Challenges + role

Overview

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

In this script, I am considering challenges in light of role, focusing on maintainers. Basically, I want to confirm/refute my suspicion that the people who selected "managing issues" and "attracting users" are largely maintainers, and these are common challenges among maintainers.

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

```
challenges_raw <- load_qualtrics_data("clean_data/challenges_Q9.tsv")
roles_raw <- load_qualtrics_data("clean_data/contributor_roles_Q4.tsv")
```

Wrangle data

```
roles_and_chall <- cbind(roles_raw, challenges_raw)
```

Remove rows that contain any empty entries.

```
nrow(roles_and_chall)
```

```
[1] 332
```

```
roles_and_chall <- exclude_empty_rows(roles_and_chall, strict = TRUE) # from scripts/utils.R
nrow(roles_and_chall)
```

```
[1] 233
```

Double-check that none of the rows sum to 0 for the roles columns, which would indicate that the participant didn't answer the question.

```
roles_vec <- names(roles_raw)
roles_and_chall %>%
    select(all_of(roles_vec)) %>%
    filter(rowSums(across(where(is.numeric))) == 0)
```

```
 [1] Maintainer                 Contributor                Bug/Issue Reporter
 [4] Community Manager          Educator                   Other
 [7] Supervisor                 IT/Systems administrator UI/UX Designer
[10] Technical support
<0 rows> (or 0-length row.names)
```

Let's reshape these data frames to long data.

```
maintainers <- roles_and_chall %>%
  filter(Maintainer == 1) %>%
  select(-one_of(roles_vec))
non_maintainers <- roles_and_chall %>%
  filter(Maintainer == 0) %>%
  select(-one_of(roles_vec))

maintainers_long <- maintainers %>%
  pivot_longer(
    cols = everything(),
    names_to = "challenge",
    values_to = "frequency"
  )

non_maintainers_long <- non_maintainers %>%
  pivot_longer(
    cols = everything(),
    names_to = "challenge",
    values_to = "frequency"
  )
```

Exploratory stats

Maintainers

First, let's look at the maintainers' top challenges, using a coded "points" system.

```
maintainers_long <- maintainers_long %>%
  mutate(
    score = recode(
      frequency,
      "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.
```

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

summary_df_maint <- maintainers_long %>%
  group_by(challenge) %>%
  summarise(
    total  = sum(score),
    mean   = mean(score, na.rm = TRUE),
    median = median(score),
    mode   = get_mode(score),
    st_dev = sd(score, na.rm = TRUE)
  ) %>%
  ungroup()

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

summary_df_maint
```

```
# A tibble: 14 x 6
   challenge             total  mean median  mode st_dev
   <chr>                 <int> <dbl>  <dbl> <int>  <dbl>
 1 Documentation time      422  3.15      3     3  0.914
 2 Coding time             395  2.95      3     4  0.960
 3 Education time          315  2.35      2     3  1.22
 4 Attracting users        312  2.33      2     2  1.34
 5 Managing issues         312  2.33      2     3  1.12
 6 Securing funding        294  2.19      3     4  1.72
 7 Finding funding         281  2.10      2     4  1.64
 8 Recognition             222  1.66      2     0  1.34
 9 Educational resources   210  1.57      1     1  1.19
10 Hiring                  205  1.53      1     0  1.61
11 Security                201  1.5       1     0  1.30
12 Finding mentors         195  1.46      1     0  1.34
13 Legal                   195  1.46      1     1  1.19
14 Finding peers           184  1.37      1     1  1.15
```

What percent of maintainers selected “frequently” or “always” for these two issues?

