Voice Reflection Replication

Chris Yan

2023-07-01

1) First create a graph that represents the proportion of students who self- explained by voice in week 9 for treatment condition.

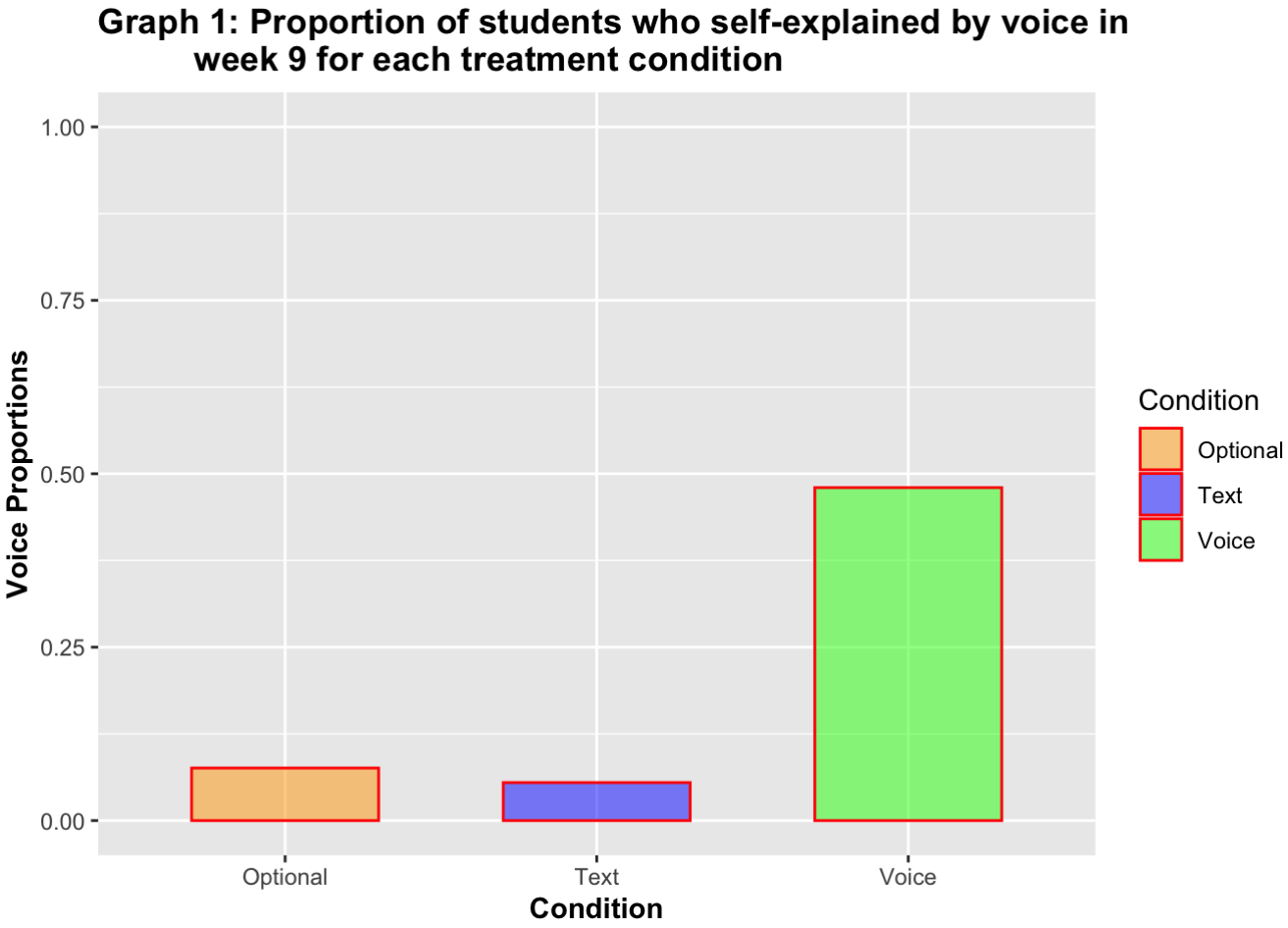
```
# let's first import libraries we will be using
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.4.0     ✔ purrr   1.0.1
## ✔ tibble  3.2.1     ✔ dplyr   1.1.1
## ✔ tidyr   1.3.0     ✔ stringr 1.5.0
## ✔ readr   2.1.3     ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)

# let's now import our data we will be using
d <- read_csv("d_report.csv")
```

```
## Rows: 248 Columns: 36
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): condition, conditionForWeek9, hashed_id
## dbl (33): week4-inter_voiceCount, week4-inter_textCount, week4-inter_complet...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# filter our data so that we get our desired variables
d_filtered <- d[ , c("condition", "week9-normal_voiceCount",
                     "week9-normal_textCount", "week9-normal_completed")]

# get a 3x4 dataframe that has the variables of condition, total_voice, total_
# text, total_count. as well as observations for Optional, Voice, and Text.
# in the end we want to get the sums of each permutation.
grouped_summary <- d_filtered %>%
  group_by(condition) %>%
  mutate(total_voice = ifelse(`week9-normal_voiceCount` >= 1, 1, 0),
         total_text = ifelse(`week9-normal_textCount` >= 1, 1, 0),
         total_count = ifelse(`week9-normal_voiceCount` +
                                `week9-normal_textCount` >= 1, 1, 0)) %>%
  summarize(total_voice = sum(total_voice),
            total_text = sum(total_text),
            total_count = total_voice + total_text)

# add the proportions (ratios) as variables to the dataframe
grouped_summary <- grouped_summary %>%
  mutate(voice_proportion = total_voice / total_count,
         text_proportion = total_text / total_count)

# plot the graph as a bar graph
ggplot(grouped_summary, aes(x=condition, y=voice_proportion, fill=condition)) +
  geom_col(color="red", width=0.6, alpha=0.5) +
  ylim(0, 1) +
  xlab("Condition") +
  ylab("Voice Proportions") +
  ggtitle("Graph 1: Proportion of students who self-explained by voice in
          week 9 for each treatment condition") +
  theme(plot.title = element_text(face="bold"),
        axis.title.x = element_text(face="bold"),
        axis.title.y = element_text(face="bold")) +
  scale_fill_manual(values = c("Optional"="orange", "Text"="blue", "Voice"=
                                 "green"),
                    guide=guide_legend(title="Condition"))
```


