PHW251 Problem Set 6

Clara Voong

10/14/2024

Part 1

For this part we will work with fictional data comparing the efficacy of two interventions. The interventions took place across several states and cities, with slight variations in dates. The outcome is a continuous variable.

Question 1

There's missing data in this data set. Can you identify them? In the next question you will re-code these values to NA.

```
unique(df)
```

```
## # A tibble: 49 x 7
##    date       city        state intervention gender orientation       outcome
##    <chr>      <chr>       <chr>        <dbl> <chr>  <chr>               <dbl>
##  1 25/05/2018 atlanta     GA               1 -999   heterosexual           10
##  2 25/05/2018 Atlanta     gA               1 -999   heterosexual            6
##  3 25/05/2018 atlAnTa     TX               2 female lesbian/gay woman       3
##  4 25/02/2019 San Antonio TX               1 -1     heterosexual            9
##  5 25/02/2019 austin      tX               2 male   heterosexual            1
##  6 25/03/2018 oakland     ca               2 male   heterosexual            2
##  7 25/03/2018 oakland     ca               2 female heterosexual           NA
##  8 25/03/2018 Hayward     CA              NA <NA>   heterosexual            4
##  9 25/03/2018 hayward     GA               1 male   gay                     9
## 10 25/03/2018 hayward     TX               1 female heterosexual           10
## # i 39 more rows
```

```
summary(df)
```

```
##      date               city              state            intervention
##  Length:49          Length:49          Length:49          Min.   :1.000
##  Class :character   Class :character   Class :character   1st Qu.:1.000
##  Mode  :character   Mode  :character   Mode  :character   Median :1.000
##                                                           Mean   :1.421
##                                                           3rd Qu.:2.000
##                                                           Max.   :2.000
##                                                           NA's   :11
##     gender          orientation           outcome
##  Length:49          Length:49          Min.   : 1.000
##  Class :character   Class :character   1st Qu.: 3.000
##  Mode  :character   Mode  :character   Median : 6.000
##                                        Mean   : 5.381
##                                        3rd Qu.: 7.000
```

```
##                                          Max.   :10.000
##                                          NA's   :7
```

```
sum(colSums(is.na(df)))
```

```
## [1] 32
```

```
df %>%
  map(~ unique(.))
```

```
## $date
## [1] "25/05/2018" "25/02/2019" "25/03/2018"
##
## $city
##  [1] "atlanta"     "Atlanta"     "atlAnTa"     "San Antonio" "austin"
##  [6] "oakland"     "Hayward"     "hayward"     "san Antonio" "iakland"
## [11] "Haywarf"
##
## $state
##  [1] "GA"  "gA"  "TX"  "tX"  "ca"  "CA"  NA    "ga"  "tx"  "C A" "G A" "CA_"
##
## $intervention
## [1]  1  2 NA
##
## $gender
## [1] "-999"   "female" "-1"     "male"   NA
##
## $orientation
## [1] "heterosexual"      "lesbian/gay woman" "gay"
## [4] "-999"              "-1"                NA
## [7] "other"
##
## $outcome
##  [1] 10  6  3  9  1  2 NA  4  7  8  5
```

1A. How many NAs did you find?

32

1B. Are there other values you think may count as NA?

Yes, for the gender and orientation variables, there are other values like “-999” and “-1” that may consitute as NAs.

