Assignment 1

Chiebuka Onwuzurike

Instructions:

You can use it to create a document that consists of your answers to the questions and your R code. Please change your name in the above text from “Your Name Here” to your name.

About R Markdown Notebooks

This is an R Markdown (http://rmarkdown.rstudio.com) 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
```

Problem 1:

1.1

Our first task will be to read the dataset called ‘class_data.csv’. To do so, 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 #
class_data <- fread('student_list_msba2021spring.csv')
```

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

Let's print the first few lines using the 'head' command.

```
head(class_data)
```

```
##              name  user_id           email treatment_group section
## 1: Patel, Manushi manuship manuship@bu.edu               2      B1
## 2:      Yu, Lequn    lequn    lequn@bu.edu               4      B1
## 3:   Sun, Zhiyuan jacobszy jacobszy@bu.edu               1      B1
## 4:         Li, Bo boli0315 boli0315@bu.edu               0      B1
## 5:  Leng, Linghan   lenglh   lenglh@bu.edu               3      B1
## 6:       Guo, Kai kaiguo96 kaiguo96@bu.edu               0      B1
```

1.2 Reading your name

Our next goal is to find the row associated with your name. Each row has a number, and we can isolate it by that number. We can also specify which columns show up. The code below gets the first row, for example. Modify it so that your name shows up (use the columns for name and user_id).

```
class_data[30, list(name, user_id)]
```

```
##              name  user_id
## 1: Veytsman, Ryan veytsman
```

We can also reference a row by the value of that row. The code below isolates the rows for which the college is QST. Modify it so that it finds the row whose first_name is your name.

```
head(class_data[section == 'A1'])
```

```
##             name  user_id           email treatment_group section
## 1:    Zhu, Shuyi shuyizhu shuyizhu@bu.edu               0      A1
## 2:   Wang, Jiaqi jwang311 jwang311@bu.edu               3      A1
## 3:   Zhang, Ying   yingyz   yingyz@bu.edu               2      A1
## 4:   Li, Mingwei mingweil mingweil@bu.edu               1      A1
## 5: Chen, Chuning chuningc chuningc@bu.edu               1      A1
## 6:  Li, Jiazheng jli12320 jli12320@bu.edu               0      A1
```

1.3

Find the treatment group associated with your name.

```
class_data[name == 'Onwuzurike, Chiebuka',list(name, treatment_group)]
```

```
##                   name treatment_group
## 1: Onwuzurike, Chiebuka               3
```

1.4

Create a new column called ‘section_code’ that takes the value of 1 if the student is in section ‘A1’ and 0 otherwise. The purpose is to use this column to calculate what share of students is in each section.

```
### How to create a new column:
# class_data[, test_column1 := 1]
# class_data$test_column2 <- 2
# Verify that these two columns were created
# class_data[, list(test_column1, test_column2)]
# class_data[, section_code := NULL]
```

To create a conditional column 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 your user_id starts with the letter w and 0 otherwise. Note, we use the ‘substr’ function to get the first letter.

```
class_data[, starts_with_w := ifelse(substr(user_id, 1, 1)  == 'w', 1, 0)]
# Check that it works:
class_data[1:6, list(starts_with_w, user_id)]
```

```
##    starts_with_w  user_id
## 1:             0 manuship
## 2:             0    lequn
## 3:             0 jacobszy
## 4:             0 boli0315
## 5:             0   lenglh
## 6:             0 kaiguo96
```

Create a new column called ‘section_code’ that takes the value of 1 if the student is in section ‘A1’ and 0 otherwise.

```
class_data[, section_code := ifelse(section == "A1",1,0)]
head(class_data)
```

```
##              name  user_id           email treatment_group section
## 1: Patel, Manushi manuship manuship@bu.edu               2      B1
## 2:      Yu, Lequn    lequn    lequn@bu.edu               4      B1
## 3:   Sun, Zhiyuan jacobszy jacobszy@bu.edu               1      B1
## 4:         Li, Bo boli0315 boli0315@bu.edu               0      B1
## 5:  Leng, Linghan   lenglh   lenglh@bu.edu               3      B1
## 6:       Guo, Kai kaiguo96 kaiguo96@bu.edu               0      B1
##    starts_with_w section_code
## 1:             0            0
## 2:             0            0
## 3:             0            0
## 4:             0            0
## 5:             0            0
## 6:             0            0
```