Graph 1: Proportion of students who self-explained by voice in week 9 for each treatment condition
1.00
0.75
0.50
0.25
0.00
Voice Proportions
Optional
Text
Voice
Condition
Condition
Optional
Text
Voice

2a) Find the number of people who rated each score -3 (strong disagree), -2, -1, 0, +2, +3 (strongly agree), for each pre-survey Likert items, and represent it using the given template.

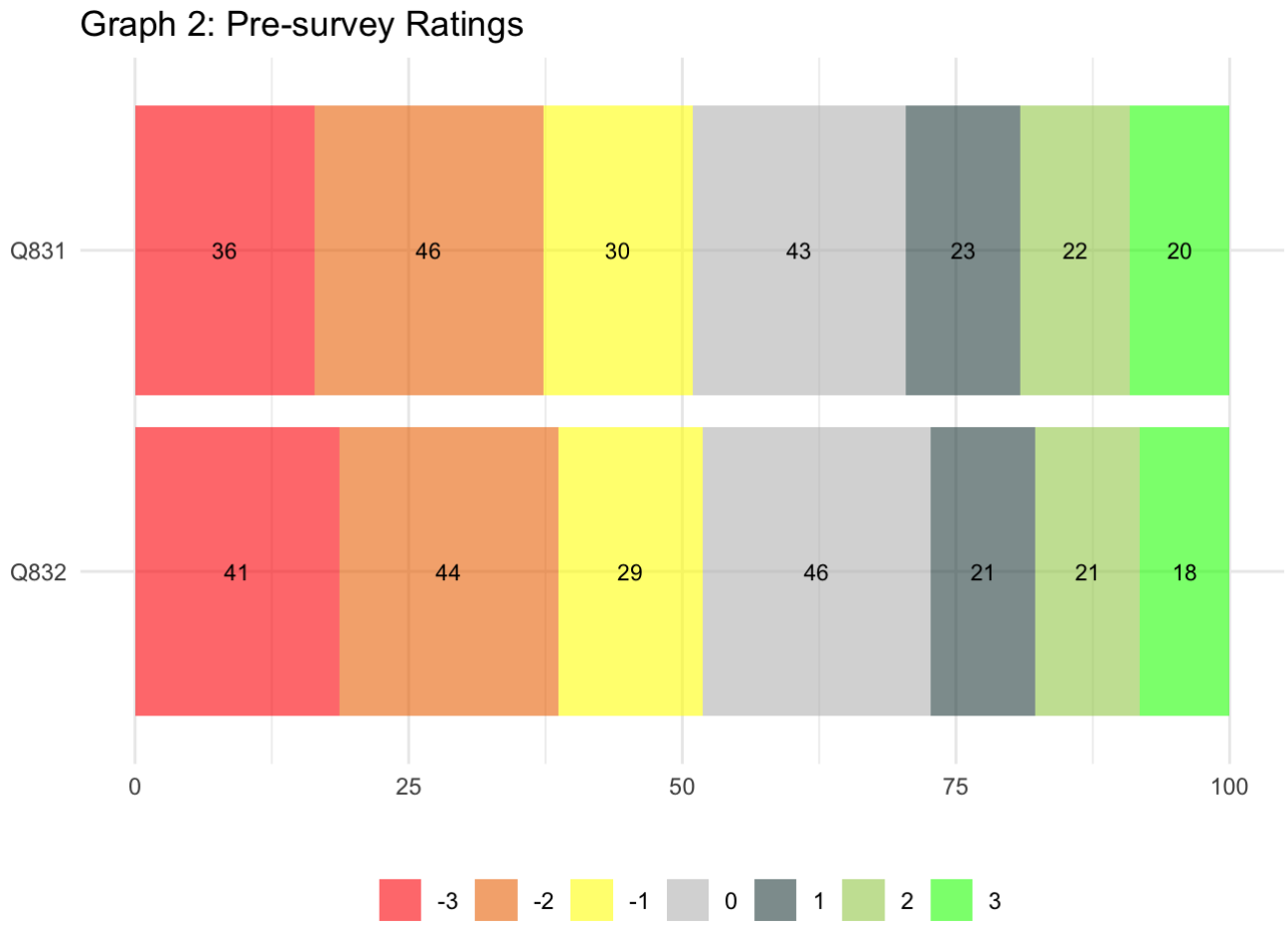
```
library(tidyverse)
library(ggplot2)

# Read the pre-survey data
d_survey <- read_csv('d_pre_survey.csv')
```

```
## Rows: 224 Columns: 59
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (28): StartDate, EndDate, Status, RecordedDate, ResponseId, Distribution...
## dbl (22): Progress, Duration (in seconds), timing_intro_First Click, timing_...
## lgl  (9): Finished, RecipientLastName, RecipientFirstName, RecipientEmail, E...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Filter and transform the data
d_survey_filtered <- d_survey %>%
  select(Q831, Q832) %>%
  filter(!is.na(Q831) & !Q831 %in% c("Prefer not to answer") &
           !is.na(Q832) & !Q832 %in% c("Prefer not to answer")) %>%
  mutate(
    Q831_score = case_when(
      Q831 == "Strongly disagree" ~ -3,
      Q831 == "Disagree" ~ -2,
      Q831 == "Somewhat disagree" ~ -1,
      Q831 == "Neither agree nor disagree" ~ 0,
      Q831 == "Somewhat agree" ~ 1,
      Q831 == "Agree" ~ 2,
      Q831 == "Strongly agree" ~ 3
    ),
    Q832_score = case_when(
      Q832 == "Strongly disagree" ~ -3,
      Q832 == "Disagree" ~ -2,
      Q832 == "Somewhat disagree" ~ -1,
      Q832 == "Neither agree nor disagree" ~ 0,
      Q832 == "Somewhat agree" ~ 1,
      Q832 == "Agree" ~ 2,
      Q832 == "Strongly agree" ~ 3
    )
  )

# Combine the data for Q831 and Q832
combined_data <- rbind(
  data.frame(question = "Q831", score = d_survey_filtered$Q831_score),
  data.frame(question = "Q832", score = d_survey_filtered$Q832_score)
)

# Count the number of people for each score and question
count_data <- combined_data %>%
  group_by(question, score) %>%
  summarize(count = n())
```