Question 2

2A. For the other values you believe may also be NAs, re-code them as NA.

```
df <- df %>%
  mutate(
    gender = ifelse(gender %in% c("-999", "-1"), NA, gender),
    orientation = ifelse(orientation %in% c("-999", "-1"), NA, orientation)
  )

df %>%
  map(~ unique(.))
```

```
## $date
## [1] "25/05/2018" "25/02/2019" "25/03/2018"
##
## $city
##  [1] "atlanta"     "Atlanta"     "atlAnTa"     "San Antonio" "austin"
##  [6] "oakland"     "Hayward"     "hayward"     "san Antonio" "iakland"
## [11] "Haywarf"
##
## $state
##  [1] "GA"  "gA"  "TX"  "tX"  "ca"  "CA"  NA    "ga"  "tx"  "C A" "G A" "CA_"
##
## $intervention
## [1]  1  2 NA
##
## $gender
## [1] NA       "female" "male"
##
## $orientation
## [1] "heterosexual"      "lesbian/gay woman" "gay"
## [4] NA                  "other"
##
## $outcome
##  [1] 10  6  3  9  1  2 NA  4  7  8  5
```

2B. Print the head() of the dataframe

```
head(df)
```

```
## # A tibble: 6 x 7
##   date       city        state intervention gender orientation       outcome
##   <chr>      <chr>       <chr>        <dbl> <chr>  <chr>               <dbl>
## 1 25/05/2018 atlanta     GA               1 <NA>   heterosexual           10
## 2 25/05/2018 Atlanta     gA               1 <NA>   heterosexual            6
## 3 25/05/2018 atlAnTa     TX               2 female lesbian/gay woman       3
## 4 25/02/2019 San Antonio TX               1 <NA>   heterosexual            9
## 5 25/02/2019 austin      tX               2 male   heterosexual            1
## 6 25/03/2018 oakland     ca               2 male   heterosexual            2
```

Question 3

Now that we've fixed our NA values, let's address the errors we see with city and state names. Let's fix these entries to have uniform naming where cities are properly capitalized and state abbreviations are in all capital letters. For example, we want to see "San Antonio" and "TX" rather than "san Antonio" and "tx".

3A. Use `distinct()` and `pull()` to see all the variations you need to account for.

```
df %>%
  distinct(city) %>%
  pull(city)
```

```
##  [1] "atlanta"     "Atlanta"     "atlAnTa"     "San Antonio" "austin"
##  [6] "oakland"     "Hayward"     "hayward"     "san Antonio" "iakland"
## [11] "Haywarf"
```

```
df %>%
  distinct(state) %>%
  pull(state)
```

```
##  [1] "GA"  "gA"  "TX"  "tX"  "ca"  "CA"  NA    "ga"  "tx"  "C A" "G A" "CA_"
```

3B. Then, use `case_when()` to fix the values.

We have provided the code to fix the variation for Georgia and Texas using `case_when()`. Expand this code to fix the state abbreviations for California and all the city names.

```
df <- df %>%
  mutate(state = case_when(
    state %in% c("GA", "gA", "ga", "G A") ~ "GA",
    state %in% c("TX", "tX", "tx") ~ "TX",
    state %in% c("ca", "CA", "C A", "CA_") ~ "CA",
    TRUE ~ NA
  ))

df <- df %>% mutate(
  city= case_when(
    city %in% c("atlanta", "Atlanta", "atlAnTa") ~ "Atlanta",
    city %in% c("San Antonio", "san Antonio") ~ "San Antonio",
    city == "austin" ~ "Austin",
    city %in% c("oakland", "iakland") ~ "Oakland",
    city %in% c("Hayward", "hayward", "Haywarf") ~ "Hayward",
    TRUE ~ NA
  )
)

df %>%
  distinct(city) %>%
  pull(city)
```

```
## [1] "Atlanta"     "San Antonio" "Austin"      "Oakland"     "Hayward"
```

```
df %>%
  distinct(state) %>%
  pull(state)
```

```
## [1] "GA" "TX" "CA" NA
```

Question 4

4A. Format the date column into a date format using a lubridate function.

