HW3 Part 2

July 31, 2021

1 MGT6203 HW3 Part 2 (60 Points Total)

1.1 Instructions:

For Homework 3 Part 2, please use this R notebook in Vocareum to submit your solutions. Vocareum is an educational cloud platform for programming in several languages; it is based on the Jupyter notebook environment. This platform allows us to move homework assignments to the cloud. The advantages are that all of you will be working in the same coding environment AND peer reviewers will be able to run your R code easily. This way we eliminate some issues we might encounter when working on an individual/local basis, such as library installations and Rstudio OS requirements; R notebooks work on mobile platforms and tablets.

With R notebooks, you will be learning a new way of presenting data analysis reports, that is neat and flexible, where formatted (English) text and (R) code can easily coexist on the same page. Notebooks can be also collaborative when needed. For now, we are asking each of you to do your own work for homework. Think of R notebooks as interactive program-based Google docs or MS-Office 360 docs; these are gradually replacing local files on our computers.

Many of you are new to the R notebooks and Vocareum platforms. We will provide TA help in Piazza with specific code if you have questions. Here we list some important things to get you started. Please read through them carefully.

Even though we are moving from your local envrionment to the cloud, **our expectations from your homework will remain the same**. Same goes for the rubrics.

Vocareum has its own cloud based file system, the data files you will be using for the assignments will be stored in the cloud with path **"../resource/asnlib/publicdata/FILENAME.csv"**. You will be able to import them with the same method as you do in RStudio, simply substitute the path name to the one specified in the instructions. You won't be able to modify these data files.

You will be able to find the data files on Canvas/EdX if you would like to explore them offline.

For coding questions, you will be graded on the R code as well as the output in your submission.

For interpretations or short response questions, please type the answers in the notebook's markdown cells. To change a code cell to a markdown cell, click on the cell, and in the dropdown menu above, switch the type of the cell block from "code" to "markdown". **Adding print statements to code cells for short response/interpretation questions is also fine, as long as we can clearly see the output of your response**.

You don't need to, but if you would like to learn more about how to format your markdown cells, visit the following site: https://www.earthdatascience.org/courses/intro-to-earth-data-science/file-formats/use-text-files/format-text-with-markdown-jupyter-notebook/. Jupyter notebook also support LaTeX.

Feel free to delete or add as many additional cells as you need. But please try to keep your notebook clean and keep your solution to a question directly under that question to avoid confusions.

You may delete the #SOLUTION BEGINS/ENDS HERE comments from the cell blocks, they are just pointers that indicates where to put you solutions.

When you have finished the assignment, remember to rerun your notebook to check if it runs correctly. You can do so by going to **Kernel-> Restart & Run All**. You may lose points if your solutions does not run successfully.

Click the "Submit" button on the top right corner to turn in your assignment. Your assignment will enter the next phase for peer review.

You are allowed a total of 2 submissions for this assignment. So make sure that you submit your responses carefully. You will be able to come back and resubmit your assignment as long as it is before the start of the peer review period.

Please remember to finish the peer reviews after you have submitted your assignment. You are responsible for grading the work of three of your peers thoroughly, and in addherence to the rubrics. And you will be held accountable for peer grading. **There will be a 30% penalty to your grade if you fail to complete one or more peer reviews in proper fashion.**

Feel free to address your questions, concerns, and provide any feedback on Piazza. We will continuously try to improve going forward.

Good Luck!

2 About Package Installation:

Most of the packages (if not all) that you will need to complete this assignment are already installed in this environment. An easy way to check this is to run the command: library(PackageName). If this command runs successfully then the package was already installed and has been successfully attached to the code. If the command gave an error saying the Package was not found then follow the steps below to successfully install the package and attach it to the code:

Use *installed.packages()* command to return a table of the packages that are preinstalled in the environment.

To attach a preinstalled library in Vocareum, simply use *library(PackageName)*

To install a package that does not come with the provided environment, please use the following syntax:

install.packages("PackageName", lib="../work/")

To attach a library you just installed, use the following syntax:

library(PackageName, lib.loc="../work/")

Make sure the file location is the same as the above code snippets *("../work/")*

2.1 Instructions for Q1 to Q2:

Please use the Facebook Ad dataset 'KAG.csv' for the next set of questions. We advise solving these questions using R (preferably using dplyr library wherever applicable) after reviewing the code provided for Week 11 and other resources provided for learning dplyr in R Learning Guide.

Load the dataset as below:

data <- read.csv("../resource/asnlib/publicdata/KAG.csv", stringsAsFactors = FALSE)

IMPORTANT NOTE: For no clicks and no amount spent, please consider CPC as 0.