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

```
# Calculate the percentages
count_data <- count_data %>%
  group_by(question) %>%
  mutate(percentage = count / sum(count) * 100)

# Define the order and labels for the legend
response_order <- c("-3", "-2", "-1", "0", "1", "2", "3")

# Reorder the levels of the "score" variable
count_data$score <- factor(count_data$score, levels = rev(response_order))

# Reorder the levels of the "question" variable
count_data$question <- factor(count_data$question, levels = c("Q832", "Q831"))


# Plot the horizontal bar graph with reversed x-axis
ggplot(count_data, aes(x = percentage, y = question, fill = score)) +
  geom_col(alpha = 0.6) +
  xlab("") +
  ylab("") +
  ggtitle("Graph 2: Pre-survey Ratings") +
  theme_minimal() +
  scale_fill_manual(
    values = c(
      "-3" = "red",
      "-2" = "darkorange2",
      "-1" = "yellow",
      "0" = "grey",
      "1" = "darkslategrey",
      "2" = "darkolivegreen3",
      "3" = "green"
    ),
    labels = c(3,2,1,0,-1,-2,-3),
    guide = guide_legend(title = "", nrow=1, reverse=TRUE)) +
  theme(legend.position = "bottom") +
  geom_text(aes(label = count), position = position_stack(vjust = 0.5),
            color = "black", size = 3)
```



2b)

Find the number of people who rated each score -3 (strongly disagree), -2, -1, 0, +1, +2, +3 (strongly agree) for each of these post-survey Likert items, and represent it using the given template.

