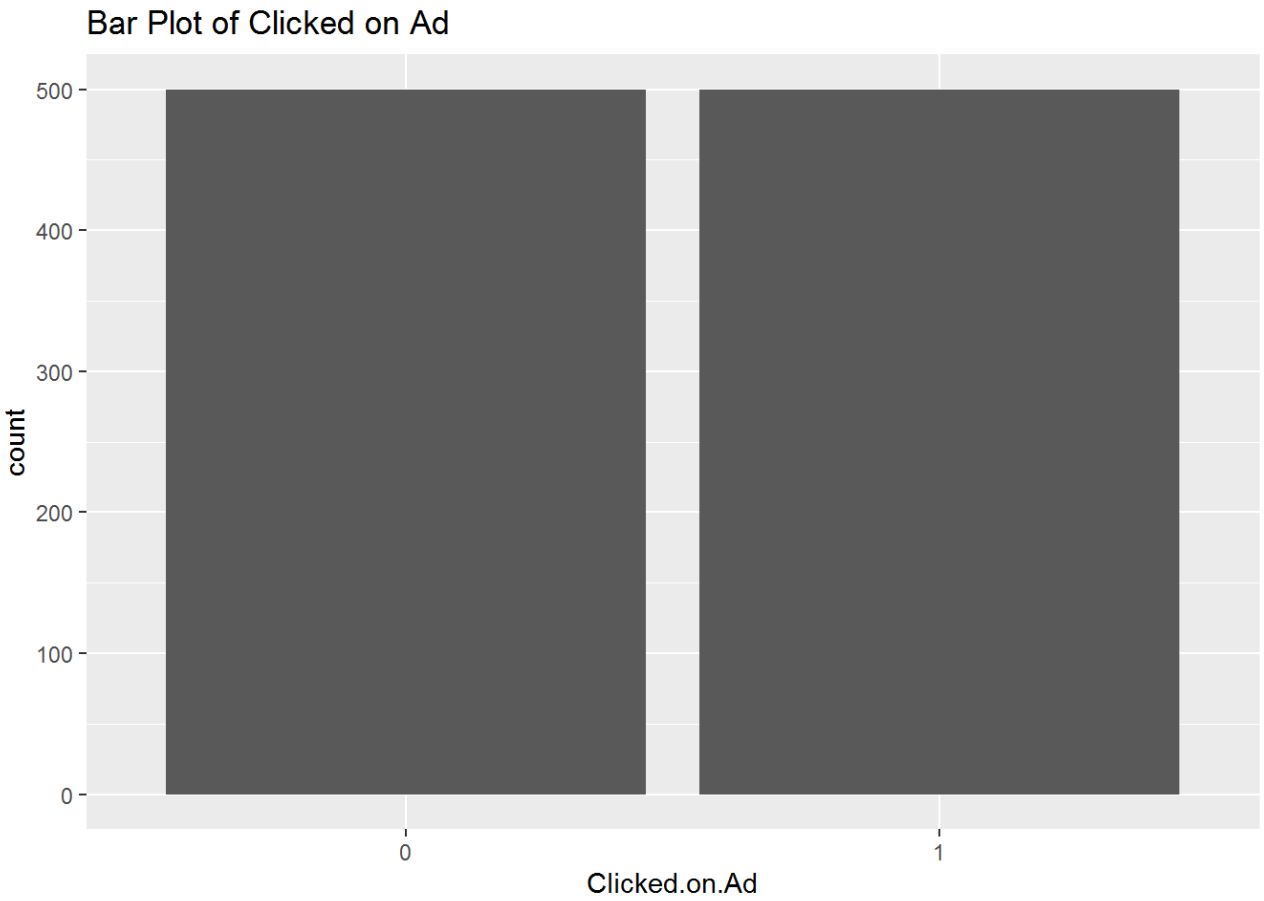
```
ggplot(advertising, aes(x = Age)) +geom_bar()+ labs(title = "Bar Plot of Age")
```




```
ggplot(advertising, aes(x = Male)) +geom_bar()+ labs(title = "Bar Plot of Mal
e")
```

