

Challenges

Overview

Initial 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")))
```

Define functions

```
multiple_plots <- function(df, title_codes, cols_of_interest) {
  for (ch in cols_of_interest) {
    df_ch <- filter(df, challenge == ch)
    plot_title <- title_codes[[ch]]
    p <- basic_bar_chart(
      df_ch,
      x_var = "challenge_level",
      y_var = "total",
      title = plot_title,
      show_grid = TRUE
    )
  }
}
```

```

    print(p)
  }
}

```

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"))
```

My assumption above was correct; the options are ordered as expected. Let’s rename the columns accordingly.

```
challenge_codes <- c(
  "Coding time" = "challenges_1",
  "Documentation time" = "challenges_2",
  "Managing issues" = "challenges_3",
  "Attracting users" = "challenges_4",
  "Recognition" = "challenges_5",
  "Hiring" = "challenges_6",
  "Security" = "challenges_7",
  "Finding peers" = "challenges_8",
  "Finding mentors" = "challenges_9",
  "Education time" = "challenges_10",
  "Educational resources" = "challenges_11",
  "Legal" = "challenges_12",
  "Finding funding" = "challenges_13",
  "Securing funding" = "challenges_14"
)
challenges <- rename(challenges, challenge_codes)
```

Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.

```
i Please use `all_of()` or `any_of()` instead.
# Was:
data %>% select(challenge_codes)

# Now:
data %>% select(all_of(challenge_codes))
```

See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.

Next, remove empty rows, i.e. rows from respondents who didn't receive this question. As with many questions in this survey, we can cut some corners in the code because the question was mandatory. For example, no need to worry about incomplete answers.

```
nrow(challenges)
```

```
[1] 332
```

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

```
[1] 233
```

Let's reshape the data from wide to long format for easier plotting later.

```
long_data <- challenges %>%
  pivot_longer(
    cols = everything(),
    names_to = "challenge",
    values_to = "challenge_level"
  )

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.

```

```
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

```

Next, let's calculate some simple descriptive statistics. I will choose: * The total "score", that is, the total number of "points" a challenge received ("Never" = 0, "Non-applicable" = 0, "Rarely" = 1, "Occasionally" = 2, "Frequently" = 3, "Always" = 4) * The mean (which might be misleading if 0s drag it down, and also, who's to say what a 2.5 really means? Are the distances between the Likert points equal? We don't know.) * The mode * The standard deviation

```

# 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
```

Create a plot for each “challenge”. After inspecting the plots, I attempted to order them into groups based on the shape of their distribution. These are the shapes I observed (this is extremely subjective):