1.4

Calculate the share of students in each section and treatment group In order to do this, we will use aggregation features of data.table. This is like SQL, if you've used it before. In a data.table, we can group by variables (after the second comma) and count them. The code below counts the students by college.

```
# list(num_students = .N) creates a variable called 'num_students' that counts students by colle
ge
agg_data <- class_data[, list(num_students = .N), list(section)]
agg_data
```

```
##    section num_students
## 1:      B1           34
## 2:      A1           40
```

Note, we now have two datasets, the original dataset 'class_data' and the aggregate data 'agg_data'. We can then calculate the share of students by major by dividing by the total number of students. We first calculate the total number of students across major by using the function 'sum' to sum the values of the column 'num_students'. We then create a column called 'share_students' by dividing num students by the total number of students.

```
tot_num_students <- sum(agg_data[, num_students])
agg_data[, share_students := num_students/tot_num_students]
agg_data
```

```
##    section num_students share_students
## 1:      B1           34      0.4594595
## 2:      A1           40      0.5405405
```

Below, repeat the above steps to calculate the share of students by each value of the column 'treatment_group'.

```
# Your code here:
treatment_agg_data <- class_data[, list(num_students = .N), list(treatment_group)]
treatment_agg_data[, share_students := num_students/tot_num_students]
treatment_agg_data
```

```
##    treatment_group num_students share_students
## 1:               2           16      0.2162162
## 2:               4            9      0.1216216
## 3:               1           19      0.2567568
## 4:               0           18      0.2432432
## 5:               3           12      0.1621622
```

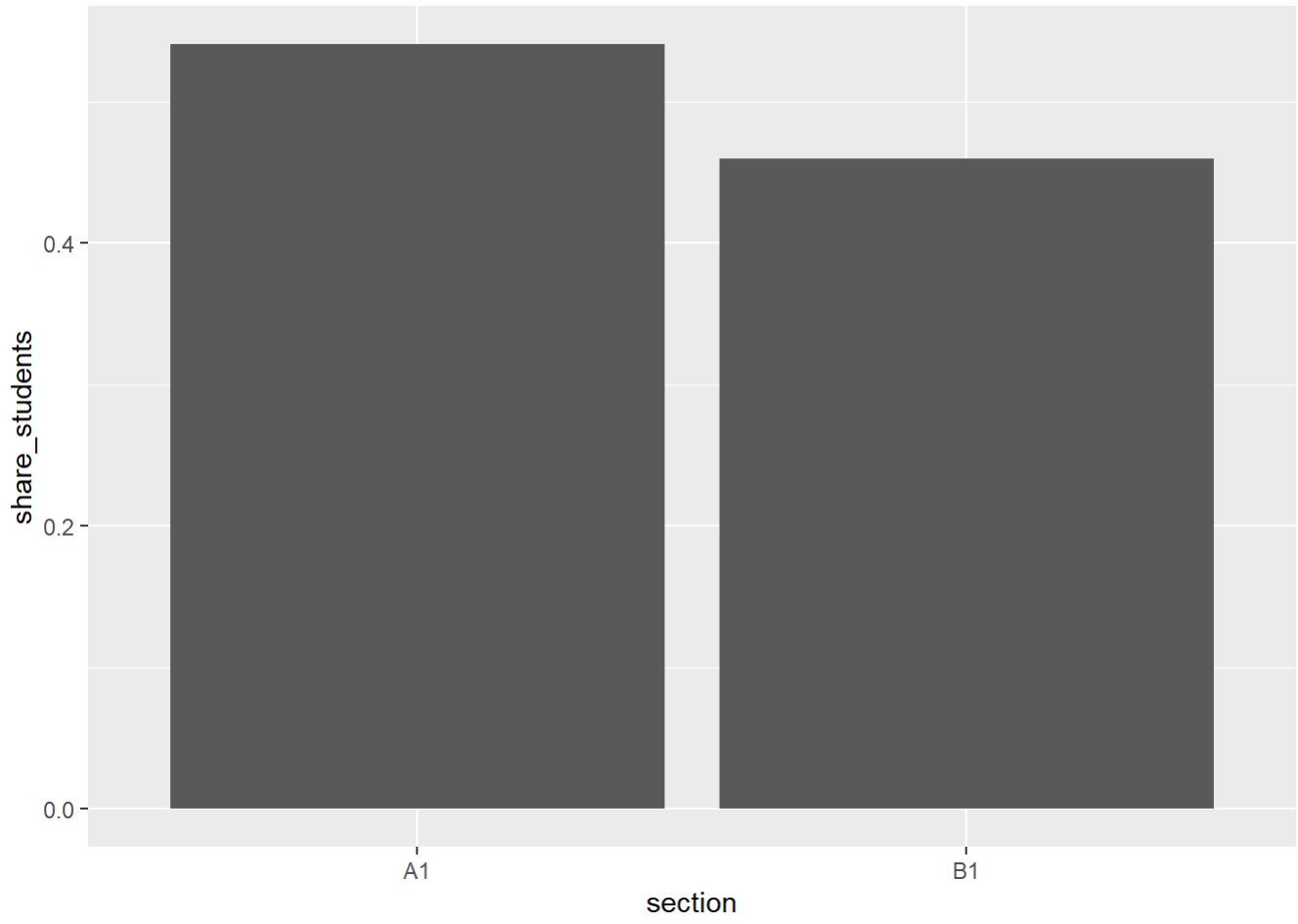
1.5 Calculate the share of students that has each treatment group (10 points)