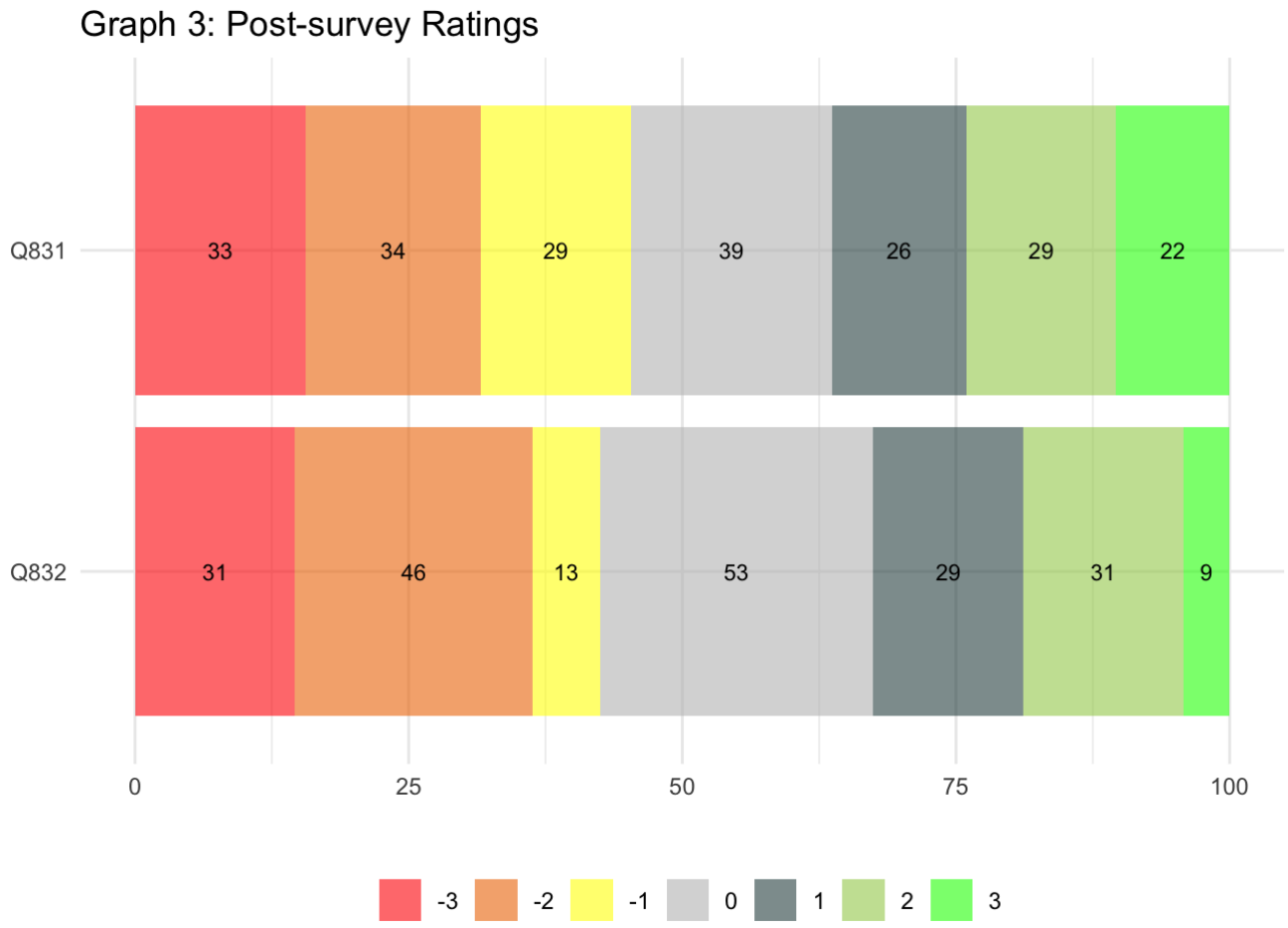
```
library(tidyverse)
library(ggplot2)

# Read the pre-survey data
post_survey <- read_csv('d_post_survey.csv')
```

```
## Rows: 216 Columns: 74
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (48): StartDate, EndDate, Status, RecordedDate, ResponseId, Distribution...
## dbl (21): Progress, Duration (in seconds), timing_intro_First Click, timing_...
## lgl  (5): Finished, RecipientLastName, RecipientFirstName, RecipientEmail, E...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Filter and transform the data
post_survey_filtered <- post_survey %>%
  select(Q831.1, Q832) %>%
  filter(!is.na(Q831.1) & !(Q831.1) %in% c("Prefer not to answer") &
           !is.na(Q832) & !Q832 %in% c("Prefer not to answer")) %>%
  mutate(
    Q831_score = case_when(
      Q831.1 == "Strongly disagree" ~ -3,
      Q831.1 == "Disagree" ~ -2,
      Q831.1 == "Somewhat disagree" ~ -1,
      Q831.1 == "Neither agree nor disagree" ~ 0,
      Q831.1 == "Somewhat agree" ~ 1,
      Q831.1 == "Agree" ~ 2,
      Q831.1 == "Strongly agree" ~ 3
    ),
    Q832_score = case_when(
      Q832 == "Strongly disagree" ~ -3,
      Q832 == "Disagree" ~ -2,
      Q832 == "Somewhat disagree" ~ -1,
      Q832 == "Neither agree nor disagree" ~ 0,
      Q832 == "Somewhat agree" ~ 1,
      Q832 == "Agree" ~ 2,
      Q832 == "Strongly agree" ~ 3
    )
  )

# Combine the data for Q831 and Q832
combined_data <- rbind(
  data.frame(question = "Q831", score = post_survey_filtered$Q831_score),
  data.frame(question = "Q832", score = post_survey_filtered$Q832_score)
)

# Count the number of people for each score and question
count_data <- combined_data %>%
  group_by(question, score) %>%
  summarize(count = n())
```

