Assignment 1

Shu Wang

Instructions:

You can use R markdown to create a document that consists of your answers to the questions and your R code.

Data Description:

The data for this exercise come from an experiment conducted by the Upworthy, a company famous for pioneering the use of experiments to find 'clickbait' headlines that generate the most user engagement. The dataset contains rows corresponding to impressions (which occur when a user sees the link headline on social media). It also contains data on whether the impression lead to a click. Our job will be to find the headline that resulted in the most clicks.

Problem 1:

1.1 Your name

Please change your name in the header from "Your Name Here" to your name.

1.2 Investigating the dataset

We first need to read the dataset into R. The dataset is called 'data_upworthy_exp.csv'. We will use the function 'fread' from the package 'data.table'. Please run the chunk below to load the library and read the data. Run this chunk:
Note you may need to install these packages using *install.packages('package_name')*.

```
{r warning = FALSE, message = FALSE} install.packages("data.table")
```

```
install.packages("purrr")
```

```r
# Load library 'data.table'
library(data.table)
library(purrr)
# Read the data #
data_upworthy <- fread('data_upworthy_exp.csv')
data_upworthy[, slug := substr(slug, 1, 6)]
data_upworthy[, eyecatcher_id := NULL]
```

```r
library(knitr)
```

To check that the data has been read successfully, you can type the name of the data structure (data_upworthy) into the console. Or look at the environment tab in Rstudio.

Let's verify that the column names in the dataset are correct by using the function 'names':

```r
names(data_upworthy)
```

```
[1] "headline" "slug"     "clicked"
```

Let's print the first 5 lines using the syntax of data.table. Note that '1:5' here means all rows between 1 and 5. Note that since the columns are strings, you may need to scroll right to see all of them.

```r
data_upworthy[1:5]
```

```
                                                                                          hea
1: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, I
2: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, I
3: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, I
4: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, I
5: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, I
     slug clicked
1: let-s-       1
2: let-s-       1
3: let-s-       1
4: let-s-       1
5: let-s-       1
```

1.3 Find the headline corresponding to row 4203:

We can use the data.table syntax to find specific rows and columns. For example, the code below returns whether a user in row 200 clicked and the url (slug) for that user.

```
data_upworthy[200, list(clicked, slug)]
```

Modify the above code to find the headline seen by the impression in row 4203.

```
this_headline <-data_upworthy[4203, list(headline)]
this_headline
```

```
                                                 headline
1: $3 Million Is What It Takes For A State To Legally Kill Someone
```

1.4 Select the set of rows for which clicked equals 1. Use the 'dim' function to see how many rows this is:

We can also reference a row by the value of that row. The code below isolates the rows for which the slug is 'ill-sa'. It then uses the function 'dim' to get the dimensions of the data table. Modify it so that it finds the rows for which clicked is equal to 1.

```
data_upworthy[clicked == '1']
```

```
  1: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice
  2: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice
  3: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice
  4: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice
  5: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice
 ---
 99:                                   I'll Say It: It's Not OK For States To Legally Murder
100:                                   I'll Say It: It's Not OK For States To Legally Murder
101:                                   I'll Say It: It's Not OK For States To Legally Murder
102:                                   I'll Say It: It's Not OK For States To Legally Murder
103:                                   I'll Say It: It's Not OK For States To Legally Murder
       slug clicked
  1: let-s-       1
  2: let-s-       1
  3: let-s-       1
  4: let-s-       1
  5: let-s-       1
 ---
 99: ill-sa       1
100: ill-sa       1
101: ill-sa       1
102: ill-sa       1
103: ill-sa       1
```

```
dim(data_upworthy[clicked == '1'])
```

```
[1] 103   3
```

1.5 How many unique headlines are there?

We'd like to know how many headlines there are. We can use the function 'unique' to learn the number of unique values. For example, the code below returns the number of unique values of the column 'clicked'. Please modify it so that it returns the number of unique headlines in the dataset. How many unique headlines are there?

```
### Unique values of the column headline:
unique(data_upworthy[, headline])
```

```
[1] "Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice
[2] "$3 Million Is What It Takes For A State To Legally Kill Someone"
[3] "The Fact That Sometimes Innocent People Are Executed Is Enough To End The Death Penalty
[4] "Reason #351 To End The Death Penalty: It Costs $3 Million Per Case."
[5] "I Was Already Against The Death Penalty, But Now That I See What It Costs Us All? Ahem.
[6] "I'll Say It: It's Not OK For States To Legally Murder People."
```

```
### Number of unique values of the column headline:
uniqueN(data_upworthy[, headline])
```

```
[1] 6
```

1.6 Create a new column called reason, which takes the value of 1 when the headline is "Reason #351 To End The Death Penalty: It Costs $3 Million Per Case." and 0 otherwise.