- * Right-skewed: Documentation time, coding time, education time
- * Interpretation: Common tasks that are frequently challenging
- * Highly bimodal: Securing funding, identifying funding, hiring
- * Interpretation: Tasks that are not as common, but they are often challenging for the people tasked with them.
- * Normal: Educational resources, Legal
- * Interpretation: Moderately common tasks that are challenging with moderate frequency.
- * NA-skewed but otherwise normal: Attracting users, Receiving recognition, finding mentors, managing security risks, managing issues
- * Interpretation: Less-common tasks that are challenging with moderate frequency.
- * Left-skewed: Finding peers
- * Interpretation: Moderately common tasks that are infrequently challenging.

```
titles <- list(
  "Coding time" = "Limited time for writing new code",
  "Documentation time" = "Limited time for writing documentation",
  "Managing issues" = "Managing issues and pull requests",
  "Attracting users" = "Attracting users and/or contributors",
  "Recognition" = "Receiving recognition for my contributions",
  "Hiring" = "Finding and hiring qualified personnel",
  "Security" = "Managing security risks",
  "Finding peers" = "Finding a community of peers who share my interests",
```



```

    "Finding mentors" = "Finding mentors",
    "Education time" = "Finding time to educate myself",
    "Educational resources" = "Identifying helpful educational resources",
    "Legal" = "Navigating licensing and other legal issues",
    "Finding funding" = "Identifying potential funding sources\nfor my open source projects"
    "Securing funding" = "Securing funding for my open source projects"
)

right_skewed <- c(
  "Coding time",
  "Documentation time",
  "Education time"
)

bimodal <- c(
  "Finding funding",
  "Securing funding",
  "Hiring"
)

normal <- c(
  "Educational resources",
  "Legal"
)

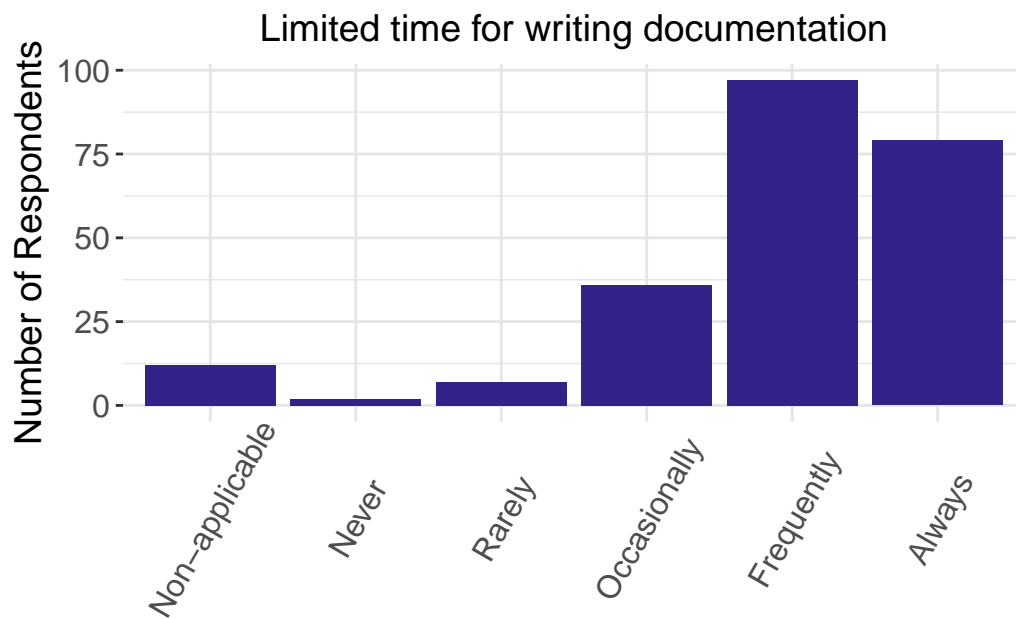
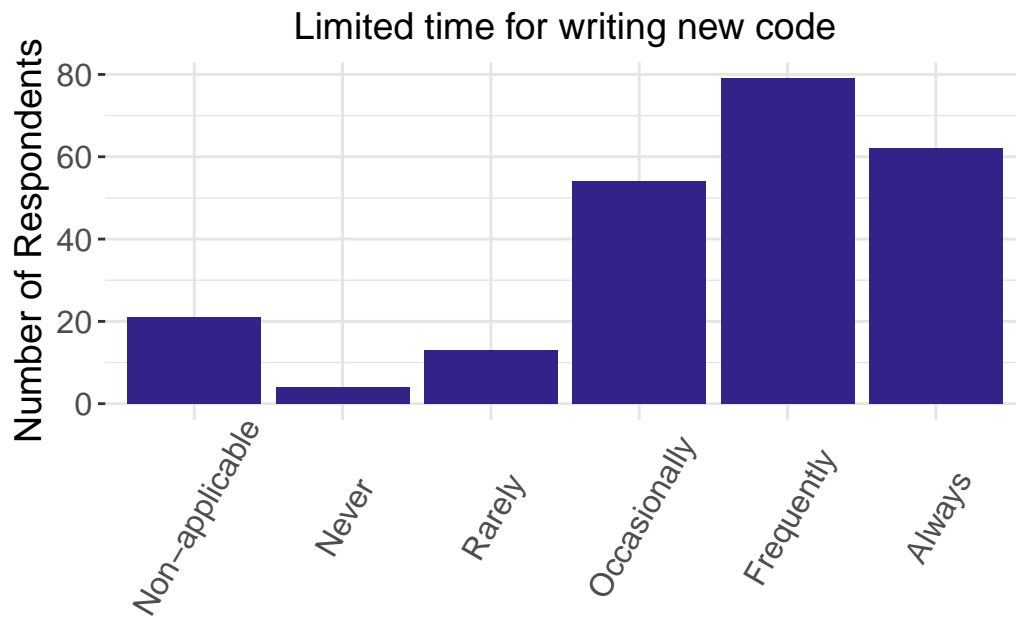
na_skewed <- c(
  "Managing issues",
  "Attracting users",
  "Recognition",
  "Security",
  "Finding mentors"
)

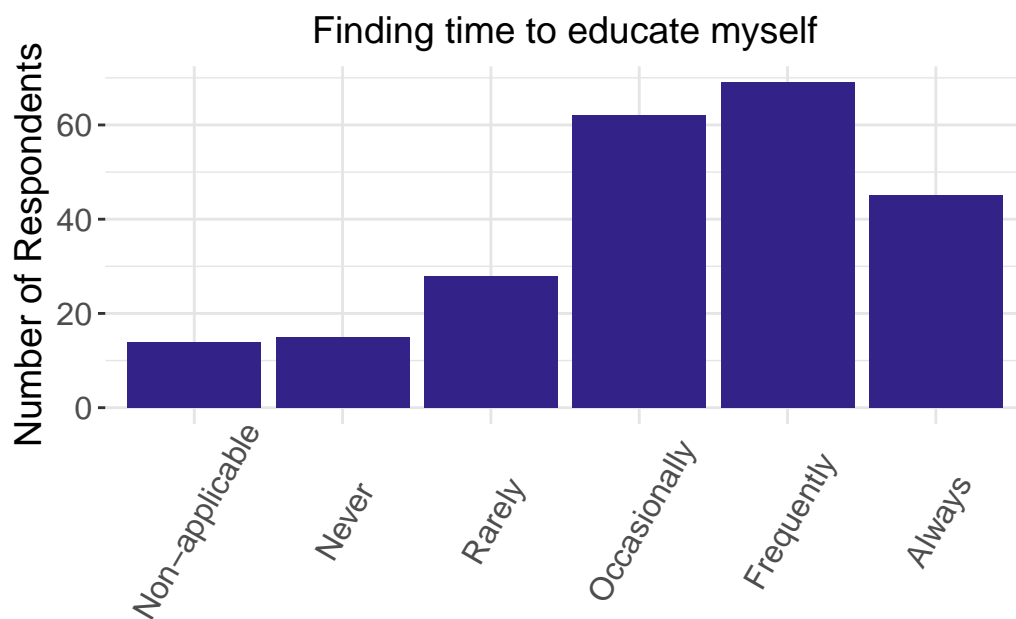
left_skewed <- c(
  "Finding peers"
)