Ominously, these interventions all occurred on the 25th day of the month.

```
df <- df %>% mutate(
  date = dmy(date)
)

df
```

```
## # A tibble: 49 x 7
##    date       city        state intervention gender orientation       outcome
##    <date>     <chr>       <chr>         <dbl> <chr>  <chr>               <dbl>
##  1 2018-05-25 Atlanta     GA                1 <NA>   heterosexual           10
##  2 2018-05-25 Atlanta     GA                1 <NA>   heterosexual            6
##  3 2018-05-25 Atlanta     TX                2 female lesbian/gay woman       3
##  4 2019-02-25 San Antonio TX                1 <NA>   heterosexual            9
##  5 2019-02-25 Austin      TX                2 male   heterosexual            1
##  6 2018-03-25 Oakland     CA                2 male   heterosexual            2
##  7 2018-03-25 Oakland     CA                2 female heterosexual           NA
##  8 2018-03-25 Hayward     CA               NA <NA>   heterosexual            4
##  9 2018-03-25 Hayward     GA                1 male   gay                     9
## 10 2018-03-25 Hayward     TX                1 female heterosexual           10
## # i 39 more rows
```

Question 5

You may have noticed that some of the cities don't match their state. We can't, at least from our data, distinguish which value is correct (the city or the state). The correct city and state pairings are:

- Atlanta, GA
- Austin, TX
- San Antonio, TX
- Hayward, CA
- Oakland, CA

5A. Drop the rows with this city/state inconsistency.

One suggestion is to create a variable indicating whether to drop the row. If you performed this step correctly you should have 33 rows.

```
df <- df %>% mutate(
  keep = if_else(
    (city == "Atlanta" & state == "GA") |
    (city == "Austin" & state == "TX") |
    (city == "San Antonio" & state == "TX") |
    (city == "Hayward" & state == "CA") |
    (city == "Oakland" & state == "CA"),
    1,
    0
  )
)

df <- df %>% filter(keep==1)

dim(df)
```

```
## [1] 33  8
```

5B. Print the unique combinations of city and state that are now in the data frame

Use the code below and modify if needed.

```
unique(df[,c("city", "state")])
```

```
## # A tibble: 5 x 2
##   city        state
##   <chr>       <chr>
## 1 Atlanta     GA
## 2 San Antonio TX
## 3 Austin      TX
## 4 Oakland     CA
## 5 Hayward     CA
```

Question 6

Another issue: our interventions column has missing data. We have two interventions that occurred in these locations:

- Intervention 1: Hayward, Atlanta, San Antonio
- Intervention 2: Oakland, Atlanta, Austin

For all of the cities except Atlanta it's clear what intervention took place.

6A. In these clear instances, replace NAs with the appropriate intervention.

```
df <- df %>% mutate(
  intervention = case_when(
    city %in% c("Hayward", "San Antonio") ~ 1,
    city %in% c("Oakland", "Austin") ~ 2,
    TRUE ~ intervention
  )
)

head(df)
```

```
## # A tibble: 6 x 8
##   date       city        state intervention gender orientation  outcome  keep
##   <date>     <chr>       <chr>        <dbl> <chr>  <chr>          <dbl> <dbl>
## 1 2018-05-25 Atlanta     GA               1 <NA>   heterosexual      10     1
## 2 2018-05-25 Atlanta     GA               1 <NA>   heterosexual       6     1
## 3 2019-02-25 San Antonio TX               1 <NA>   heterosexual       9     1
## 4 2019-02-25 Austin      TX               2 male   heterosexual       1     1
## 5 2018-03-25 Oakland     CA               2 male   heterosexual       2     1
## 6 2018-03-25 Oakland     CA               2 female heterosexual      NA     1
```

```
dim(df)
```

```
## [1] 33  8
```

6B. For Atlanta, drop the observations with missing intervention data since we cannot determine which intervention occurred.

```
df <- df %>%
  filter(!(city == "Atlanta" & is.na(intervention)))

dim(df)
```

```
## [1] 31  8
```

6C. How many observations did you drop?

2

Question 7

We have a few NAs in the outcome column. Our on-site researchers informed us that when a score of "0" was provided, the data collection team left the cell blank.