To create a new column in a data.table we can use the syntax below. This creates a column called 'ones' that is always equal to 1.

```
data_upworthy[, ones := 1]
```

Your task is to create a new column called reason, that takes the value of 1 when the headline is "Reason #351 To End The Death Penalty: It Costs $3 Million Per Case." and 0 otherwise. To do so, we can use the 'ifelse' function. The ifelse function has three parts: a. The first part determines the condition. b. The part after the first comma determines what happens if a) is true. c. The part after the second comma determines what happens if b) is true. Let's try this! The code below create a column that takes the value 1 if the slug is 'ill-sa' and 0 otherwise.

```
data_upworthy[, slug_legally := ifelse(slug == 'ill-sa', 1, 0)]
# Check that it takes on the value 1 when appropriate:
data_upworthy[slug == 'ill-sa', list(slug_legally, slug)]

# Check that it takes on the value 0 when appropriate:
data_upworthy[slug != 'ill-sa', list(slug_legally, slug)]
```

Create a new column called reason, which takes the value of 1 when the headline is "Reason #351 To End The Death Penalty: It Costs $3 Million Per Case." and 0 otherwise.

```
data_upworthy[, reason := ifelse(headline == "Reason #351 To End The Death Penalty:
                                 It Costs $3 Million Per Case.", 1, 0)]
```

1.7 Calculate the share of impressions that see each headline.

In order to do this, we will use the aggregation features of data.table. They work like SQL, if you've used it before. In a data.table, we can group by variables (the grouping is specified after the second comma) and apply functions to each group (after the first comma). The code below counts the number of impressions by whether the slug_legally variable is equal to 1. Note that '.N' is a special function in data.table that counts the number of rows.

```
# list(num_students = .N) creates a variable when
# the number of rows in each group (slug_legally = 1, slug_legally = 0)
agg_data <- data_upworthy[, list(num_impression = .N), list(slug_legally)]
```

```
agg_data
```

Note, we now have two datasets, the original dataset 'data_upworthy' and the aggregate data 'agg_data'. Let's continue working with agg_data. To calculate the total number of impressions we can use the sum function.

```
tot_impressions <- sum(agg_data[, num_impression])
agg_data[, share_impressions := num_impression/tot_impressions]
```

Below, repeat the above steps to calculate the share of impressions by headline.

```
# Your code here:
library(kableExtra)
library(knitr)
agg_data2 <- data_upworthy[, list(num_impression = .N), list(headline)]
```

```
tot_impressions2 <- sum(agg_data2[, num_impression])
agg_data2[, share_impressions := num_impression/tot_impressions2]
agg_data2
```

```
1:    Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choic
2:                                   $3 Million Is What It Takes For A State To Legally Kil
3: The Fact That Sometimes Innocent People Are Executed Is Enough To End The Death Penalty. 
4:                               Reason #351 To End The Death Penalty: It Costs $3 Million 
5:            I Was Already Against The Death Penalty, But Now That I See What It Costs Us A
6:                                     I'll Say It: It's Not OK For States To Legally Murde
   num_impression share_impressions
1:           3118         0.1711588
2:           3017         0.1656145
3:           2974         0.1632541
4:           3050         0.1674260
5:           3061         0.1680299
6:           2997         0.1645167
```

1.8 Calculate the click rate by headline.

If we're Upworthy, we'd like to know which headline results in the most clicks so that we can show that headline to everyone in the future. To calculate the mean, we use the 'mean' function.

For example, the code below calculates the mean of ‘clicked’ for the entire data sample. Modify it to calculate the mean by headline. Which headline has the highest conversion rate?

```
# Mean and count of impressions by 'headline'. Note we can generate multiple aggregate var
data_click_rate <- data_upworthy[, list(click_rate = mean(clicked), num_impressions = .N),
                                  by = list(headline)]

data_click_rate
```

```
1:    Let’s See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choic
2:                                  $3 Million Is What It Takes For A State To Legally Kil
3: The Fact That Sometimes Innocent People Are Executed Is Enough To End The Death Penalty. 
4:                              Reason #351 To End The Death Penalty: It Costs $3 Million 
5:            I Was Already Against The Death Penalty, But Now That I See What It Costs Us A
6:                                     I'll Say It: It's Not OK For States To Legally Murde
    click_rate num_impressions
1: 0.002565747            3118
2: 0.006297647            3017
3: 0.008742434            2974
4: 0.003278689            3050
5: 0.006533812            3061
6: 0.006673340            2997
```

```
# Headline"The Fact That Sometimes Innocent People Are Executed Is Enough To End The Death
# Penalty.But this?" has the highest conversion rate.
```

1.9 Plot the click rate by headline.

To plot the data, we use the package ‘ggplot2’. We can load this package by using the command library as below. Remember, you must tell R to load specific packages such as ggplot2 and data.table.