```

“right-skewed”

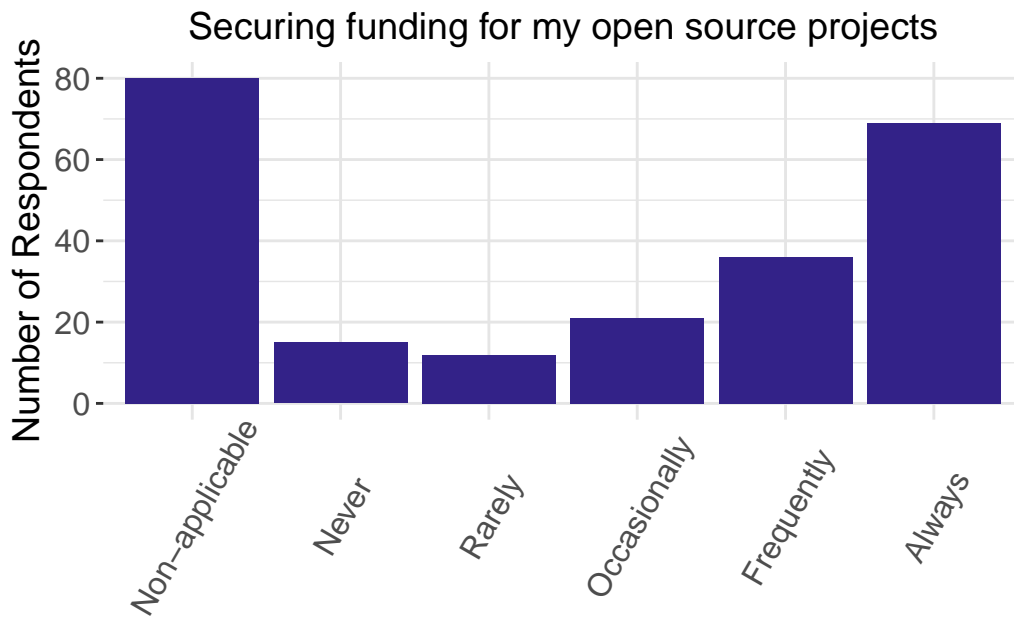
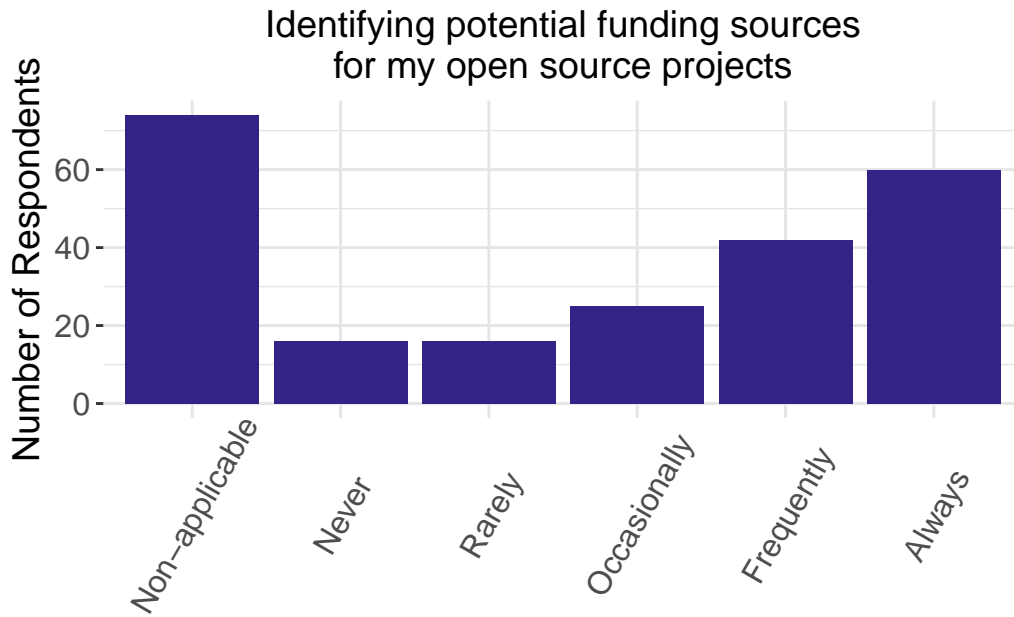
```
multiple_plots(to_plot, titles, right_skewed)
```





“highly bimodal”