2.1.1 Q1. (8 Points)

a. Among the ads that have the least CPC, which one ad leads to the most impressions? (provide ad_id as the answer) (4 Points)

```
In [1]: # SOLUTION BEGINS HERE
```

```
In [2]: if (!require("dplyr")) {install.packages("dplyr")
                                library("dplyr")}
```

```
Loading required package: dplyr


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
```

```
In [3]: data = read.csv("../resource/asnlib/publicdata/KAG.csv", stringsAsFactors = FALSE)
        str(data)
        head(data)
        unique(data$campaign_id)
```

```
'data.frame':        1143 obs. of  15 variables:
 $ X                  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ ad_id              : int  708746 708749 708771 708815 708818 708820 708889 708895 708953 70
 $ campaign_id        : int  916 916 916 916 916 916 916 916 916 916 ...
 $ age                : int  32 32 32 32 32 32 32 32 32 32 ...
 $ gender             : int  0 0 0 0 0 0 0 0 0 0 ...
 $ interest           : int  15 16 20 28 28 29 15 16 27 28 ...
 $ Impressions        : int  7350 17861 693 4259 4133 1915 15615 10951 2355 9502 ...
 $ Clicks             : int  1 2 0 1 1 0 3 1 1 3 ...
 $ Spent              : num  1.43 1.82 0 1.25 1.29 ...
 $ Total_Conversion   : int  2 2 1 1 1 1 1 1 1 1 ...
 $ Approved_Conversion : int  1 0 0 0 1 1 0 1 0 0 ...
 $ CTR                : num  0.0136 0.0112 0 0.0235 0.0242 0 0.0192 0.0091 0.0425 0.0316 ...
 $ CPC                : num  1.43 0.91 0 1.25 1.29 ...
 $ CostPerConv_Total  : num  0.715 0.91 0 1.25 1.29 0 4.77 1.27 1.5 3.16 ...
 $ CostPerConv_Approved: num  1.43 1.82 0 1.25 1.29 ...
```

A data.frame: 6 Œ 15

	X <int>	ad_id <int>	campaign_id <int>	age <int>	gender <int>	interest <int>	Impressions <int>	Clicks <int>
1	1	708746	916	32	0	15	7350	1
2	2	708749	916	32	0	16	17861	2
3	3	708771	916	32	0	20	693	0
4	4	708815	916	32	0	28	4259	1
5	5	708818	916	32	0	28	4133	1
6	6	708820	916	32	0	29	1915	0

1. 916 2. 936 3. 1178

```
In [4]: summarize_at(data[data$Clicks == 0 & data$Spent == 0,], vars(CTR, CPC), list(mean))
```

A data.frame: 1 Œ 2

CTR <dbl>	CPC <dbl>
0	0

```
In [5]: data %>% filter(CPC == min(CPC)) %>% filter(Impressions == max(Impressions))
```

A data.frame: 1 Œ 15

X <int>	ad_id <int>	campaign_id <int>	age <int>	gender <int>	interest <int>	Impressions <int>	Clicks <int>	Sp <
521	1121094	1178	32	0	10	24362	0	0

```
In [6]: # SOLUTION ENDS HERE
```

b. What campaign (provide campaign_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)? (4 Points) $i.e.\ to\ calculate\ most\ Cost\ Per\ Mille\ or\ CPM, use\ :$ $\frac{total\ cost\ for\ the\ campaign}{total\ impressions\ in\ thousands}$

```
In [7]: # SOLUTION BEGINS HERE

In [8]: data %>% mutate(CPM = Spent/(Impressions/1000)) %>%
        group_by(campaign_id) %>%
        summarize(mean_CPM = mean(CPM), n(), .groups = 'drop') %>%
        filter(mean_CPM == max(mean_CPM))
```

A tibble: 1 Œ 3

campaign_id <int>	mean_CPM <dbl>	n() <int>
1178	0.2505577	625

```
In [9]: # SOLUTION ENDS HERE
```

2.1.2 Q2. (8 Points)

Assume each conversion ('Total_Conversion') is worth $10, each approved conversion ('Approved_Conversion') is worth $50. ROAS (return on advertising spent) is revenue divided by advertising spent. Calculate ROAS and round it to two decimals. (Use 'Spent' as the Cost in the given ROAS formula, remove data points where Spent=0)

$$Hint: ROAS = \frac{Revenue}{Spent} = \frac{10 \times TotalConversion + 50 \times ApprovedConversion}{Spent} \times 100\%$$

a. Make a boxplot of the ROAS grouped by gender for interest_id = 15, 21, 101 in one graph. Try to use the function '+ scale_y_log10()' in ggplot to make the visualization look better. The x-axis label should be 'Interest ID' while the y-axis label should be ROAS; and each interest_id will have two boxplots (one boxplot for each gender). (4 Points) Hint: Remember to filter the advertisements where there is no advertising spent.
Hint: ROAS should be rounded to the second decimal.

```
In [10]: # SOLUTION BEGINS HERE
```