7A. Re-code the NAs to 0.

```
df <- df %>% mutate(
  outcome = if_else(
    is.na(outcome), 0, outcome
  )
)

head(df)
```

```
## # A tibble: 6 x 8
##   date       city        state intervention gender orientation  outcome  keep
##   <date>     <chr>       <chr>        <dbl> <chr>  <chr>          <dbl> <dbl>
## 1 2018-05-25 Atlanta     GA               1 <NA>   heterosexual      10     1
## 2 2018-05-25 Atlanta     GA               1 <NA>   heterosexual       6     1
## 3 2019-02-25 San Antonio TX               1 <NA>   heterosexual       9     1
## 4 2019-02-25 Austin      TX               2 male   heterosexual       1     1
## 5 2018-03-25 Oakland     CA               2 male   heterosexual       2     1
## 6 2018-03-25 Oakland     CA               2 female heterosexual       0     1
```

7B. Use code to confirm that there are no longer any NAs in the outcome column.

```
sum(is.na(df$outcome))
```

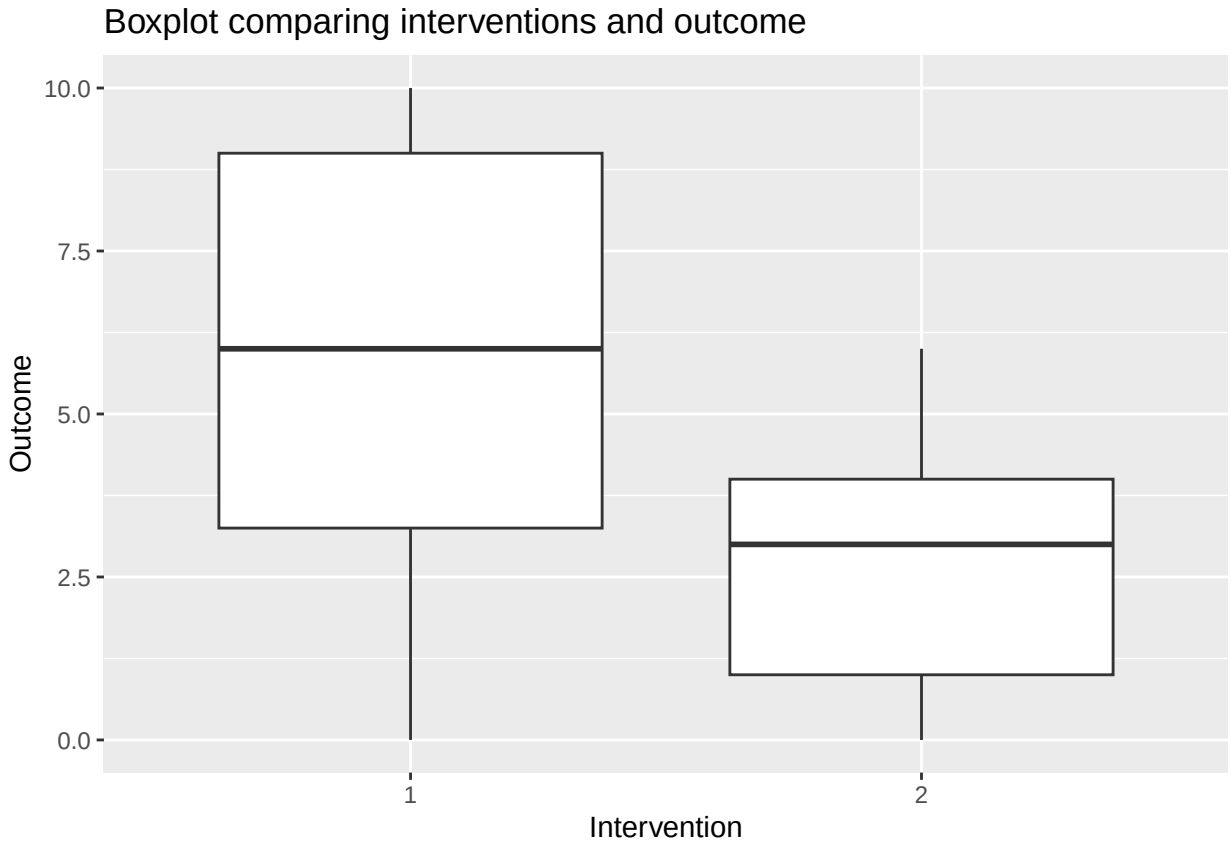
```
## [1] 0
```

Question 8

8A Use ggplot to create a box plot comparing the two interventions and their outcome.