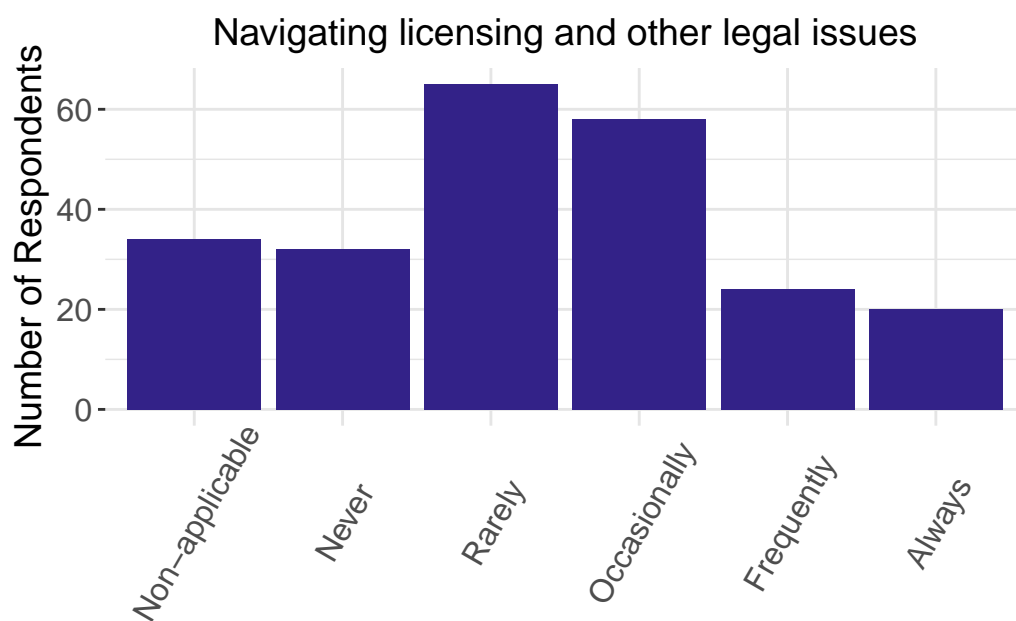
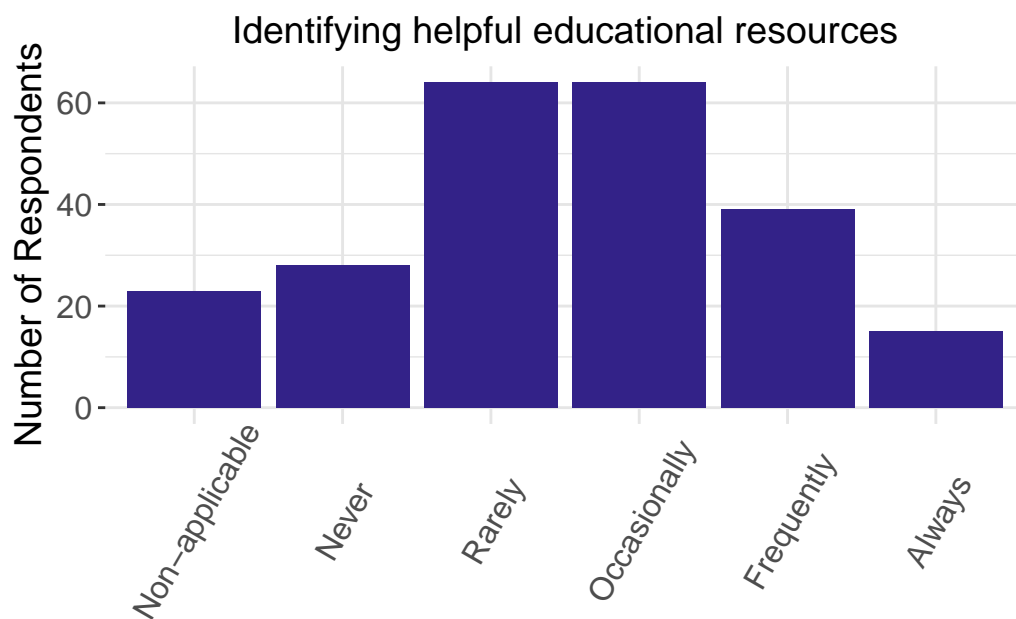
```
multiple_plots(to_plot, titles, bimodal)
```