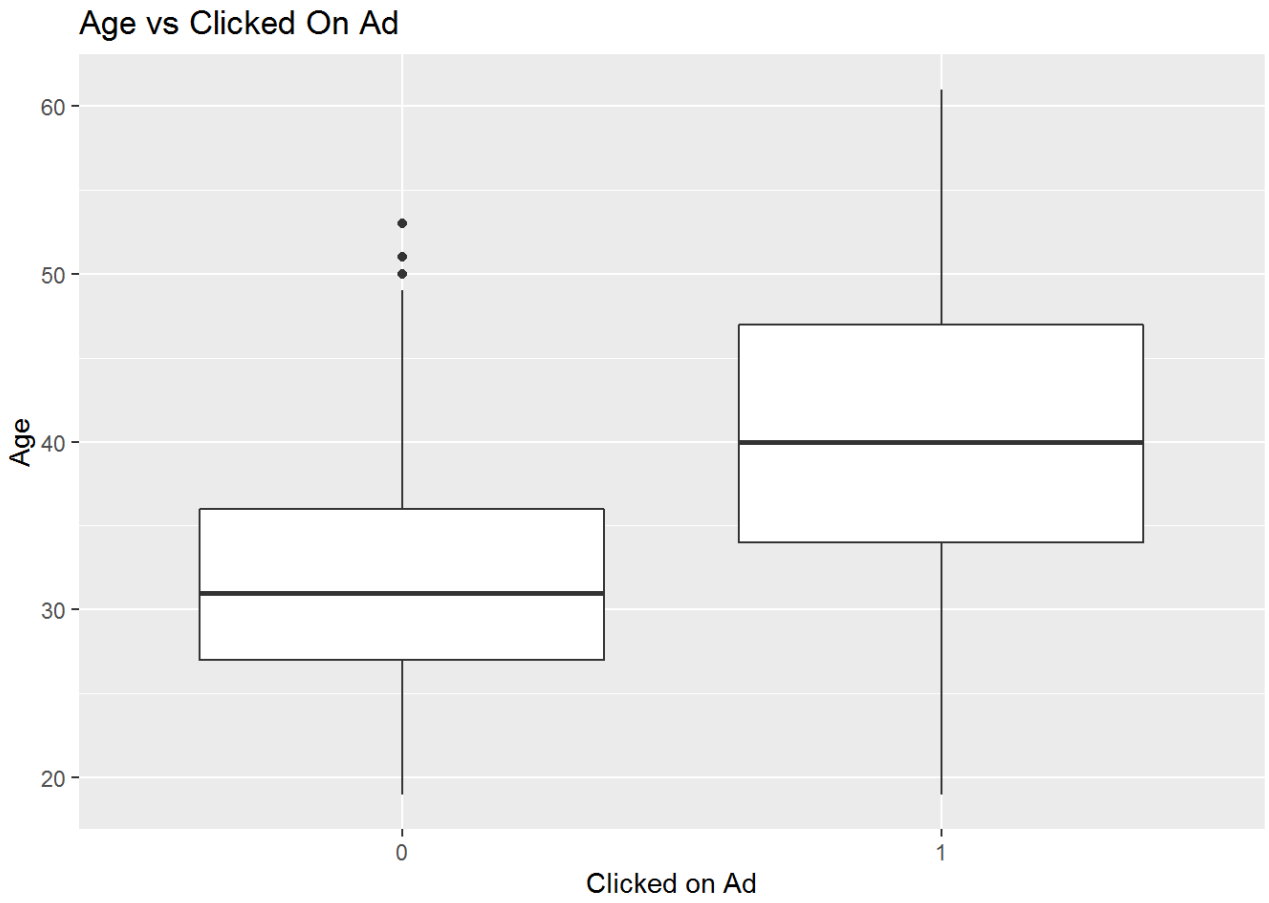



```
ggplot(advertising, aes(x = Clicked.on.Ad)) +geom_bar()+ labs(title = "Bar Plot
of Clicked on Ad")
```