The outcome is a continuous variable from 0 to 10. You may need to factor one of your variables. Look at the visualization cheatsheet if you don't know the "geom" for creating a boxplot.

```
ggplot(data=df, mapping=aes(x=factor(intervention), y=outcome)) +
  geom_boxplot() +
  labs(
    title="Boxplot comparing interventions and outcome",
    x = "Intervention",
    y= "Outcome"
  )
```


Boxplot comparing interventions and outcome
10.0
7.5
5.0
2.5
0.0
Outcome
1
2
Intervention

Part 2

For this part 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:

NPR: Is A Placebo A Sham If You Know It's A Fake And It Still Works?

Nature Communications: Placebos without deception reduce self-report and neural measures of emotional distress

In this fictional data we conducted an experiment across two university 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 induction task and provided a 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 induction task (i.e., placebo or non-deceptive placebo ratings).

Data coding:

- ID: Contains participant ID number, a letter to indicate group, and pre or post tags.

  C = Control P = Placebo N = Non-deceptive
- LOCATION: Research Site
- PAIN RATING: Self report of pain based on a 0-10 scale
- DATE: Date of observation

Question 9

9A 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. Please use "placebo_df" as the name of your data frame.

```
library(readxl)
placebo_df <-
  read_xlsx(
    "~/PHW251_2024/problem_sets/problem_set_6/data/one_level_deeper/non_deceptive_placebo.xlsx")
```

Question 10

It's a bit difficult to tell what group (control, placebo, or non-deceptive placebo) each participant is in with their IDs combined with their grouping.

10A. Create a new column called "GROUP" based on the letter assignment for IDs.

The stringr function 'str_detect()' will be useful here!

```
placebo_df <- placebo_df %>% mutate(
  GROUP = case_when(
    str_detect(ID, "P") ~ "Placebo",
    str_detect(ID, "C") ~ "Control",
    str_detect(ID, "N") ~ "Non-deceptive placebo",
    TRUE ~ NA
  )
)
```

10B. Print the head() of the dataframe

```
head(placebo_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 placebo
## 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 placebo
```

Question 11

We have a similar issue telling apart the pre- and post- observations.

11A. 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.

```
placebo_df <- placebo_df %>%
  mutate(TEST=
           case_when(
             str_detect(ID, "pre") ~ "pre",
             str_detect(ID, "post") ~ "post",
             str_detect(ID, "PRE") ~ "pre",
             str_detect(ID, "POST") ~ "post",
           ))
```

11B. Print the head() of the dataframe

```
head(placebo_df)
```

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

Question 12

There were differences in the formatting for dates across the two research sites.

12A. 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

```
placebo_df <- placebo_df %>% mutate(
  DATEFIX= parse_date_time(DATE, orders = c("B d, Y", "d-b-y")) %>% as.Date()
)
```

12B. Print the head() of the dataframe

```
head(placebo_df)
```

```
## # A tibble: 6 x 7
##   ID       LOCATION DATE                PAIN_RATE GROUP         TEST  DATEFIX
##   <chr>    <chr>    <chr>                   <dbl> <chr>         <chr> <date>
## 1 C101_pre UCLA     January 31st, 2018          8 Control       pre   2018-01-31
## 2 P102_pre UCLA     February 25th, 2018         7 Placebo       pre   2018-02-25
## 3 N103_pre UCLA     January 17th, 2018          7 Non-deceptiv~ pre   2018-01-17
## 4 C104_pre UCLA     January 31st, 2018          8 Control       pre   2018-01-31
## 5 P105_pre UCLA     February 25th, 2018         6 Placebo       pre   2018-02-25
## 6 N106_pre UCLA     January 17th, 2018          8 Non-deceptiv~ pre   2018-01-17
```