```
maint_counts <- maintainers_long %>%
  count(challenge, frequency, name = "n")

maint_counts <- maint_counts %>%
  group_by(challenge) %>%
  mutate(perc_total = round(100 * n / sum(n), 1)) %>%
  ungroup()

# Total number of maintainers
nrow(maintainers)
```

```
[1] 134
```

```
maint_counts
```

```
# A tibble: 82 x 4
   challenge        frequency          n perc_total
   <chr>            <chr>          <int>      <dbl>
 1 Attracting users Always            31       23.1
 2 Attracting users Frequently        35       26.1
 3 Attracting users Never             10        7.5
 4 Attracting users Non-applicable    10        7.5
 5 Attracting users Occasionally      35       26.1
 6 Attracting users Rarely            13        9.7
 7 Coding time      Always            46       34.3
 8 Coding time      Frequently        45       33.6
 9 Coding time      Non-applicable     2        1.5
10 Coding time      Occasionally      35       26.1
# i 72 more rows
```

```
maint_concise <- maintainers_long %>%
    select(-score) %>%
    mutate(
        frequency = recode(
            frequency,
            "Always" = "Always or Frequently",
            "Frequently" = "Always or Frequently",
            "Rarely" = "Never or Rarely",
            "Never" = "Never or Rarely"
        )
        )

maint_concise <- maint_concise %>%
  count(challenge, frequency, name = "n")

maint_concise <- maint_concise %>%
  group_by(challenge) %>%
  mutate(perc_total = round(100 * n / sum(n), 1)) %>%
  ungroup()

# Total number of maintainers
nrow(maintainers)
```

```
[1] 134
```

```
maint_concise
```

```
# A tibble: 56 x 4
   challenge          frequency                n perc_total
   <chr>              <chr>                <int>      <dbl>
 1 Attracting users   Always or Frequently    66       49.3
 2 Attracting users   Never or Rarely         23       17.2
 3 Attracting users   Non-applicable          10        7.5
 4 Attracting users   Occasionally            35       26.1
 5 Coding time        Always or Frequently    91       67.9
 6 Coding time        Never or Rarely          6        4.5
 7 Coding time        Non-applicable           2        1.5
 8 Coding time        Occasionally            35       26.1
 9 Documentation time Always or Frequently   110       82.1
10 Documentation time Never or Rarely          4        3
# i 46 more rows
```

Non-maintainers

Let's look at the same data for non-maintainers.

```
non_maintainers_long <- non_maintainers_long %>%
  mutate(
    score = recode(
      frequency,
      "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.
```

```
summary_df_non_maint <- non_maintainers_long %>%
  group_by(challenge) %>%
  summarise(
    total  = sum(score),
    mean   = mean(score, na.rm = TRUE),
    median = median(score),
    mode   = get_mode(score),
    st_dev = sd(score, na.rm = TRUE)
  ) %>%
  ungroup()

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

summary_df_non_maint
```

```
# A tibble: 14 x 6
   challenge              total  mean median  mode st_dev
   <chr>                  <int> <dbl>  <int> <int>  <dbl>
 1 Documentation time       264 2.67       3     3   1.21
 2 Education time           224 2.26       2     3   1.33
 3 Coding time              211 2.13       3     3   1.41
 4 Educational resources    159 1.61       2     2   1.19
 5 Finding funding          151 1.53       0     0   1.67
 6 Securing funding         144 1.45       0     0   1.67
 7 Managing issues          139 1.40       2     0   1.32
 8 Legal                    138 1.39       1     0   1.31
 9 Attracting users         130 1.31       1     0   1.40
10 Finding mentors          128 1.29       1     0   1.27
11 Recognition              112 1.13       1     0   1.32
12 Security                 106 1.07       0     0   1.30
13 Hiring                    86 0.869      0     0   1.34
14 Finding peers             83 0.838      0     0   1.02
```

As we would expect, "Managing issues" and "Attracting users" are not as high on the list as they were for maintainers. (Actually, in my opinion, they're still surprisingly high.)

```r
non_maint_counts <- non_maintainers_long %>%
  count(challenge, frequency, name = "n")

non_maint_counts <- non_maint_counts %>%
  group_by(challenge) %>%
  mutate(perc_total = round(100 * n / sum(n), 1)) %>%
  ungroup()

# Total number of non-maintainers
nrow(non_maintainers)
```

```
[1] 99
```

```r
non_maint_counts
```

```
# A tibble: 84 x 4
   challenge        frequency          n perc_total
   <chr>            <chr>          <int>      <dbl>
 1 Attracting users Always             8        8.1
 2 Attracting users Frequently        17       17.2
 3 Attracting users Never              5        5.1
 4 Attracting users Non-applicable    40       40.4
 5 Attracting users Occasionally      18       18.2
 6 Attracting users Rarely            11       11.1
 7 Coding time      Always            16       16.2
 8 Coding time      Frequently        34       34.3
 9 Coding time      Never              4        4  
10 Coding time      Non-applicable    19       19.2
# i 74 more rows
```