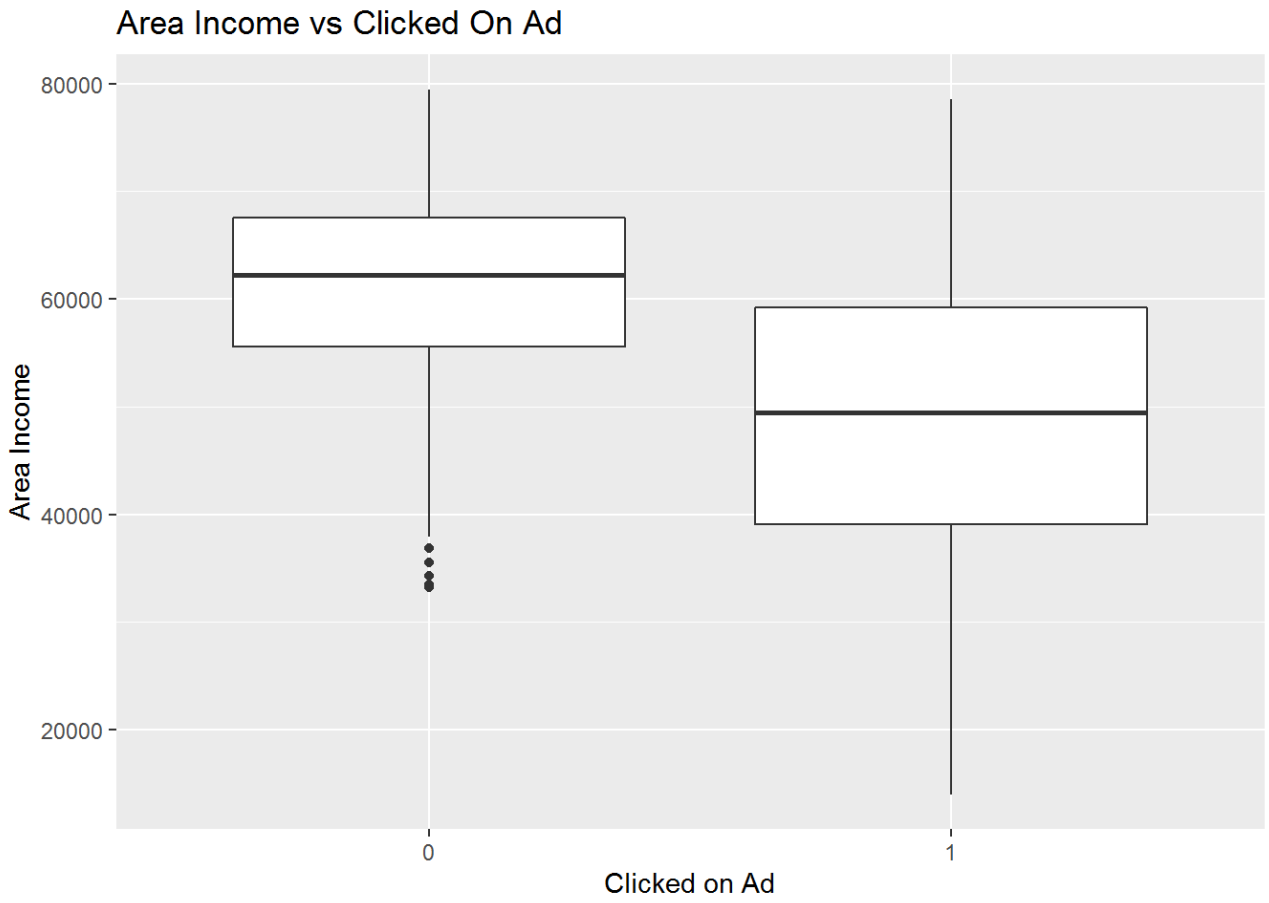


c. Plot boxplots for Age, Area.Income, Daily.Internet.Usage and Daily.Time.Spent.on.Site separated by the variable Clicked.on.Ad. To clarify, we want to create 4 plots, each of which has 2 boxplots: 1 for people who clicked on the ad, one for those who didn't. [2 points]

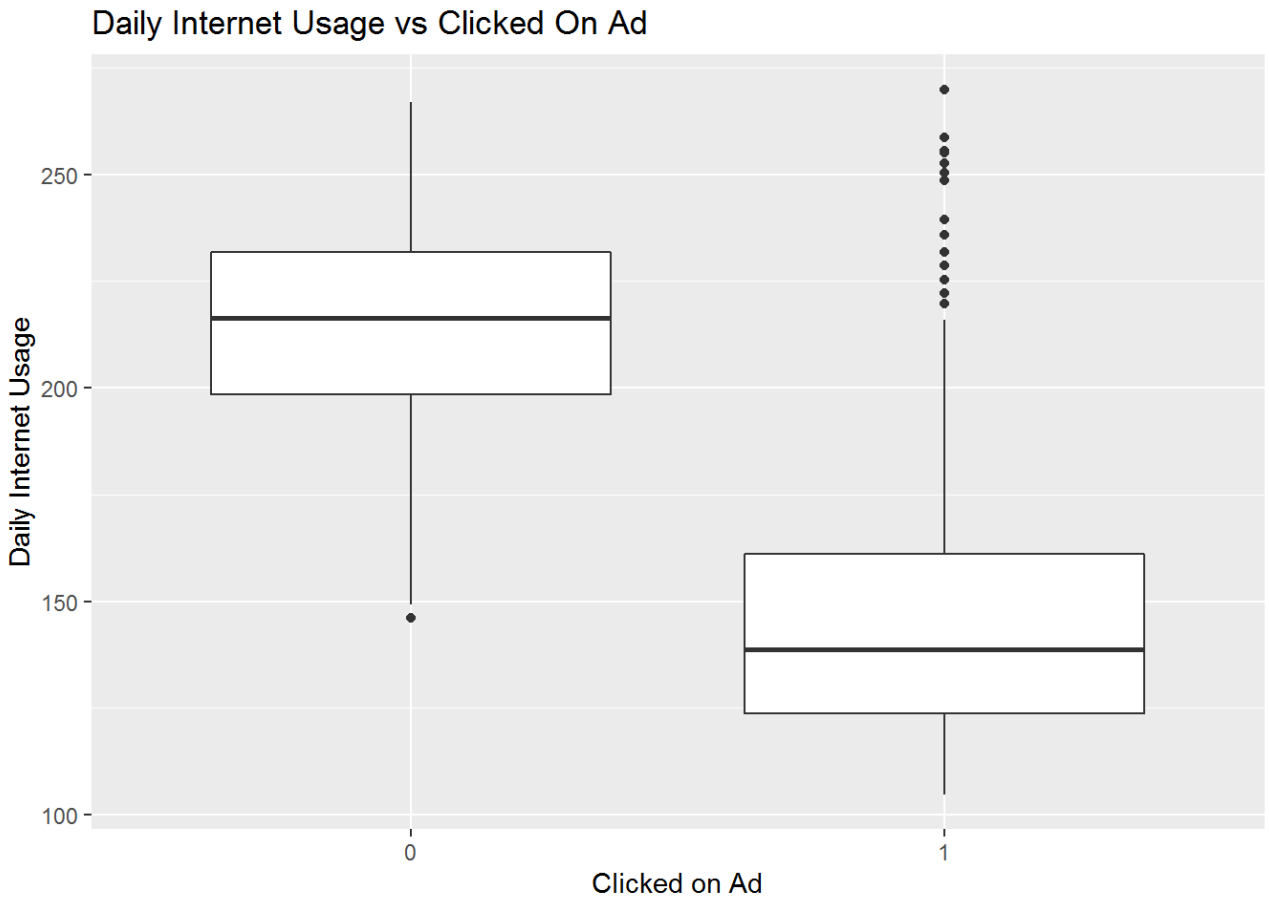
```
#Age vs Clicked On Ad
ggplot(data = advertising, mapping = aes(x = Clicked.on.Ad, y = Age)) + geom_bo
xplot() + labs(title = "Age vs Clicked On Ad", x = "Clicked on Ad")
```



