Assignment 1

Yuxuan Mei

Instructions:

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

About R Markdown Notebooks

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code. It can also be used to generate PDFs of your results. Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*. The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

R markdown instruction

In R markdown files, we put code into 'chunks', like the one below. You can click the 'run' buttom on the code chunk below to run it. As you can see, the code chunk returns the output of the R code.

```
variable_1 <- 1
variable_2 <- 2
variable_1 + variable_2
```

```
## [1] 3
```

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 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:

```
# Load library 'data.table'
library(data.table)
library(purrr)
```

```
##
## Attaching package: 'purrr'
```

```
## The following object is masked from 'package:data.table':
##
##     transpose
```

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

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':

```
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.

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

```
##                                                                                 headline
## 1: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Right?
## 2: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Right?
## 3: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Right?
## 4: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Right?
## 5: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Right?
##      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)]
```

```
##    clicked   slug
## 1:       0 let-s-
```

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

```
### Your code here:
data_upworthy[4203, headline]
```

```
## [1] "$3 Million Is What It Takes For A State To Legally Kill Someone"
```

1.4 Select a subset of the data by value. 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]
```

```
##                                                                                                  headline
##   1: Let’s See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Right?
##   2: Let’s See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Right?
##   3: Let’s See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Right?
##   4: Let’s See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Right?
##   5: Let’s See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Right?
##  ---
##  99:                                    I'll Say It: It's Not OK For States To Legally Murder People.
## 100:                                    I'll Say It: It's Not OK For States To Legally Murder People.
## 101:                                    I'll Say It: It's Not OK For States To Legally Murder People.
## 102:                                    I'll Say It: It's Not OK For States To Legally Murder People.
## 103:                                    I'll Say It: It's Not OK For States To Legally Murder People.
##        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 clicked:
unique(data_upworthy[, headline])
```

```
## [1] "Let’s See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Right?"
## [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. But Thi
## [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 clicked:
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[, reason := ifelse(headline == 'Reason #351 To End The Death
                                  Penalty: It Costs $3 Million Per Case.', 1, 0)]
# Check that it takes on the value 1 when appropriate:
data_upworthy[headline == 'Reason #351 To End The Death Penalty:
              It Costs $3 Million Per Case.', list(headline, reason)]
```

```
## Empty data.table (0 rows and 2 cols): headline,reason
```

```
# Check that it takes on the value 0 when appropriate:
data_upworthy[headline != 'Reason #351 To End The Death
              Penalty: It Costs $3 Million Per Case.', list(headline, reason)]
```

```
##                                                                                                 headli
##     1: Let’s See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Righ
##     2: Let’s See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Righ
##     3: Let’s See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Righ
##     4: Let’s See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Righ
##     5: Let’s See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choice, Righ
##    ---
## 18213:                                  I'll Say It: It's Not OK For States To Legally Murder People
## 18214:                                  I'll Say It: It's Not OK For States To Legally Murder People
## 18215:                                  I'll Say It: It's Not OK For States To Legally Murder People
## 18216:                                  I'll Say It: It's Not OK For States To Legally Murder People
## 18217:                                  I'll Say It: It's Not OK For States To Legally Murder People
##        reason
##     1:      0
##     2:      0
##     3:      0
##     4:      0
##     5:      0
##    ---
## 18213:      0
## 18214:      0
## 18215:      0
## 18216:      0
## 18217:      0
```

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.

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 <- data_upworthy[, list(num_impression = .N), list(headline)]
```

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:
agg_data <- data_upworthy[, list(num_impression = .N), list(headline)]
tot_impressions <- sum(agg_data[, num_impression])
agg_data[, share_impressions := num_impression/tot_impressions]
```

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 over the entire sample
#data_upworthy[, mean(clicked), list(headline)]

