Contributor roles

Overview

This script explores Q4: "Which of these open source contributor roles has ever applied to you?".

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

```
get_mode <- function(v) {
  # Get unique values
  uniques <- unique(v)

  # Tabulate the frequencies of unique values
  freqs <- tabulate(match(v, uniques))

  # Find the value(s) with the maximum frequency
  modes <- uniques[freqs == max(freqs)]
```

```
  return(modes)
}
```

Load data

```
roles_raw <- load_qualtrics_data("clean_data/contributor_roles_Q4.tsv")
other_quant <- load_qualtrics_data("clean_data/other_quant.tsv")
head(roles_raw)
```

```
  Maintainer Contributor Bug/Issue Reporter Community Manager Educator Other
1          1           1                  1                  1        1     0
2          0           1                  0                  0        0     0
3          1           1                  1                  0        1     0
4          1           1                  1                  0        1     0
5          1           1                  1                  0        1     0
6          0           0                  0                  0        0     0
  Supervisor IT/Systems administrator UI/UX Designer Technical support
1          1                        0              0                  1
2          0                        0              0                  0
3          0                        0              1                  0
4          1                        0              0                  0
5          1                        0              0                  0
6          0                        0              0                  0
```

```
nrow(roles_raw)
```

```
[1] 332
```

Wrangle data

Drop rows with all 0s (NAs were already converted to 0s during data cleanup).

```
roles <- filter_all(roles_raw, any_vars(. != 0))
nrow(roles)
```

```
[1] 233
```

Explore data

First, some descriptive statistics on the overall number of roles per person.

```
roles_by_person <- rowSums(roles)
# This should be false
any(roles_by_person == 0)
```

```
[1] FALSE
```

```
mean(roles_by_person)
```

```
[1] 4.283262
```

```
median(roles_by_person)
```

```
[1] 4
```

```
get_mode(roles_by_person)
```

```
[1] 4
```

```
counts <- colSums(roles)
counts <- data.frame(Role = names(counts), Count = as.integer(counts), row.names = NULL)
counts <- counts %>%
    arrange(desc(Count))

counts
```

```
                      Role Count
1        Bug/Issue Reporter   189
2               Contributor   187
3                  Educator   138
4                Maintainer   134
5         Technical support   102
6                Supervisor    96
7         Community Manager    49
8            UI/UX Designer    46
9  IT/Systems administrator    44
10                    Other    13
```

Save a file for supplement

```
write_df_to_file(counts, "supplementary_tables/role_counts.tsv")
```

That's mildly interesting. There's about 2.7-3 maintainers for every community manager, sys admin, or UI designer (keeping in mind that people might be both). I'm curious how many people are EXCLUSIVELY one thing. Mainly interested in how many people are EXCLUSIVELY a maintainer, contributor, bug reporter, or supervisor.

```
indices <- which(rowSums(roles) == 1 )
exclusive <- roles[indices,]
nrow(exclusive)
```

```
[1] 16
```

Let's repeat what we did for the whole data frame, but now with just people who filled one role.

```
counts_exc <- colSums(exclusive)
counts_exc <- data.frame(Role = names(counts_exc), Count = as.integer(counts_exc), row.names
counts_exc <- counts_exc %>%
    arrange(desc(Count))

counts_exc
```

```
                      Role Count
1        Bug/Issue Reporter     5
2               Contributor     4
3                Supervisor     3
4                  Educator     2
5                Maintainer     1
6         Community Manager     1
7                     Other     0
8  IT/Systems administrator     0
9            UI/UX Designer     0
10        Technical support     0
```

So, out of 233 contributors, only 16 identified with exactly one contributor role. 5 of those were Bug/Issue Reporter, the most common exclusive-role.

```r
# Calculate total roles for each participant
roles2 <- cbind(roles, total_roles = rowSums(roles))

# Get mean number of roles per column (role)
get_mean_num_roles <- function(df, colnum) {
    # drop rows where the entry in this col is 0
    filtered <- df[df[[colnum]] != 0, ]
    return(
        mean(filtered$total_roles)
    )
}

# Get median number of roles per column (role)
get_median_num_roles <- function(df, colnum) {
    # drop rows where the entry in this col is 0
    filtered <- df[df[[colnum]] != 0, ]
    return(
        median(filtered$total_roles)
    )
}

# Get mode number of roles per column (role)
get_mode_num_roles <- function(df, colnum) {
    # drop rows where the entry in this col is 0
    filtered <- df[df[[colnum]] != 0, ]
    # Some roles have multiple modes.
    # Make this a character col, and group the multiple
    # modes into a single string. This is awful for math,
    # but we're just printing it for the supplement.
    return(
      paste(get_mode(filtered$total_roles), collapse = ", ")
    )
}

num_roles <- data.frame(
  role = names(roles),
  mean_num_roles_per_participant = sapply(
    seq(ncol(roles)),
    function(x) get_mean_num_roles(roles2, x)
  ),
  median_num_roles_per_participant = sapply(
    seq(ncol(roles)),
```

```
    function(x) get_median_num_roles(roles2, x)
  ),
  mode_num_roles_per_participant = sapply(
    seq(ncol(roles)),
    function(x) get_mode_num_roles(roles2, x)
  )
)

num_roles <- num_roles[
  order(
    num_roles$mean_num_roles_per_participant,
    decreasing = TRUE
  ),
]

num_roles
```

```
                      role mean_num_roles_per_participant
8  IT/Systems administrator                      6.386364
6                    Other                      6.384615
9            UI/UX Designer                      6.369565
4         Community Manager                      6.163265
10        Technical support                      5.715686
7                Supervisor                      5.583333
5                  Educator                      5.195652
1                Maintainer                      5.194030
2               Contributor                      4.737968
3        Bug/Issue Reporter                      4.682540
   median_num_roles_per_participant mode_num_roles_per_participant
8                                 7                              8
6                                 7                              8
9                                 6                           6, 7
4                                 6                              5
10                                6                              6
7                                 5                              5
5                                 5                              5
1                                 5                              4
2                                 4                              4
3                                 4                              4
```