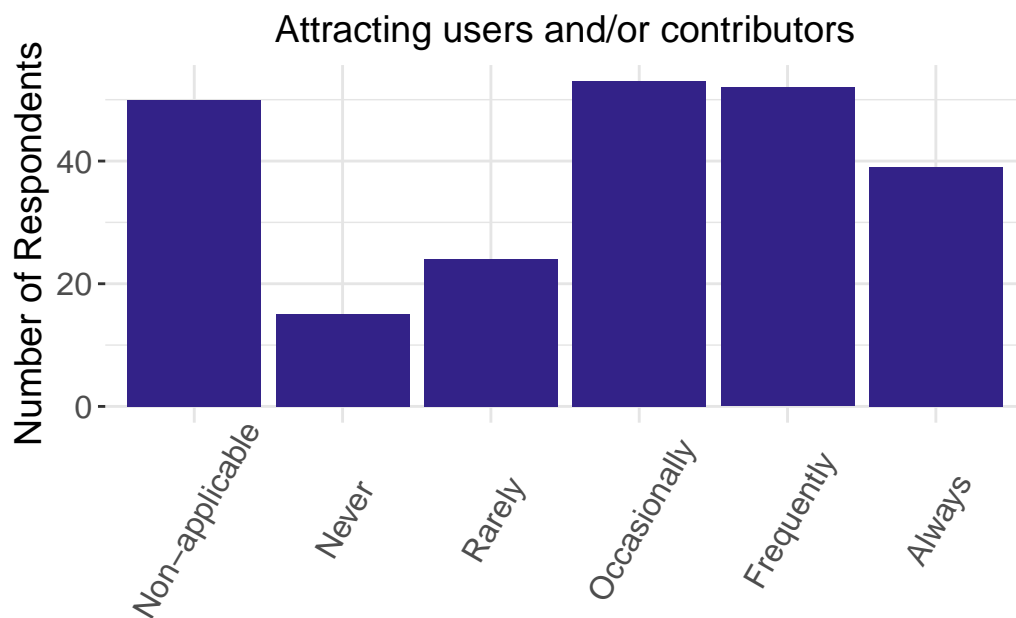
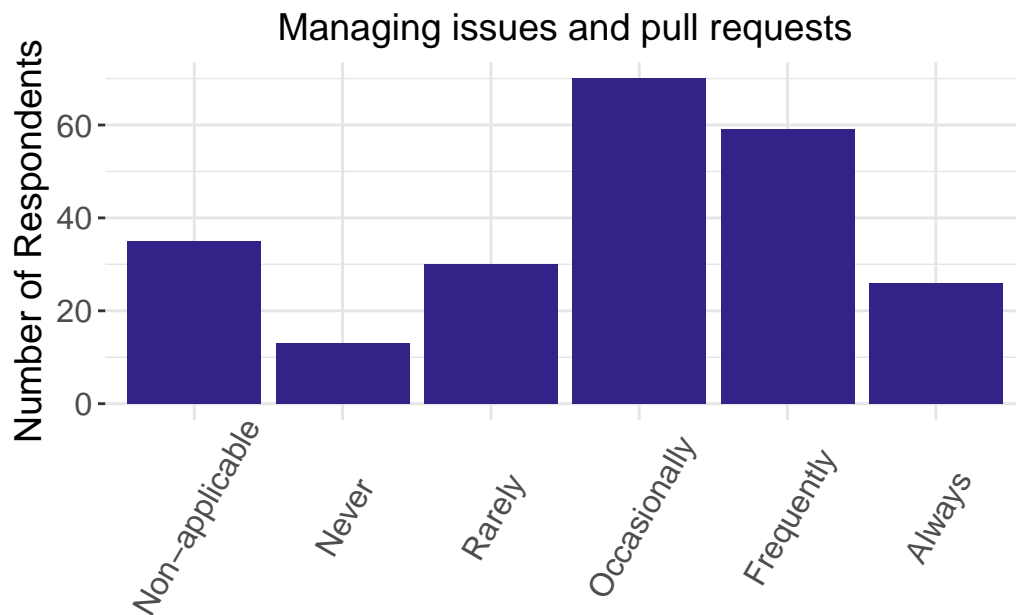
“normal”