```r
non_maint_concise <- non_maintainers_long %>%
    select(-score) %>%
    mutate(
        frequency = recode(
            frequency,
            "Always" = "Always or Frequently",
            "Frequently" = "Always or Frequently",
            "Rarely" = "Never or Rarely",
            "Never" = "Never or Rarely"
        )
        )
```

```r
non_maint_concise <- non_maint_concise %>%
  count(challenge, frequency, name = "n")

non_maint_concise <- non_maint_concise %>%
  group_by(challenge) %>%
  mutate(perc_total = round(100 * n / sum(n), 1)) %>%
  ungroup()

non_maint_concise
```

```
# A tibble: 56 x 4
   challenge          frequency                 n perc_total
   <chr>              <chr>                 <int>      <dbl>
 1 Attracting users   Always or Frequently     25       25.3
 2 Attracting users   Never or Rarely          16       16.2
 3 Attracting users   Non-applicable           40       40.4
 4 Attracting users   Occasionally             18       18.2
 5 Coding time        Always or Frequently     50       50.5
 6 Coding time        Never or Rarely          11       11.1
 7 Coding time        Non-applicable           19       19.2
 8 Coding time        Occasionally             19       19.2
 9 Documentation time Always or Frequently     66       66.7
10 Documentation time Never or Rarely           5        5.1
# i 46 more rows
```

Write results to file

Let’s combine these results into one pretty data table.

```r
out <- bind_rows(
  maint_concise %>% mutate(group = "Maintainers"),
  non_maint_concise %>% mutate(group = "Non-maintainers")
) %>%
  # Capitalize titles
  rename(Challenge = challenge, Frequency = frequency) %>%
  pivot_wider(
    id_cols = c(Challenge, Frequency),
    names_from = group,
    values_from = c(n, perc_total),
    values_fill = list(n = 0L, perc_total = 0),
    # Lets you format the new column names with a Glue string
    # e.g. n + Maintainers = n Maintainers
    names_glue = "{.value} {group}"
  ) %>%
  # Prettify
  rename(
    `N Maintainers` = `n Maintainers`,
    `N Non-maintainers` = `n Non-maintainers`,
    `Percent of Maintainers` = `perc_total Maintainers`,
    `Percent of Non-Maintainers` = `perc_total Non-maintainers`
  ) %>%
  # Re-order factor levels
  mutate(
    Frequency = factor(
      Frequency,
      levels = c(
        "Always or Frequently",
        "Occasionally",
        "Never or Rarely",
        "Non-applicable"
      )
    )
  ) %>%
  arrange(Challenge, Frequency)

# Sanity check: should be 0 rows
# Keeps rows in x (e.g. maint_concise)
# that do not have a match in y (e.g. non_maint_concise)
# on the keys in by
anti_join(maint_concise, non_maint_concise, by = c("challenge", "frequency"))
```

```
# A tibble: 0 x 4
# i 4 variables: challenge <chr>, frequency <chr>, n <int>, perc_total <dbl>
```

```
anti_join(non_maint_concise, maint_concise, by = c("challenge", "frequency"))
```

```
# A tibble: 0 x 4
# i 4 variables: challenge <chr>, frequency <chr>, n <int>, perc_total <dbl>
```

```r
out
```

```
# A tibble: 56 x 6
   Challenge          Frequency            `N Maintainers` `N Non-maintainers`
   <chr>              <fct>                          <int>               <int>
 1 Attracting users   Always or Frequently              66                  25
 2 Attracting users   Occasionally                      35                  18
 3 Attracting users   Never or Rarely                   23                  16
 4 Attracting users   Non-applicable                    10                  40
 5 Coding time        Always or Frequently              91                  50
 6 Coding time        Occasionally                      35                  19
 7 Coding time        Never or Rarely                    6                  11
 8 Coding time        Non-applicable                     2                  19
 9 Documentation time Always or Frequently             110                  66
10 Documentation time Occasionally                      17                  19
# i 46 more rows
# i 2 more variables: `Percent of Maintainers` <dbl>,
#   `Percent of Non-Maintainers` <dbl>
```