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

```
# Calculate the percentages
count_data <- count_data %>%
  group_by(question) %>%
  mutate(percentage = count / sum(count) * 100)

# Define the order and labels for the legend
response_order <- c("-3", "-2", "-1", "0", "1", "2", "3")

# Reorder the levels of the "score" variable
count_data$score <- factor(count_data$score, levels = rev(response_order))

# Reorder the levels of the "question" variable
count_data$question <- factor(count_data$question, levels = c("Q832", "Q831"))


# Plot the horizontal bar graph with reversed x-axis
ggplot(count_data, aes(x = percentage, y = question, fill = score)) +
  geom_col(alpha = 0.6) +
  xlab("") +
  ylab("") +
  ggtitle("Graph 3: Post-survey Ratings") +
  theme_minimal() +
  scale_fill_manual(
    values = c(
      "-3" = "red",
      "-2" = "darkorange2",
      "-1" = "yellow",
      "0" = "grey",
      "1" = "darkslategrey",
      "2" = "darkolivegreen3",
      "3" = "green"
    ),
    labels = c(3,2,1,0,-1,-2,-3),
    guide = guide_legend(title = "", nrow=1, reverse=TRUE)) +
  theme(legend.position = "bottom") +
  geom_text(aes(label = count), position = position_stack(vjust = 0.5),
            color = "black", size = 3)
```


Graph 3: Post-survey Ratings
Q831
33
34
29
39
26
29
22
Q832
31
46
13
53
29
31
9
0
25
50
75
100
-3
-2
-1
0
1
2
3

3) Creating a box plot that shows the ratings of people who preferred each medium type, based on their original treatment condition.

```
library(tidyverse)
library(ggplot2)

d_report <- read_csv('d_report.csv')
```