We can now plot the shares. We do so by using the package 'ggplot2'. We can load this package by using the command library as below. Remember, you must tell R to load packages.

```
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 section which will be on the x axis, y is the share_students and will be on the y axis). We then add the plot type: 'geom_bar(stat = 'identity')'

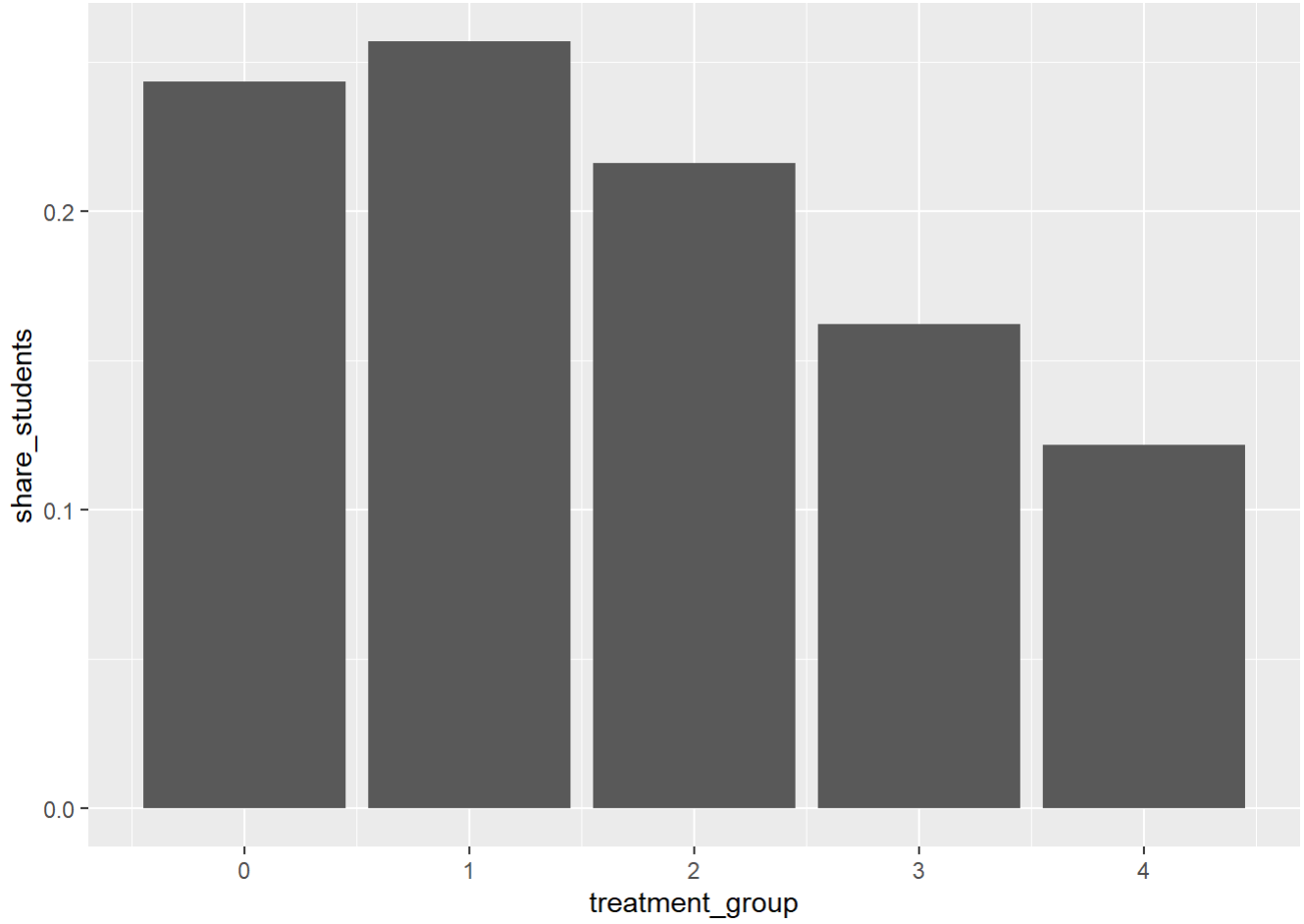
```
this_plot <- ggplot(agg_data, aes(x = section, y = share_students)) + geom_bar(stat = 'identity'
)
this_plot
```



Repeat the above, but plot the share_students by treatment group

1.6 Plot a histogram of the treatment groups using the ggplot function. (10 points)