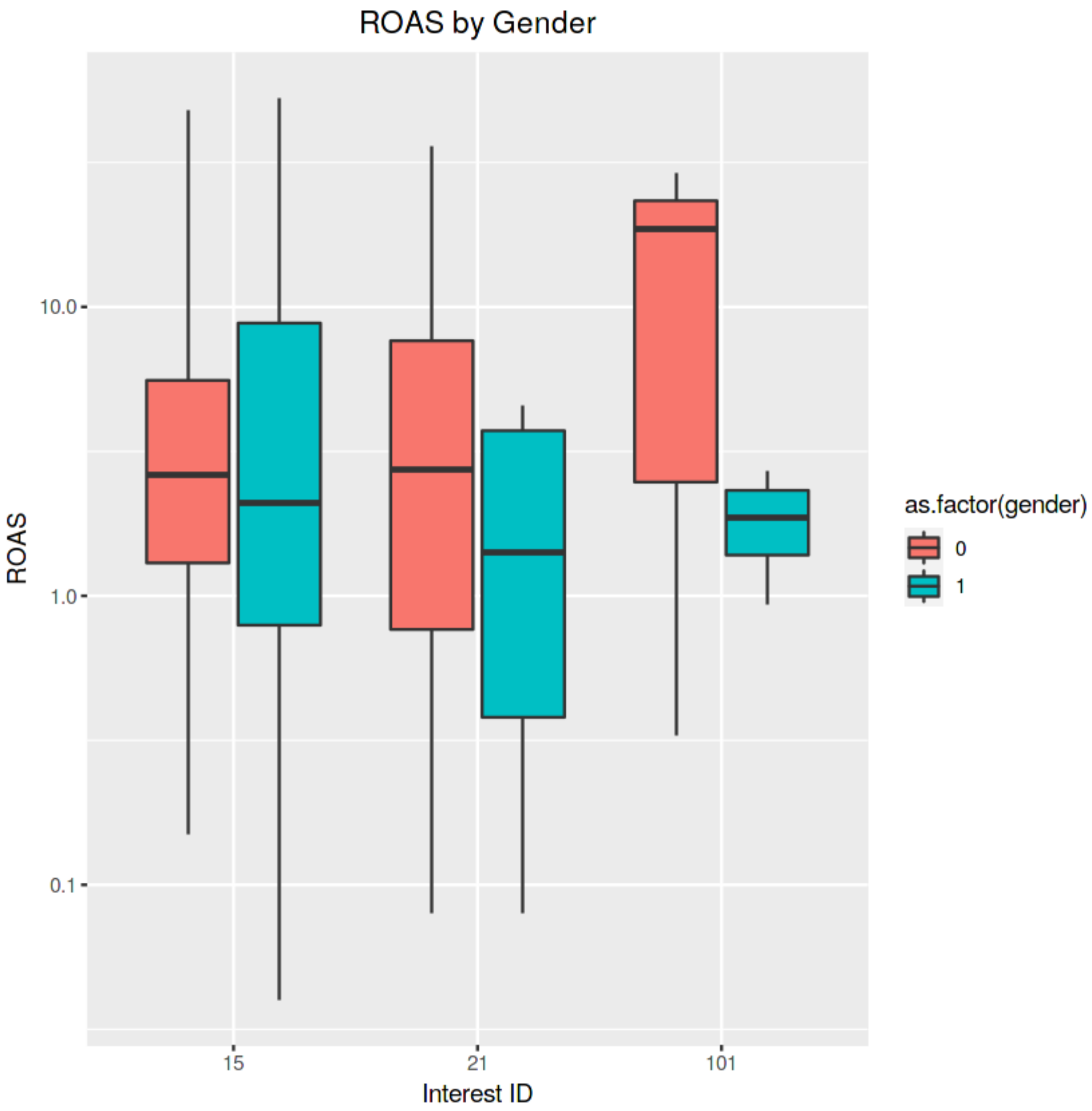
```
In [11]: data = data %>% mutate(ROAS = round((5*Total_Conversion + 50*Approved_Conversion)/Sper
         data_box = data %>% filter(Spent != 0) %>%
             filter(interest == 15|interest == 21|interest == 101)
         str(data_box)
```

```
'data.frame':        71 obs. of  16 variables:
 $ X                  : int  1 7 19 22 35 39 43 50 57 107 ...
 $ ad_id              : int  708746 708889 709179 709323 710360 710617 710763 711623 734215 73
 $ campaign_id        : int  916 916 916 916 916 916 916 916 936 936 ...
 $ age                : int  32 32 37 37 47 32 32 42 32 47 ...
 $ gender             : int  0 0 0 0 0 1 1 1 0 1 ...
 $ interest           : int  15 15 15 15 21 15 15 15 15 15 ...
 $ Impressions        : int  7350 15615 4627 7132 2182 5014 5369 3812 13659 69708 ...
 $ Clicks             : int  1 3 1 2 1 1 1 1 3 20 ...
 $ Spent              : num  1.43 4.77 1.69 2.61 1.53 ...
 $ Total_Conversion   : int  2 1 1 1 1 1 1 2 1 1 ...
 $ Approved_Conversion : int  1 0 0 0 1 0 0 1 0 0 ...
 $ CTR                : num  0.0136 0.0192 0.0216 0.028 0.0458 0.0199 0.0186 0.0262 0.022 0.0
 $ CPC                : num  1.43 1.59 1.69 1.3 1.53 ...
 $ CostPerConv_Total   : num  0.715 4.77 1.69 2.61 1.53 ...
 $ CostPerConv_Approved: num  1.43 4.77 1.69 2.61 1.53 ...
 $ ROAS               : num  41.96 1.05 2.96 1.92 35.95 ...
```

```
In [12]: if (!require("ggplot2")) {install.packages("ggplot2")
                                 library("ggplot2")}
```

```
Loading required package: ggplot2
```

```
In [13]: ggplot(data_box,
                aes(x = as.factor(interest), y = ROAS, fill = as.factor(gender))) +
         geom_boxplot() +
         scale_y_log10() +
         xlab("Interest ID") +
         ylab("ROAS") +
         ggtitle("ROAS by Gender") +
         theme(plot.title = element_text(hjust = 0.5))
```



```
In [14]: # SOLUTION ENDS HERE
```