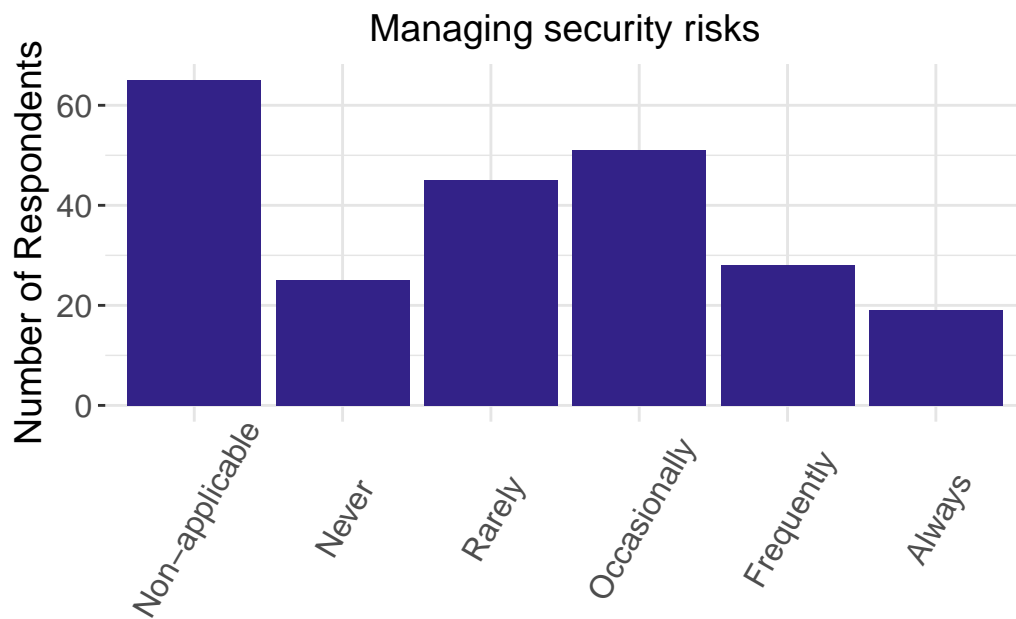
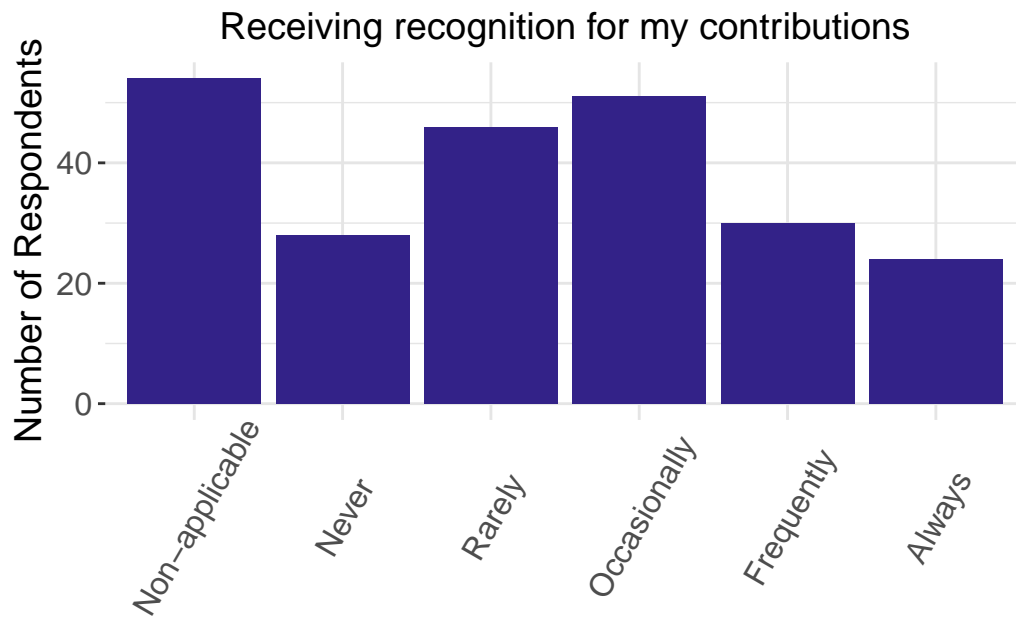
```
multiple_plots(to_plot, titles, normal)
```

