

Challenges

Overview

Unfinished analysis of survey Q9: “How frequently have you encountered the following challenges while working on open-source projects?”

Import packages and utilities

```
project_root <- here::here() # requires that you be somewhere in the
# project directory (not above it)
# packages
suppressMessages(source(file.path(project_root, "scripts/packages.R")))
# functions and objects used across scripts
suppressMessages(source(file.path(project_root, "scripts/utils.R")))
```

Load data

```
data <- load_qualtrics_data("deidentified_no_qual.tsv")
```

Wrangle data

```
challenges <- data %>%
  select(
    starts_with("challenges")
  )

head(challenges)
```

	challenges_1	challenges_2	challenges_3	challenges_4	challenges_5	challenges_6
1	Always	Always	Always	Always	Always	Always
2	Frequently	Occasionally	Occasionally	Occasionally	Occasionally	Rarely
3	Frequently	Always	Occasionally	Always	Occasionally	Frequently
4	Always	Always	Frequently	Occasionally	Frequently	Always
5	Always	Always	Rarely	Occasionally	Frequently	Never
6						

	challenges_7	challenges_8	challenges_9	challenges_10	challenges_11
1	Always	Always	Always	Always	Always
2	Frequently	Occasionally	Frequently	Frequently	Frequently
3	Frequently	Occasionally	Occasionally	Rarely	Rarely
4	Occasionally	Rarely	Rarely	Frequently	Rarely
5	Never	Never	Never	Always	Occasionally
6					

	challenges_12	challenges_13	challenges_14
1	Always	Always	Always
2	Frequently	Frequently	Occasionally
3	Always	Always	Always
4	Occasionally	Frequently	Frequently
5	Occasionally	Rarely	Always
6			

STOP!! Presumably, “challenges_1” corresponds to the first option, “challenges_2” corresponds to the second option, etc. But we still need to check. I am manually comparing the answers in this data frame to those in the Qualtrics interface, which shows the whole response, i.e. “Limited time for writing new code”, not just “challenges_1”. To be extra confident that I am comparing the same rows between the two tables, I am looking at responses associated with a particular email. After this code chunk, I go back to using the data frame that doesn’t contain the emails.

Since this code only needed to be run once, I’ve commented it out.

```
# pii <- load_qualtrics_data("pii.tsv")
# emails <- pii %>%
#   select(starts_with("stay_in_touch_email"))

# t <- cbind(emails, challenges)

# # Next, I run this line repeatedly with different emails,
# # to make sure that this person's response to "challenges_1"
# # matches their response to "Limited time for writing new code", etc.
# subset(t, startsWith(stay_in_touch_email, "PERSON_EMAIL_HERE"))
```

Next, remove empty rows, i.e. rows from respondents who didn't receive this question.

```
nrow(challenges)
```

```
[1] 332
```

```
challenges <- exclude_empty_rows(challenges) # from scripts/utils.R  
nrow(challenges)
```

```
[1] 233
```

```
long_data <- challenges %>%  
  pivot_longer(  
    cols = starts_with("challenges"),  
    names_to = "challenge",  
    values_to = "challenge_level"  
  )  
  
long_data <- long_data %>%  
  mutate(  
    challenge = recode(  
      challenge,  
      "challenges_1" = "Coding time",  
      "challenges_2" = "Documentation time",  
      "challenges_3" = "Managing issues",  
      "challenges_4" = "Attracting users",  
      "challenges_5" = "Recognition",  
      "challenges_6" = "Hiring",  
      "challenges_7" = "Security",  
      "challenges_8" = "Finding peers",  
      "challenges_9" = "Finding mentors",  
      "challenges_10" = "Education time",  
      "challenges_11" = "Educational resources",  
      "challenges_12" = "Legal",  
      "challenges_13" = "Finding funding",  
      "challenges_14" = "Securing funding"  
    )  
  )  
  
long_data <- long_data %>%  
  mutate(  
    
```

```

    challenge_score = recode(
      challenge_level,
      "Never"       = 0L,
      "Non-applicable" = 0L,
      "Rarely"      = 1L,
      "Occasionally" = 2L,
      "Frequently"  = 3L,
      "Always"      = 4L
    )
  )
# Using interger literals 0L, 1L, etc., ensures that
# the new column will be integers, not doubles.
# We can get away with not checking for unmapped values
# because this question was mandatory.

long_data