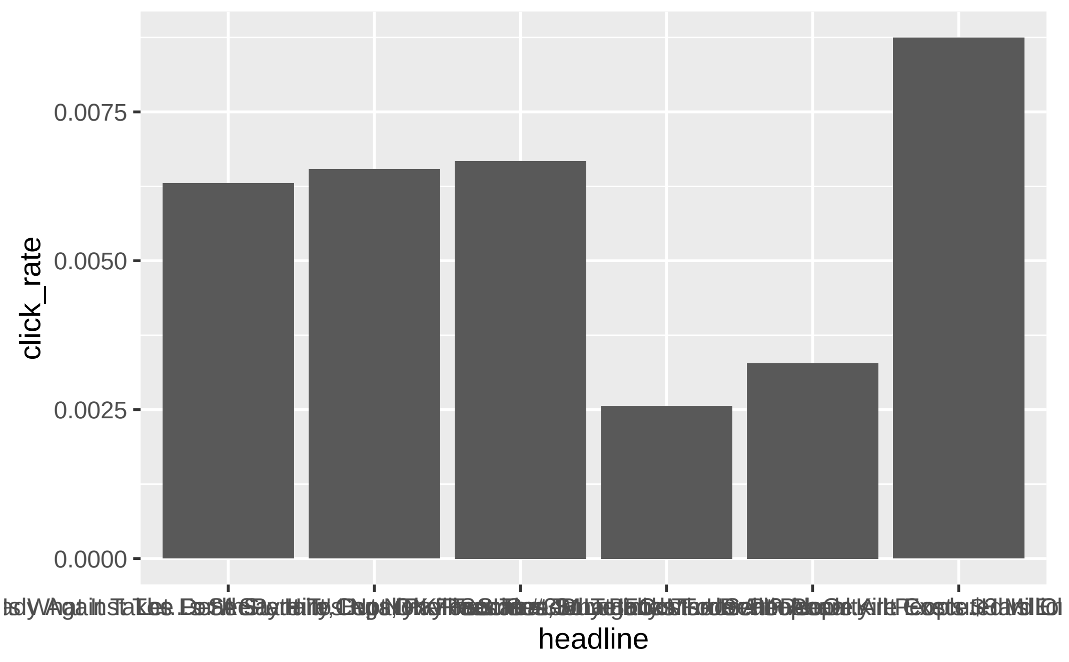
```
library(ggplot2)
```

Now, let’s create a bar plot. The ggplot function takes in a dataset (the first part of the function), and the values you are going to plot (x is the variable which will be on the x axis, y will be on the y axis). We then add the plot type: ‘geom_bar(stat = ’identity’)’ to tell it to make it a bar plot.

```r
this_plot <- ggplot(agg_data, aes(x = slug_legally, y = share_impressions)) +
  geom_bar(stat = 'identity') + coord_flip()
this_plot
```

Modify the above code to plot the click rate by headline:

```r
plot2 <- ggplot(data_click_rate, aes(x = headline, y = click_rate)) +
  geom_bar(stat = 'identity')
plot2
```