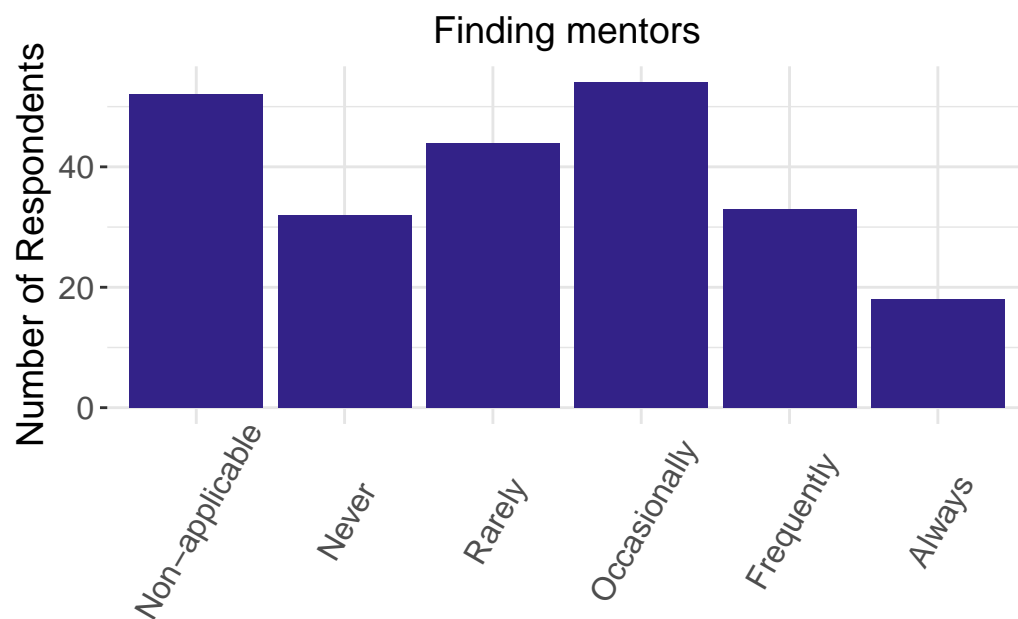


“na-skewed”

```
multiple_plots(to_plot, titles, na_skewed)
```

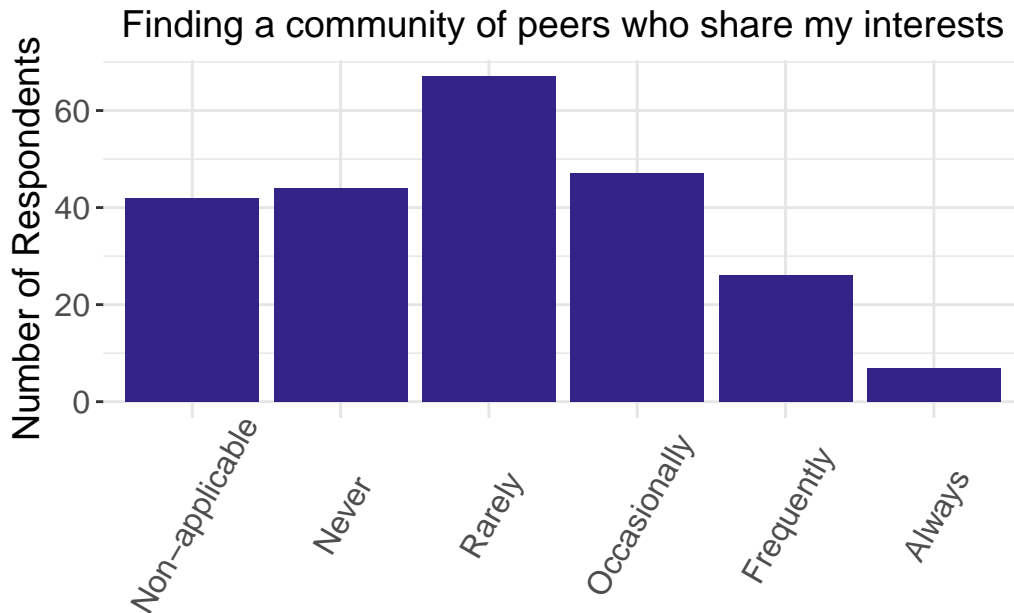






“left-skewed”

```
multiple_plots(to_plot, titles, left_skewed)
```



Next, I should probably divide these groups up using some more rigorous method. Maybe clustering the different challenges (e.g. k-means).

Dimensionality reduction (Shelved, for now)

In the above sections, I was looking at each challenge as its own data point, and effectively clustering challenges. Here, I started regarding each person as a data point, and clustering people. But this didn't seem too promising, and I don't know if I care enough to pursue it. NOTE THE DIFFERENT CODING SCHEME

```
challnumeric <- challenges %>%  
  mutate(  
    across(  
      everything(),  
      ~ recode(  
        .x,  
        "Never" = 0L,  
        "Non-applicable" = -1L, # THIS IS DIFFERENT (-1, not 0)  
        "Rarely" = 1L,  
        "Occasionally" = 2L,  
        "Frequently" = 3L,  
        "Always" = 4L
```

Importance of components:

```
set.seed(42)
km <- kmeans(scale(challnumeric), centers = 5)
challnumeric$cluster_km <- km$cluster
```