Assignment 1

Prateek Naharia

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')*.

```
# 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':

```
data_upworthy
```

```
    1: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choi
    2: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choi
    3: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choi
    4: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choi
    5: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choi
   ---
18213:                                  I'll Say It: It's Not OK For States To Legally Murd
18214:                                  I'll Say It: It's Not OK For States To Legally Murd
18215:                                  I'll Say It: It's Not OK For States To Legally Murd
18216:                                  I'll Say It: It's Not OK For States To Legally Murd
18217:                                  I'll Say It: It's Not OK For States To Legally Murd
         slug clicked
    1: let-s-       1
    2: let-s-       1
    3: let-s-       1
```

```
    4: let-s-       1
    5: let-s-       1
   ---
18213: ill-sa       0
18214: ill-sa       0
18215: ill-sa       0
18216: ill-sa       0
18217: ill-sa       0
```

```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.

```r
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.

```
data_upworthy[4203, list(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[slug == 'ill-sa']
```

```
                                                            headline   slug
   1: I'll Say It: It's Not OK For States To Legally Murder People. ill-sa
   2: I'll Say It: It's Not OK For States To Legally Murder People. ill-sa
   3: I'll Say It: It's Not OK For States To Legally Murder People. ill-sa
   4: I'll Say It: It's Not OK For States To Legally Murder People. ill-sa
   5: I'll Say It: It's Not OK For States To Legally Murder People. ill-sa
  ---
2993: I'll Say It: It's Not OK For States To Legally Murder People. ill-sa
2994: I'll Say It: It's Not OK For States To Legally Murder People. ill-sa
2995: I'll Say It: It's Not OK For States To Legally Murder People. ill-sa
2996: I'll Say It: It's Not OK For States To Legally Murder People. ill-sa
2997: I'll Say It: It's Not OK For States To Legally Murder People. ill-sa
      clicked
   1:       1
   2:       1
   3:       1
   4:       1
   5:       1
  ---
2993:       0
```

```
2994:       0
2995:       0
2996:       0
2997:       0
```

```r
dim(data_upworthy[slug == 'ill-sa'])
```

```
[1] 2997    3
```

```r
data_upworthy[clicked == 1]
```

```
                                                                                                  I
  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
```

```r
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[, clicked])
```

```
[1] 1 0
```

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

```
[1] 2
```

```
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."
```

```
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)]
```

```
      slug_legally   slug
   1:            1 ill-sa
   2:            1 ill-sa
   3:            1 ill-sa
   4:            1 ill-sa
   5:            1 ill-sa
  ---
2993:            1 ill-sa
2994:            1 ill-sa
2995:            1 ill-sa
2996:            1 ill-sa
2997:            1 ill-sa
```

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

```
       slug_legally   slug
    1:            0 let-s-
    2:            0 let-s-
    3:            0 let-s-
    4:            0 let-s-
    5:            0 let-s-
   ---
15216:            0 i-was-
15217:            0 i-was-
15218:            0 i-was-
```

```
15219:             0 i-was-
15220:             0 i-was-
```

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 := 1]
```

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

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

```
      reason
   1:      0
   2:      0
   3:      0
   4:      0
   5:      0
  ---
3046:      0
3047:      0
3048:      0
3049:      0
3050:      0
                                                                    headline
   1: Reason #351 To End The Death Penalty: It Costs $3 Million Per Case.
   2: Reason #351 To End The Death Penalty: It Costs $3 Million Per Case.
   3: Reason #351 To End The Death Penalty: It Costs $3 Million Per Case.
   4: Reason #351 To End The Death Penalty: It Costs $3 Million Per Case.
   5: Reason #351 To End The Death Penalty: It Costs $3 Million Per Case.
  ---
3046: Reason #351 To End The Death Penalty: It Costs $3 Million Per Case.
3047: Reason #351 To End The Death Penalty: It Costs $3 Million Per Case.
3048: Reason #351 To End The Death Penalty: It Costs $3 Million Per Case.
3049: Reason #351 To End The Death Penalty: It Costs $3 Million Per Case.
```

```
3050: Reason #351 To End The Death Penalty: It Costs $3 Million Per Case.
```

```
data_upworthy[headline !=
                'Reason #351 To End The Death Penalty: It Costs $3 Million Per Case.',
              list(reason, headline)]
```

```
       reason
    1:      0
    2:      0
    3:      0
    4:      0
    5:      0
   ---
15163:      0
15164:      0
15165:      0
15166:      0
15167:      0

    1: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choi
    2: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choi
    3: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choi
    4: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choi
    5: Let's See … Hire Cops, Pay Teachers, Buy Books For Schools. Or Kill People. Hard Choi
   ---
15163:                                  I'll Say It: It's Not OK For States To Legally Murd
15164:                                  I'll Say It: It's Not OK For States To Legally Murd
15165:                                  I'll Say It: It's Not OK For States To Legally Murd
15166:                                  I'll Say It: It's Not OK For States To Legally Murd
15167:                                  I'll Say It: It's Not OK For States To Legally Murd
```