```
that_plot <- ggplot(treatment_agg_data, aes(x = treatment_group, y = share_students)) + geom_bar
(stat = 'identity')
that_plot
```



1.7 Randomly assign to treatment using simple randomization:

Simple randomization amounts to flipping a coin separately for each person to determine who gets treated. Let's flip a coin so that 40% of the students are in a new treatment group. To do so, we'll use the rbernoulli function. Use ?rbernoulli to look at how it works.

```
# 5 coin flips with probability 20% of TRUE.
rbernoulli(5, .20)
```

```
## [1] FALSE FALSE  TRUE FALSE FALSE
```

```
# Let's check that the mean is close to 20%. Here we use the function mean to take the average o
f the sample.
mean(rbernoulli(1000, .20))
```

```
## [1] 0.212
```

We now want to create a column in our dataset that has a 40% of people in the treatment. To do so, we need to tell rbernoulli how many times to flip the coin (it's the number of people in the dataset) and the probablity. Write the correct code based on the column below:

```
# class_data[, treatment := rbernoulli(number of people goes here, probability goes here)]
# Your code here:
class_data[, treatment := rbernoulli(tot_num_students, .4)]
```

Finally, let's look at the share of people in section A1 in the treatment group. What do you get?

```
class_data[treatment == 1, mean(section == 'A1')]
```

```
## [1] 0.4324324
```

What about the share of students in section B1 in the control group?

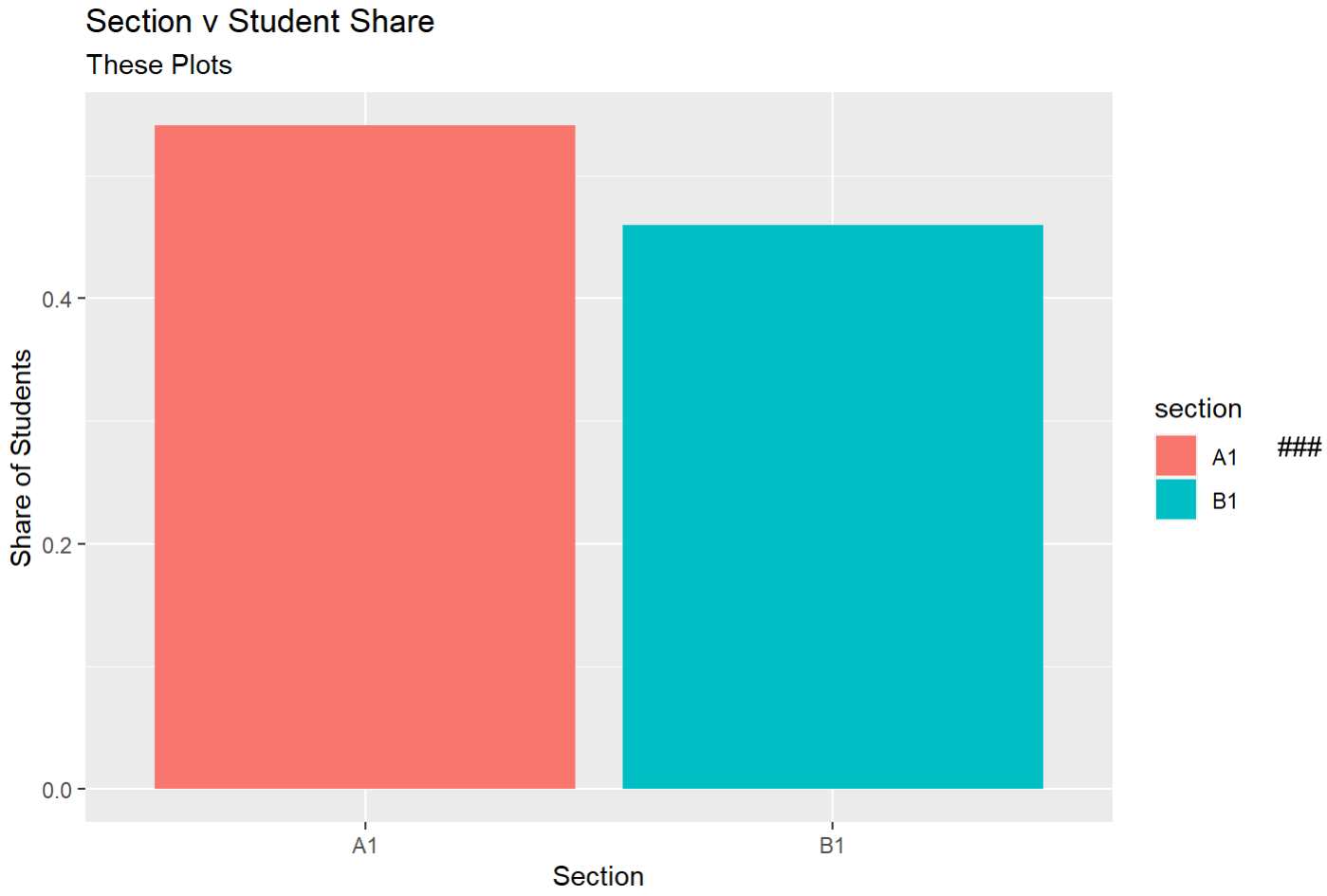
```
# Your code here:
class_data[treatment == 1, mean(section == 'B1')]
```