b. Summarize the median and mean of ROAS by genders when campign_id == 1178. (4 Points)
Hint: Remember to filter the advertisements where there is no advertising spent. Hint: Your answer should include all interest_ids
Hint: ROAS should be rounded to the second decimal.

```
In [15]: # SOLUTION BEGINS HERE
```

```
In [16]: data %>% filter(campaign_id == 1178) %>%
           filter(Spent != 0) %>%
           group_by(gender) %>%
           summarize("Mean ROAS" = round(mean(ROAS), 2), "Median ROAS" = median(ROAS), count
```

A tibble: 2 Œ 4

gender <int>	Mean ROAS <dbl>	Median ROAS <dbl>	count <int>
0	2.55	1.19	341
1	1.58	0.73	272

```
In [17]: # SOLUTION ENDS HERE
```

2.2 Instructions for Q3 to Q5:

Using the Advertising.csv dataset and the following setup instructions to solve the questions.

```
In [18]: library(pROC)
         library(caret)
         library(dplyr)
         library(ggplot2)

         data <- read.csv("../resource/asnlib/publicdata/Advertising.csv", header = TRUE, stri
         data$Clicked.on.Ad <- as.factor(data$Clicked.on.Ad)
         head(data)
```

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


Attaching package: pROC


The following objects are masked from package:stats:

    cov, smooth, var


Loading required package: lattice
```

A data.frame: 6 Œ 10

	Daily.Time.Spent.on.Site <dbl>	Age <int>	Area.Income <dbl>	Daily.Internet.Usage <dbl>	Ad.Topic. <chr>
1	68.95	35	61833.90	256.09	Cloned 5t
2	80.23	31	68441.85	193.77	Monitore
3	69.47	26	59785.94	236.50	Organic b
4	74.15	29	54806.18	245.89	Triple-bu
5	68.37	35	73889.99	225.58	Robust lo
6	59.99	23	59761.56	226.74	Sharable