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
```

```
   slug_legally num_impression
1:            0          15220
2:            1           2997
```

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]
tot_impressions
```

```
[1] 18217
```

```
agg_data
```

```
   slug_legally num_impression share_impressions
1:            0          15220         0.8354833
2:            1           2997         0.1645167
```

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

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

```
tot_impressions
```

```
[1] 18217
```

```
agg_data
```

```
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
```

```
names(agg_data)
```

```
[1] "headline"          "num_impression"    "share_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)]
```

```
[1] 0.005654059
```

```
# Mean and count of impressions by 'slug_legally'. Note we can generate multiple aggregate
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
```

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.

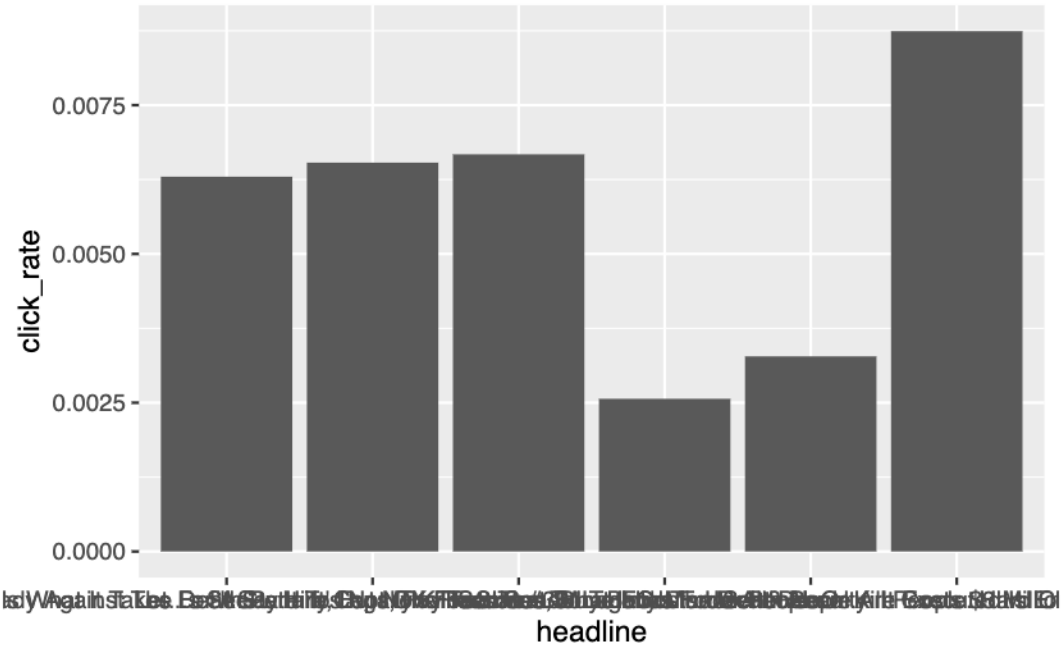
```
library(ggplot2)
```

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.