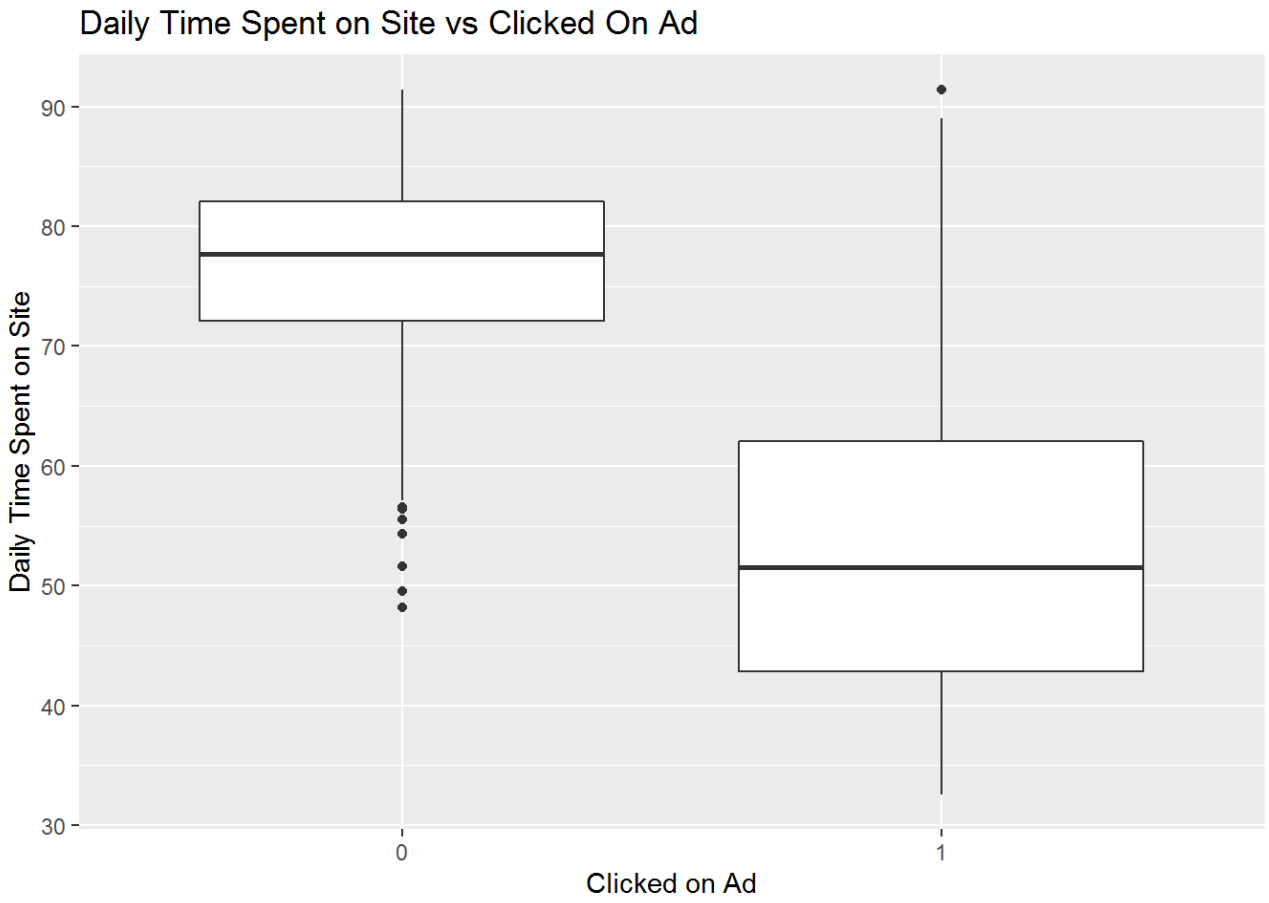

```
#Area Income vs Clicked On Ad

ggplot(data = advertising, mapping = aes(x = Clicked.on.Ad, y = Area.Income))
+ geom_boxplot() + labs(title = "Area Income vs Clicked On Ad", x = "Clicked o
n Ad",y = "Area Income")
```




```
#Daily Internet Usage vs Clicked On Ad
ggplot(data = advertising, mapping = aes(x = Clicked.on.Ad, y = Daily.Internet.
Usage)) + geom_boxplot() + labs(title = "Daily Internet Usage vs Clicked On A
d", x = "Clicked on Ad",y = "Daily Internet Usage")
```