```
In [19]: str(data)
```

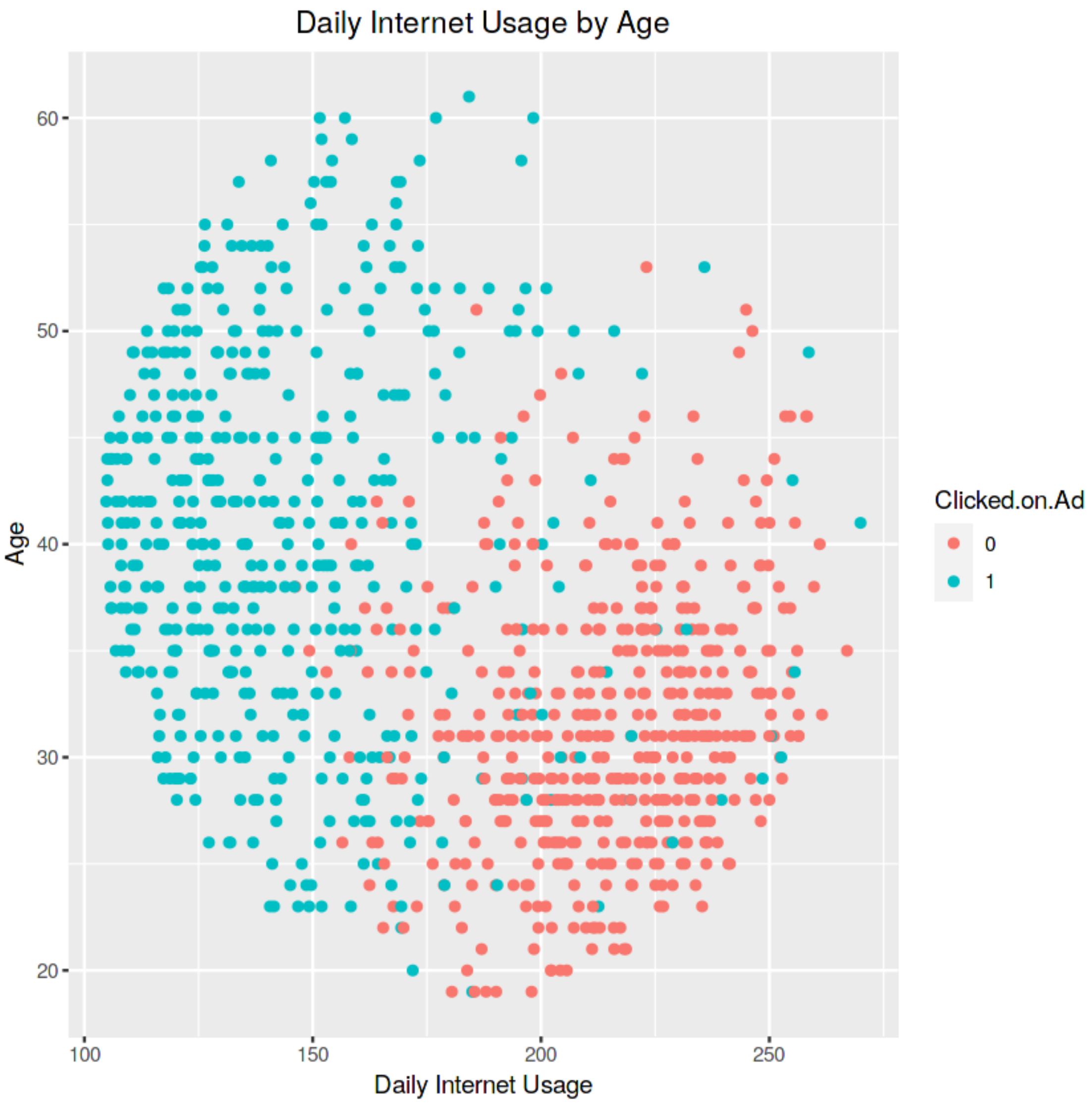
```
'data.frame':	1000 obs. of  10 variables:
 $ Daily.Time.Spent.on.Site: num  69 80.2 69.5 74.2 68.4 ...
 $ Age                     : int  35 31 26 29 35 23 33 48 30 20 ...
 $ Area.Income             : num  61834 68442 59786 54806 73890 ...
 $ Daily.Internet.Usage    : num  256 194 236 246 226 ...
 $ Ad.Topic.Line           : chr  "Cloned 5thgeneration orchestration" "Monitored national star
 $ 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
 $ Clicked.on.Ad           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
```

2.2.1 Q3. (7 Points)

Make a scatter plot for 'Daily.Internet.Usage' against 'Age'. Separate the datapoints by different shapes and/or color based on if the datapoint has clicked on the ad or not (Clicked.on.Ad=0 means no, and Clicked.on.Ad=1 means yes). Based off the general trends in the scatter plot you created, consider a new data point where an individual has a 'Daily.Internet.Usage' less than or equal to 150, and an age of 40. Would this new individual be likely to click the ad or not click the ad?

```
In [20]: # SOLUTION BEGINS HERE
```

```
In [21]: ggplot(data,
                aes(x = Daily.Internet.Usage, y = Age)) +
         geom_point(aes(color = Clicked.on.Ad)) +
         xlab("Daily Internet Usage") +
         ylab("Age") +
         ggtitle("Daily Internet Usage by Age") +
         theme(plot.title = element_text(hjust = 0.5))
```



With daily internet usage <=150 and age =40, the individual would be likely to click on the ad.