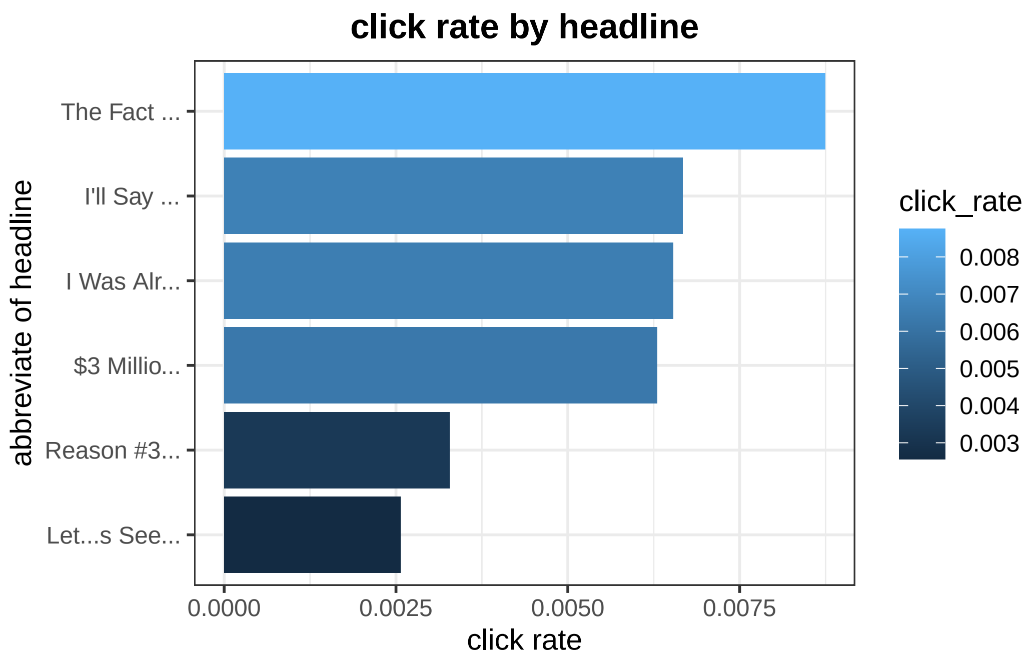


1.10 BONUS: Make a pretty plot by labeling the axes and tweaking the theme.

```r
click_rate2 <- transform(data_click_rate, headline = reorder(headline, click_rate))
```

```r
suppressWarnings(require(RColorBrewer))
myColors <- brewer.pal(6, "GnBu")
```

```r
plot3 <- ggplot(click_rate2, aes(x = click_rate, y = headline,fill = click_rate)) +
  geom_bar(stat = 'identity') +
  scale_y_discrete(label = function(x) stringr::str_trunc(x, 12)) +
    theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)) + theme_bw() +
  scale_colour_manual(values=myColors) +
  xlab("click rate") + ylab("abbreviate of headline")+ ggtitle("click rate by headline") +
  theme(plot.title=element_text(face="bold",hjust = 0.5))
plot3
```



Problem 2

Question a

```
# We are expected to generate 600 revenue from user2 if the customer didn't see the Q&A.
# We are expected to generate 100 revenue from user2 if the customer saw the Q&A.
# In other words,we would expect a large decrease in revenue if he/she saw the Q&A.
```

Question b

```
Name <-c("user1","user2","user3","user4","user5","user6","user7","user8","user9","user10")

If_user_saw_QA <-c(1100,100,500,100,1600,2000,1200,700,1100,140)

If_user_didnot_saw_QA <-c(1100,600,500,900,700,2000,1200,700,300,140)
```

```
df<-data.frame(Name,If_user_saw_QA,If_user_didnot_saw_QA)
df$treatment_effect = If_user_saw_QA - If_user_didnot_saw_QA
```

```
df
```

```
     Name If_user_saw_QA If_user_didnot_saw_QA treatment_effect
1   user1           1100                  1100                0
2   user2            100                   600             -500
3   user3            500                   500                0
4   user4            100                   900             -800
5   user5           1600                   700              900
6   user6           2000                  2000                0
7   user7           1200                  1200                0
8   user8            700                   700                0
9   user9           1100                   300              800
10 user10            140                   140                0
```

Question c

```
# Among 10 users,6 of them shows indifference between showing Q&A and not showing Q&A.
# 2 users show decrease in revenue and 2 users show increase
# in revenue when we give Q&A compared to without Q&A.

#The reason maybe that different users have different attitudes towards Q&A survey.
#Some may find it attracting while others may be disturbed

# by these Q&A and some others would just ignore it.
```

Question d

```
ATE = mean(If_user_saw_QA) - mean(If_user_didnot_saw_QA)
ATE
```

```
[1] 40
```

Question e

```
estimate_QA = 1/5*(1100+ 500+1600+1200+1100)
estimate_noQA = 1/5 *(600+900+2000+700 + 140)
estimate_ATE = estimate_QA - estimate_noQA
estimate_ATE
```

```
[1] 232
```

Question f

```
quantile(df$treatment_effect,probs = 0.3)
```

```
30%
  0
```

How long did this assignment take you to do (hours)? How hard was it (easy, reasonable, hard, too hard)?

2-3 hours, reasonable.