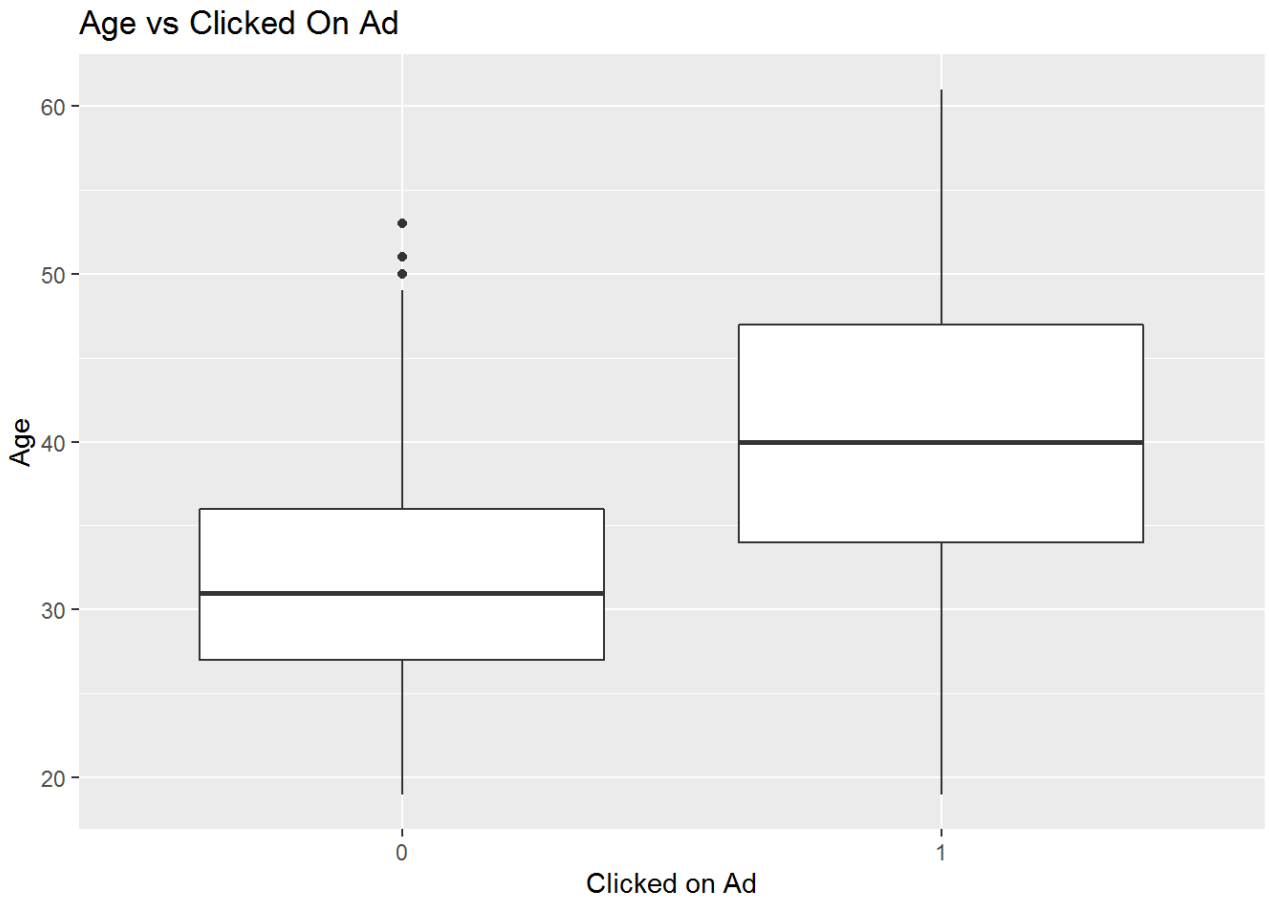

```
#Daily Time Spent on Site vs Clicked On Ad
ggplot(data = advertising, mapping = aes(x = Clicked.on.Ad, y = Daily.Time.Spen
t.on.Site)) + geom_boxplot() + labs(title = "Daily Time Spent on Site vs Clicke
d On Ad", x = "Clicked on Ad",y = "Daily Time Spent on Site")
```



d. Based on our preliminary boxplots, would you expect an older person to be more likely to click on the ad than someone younger? [2 points]

Answer: Looking at the Age vs Clicked On Ad Box Plot below, the median age of users clicking the Ad is higher than the Median age of users not clicking the AD.The maximum age for users clicking the Ad is also higher than that of the ones not clicking the AD when outliers are ignored.From this we can conclude that the tendency of older person clicking the AD is higher.

```
#Age vs Clicked On Ad
ggplot(data = advertising, mapping = aes(x = Clicked.on.Ad, y = Age)) + geom_bo
xplot()+ labs(title = "Age vs Clicked On Ad", x = "Clicked on Ad")
```