Question 13

You realize there was a strange error in your excel file that, for every date, pushed the date forward by 1 year. Rather than editing your excel sheet and potentially making an incorrect permanent change to your raw data you decide to fix the error in R.

13A. Create a new column called "DATE_FIX_2" that fixes the date.

```
placebo_df <- placebo_df %>% mutate(
  DATE_FIX_2 = DATEFIX - years(1)
)

placebo_df
```

```
## # A tibble: 150 x 8
##    ID       LOCATION DATE            PAIN_RATE GROUP TEST  DATEFIX    DATE_FIX_2
##    <chr>    <chr>    <chr>               <dbl> <chr> <chr> <date>     <date>
##  1 C101_pre UCLA     January 31st, ~         8 Cont~ pre   2018-01-31 2017-01-31
##  2 P102_pre UCLA     February 25th,~         7 Plac~ pre   2018-02-25 2017-02-25
##  3 N103_pre UCLA     January 17th, ~         7 Non-~ pre   2018-01-17 2017-01-17
##  4 C104_pre UCLA     January 31st, ~         8 Cont~ pre   2018-01-31 2017-01-31
##  5 P105_pre UCLA     February 25th,~         6 Plac~ pre   2018-02-25 2017-02-25
##  6 N106_pre UCLA     January 17th, ~         8 Non-~ pre   2018-01-17 2017-01-17
##  7 C107_pre UCLA     January 31st, ~         7 Cont~ pre   2018-01-31 2017-01-31
##  8 P108_pre UCLA     February 25th,~         8 Plac~ pre   2018-02-25 2017-02-25
##  9 N109_pre UCLA     January 17th, ~         3 Non-~ pre   2018-01-17 2017-01-17
## 10 C110_pre UCLA     January 31st, ~         8 Cont~ pre   2018-01-31 2017-01-31
## # i 140 more rows
```

Question 14

14A. Clean up the data frame by removing DATE and DATE_FIX.

14B. Afterwards, rename DATE_FIX2 to DATE

```
placebo_df <- placebo_df %>%
  select(ID, LOCATION, DATE_FIX_2, PAIN_RATE, GROUP, TEST) %>%
  rename(DATE = DATE_FIX_2)
```

14C. Print the head() of the dataframe

```
head(placebo_df)
```

```
## # A tibble: 6 x 6
##   ID       LOCATION DATE       PAIN_RATE GROUP                 TEST
##   <chr>    <chr>    <date>         <dbl> <chr>                 <chr>
## 1 C101_pre UCLA     2017-01-31         8 Control               pre
## 2 P102_pre UCLA     2017-02-25         7 Placebo               pre
## 3 N103_pre UCLA     2017-01-17         7 Non-deceptive placebo pre
## 4 C104_pre UCLA     2017-01-31         8 Control               pre
## 5 P105_pre UCLA     2017-02-25         6 Placebo               pre
## 6 N106_pre UCLA     2017-01-17         8 Non-deceptive placebo pre
```

Question 15

We're interested in plotting our data to begin digging into the results. Below is dplyr and ggplot code to do this.

15A. Uncomment and run the following code as-is (visualization is not the focus of this problem set).