# Mean and count of impressions by 'slug_legally'. Note we can generate multiple aggregate variables at
#data_click_rate <- data_upworthy[, list(click_rate = mean(clicked), num_impressions = .N), by = list(h
agg_data <- data_upworthy[, list(click_rate = mean(clicked),
                                 share_impressions = .N/tot_impressions), list(headline)]
# shorten the headlines
agg_data[, headline:= substr(headline, 0, 12)]
```

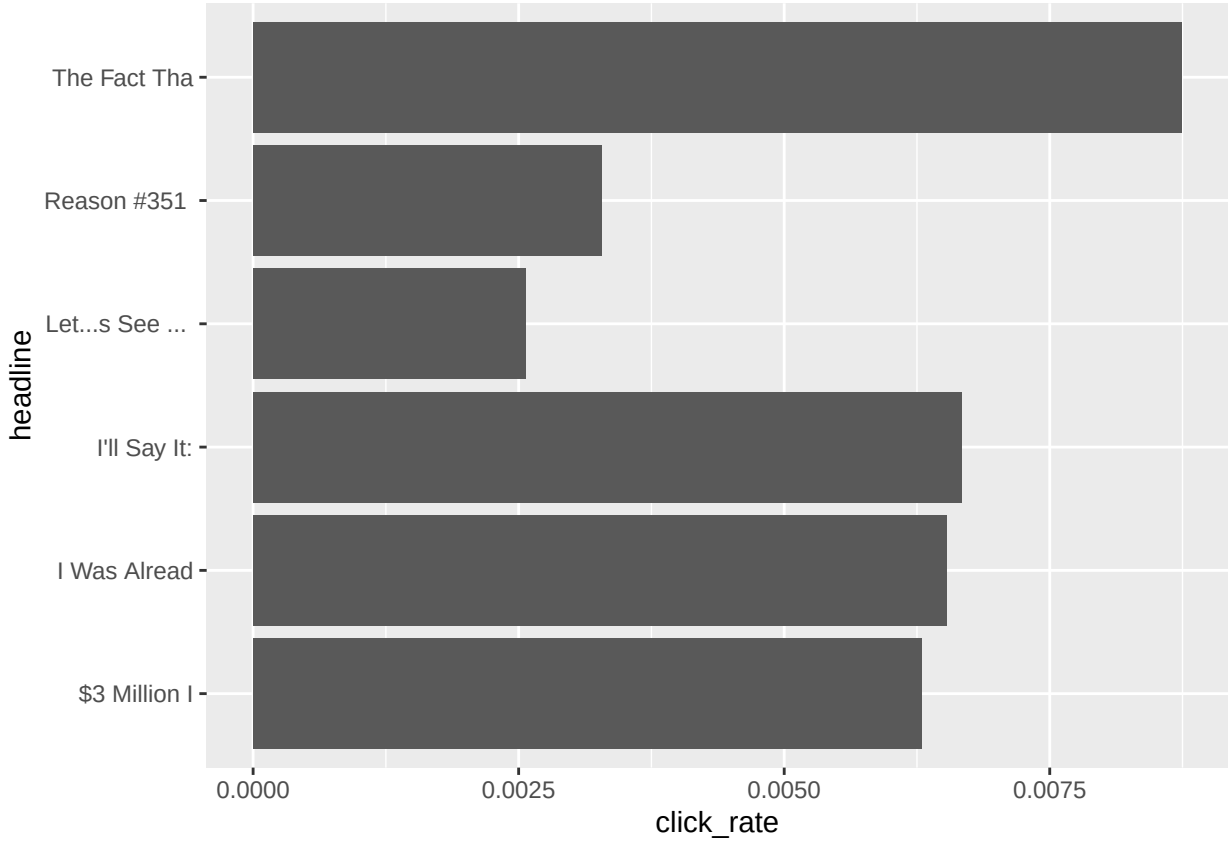
1.9 Plot the click rate by headline.

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)
```