```
## [1] 0.5675676
```

1.8 BONUS

```
these_plot <- ggplot(agg_data, aes(x = section, y = share_students, fill = section)) +
  geom_bar(stat = 'identity') +
  labs (x= "Section", y= "Share of Students", title= "Section v Student Share",colour = "sectio
n", subtitle = "These Plots")

these_plot
```



Problem 2

```
# R code if needed
PO <- data.table("Users" = seq(1,10))
PO[,"Yes_QA":= c(1100,100,500,900,1600,2000,1200,700,1100,140)]
PO[,"No_QA" := c(1100,600,500,900,700,2000,1200,700,1000,140)]
PO$TE <- PO$Yes_QA -PO$No_QA
PO
```

```
##     Users Yes_QA No_QA   TE
##  1:     1   1100  1100    0
##  2:     2    100   600 -500
##  3:     3    500   500    0
##  4:     4    900   900    0
##  5:     5   1600   700  900
##  6:     6   2000  2000    0
##  7:     7   1200  1200    0
##  8:     8    700   700    0
##  9:     9   1100  1000  100
## 10:    10    140   140    0
```

Question a

The treatment for this situation is a User saw the Q&A page on a Wayfair's furniture page. The outcomes are the reveune they would spend if they had the treatment (User Saw Q&A) or if they didn't have the treatment (User did not see Q&A).

If User 2 recieved the treatment (Saw Q&A) then they would spend 100 if USer 2 did not recieve the treatment (Did not See Q&A) then they would spend 600.

Question b

```
PO[,list(Users,TE)]
```

```
##     Users   TE
##  1:     1    0
##  2:     2 -500
##  3:     3    0
##  4:     4    0
##  5:     5  900
##  6:     6    0
##  7:     7    0
##  8:     8    0
##  9:     9  100
## 10:    10    0
```

Based on the data, Users 1,3,4,6,7,8,and 10 had no treatment effect. Users 2 had a negative treatment effect of -500. Users 5 and 9 had a positive treatment effect of 900 and 100 respectively.

Question c

User with 0 treatment effect were not influenced positvely or negatively by seeing the Q&A page. These users could have no questions or very set on purchasing what they purchases. Users with a positive treatment effect could have so an answer to a question they had and been swayed to buy a product they were on the edge about. Users with a negative treatment effect could have either not had seen an answer to their question or seen an answers. Ultimately they were swayed to buying less products than if they hadn't seen the Q&A page.

Question d

```
avg <- c(mean(PO$Yes_QA),mean(PO$No_QA),mean(PO$TE))
avg
```

```
## [1] 934 884  50
```

The observed average for the treatment was 934. The observed average for no treatment was 884. The impact of the treatment was 50. Due to seeing the Q&A page we can see an increase in 50.

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

It took me the class time. I thought it was reasonable. However, I'm assuming this material was easier given the fact this is in the beginning. I also don't know how much you will be guiding us on applying the course concepts in future assignments.