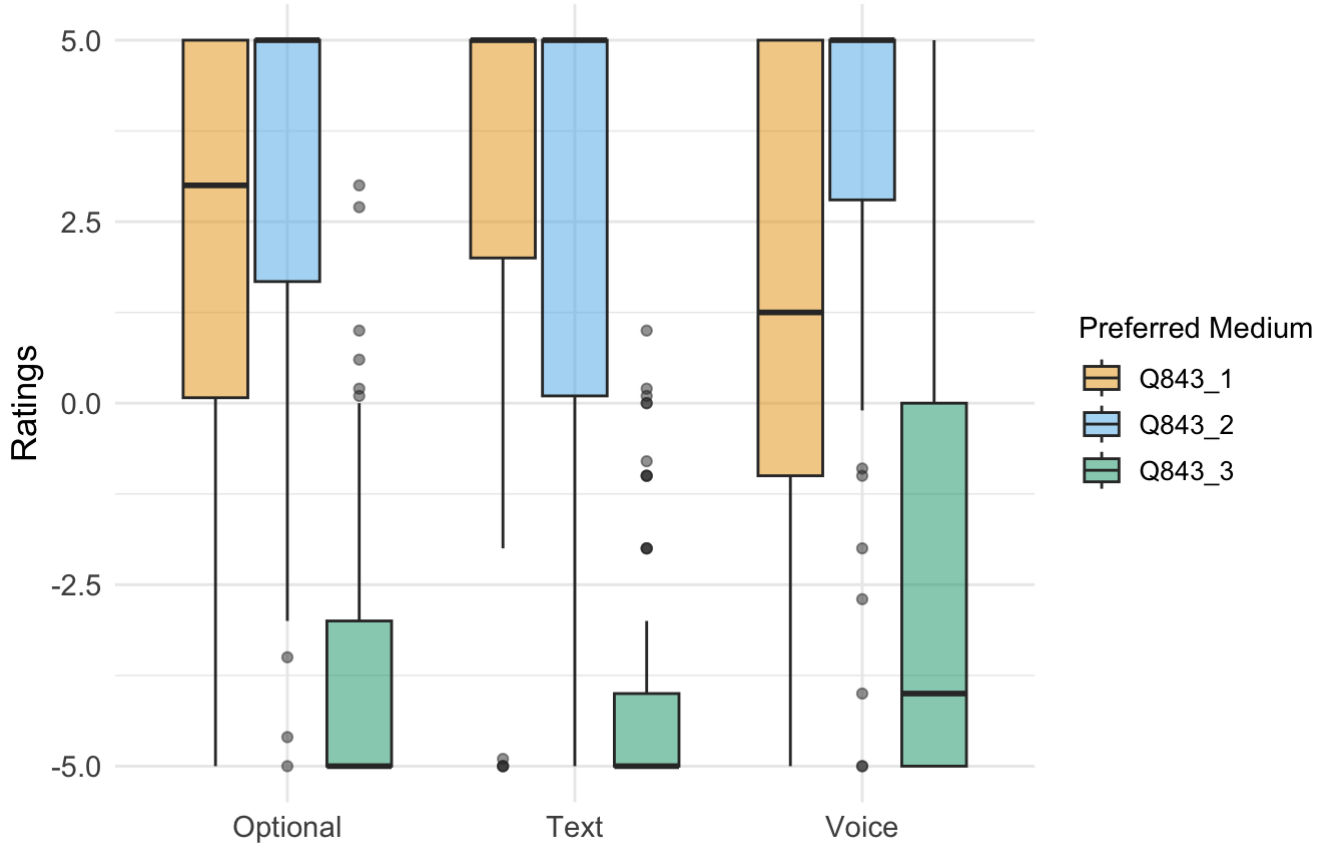
```
## Rows: 248 Columns: 36
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): condition, conditionForWeek9, hashed_id
## dbl (33): week4-inter_voiceCount, week4-inter_textCount, week4-inter_complet...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
d_post_survey <- read_csv('d_post_survey.csv')
```

```
## Rows: 216 Columns: 74
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (48): StartDate, EndDate, Status, RecordedDate, ResponseId, Distribution...
## dbl (21): Progress, Duration (in seconds), timing_intro_First Click, timing_...
## lgl  (5): Finished, RecipientLastName, RecipientFirstName, RecipientEmail, E...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# lets get the common hashed_id values
common_hashed_id <- intersect(d_report$hashed_id, d_post_survey$hashed_id)

# now filter the data based on common hashed_id values
filtered_d_report <- d_report[d_report$hashed_id %in% common_hashed_id, ]
filtered_d_post_survey <- d_post_survey[d_post_survey$hashed_id %in%
                                        common_hashed_id, ]

# then select the desired columns
selected_d_report <- filtered_d_report[, c("hashed_id", "condition")]
selected_d_post_survey <- filtered_d_post_survey[, c("hashed_id", "Q843_1",
                                                     "Q843_2", "Q843_3")]

# now merge the selected dataframes based on hashed_id
combined <- merge(selected_d_report, selected_d_post_survey, by = "hashed_id")