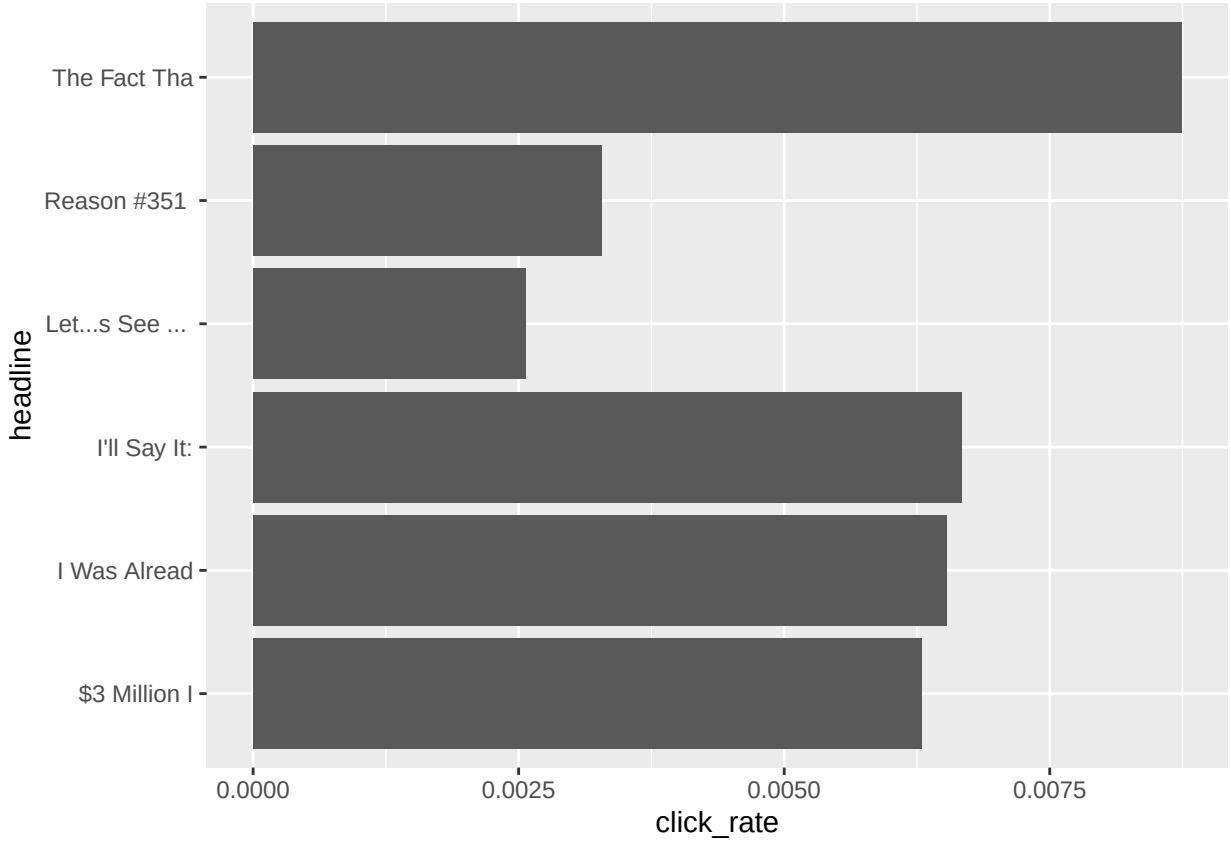
Now, let's create a bar plot. The ggplot function takes in a dataset (the first part of the function), and the values you are going to plot (x is the variable which will be on the x axis, y will be on the y axis). We then add the plot type: 'geom_bar(stat = 'identity')' to tell it to make it a bar plot.

```
this_plot <- ggplot(agg_data, aes(x = headline, y = click_rate)) +
             geom_bar(stat = 'identity') + coord_flip()
this_plot
```