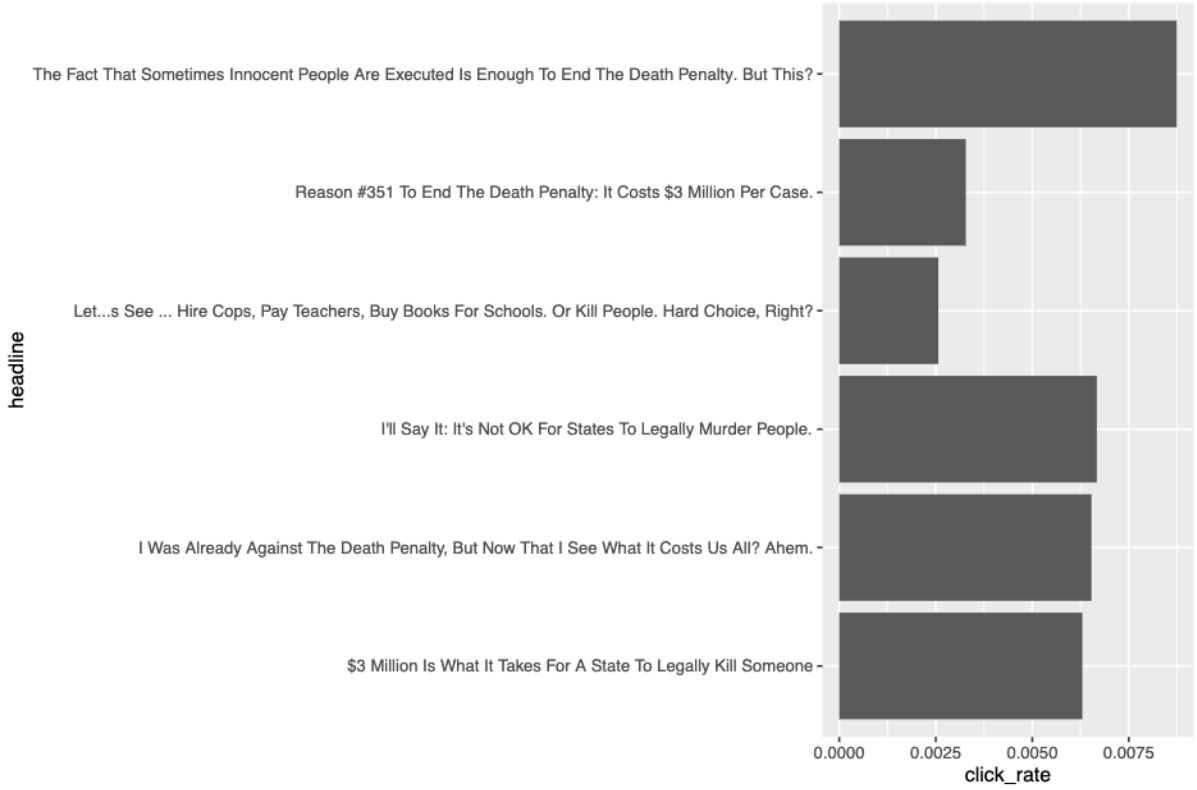
```
#this_plot <- ggplot(agg_data, aes(x = slug_legally, y = share_impressions)) + geom_bar(st
#this_plot
```