```
In [22]: # SOLUTION ENDS HERE
```

2.2.2 Q4. (7 Points)

Create a logistic regression model using the variables ’Daily.Time.Spent.on.Site’, ’Age’, and ’Area.Income’ to predict the variable ’Clicked.on.Ad’. Display the summary output of this logistic regression model.

Now that we have created our logistic regression model, 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 insure 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 greater than or equal to 80% should be treated as a click prediction. Now, using the caret package, create a confusion matrix for the model predictions and actual clicks. Print and/or plot this output.

```
In [23]: # SOLUTION BEGINS HERE
```

```
In [24]: model = glm(Clicked.on.Ad~Daily.Time.Spent.on.Site+Age+Area.Income, data, family = bi
         summary(model)
```

```
Call:
glm(formula = Clicked.on.Ad ~ Daily.Time.Spent.on.Site + Age + 
    Area.Income, family = binomial, data = data)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-2.59567  -0.30198  -0.06632   0.09470   2.84467  

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)    
(Intercept)               1.504e+01  1.443e+00  10.420   <2e-16 ***
Daily.Time.Spent.on.Site -2.048e-01  1.565e-02 -13.085   <2e-16 ***
Age                       1.630e-01  1.785e-02   9.132   <2e-16 ***
Area.Income              -1.173e-04  1.274e-05  -9.206   <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: 1386.29  on 999  degrees of freedom
Residual deviance:  391.98  on 996  degrees of freedom
AIC: 399.98

Number of Fisher Scoring iterations: 7
```

```
In [25]: data$predicted = predict(model, data, type="response")
         data$Clicked.on.Ad.Predicted = as.factor(ifelse(data$predicted >= 0.8, 1, 0))
         str(data)
```

```
'data.frame':        1000 obs. of  12 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 star
 $ 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
 $ Clicked.on.Ad           : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
 $ predicted               : num  0.3462 0.0125 0.1225 0.1354 0.1266 ...
 $ Clicked.on.Ad.Predicted : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
```

```
In [26]: confusionMatrix(data = data$Clicked.on.Ad.Predicted, reference = data$Clicked.on.Ad)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 488  87
         1  12 413

               Accuracy : 0.901
                 95% CI : (0.8808, 0.9188)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.802

 Mcnemar's Test P-Value : 1.028e-13

            Sensitivity : 0.9760
            Specificity : 0.8260
         Pos Pred Value : 0.8487
         Neg Pred Value : 0.9718
             Prevalence : 0.5000
         Detection Rate : 0.4880
   Detection Prevalence : 0.5750
      Balanced Accuracy : 0.9010

       'Positive' Class : 0
```

```
In [27]: # SOLUTION ENDS HERE
```

2.2.3 Q5. (7 Points)

Given the output above, 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. Using the Proc() library, use the roc() function to create and plot a ROC curve of our predictions and true labels.

