Problem Set 3

YOUR NAME HERE

DATE

- Due date: Monday, September 21
- Submission process: Please submit your assignment directly to Gradescope. You can do this by knitting your file and downloading the PDF to your computer. Then navigate to Gradescope.com or via the link on BCourses to submit your assignment.

Helpful hints:

- Knit your file early and often to minimize knitting errors! If you copy and paste code from the slides, you are bound to get an error that is hard to diagnose. Typing out the code is the way to smooth knitting. We recommend knitting your file each time after you write a few sentences/add a new code chunk, so you can detect the source of the knitting error more easily. This will save you and the teaching team time!
- Please make sure that your code does not run off the page of the knitted PDF. If it does, we can't see your work. To avoid this, have a look at your knitted PDF and ensure all the code fits in the file. When it doesn't, go back to your .Rmd file and add spaces (new lines) using hte return or enter key so that the code runs onto the next line.

```
# insert libraries that you will need
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.1
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggthemes)
```

For this problem set we will use *fictional* data inspired by research on non-deceptive or open-label placebos. Non-deceptive placebos are placebos but without the deception. Some studies have found suggestions that, despite not being tricked, participants are reporting similar benefits to what they would have with placebos! You can read more here:

https://www.npr.org/sections/health-shots/2016/10/27/499475288/is-it-still-a-placebo-when-it-works-and-you-know-its-a-placebo

https://www.nature.com/articles/s41467-020-17654-y

In this fictional data we conducted an experiment across two unversity sites to investigate whether non-deceptive placebos decreased self-report pain ratings. There were three groups: control, placebo, and non-deceptive placebo. Each participant completed a pre- and post- pain rating. All participants completed the same procedures during the pre-test. Only during the post-test did participants in the intervention arms (placebo, non-deceptive) receive additional instructions prior to the pain rating (i.e., placebo or non-deceptive placebo ratings).

Data coding:

- ID: Contains participant ID number and a letter to indicate group. C = Control, P = Placebo, N = Non-deceptive

Question 1

Read in the data! To make it slightly more challenging we have changed the format from a .csv to .xlsx and "hidden" the data one level deeper in the /data folder. Take a look at the data to get oriented.

```
# read in data with readr
df <- readxl::read_xlsx("data/one_level_deeper/non_deceptive_placebo.xlsx")
```

Question

It's a bit difficult to tell what group each participant is in with their IDs combined with their grouping. Create a new column called "GROUP" based on the letter assignment for IDs. Makes ure to following naming convetion above.

```
# grab index for each group
index_c <- str_detect(df$ID, "^C")
index_d <- str_detect(df$ID, "^P")
index_n <- str_detect(df$ID, "^N")

# use index for each group to assign group name
df$GROUP[index_c] <- "Control"
```

```
## Warning: Unknown or uninitialised column: `GROUP`.
```

```
df$GROUP[index_d] <- "Placebo"
df$GROUP[index_n] <- "Non-deceptive"

head(df)
```

```
## # A tibble: 6 x 5
##   ID       LOCATION DATE                PAIN_RATE GROUP
##   <chr>    <chr>    <chr>                   <dbl> <chr>
## 1 C101_pre UCLA     January 31st, 2018          8 Control
## 2 P102_pre UCLA     February 25th, 2018         7 Placebo
## 3 N103_pre UCLA     January 17th, 2018          7 Non-deceptive
## 4 C104_pre UCLA     January 31st, 2018          8 Control
## 5 P105_pre UCLA     February 25th, 2018         6 Placebo
## 6 N106_pre UCLA     January 17th, 2018          8 Non-deceptive
```

Question

We have a similar issue telling apart the pre- and post- observations. Create a new column called "TEST" that distinguishes whether the observation is a pre- or post-test.

Unfortunately, the two research sites were not consistent in their naming convention. You will need to consider the different cases.

```
# change all of ID case to uppercase to standardize
df$ID <- str_to_upper(df$ID)

# grab index
index_pre <- str_detect(df$ID, "PRE$")
index_post <- str_detect(df$ID, "POST$")

# use index to place correct test instance
df$TEST[index_pre] <- "Pre"
```

```
## Warning: Unknown or uninitialised column: `TEST`.
```

```
df$TEST[index_post] <- "Post"
```

Question

Again, there were difference in the formatting for dates and times across the two research sites. One university combined date and time whereas the other university separated date and time. Moreover, there is also difference in the format of dates. Create a new column called "DATE_FIX" that grabs only the date. Make sure this new date column takes the following format: yyyy-mm-dd

Hint: Check out ?parse_date_time

```
df$DATE_FIX <- parse_date_time(df$DATE, c("mdy", "dmy"))
```

Question

You realize there was a strange error in your excel file that pushed the years forward by 1 year. Rather than editing your excel sheet and potentially making a incorrect permanent change to your raw data you decide to fix the error in R. Create a new column called "DATE_FIX_2" that subtracts the year by 1.

```
df$DATE_FIX2 <- df$DATE_FIX - years(1)
```