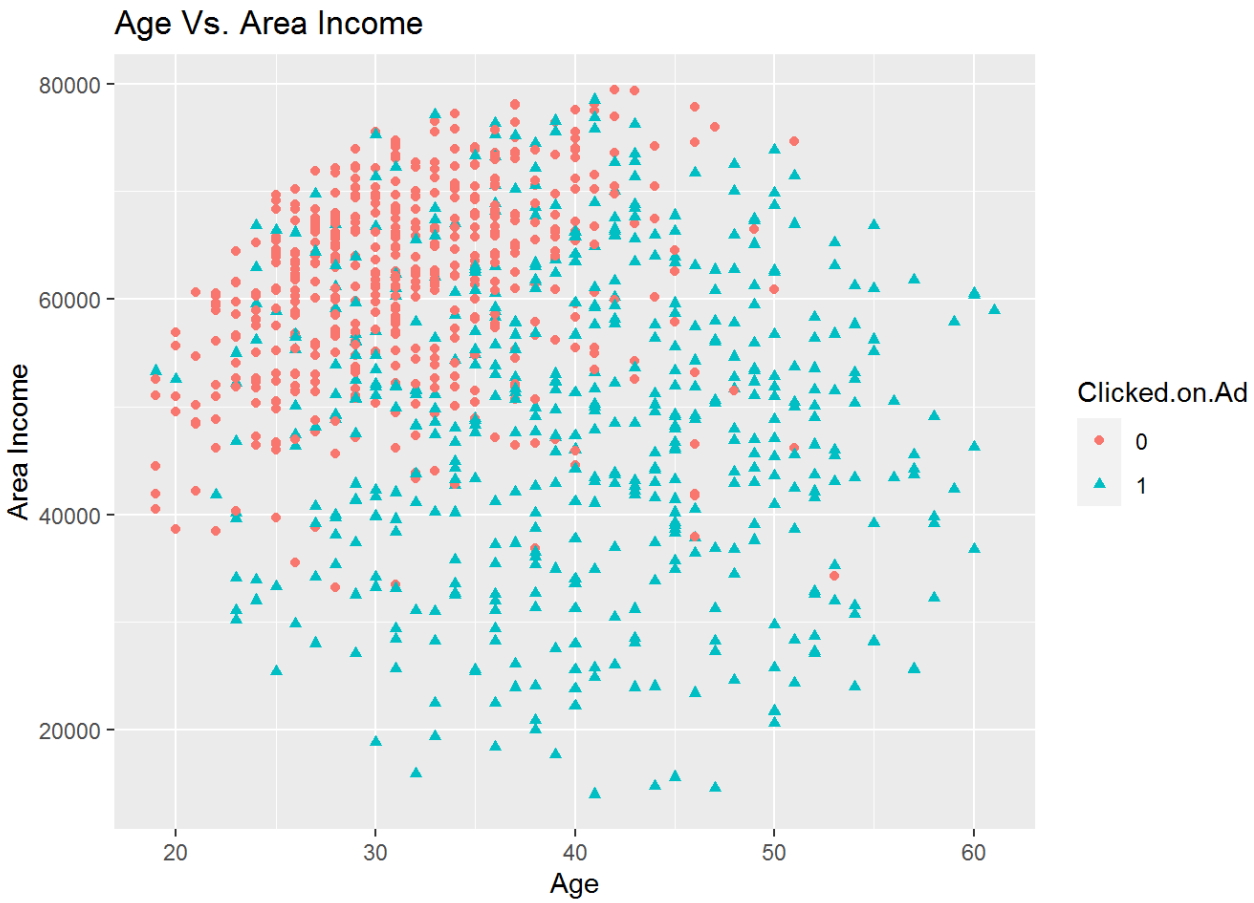


Q.6

Part (a) [3 points]

1. Make a scatter plot for Area.Income against Age. Separate the datapoints by different shapes based on if the datapoint has clicked on the ad or not.

```
ggplot(data = advertising, mapping = aes(x = Age, y = Area.Income)) + geom_poin
t(aes(shape = Clicked.on.Ad, color = Clicked.on.Ad))  + labs(title = "Age Vs. A
rea Income",y = "Area Income")
```