```
In [28]: # SOLUTION BEGINS HERE
```

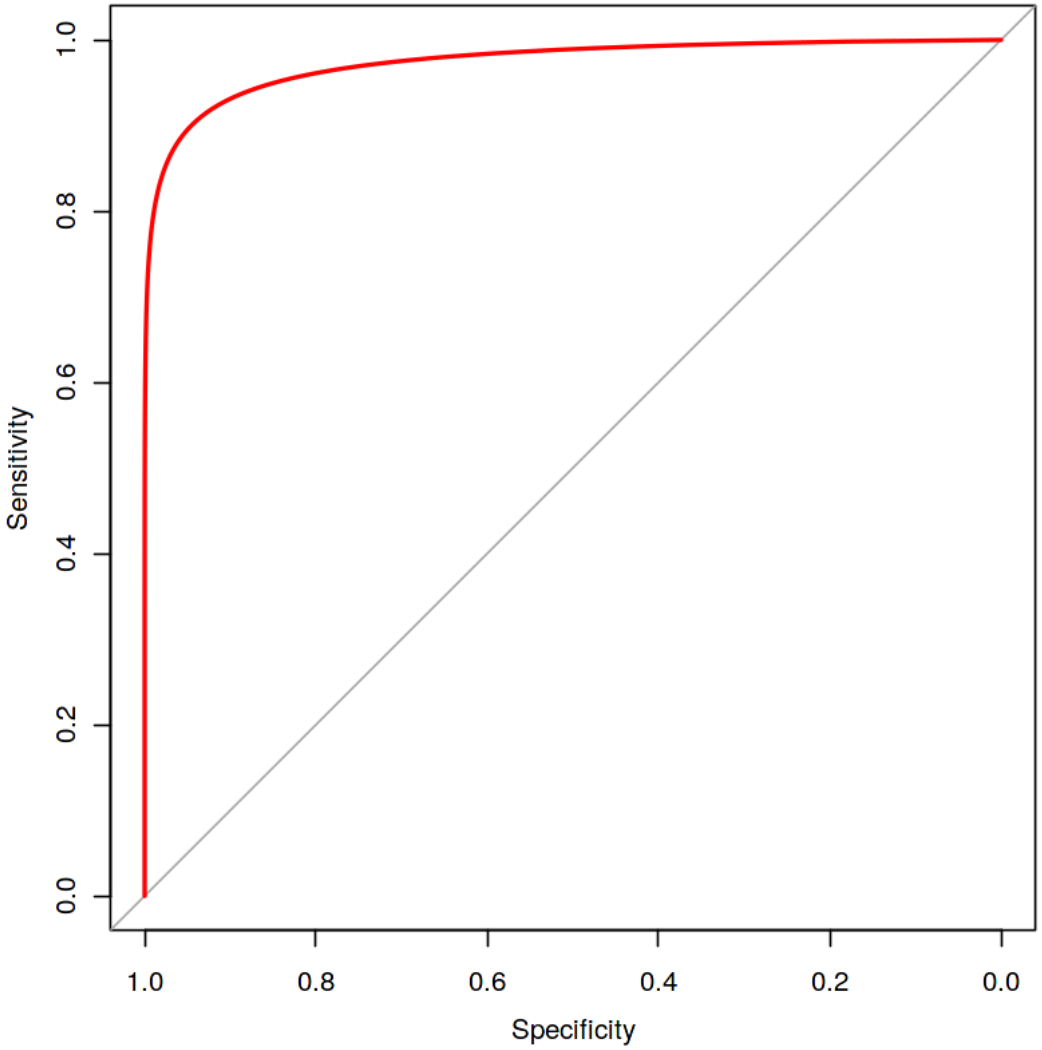
There are 87 occurences of false negatives.

```
In [29]: rc = roc(data$Clicked.on.Ad, data$predicted)
         plot(smooth(rc), col="red")
         rc$auc
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

0.974956



```
In [30]: # SOLUTION ENDS HERE
```

2.3 Instructions for Questions Q6-Q8:

In response to the ongoing pandemic, a local restaurant has implemented social distancing measures, which include the closing its in person dining areas. In order to keep the business going, the restaurant will now rely on drive through lines to handle customer ordering and service. After implementing the new ordering system, management observes that customers arrive at the rate of 62 customers per hour. Under the current system the restaurant has only 5 servers with a total service rate of 70 customers/hour.

2.3.1 Q6. (8 Points)

a. Given the description above, what is the average amount of time customers will wait in line under the current restaurant scenario? (round to nearest minute) (4 Points)

```
In [31]: # SOLUTION BEGINS HERE
```

```
In [32]: arrival = 62
         service = 70
         paste0("Average waiting time in queue: ", round((arrival/(service*(service-arrival)))*
```

'Average waiting time in queue: 7 minutes'

```
In [33]: # SOLUTION ENDS HERE
```

b. Now consider that the restaurant manager from above is wanting to reduce customer waiting times. In order to do this, the manager decides to add another server bringing the total number of servers to 6 and the total service rate to 84 customer/hour. On average, how many customers will be waiting in the queue after the manager introduces this extra server? (round to closest integer) (average arrival remains 62/hour) (4 Points)

```
In [34]: # SOLUTION BEGINS HERE
```

```
In [35]: new_service = 84
         paste0("Average number of customers in the queue: ", round((arrival^2)/(new_service*(r
```

'Average number of customers in the queue: 2 Customers'