```r
write_df_to_file(out, "supplementary_tables/maintainer_challenges.tsv")
```

```r
sessionInfo()
```

```
R version 4.4.2 (2024-10-31)
Platform: aarch64-apple-darwin20
Running under: macOS 26.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;  L

locale:
[1] C.UTF-8/C.UTF-8/C.UTF-8/C/C.UTF-8/C.UTF-8

time zone: America/Los_Angeles
tzcode source: internal

attached base packages:
[1] tools     grid      stats     graphics  grDevices datasets  utils
[8] methods   base
```

```
other attached packages:
 [1] treemapify_2.5.6      tidyr_1.3.1           svglite_2.2.1
 [4] stringr_1.5.1         scales_1.4.0          readr_2.1.5
 [7] pwr_1.3-0             patchwork_1.3.2       ordinal_2023.12-4.1
[10] lme4_1.1-37           Matrix_1.7-1          languageserver_0.3.16
[13] here_1.0.1            gtools_3.9.5          ggforce_0.5.0
[16] FSA_0.10.0            fpc_2.2-13            forcats_1.0.0
[19] factoextra_1.0.7      ggplot2_3.5.2         emmeans_1.11.2
[22] dplyr_1.1.4           corrplot_0.95         ComplexHeatmap_2.22.0
[25] cluster_2.1.8.1       BiocManager_1.30.26

loaded via a namespace (and not attached):
 [1] Rdpack_2.6.4        rlang_1.1.6         magrittr_2.0.3
 [4] clue_0.3-66         GetoptLong_1.0.5    matrixStats_1.5.0
 [7] compiler_4.4.2      flexmix_2.3-20      systemfonts_1.2.3
[10] png_0.1-8           callr_3.7.6         vctrs_0.6.5
[13] pkgconfig_2.0.3     shape_1.4.6.1       crayon_1.5.3
[16] fastmap_1.2.0       utf8_1.2.6          rmarkdown_2.29
[19] ggfittext_0.10.2    tzdb_0.5.0          ps_1.9.1
[22] nloptr_2.2.1        purrr_1.1.0         xfun_0.53
[25] modeltools_0.2-24   jsonlite_2.0.0      tweenr_2.0.3
[28] parallel_4.4.2      prabclus_2.3-4      R6_2.6.1
[31] stringi_1.8.7       RColorBrewer_1.1-3  boot_1.3-31
[34] diptest_0.77-2      numDeriv_2016.8-1.1 estimability_1.5.1
[37] Rcpp_1.1.0          iterators_1.0.14    knitr_1.50
[40] IRanges_2.40.1      splines_4.4.2       nnet_7.3-19
[43] tidyselect_1.2.1    yaml_2.3.10         doParallel_1.0.17
[46] codetools_0.2-20    processx_3.8.6      lattice_0.22-6
[49] tibble_3.3.0        withr_3.0.2         evaluate_1.0.4
[52] polyclip_1.10-7     xml2_1.4.0          circlize_0.4.16
[55] mclust_6.1.1        kernlab_0.9-33      pillar_1.11.0
[58] renv_1.1.5          foreach_1.5.2       stats4_4.4.2
[61] reformulas_0.4.1    generics_0.1.4      rprojroot_2.1.1
[64] S4Vectors_0.44.0    hms_1.1.3           minqa_1.2.8
[67] xtable_1.8-4        class_7.3-22        glue_1.8.0
[70] robustbase_0.99-4-1 mvtnorm_1.3-3       rbibutils_2.3
[73] colorspace_2.1-1    nlme_3.1-166        cli_3.6.5
[76] textshaping_1.0.1   gtable_0.3.6        DEoptimR_1.1-4
[79] digest_0.6.37       BiocGenerics_0.52.0 ucminf_1.2.2
[82] ggrepel_0.9.6       rjson_0.2.23        farver_2.1.2
[85] htmltools_0.5.8.1   lifecycle_1.0.4     GlobalOptions_0.1.2
[88] MASS_7.3-61
```