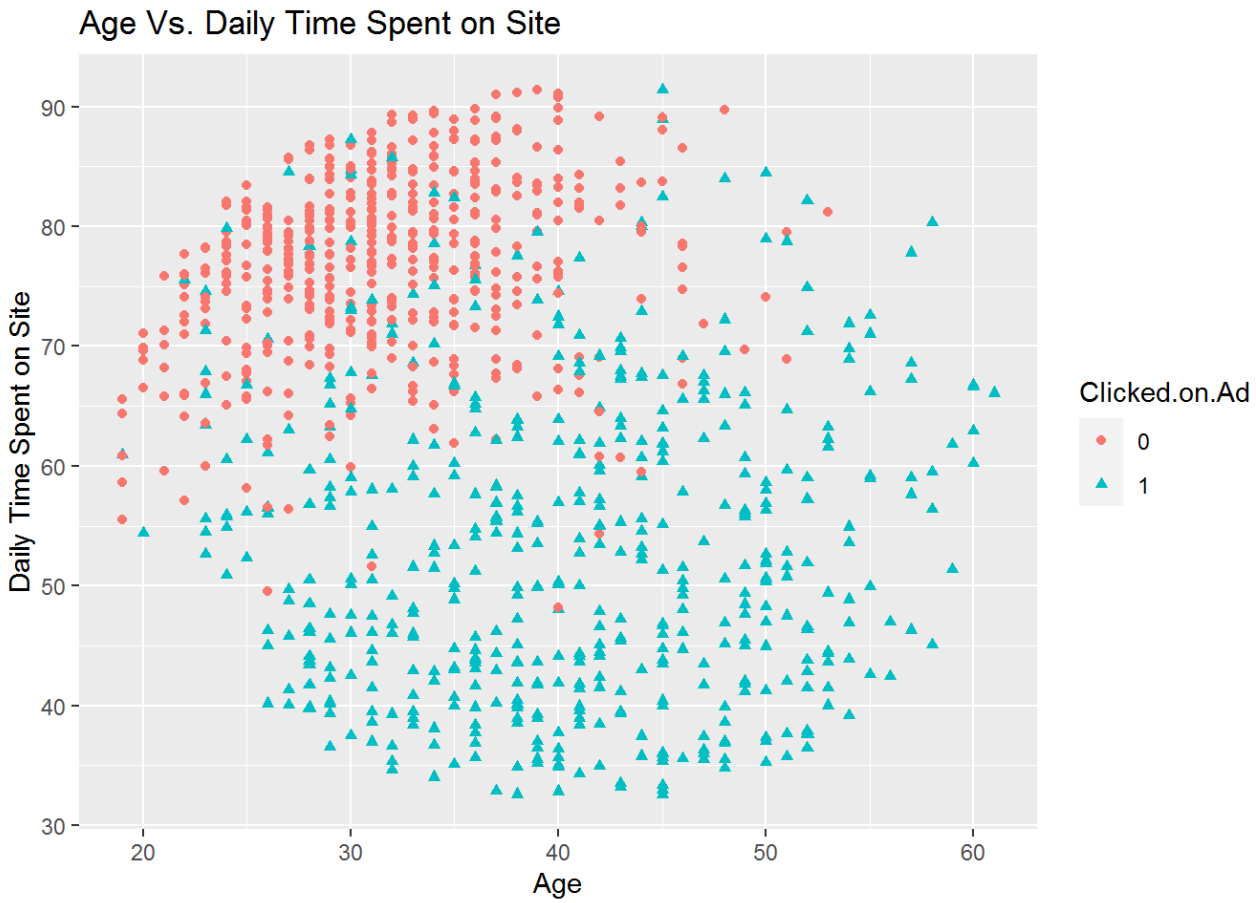


2. Based on this plot, would you expect a 31-year-old person with an Area income of $62,000 to click on the ad or not?

Answer: NO . Looking at the scatterplot, we observe that from the Age Group between 30 to 35 and income above 60,000, the number of clicks ad has reduced. So I would not expect a 31-year-old person with an Area income of $62,000 to click on the ad.

Part (b) [3 points] 1. Similar to part a), create a scatter plot for Daily.Time.Spent.on.Site against Age. Separate the datapoints by different shapes based on if the datapoint has clicked on the ad or not.

```
ggplot(data = advertising, mapping = aes(x = Age, y = Daily.Time.Spent.on.Sit
e)) + geom_point(aes(shape = Clicked.on.Ad, color = Clicked.on.Ad))  + labs(tit
le = "Age Vs. Daily Time Spent on Site" ,y = "Daily Time Spent on Site")
```



2. Based on this plot, would you expect a 50-year-old person who spends 60 minutes daily on the site to click on the ad or not?

Answer: Yes.

Q.7

Part (a) [2 points]

1. Now that we have done some exploratory data analysis to get a better understanding of our raw data, we can begin to move towards designing a model to predict advert clicks.
2. Generate a correlation funnel (using the correlation funnel package) to see which of the variable in the dataset have the most correlation with having clicked the advert.

NOTE: Here we are creating the correlation funnel in regards to HAVING clicked the advert, rather than not. This will lead to a minor distinction in your code between the 2 cases. However, it will not affect your results and subsequent variable selection.

```
ad4=advertising
ad4$Age =as.factor(ad4$Age)
ad4$Male=as.factor(ad4$Male)
ad_binarized_tbl <- ad4 %>%
  binarize()
```

```
## Warning: All elements of `...` must be named.
## Did you want `data = c(type, role, source)`?
```

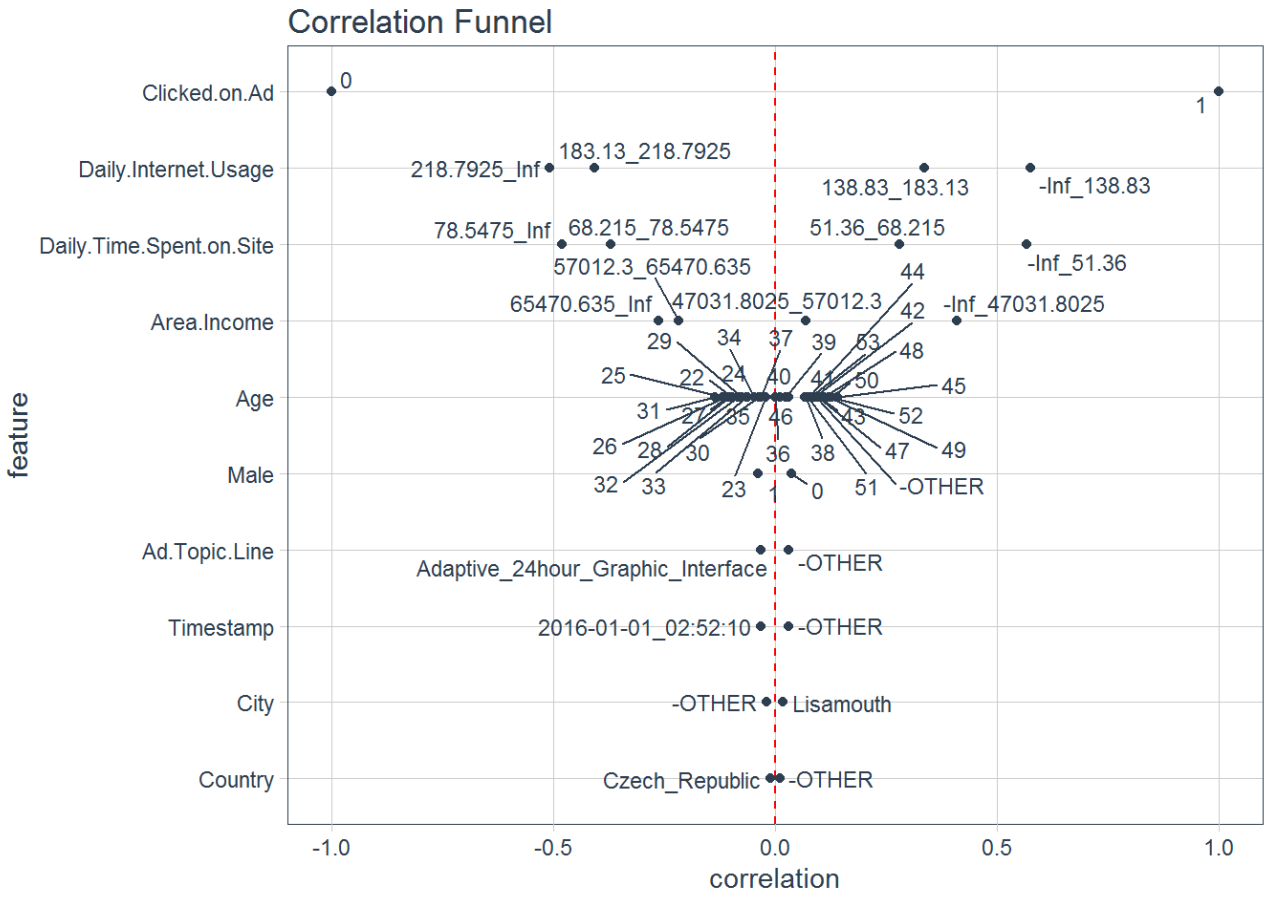
```
ad_binarized_corr_tbl <- ad_binarized_tbl %>%
  correlate(Clicked.on.Ad__1)
```

```
## Warning: The `x` argument of `as_tibble.matrix()` must have column names if
`.name_repair` is omitted as of tibble 2.0.0.
## Using compatibility `.name_repair`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
ad_binarized_corr_tbl
```

```
## # A tibble: 57 x 3
##    feature                  bin             correlation
##    <fct>                    <chr>                 <dbl>
##  1 Clicked.on.Ad            0                      -1
##  2 Clicked.on.Ad            1                       1
##  3 Daily.Internet.Usage     -Inf_138.83             0.577
##  4 Daily.Time.Spent.on.Site -Inf_51.36              0.568
##  5 Daily.Internet.Usage     218.7925_Inf           -0.508
##  6 Daily.Time.Spent.on.Site 78.5475_Inf            -0.480
##  7 Area.Income              -Inf_47031.8025         0.411
##  8 Daily.Internet.Usage     183.13_218.7925        -0.406
##  9 Daily.Time.Spent.on.Site 68.215_78.5475         -0.370
## 10 Daily.Internet.Usage     138.83_183.13           0.337
## # ... with 47 more rows
```

```
ad_binarized_corr_tbl %>%
  plot_correlation_funnel()
```