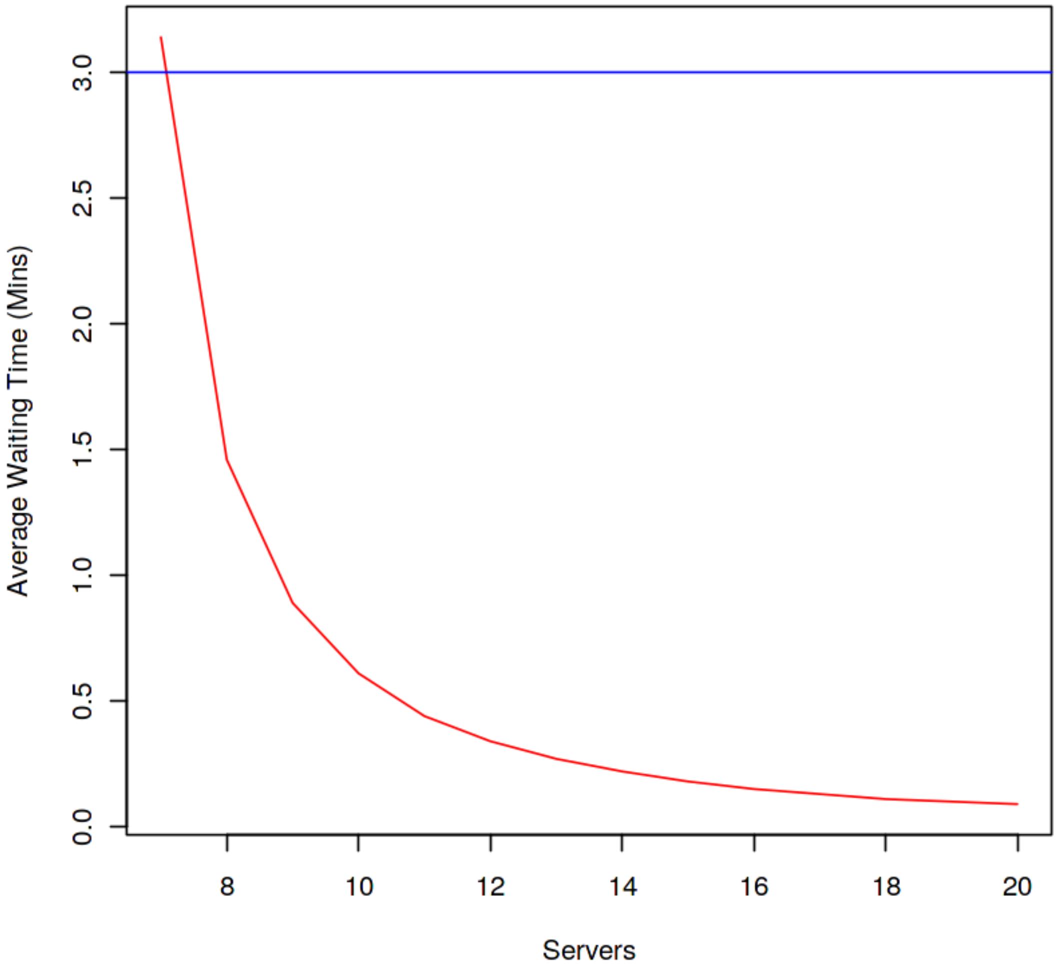
```
In [36]: # SOLUTION ENDS HERE
```

2.3.2 Q7. (7 Points)

Now consider that every server on duty for the restaurant adds a constant 14 customer/hour to the total service rate of the restaurant. For example, with 7 servers on duty the restaurant has a service rate of 98 customers/hour (7 * 14 = 98). Also the rate of arrival for the restaurant has changed to 82 customers/hour. Plot (using R) the average amount of time customers will wait in line with various numbers of servers. The minimum servers that can be on duty is 7 and the maximum is 20. Display the plot as output.

```
In [37]: # SOLUTION BEGINS HERE
```

```
In [38]: data = data.frame(servers = c(seq(7,20)))
         data = data %>% mutate(service_rate = servers*14,
                                avg_waiting_time_mins = round((82/(service_rate*(service_rate-82
         plot(data$servers,
              data$avg_waiting_time_mins,
              type="l",
              col="red",
              xlab="Servers",
              ylab = "Average Waiting Time (Mins)")
         abline(h=3, col="blue")
```



```
In [39]: # SOLUTION ENDS HERE
```

2.3.3 Q8. (8 Points)

a. Based on your plot above, at what number of servers does the average customer wait time drop below 3 minutes? (4 Points)

```
In [40]: # SOLUTION BEGINS HERE
```

```
In [41]: data
```

A data.frame: 14 Œ 3

servers	service_rate	avg_waiting_time_mins
<int>	<dbl>	<dbl>
7	98	3.14
8	112	1.46
9	126	0.89
10	140	0.61
11	154	0.44
12	168	0.34
13	182	0.27
14	196	0.22
15	210	0.18
16	224	0.15
17	238	0.13
18	252	0.11
19	266	0.10
20	280	0.09

Average waiting time drops below 3 minutes with 8 or more servers.

```
In [42]: # SOLUTION ENDS HERE
```

b. Based on your plot above, Describe the behavior of the chart and give some commentary about the relationship between the two variables. For example, is the chart increase or decreasing? What value does average wait time approach if we continue to add more and more servers (to infinity)? (4 Points)

```
In [43]: # SOLUTION BEGINS HERE
```

Clear decreasing exponential relationship between avg. waiting time and number of servers. As number of servers reaches infinity, avg. waiting time reaches close to 0 minutes.

```
In [44]: # SOLUTION ENDS HERE
```