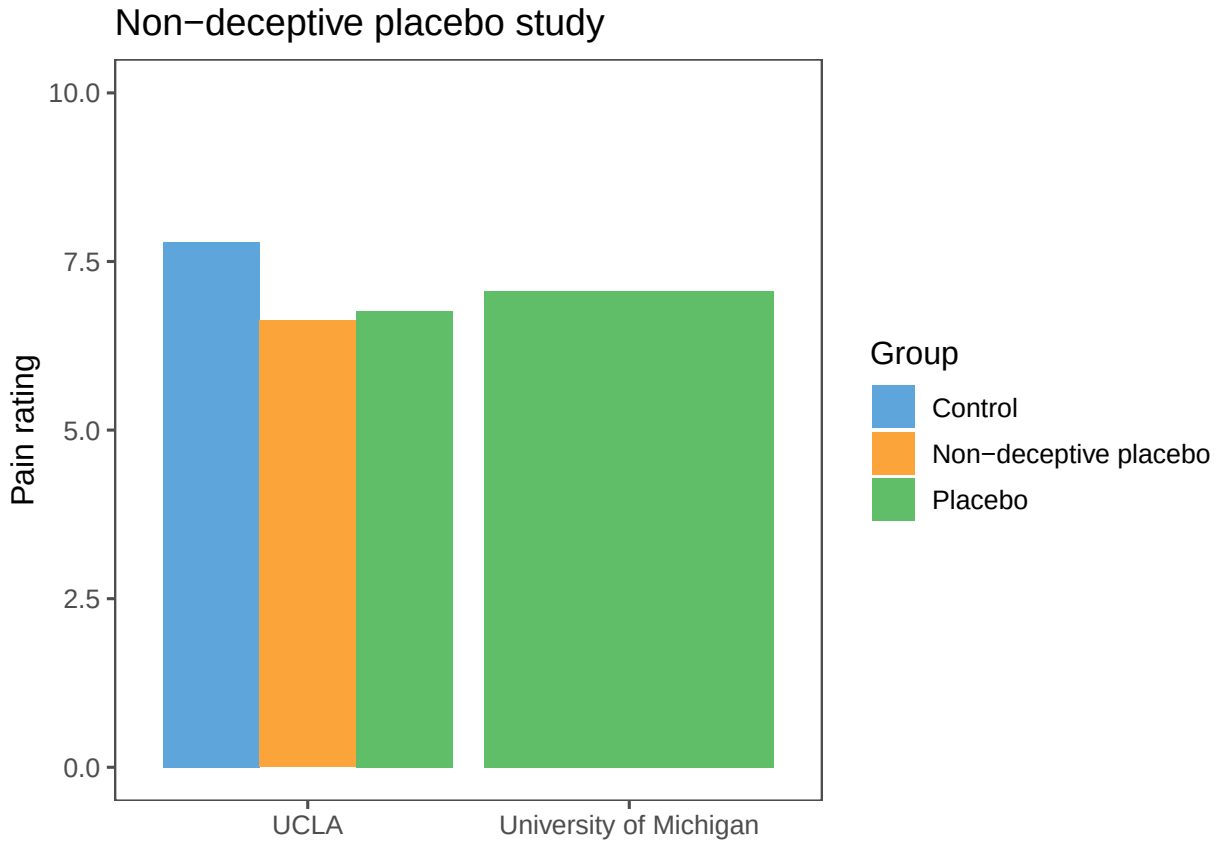
You may need to install ggthemes.

```
# install.packages("ggthemes")
library(ggthemes)

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

```
## `summarise()` has grouped output by 'GROUP'. You can override using the
## `.groups` argument.
```

```
ggplot(df_plot, aes(x = LOCATION, y = MEAN_PAIN, fill = GROUP)) +
  geom_col(position = "dodge") +
  ylim(0, 10) +
  theme_few() +
  scale_fill_few("Medium") +
  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.

15B. Make GROUP into a factor that reflects this order.