Part (b) [2 points]

1. Based on the generated correlation funnel, choose the 4 most covarying variables (with having clicked the advert) and run a logistic regression model for Clicked.on.Ad using these 4 variables. The 4 most covarying variable are

1. Daily.Time.Spent.on.Site
2. Age
3. Area.Income
4. Daily.Internet.Usage

```
advertising_logistic_regression <- glm(data=advertising,Clicked.on.Ad ~ Daily.T
ime.Spent.on.Site+Age+Area.Income+Daily.Internet.Usage, family = 'binomial')
```

2. Output the summary of this model.

```
summary(advertising_logistic_regression)
```

```
##
## Call:
## glm(formula = Clicked.on.Ad ~ Daily.Time.Spent.on.Site + Age +
##     Area.Income + Daily.Internet.Usage, family = "binomial",
##     data = advertising)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4578  -0.1341  -0.0333   0.0167   3.1961
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               2.713e+01  2.714e+00   9.995  < 2e-16 ***
## Daily.Time.Spent.on.Site -1.919e-01  2.066e-02  -9.291  < 2e-16 ***
## Age                       1.709e-01  2.568e-02   6.655 2.83e-11 ***
## Area.Income              -1.354e-04  1.868e-05  -7.247 4.25e-13 ***
## Daily.Internet.Usage     -6.391e-02  6.745e-03  -9.475  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1386.3  on 999  degrees of freedom
## Residual deviance:  182.9  on 995  degrees of freedom
## AIC: 192.9
##
## Number of Fisher Scoring iterations: 8
```

Q.8 [4 points]

Now that we have created our logistic regression model using variables of significance, we must test the model. When testing such models, it is always recommended to split the data into a training (from which we build the model) and test (on which we test the model) set. This is done to avoid bias, as testing the model on the data from which it is originally built from is unrepresentative of how the model will perform on new data. That said, for the case of simplicity, test the model on the full original dataset. Use type =“response” to ensure we get the predicted probabilities of clicking the advert Append the predicted probabilities to a new column in the original dataset or simply to a new data frame. The choice is up to you, but ensure you know how to reference this column of probabilities. Using a threshold of 80% (0.8), create a new column in the original dataset that represents if the model predicts a click or not for that person. Note this means probabilities above 80% should be treated as a click prediction. Now using the caret package, create a confusion matrix for the model predictions and actual clicks. Note you do not need to graph or plot this confusion matrix. How many false-negative occurrences do you observe? Recall false negative means the instances where the model predicts the case to be false when in reality it is true. For this example, this refers to cases where the ad is clicked but the model predicts that it isn’t

```
advertising$predictreg =predict(advertising_logistic_regression, advertising, t
ype="response")
advertising$predictvalue <- ifelse(advertising$predictreg>0.8, 1,0)
xtab <- table(advertising$Clicked.on.Ad,advertising$predictvalue)
confusionMatrix(xtab)
```

```
## Confusion Matrix and Statistics
##
##
##       0   1
##   0 497   3
##   1  36 464
##
##                Accuracy : 0.961
##                  95% CI : (0.9471, 0.9721)
##     No Information Rate : 0.533
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.922
##  Mcnemar's Test P-Value : 2.99e-07
##
##             Sensitivity : 0.9325
##             Specificity : 0.9936
##          Pos Pred Value : 0.9940
##          Neg Pred Value : 0.9280
##              Prevalence : 0.5330
##          Detection Rate : 0.4970
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.9630
##
##        'Positive' Class : 0
##
```