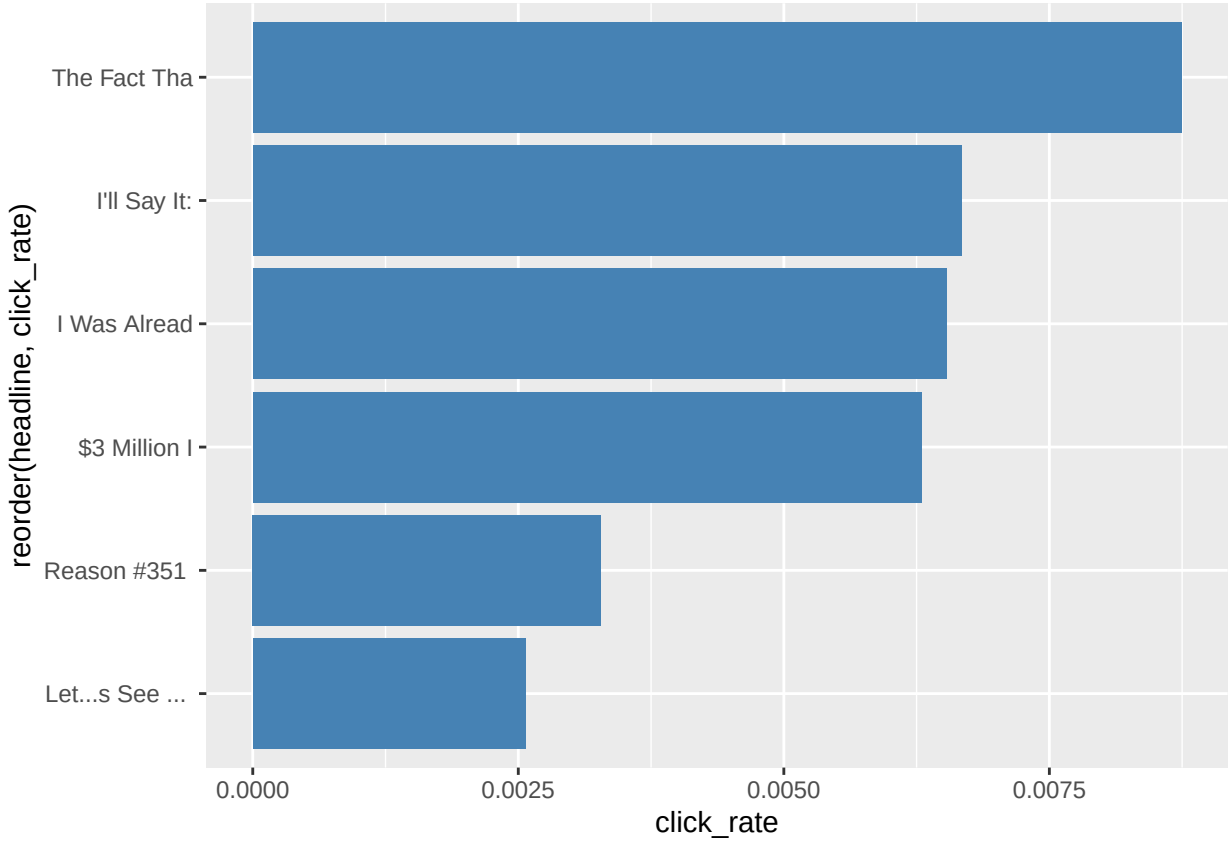
Modify the above code to plot the click rate by headline:

```
this_plot <- ggplot(agg_data, aes(x = headline, y = click_rate)) +
               geom_bar(stat = 'identity') + coord_flip()
this_plot
```



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

```
this_plot <- ggplot(agg_data, aes(x = reorder(headline, click_rate), y = click_rate)) +
              geom_bar(stat = 'identity', fill = 'steelblue') + coord_flip()
this_plot
```



Problem 2

Question a

If user 2 saw Q&A, the potential revenue would be 100, whereas the potential revenue would be 600 if the user did not see the Q&A. If the treatment is seeing Q&A, this suggests that the treatment effect for this particular user is -500.

Question b

```
data <- fread('data.csv')
# get individual true treatmetn effect
data[, TE := Treatment - Control]

data
```

```
##         user Treatment Control   TE
##  1:  User 1      1100    1100    0
##  2:  User 2       100     600 -500
##  3:  User 3       500     500    0
##  4:  User 4       900     900    0
##  5:  User 5      1600     700  900
##  6:  User 6      2000    2000    0
##  7:  User 7      1200    1200    0
##  8:  User 8       700     700    0
##  9:  User 9      1100    1000  100
## 10: User 10       140     140    0
```

Question c

Out of the 10 users, 7 of them have a treatment effect of 0, and the average treatment effect is 50. This may suggest that whether a user saw the Q&A or not may not affect his / her decision on the purchase a lot. Most users may already conducted some research on their own, and knew a lot about the product. Therefore, the Q&A might not be a crucial factor in their purchase decisions. For a smaller group of users who did not have enough knowledge on the product, their decisions could be affected more by the Q&A.

Question d

```
data[, mean(TE)]
```

```
## [1] 50
```

The true average treatment effect is 50.

Question e

```
data[, user:= .I]
treatment <- data[user%%2 == 1, Treatment]
control <- data[user%%2 == 0, Control]

ATE <- sum(treatment)/5 - sum(control)/5
ATE
```

```
## [1] 232
```

The estimate ATE is 232.

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

It took me about 3 hour. The concept is relatively easy, while the coding part is a bit rusty, and took longer than expected.