# now reshape the data into long format
combined_long <- tidyr::gather(combined, MediumType, Rating, -hashed_id,
                               -condition)

# lastly create the box plot
ggplot(combined_long, aes(x = condition, y = Rating, fill = MediumType)) +
  geom_boxplot(alpha=0.5) +
  labs(x = "", y = "Ratings",
       fill = "Preferred Medium") +
  ggtitle("Graph 4: Ratings Based on Preferred Reflection Medium") +
  scale_fill_manual(values = c("#E69F00", "#56B4E9", "#009E73")) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 16, face = "bold"),
    axis.title = element_text(size = 13),
    axis.text = element_text(size = 11),
    legend.title = element_text(size = 11),
    legend.text = element_text(size = 10)
)
```

Graph 4: Ratings Based on Preferred Reflection Medium



4)

```
library(tidyverse)

d_report <- read_csv('d_report.csv')
```

```
## Rows: 248 Columns: 36
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): condition, conditionForWeek9, hashed_id
## dbl (33): week4-inter_voiceCount, week4-inter_textCount, week4-inter_complet...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
counter <- 0
for(i in 1:nrow(d_report)) {
  if (d_report$conditionForWeek9[i] == "Optional" &&
      d_report$`week9-normal_completed`[i] == 1) {
    counter <- counter + 1
  }
}
rate <- counter/nrow(d_report)*100
rate
```

```
## [1] 83.87097
```