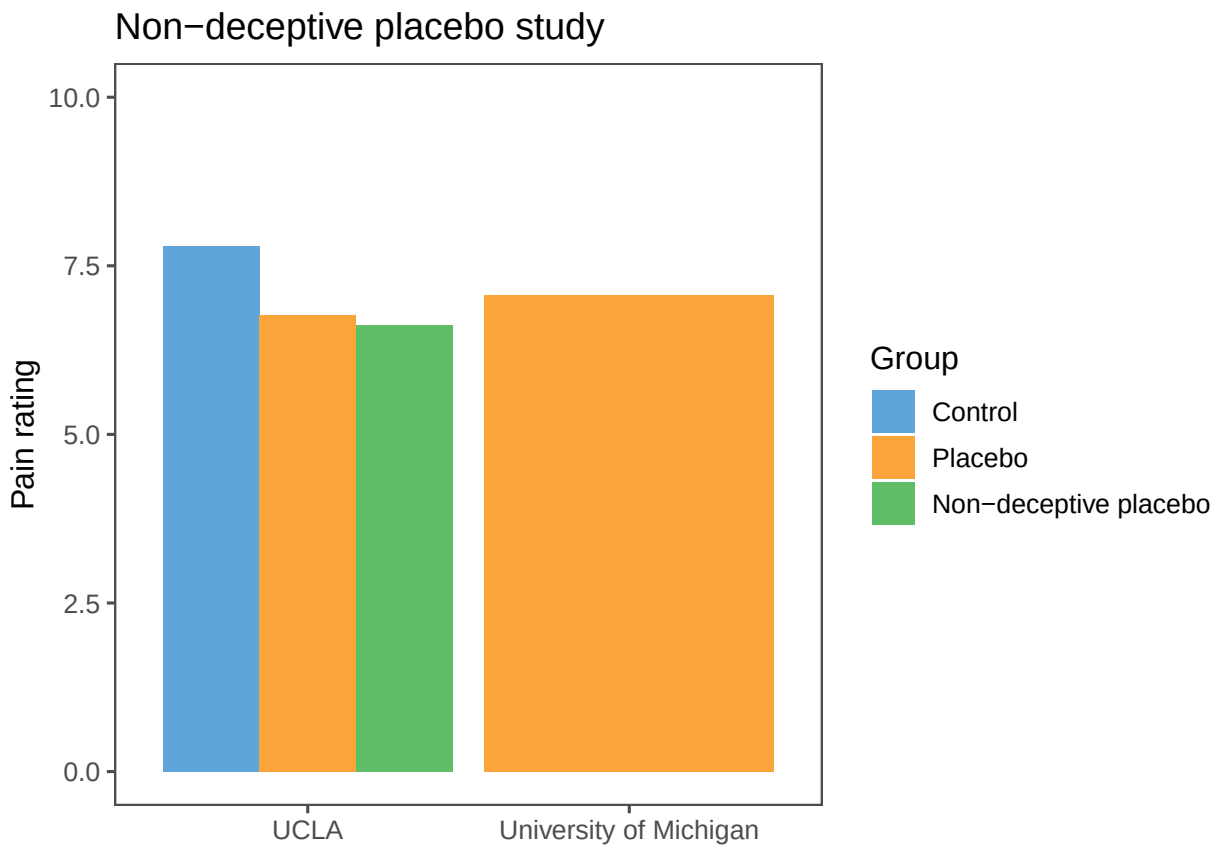
If done correctly, when you re-run the above chunk, the plot should show the bars in that order

```
placebo_df <-
  placebo_df %>%
  mutate(GROUP =
           factor(GROUP, levels = c("Control", "Placebo", "Non-deceptive placebo")))

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

```
## `summarise()` has grouped output by 'GROUP'. You can override using the
## `.groups` argument.
```

```
ggplot(df_plot, aes(x = LOCATION, y = MEAN_PAIN, fill = GROUP)) +
  geom_col(position = "dodge") +
  ylim(0, 10) +
  theme_few() +
  scale_fill_few("Medium") +
  theme(axis.title = element_blank(),
        axis.title.y = element_text()) +
  labs(fill = "Group",
       title = "Non-deceptive placebo study",
       y = "Pain rating")
```



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