Question

Let's clean up our data frame by removing DATE and DATE_FIX. Afterwards, rename DATE_FIX2 to DATE.

```
df <- subset(df, select = -c(DATE, DATE_FIX))
names(df)[6] <- "DATE"
```

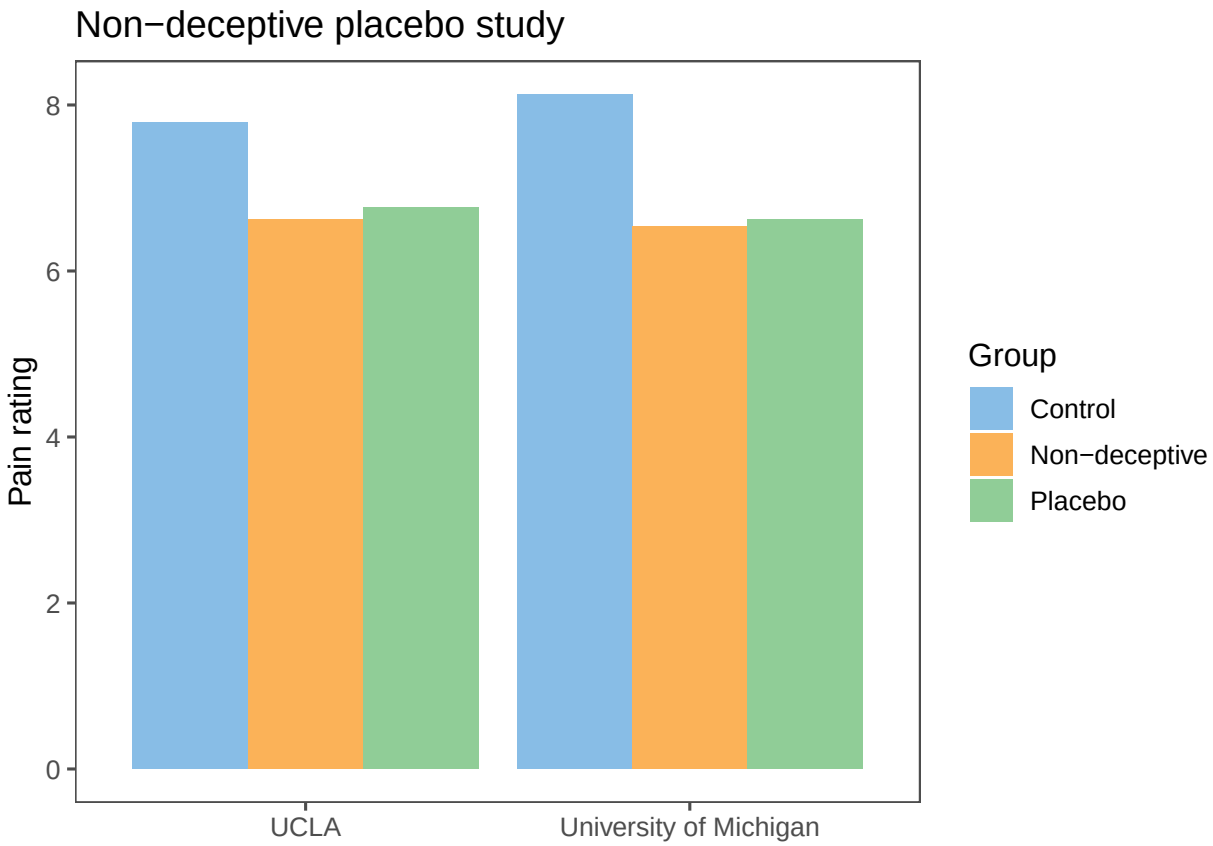
Question

We're interested in plotting our data to begin digging into the results. Below is dplyr and ggplot code that we will go over later in the course. For the time being run the following code.

```
df_plot <- df %>%
  group_by(GROUP, LOCATION) %>%
  summarize(MEAN_PAIN = mean(PAIN_RATE))
```

```
## `summarise()` regrouping output by 'GROUP' (override with `.groups` argument)
```

```
ggplot(df_plot, aes(x = LOCATION, y = MEAN_PAIN, fill = GROUP)) +
  geom_col(position = "dodge") +
  theme_few() +
  scale_fill_few("Light") +
  theme(axis.title = element_blank(),
        axis.title.y = element_text()) +
  labs(fill = "Group",
       title = "Non-deceptive placebo study",
       y = "Pain rating")
```



For a quick first pass we think this visualization isn't so bad. However, logically, we think that the order of the groups should be: Control, Placebo, Non-deceptive. Make GROUP into a factor that reflects this order. If done correctly the above plot should have the same order.

```
df$GROUP <- factor(df$GROUP,
                   levels = c("Control", "Placebo", "Non-deceptive"),
                   ordered = TRUE)
```

You're done! Please knit to pdf and upload to gradescope.