Save a file for supplement

```
write_df_to_file(num_roles, "supplementary_tables/roles_per_person.tsv")
```

Huh. So I guess no one role is particularly enriched for people who hold many roles. (I mean, we can't really get statistical significance with just one observation, and this doesn't feel worth modeling.) I guess bug reporters and contributors have the fewest roles, by a small margin, which makes sense.

Three-way comparison of maintainers, contributors, and bug reporters

I'd be interested in the 3-way venn diagram of contributors, but reporters and maintainers. (Though, if we do plot this, an UpSet plot would probably be better than a Venn diagram)

```
# maintainers, contributors, and bug reporters in that order
three_way_counts <- with(roles, {
  M <- Maintainer == 1
  C <- Contributor == 1
  B <- `Bug/Issue Reporter` == 1

  c(
    "M, !C, !B"   = sum(M & !C & !B),
    "M, C, !B"  = sum(M & C & !B),
    "M, !C, B"  = sum(M & !C & B),
    "M, C, B" = sum(M & C & B),
    "!M, !C, B"   = sum(!M & !C & B),
    "!M, C, B"  = sum(!M & C & B),
    "!M, C, !B"   = sum(!M & C & !B),
    "!M, !C, !B" = sum(!M & !C & !B)
  )
})

three_way_counts
```

```
 M, !C, !B   M, C, !B   M, !C, B    M, C, B  !M, !C, B   !M, C, B  !M, C, !B
         5         14          4        111         27         47         15
!M, !C, !B
        10
```

```r
# This should be 233, the number of experienced contributors
sum(three_way_counts)
```

```
[1] 233
```

```r
# Number of people who contributed as at least one of these three roles
at_least_one <- sum(three_way_counts)-three_way_counts["!M, !C, !B"]
at_least_one
```

```
!M, !C, !B
       223
```

```r
# Percent of total who identified as at least one of the three
round(three_way_counts/at_least_one*100, digits=1)
```

```
 M, !C, !B   M, C, !B   M, !C, B    M, C, B  !M, !C, B   !M, C, B  !M, C, !B
       2.2        6.3        1.8       49.8       12.1       21.1        6.7
!M, !C, !B
       4.5
```

Interesting. So 96% of the survey respondents identified as at least one of Maintainer, Contributor, or Bug Reporter (223/233). 50.0% of those folks identified as all three (111/223). 21.1% identified as a contributor and bug reporter, but not a maintainer, and 12.1% identified as a bug reporter only. Together, these three groups make up 50+21.1+12.2 83.3% of the total.

UpSet plot

```r
# turn named vector into a table of membership + counts
comb_tbl <- tibble::enframe(three_way_counts, name = "pattern", value = "n") %>%
  mutate(
    M = !str_detect(pattern, "!M"),
    C = !str_detect(pattern, "!C"),
    B = !str_detect(pattern, "!B")
  ) %>%
  select(M, C, B, n)

comb_tbl
```

```
# A tibble: 8 x 4
  M     C     B         n
  <lgl> <lgl> <lgl> <int>
1 TRUE  FALSE FALSE     5
2 TRUE  TRUE  FALSE    14
3 TRUE  FALSE TRUE      4
4 TRUE  TRUE  TRUE    111
5 FALSE FALSE TRUE     27
6 FALSE TRUE  TRUE     47
7 FALSE TRUE  FALSE    15
8 FALSE FALSE FALSE    10
```