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

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

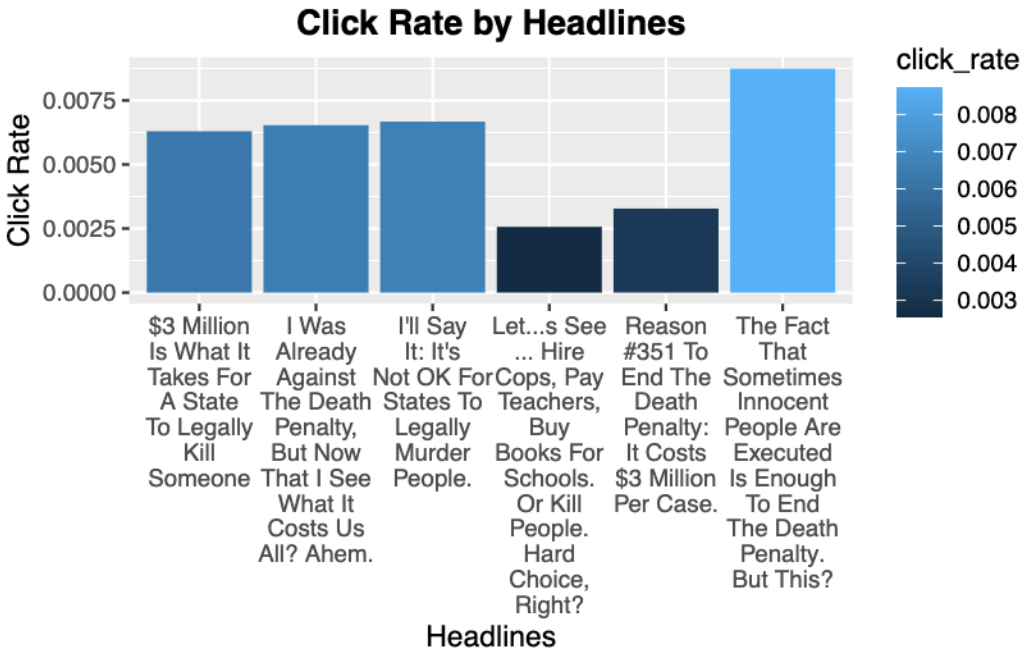



```r
this_plott <- ggplot(data_click_rate,
                     aes(x = headline,
                         y = click_rate)) +
  geom_bar(stat = 'identity') +
  coord_flip()
this_plott
```



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

```
library(RColorBrewer)
library(stringr)
#| fig.width = 11, fig.height = 10
myColors <- brewer.pal(6, "Set1")
names(myColors) <- data_click_rate$headline
data_click_rate$headline_update <- str_wrap(data_click_rate$headline, 10)
plot1 <- ggplot(data_click_rate, aes(x=headline_update,
                                     y=click_rate,
                                     fill=click_rate)) +
  geom_bar(stat="identity")+
  scale_colour_manual(values=myColors) +
  ggtitle("Click Rate by Headlines") +
  theme(plot.title=element_text(face="bold",hjust = 0.5))+
  xlab("Headlines") +
  ylab("Click Rate")
plot1
```



Problem 2

Question a

Observing - Treatment as (Seeing Q&A) & Control as (Not Seeing Q&A)

If user 2 saw Q&A, the revenue is 100 usd & if user 2 didn't see Q&A the revenue generated is 600, on the furniture page.

Question b

User	Individual Treatment Effect
User 1	0
User 2	-500
User 3	0
User 4	-800
User 5	900
User 6	0
User 7	0
User 8	0
User 9	800
User 10	0

Question c

This distribution of treatment effects are as positive treatment & negative treatment & no treatment effect. Here in, it can be seen that the Q&A section allows non-frequent users to gain information about the product/furniture listed at wayfair's website which are unknown to them, considering customer behavior. This model prevents consumers from purchasing products worst suitable on the other side gives products which are most suitable. Also leading to an increase in their likelihood of purchasing furniture. Other users may not have had questions that were answered by the Q&A, or may have already had enough information to make a purchase, leading to no change in their likelihood of purchasing furniture.

Question d

True Average Treatment Effect 854 - 814 = 40.

Question e

	If user saw Q&A	If user did not see Q&A
User 1	1100	
User 2		600
User 3	500	
User 4		900
User 5	1600	
User 6		2000
User 7	1200	
User 8		700
User 9	1100	
User 10		140
Sum	5500	4340
Avg	1100	868
ATE	232	

((Sum :Saw Q&A) - (Sum :Not see Q&A)) / 5

((1100 + 500 + 1600 + 1200 + 1100) - (600+900+2000+700+140)) / 5 = 232

The **estimate** of the ATE is 232 for the population with odd number assigned to treatment group & for the population with even number assigned to control group.

Question f

Your answer here

```
Tf <- c(0, -500, 0, -800, 900, 0, 0, 0, 800, 0)
quantile(Tf, 0.3)
```

```
30%
  0
```

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

The assignment took about 4 hours to not only write answers but to research for better graph options and syntax's. It was reasonable enough to get hands on and for being r friendly.