```

```

# A tibble: 3,262 x 3
  challenge      challenge_level challenge_score
  <chr>          <chr>             <int>
1 Coding time    Always                4
2 Documentation time Always                4
3 Managing issues Always                4
4 Attracting users Always                4
5 Recognition    Always                4
6 Hiring         Always                4
7 Security       Always                4
8 Finding peers  Always                4
9 Finding mentors Always                4
10 Education time Always                4
# i 3,252 more rows

```

```

# Helper to compute the (numeric) mode
get_mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

summary_df <- long_data %>%
  group_by(challenge) %>%
  summarise(

```

```

    total = sum(challenge_score),
    mean  = mean(challenge_score, na.rm = TRUE),
    mode  = get_mode(challenge_score),
    st_dev = sd(challenge_score, na.rm = TRUE)
  ) %>%
  ungroup()

# Order by highest total "score"
summary_df <- summary_df %>%
  arrange(desc(total))

summary_df

```

```

# A tibble: 14 x 5
  challenge      total  mean  mode st_dev
  <chr>      <int> <dbl> <int> <dbl>
1 Documentation time    686  2.94     3  1.08
2 Coding time          606  2.60     3  1.24
3 Education time        539  2.31     3  1.26
4 Managing issues       451  1.94     2  1.29
5 Attracting users      442  1.90     0  1.45
6 Securing funding      438  1.88     0  1.74
7 Finding funding       432  1.85     0  1.68
8 Educational resources  369  1.58     1  1.19
9 Recognition          334  1.43     0  1.35
10 Legal               333  1.43     0  1.24
11 Finding mentors     323  1.39     0  1.31
12 Security            307  1.32     0  1.31
13 Hiring              291  1.25     0  1.53
14 Finding peers       267  1.15     0  1.13

```

Cool! It looks like finding the time for documentation, coding, and self-education are the challenges encountered most frequently. These are the only responses that had a mode of 3 (“Frequently”) and a mean of **greater** than 2 (“Occasionally”).

Out of curiosity, how does it look when we order by variability?

```

sd_df <- summary_df %>%
  arrange(desc(st_dev))

sd_df

```

```
# A tibble: 14 x 5
  challenge      total  mean  mode st_dev
  <chr>      <int> <dbl> <int> <dbl>
1 Securing funding    438  1.88     0  1.74
2 Finding funding    432  1.85     0  1.68
3 Hiring             291  1.25     0  1.53
4 Attracting users   442  1.90     0  1.45
5 Recognition        334  1.43     0  1.35
6 Security           307  1.32     0  1.31
7 Finding mentors    323  1.39     0  1.31
8 Managing issues    451  1.94     2  1.29
9 Education time     539  2.31     3  1.26
10 Legal             333  1.43     0  1.24
11 Coding time       606  2.60     3  1.24
12 Educational resources 369  1.58     1  1.19
13 Finding peers     267  1.15     0  1.13
14 Documentation time 686  2.94     3  1.08
```

Fascinating! The greatest standard deviations are from securing funding, finding funding, and hiring. This makes sense, as these are, at least in my perception, “manager tasks”—tasks that only some people face, but they’re likely to be a big challenge for those who face them. I would guess that these might show a bimodal distribution. Let’s plot them and find out!

Plot the distributions

Prepare data for plotting

```
ordered_levels <- c(
  "Non-applicable",
  "Never",
  "Rarely",
  "Occasionally",
  "Frequently",
  "Always"
)

to_plot <- long_data %>%
  mutate(challenge_level = factor(challenge_level, levels = ordered_levels)) %>%
  count(
    challenge,
    challenge_level,
```

```

    name = "total"
  ) %>%
  ungroup()

to_plot

```

```

# A tibble: 84 x 3
  challenge      challenge_level total
  <chr>          <fct>          <int>
1 Attracting users Non-applicable    50
2 Attracting users Never              15
3 Attracting users Rarely             24
4 Attracting users Occasionally       53
5 Attracting users Frequently         52
6 Attracting users Always             39
7 Coding time      Non-applicable    21
8 Coding time      Never              4
9 Coding time      Rarely             13
10 Coding time     Occasionally       54
# i 74 more rows

```

```

t <- to_plot %>%
  filter(
    challenge == "Documentation time"
  )

basic_bar_chart(
  t,
  x_var = "challenge_level",
  y_var = "total",
  title = "Limited time for writing documentation"
)

```