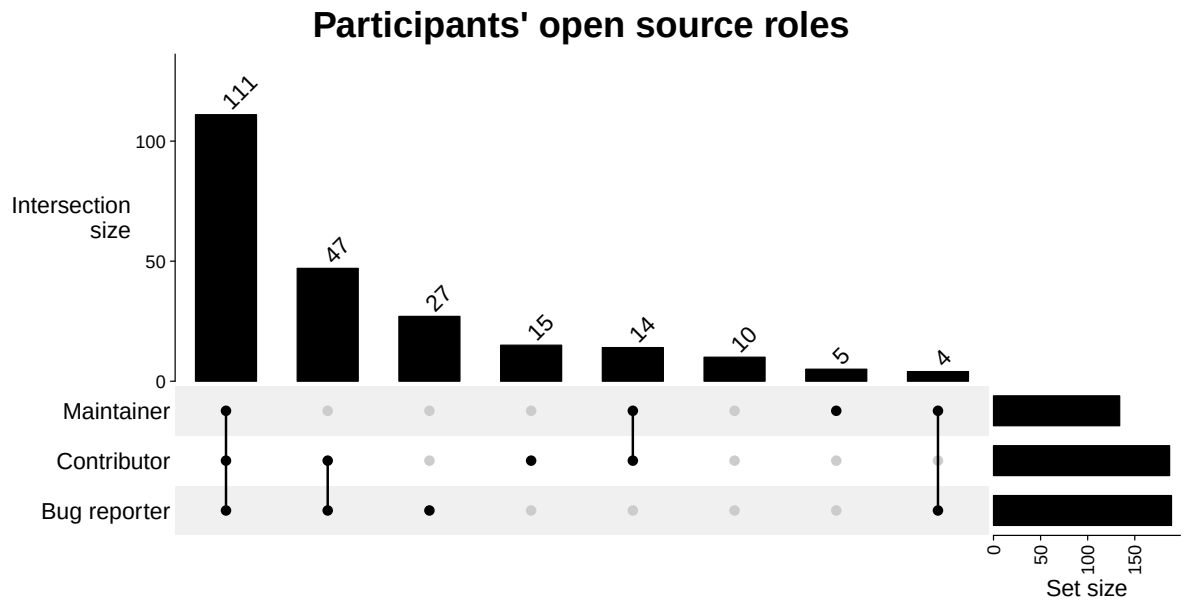
```r
# expand rows by counts (one row per element)
elem_tbl <- comb_tbl %>% uncount(n)

names(elem_tbl) <- c(
  "Maintainer",
  "Contributor",
  "Bug reporter"
)
elem_tbl
```

```
# A tibble: 233 x 3
   Maintainer Contributor `Bug reporter`
   <lgl>      <lgl>       <lgl>
 1 TRUE       FALSE       FALSE
 2 TRUE       FALSE       FALSE
 3 TRUE       FALSE       FALSE
 4 TRUE       FALSE       FALSE
 5 TRUE       FALSE       FALSE
 6 TRUE       TRUE        FALSE
 7 TRUE       TRUE        FALSE
 8 TRUE       TRUE        FALSE
 9 TRUE       TRUE        FALSE
10 TRUE       TRUE        FALSE
# i 223 more rows
```

```r
m <- make_comb_mat(
  elem_tbl[, c("Maintainer","Contributor","Bug reporter")],
  mode = "distinct"
  )
# distinct is the default and indicates that the combinations are
```

```r
# mutually exclusive; each data point appears only once in the data set

ordered_combs <- order(comb_size(m), decreasing = TRUE)
```

```r
# Since this function call has a lot of args, I'm breaking it up into chunks

# Control bars at top of plot
top_ann_args <- ComplexHeatmap::upset_top_annotation(
    m,
    add_numbers = TRUE,
    numbers_gp = gpar(fontsize = 14),  # bigger labels above bars
    height = unit(70, "mm"), # bigger bars
    annotation_name_gp = gpar(fontsize = 14), # bigger axis label ("Intersection size")
    axis_param = list(gp = gpar(fontsize = 11))  # bigger axis tick labels
)

right_ann_args <- ComplexHeatmap::upset_right_annotation(
    m,
    width = unit(40, "mm"), # width of right bars
    annotation_name_gp = gpar(fontsize = 14), # bigger axis label ("Set size")
    axis_param = list(gp = gpar(fontsize = 11)) # bigger axis tick labels
)

upset_plot <- UpSet(
  m,
  # These args control the membership block
  height = unit(32, "mm"),           # block height
  pt_size = unit(3, "mm"),           # dot size in the membership matrix
  row_names_gp = gpar(fontsize = 14), # row labels (set names)

  comb_order = ordered_combs,
  set_order = c("Maintainer","Contributor","Bug reporter"),  # control set order

  top_annotation = top_ann_args,
  right_annotation = right_ann_args,
  column_title = "Participants' open source roles",
  column_title_gp  = gpar(fontsize = 22, fontface = "bold")

)

upset_plot
```



Save the plot

```
tiff(
  file.path(FIGURE_PATH, "roles_upset.tiff"),
  width = 10,
  height = 6,
  unit = "in",
  res = 700
  )
upset_plot
dev.off()
```

```
pdf
  2
```

Further exploration with maintainers

Out of curiosity, how many maintainers also selected tech support?

```
two_way_counts <- with(roles, {
  M <- Maintainer == 1
  TS <- `Technical support` == 1

  c(
    "M, !T"   = sum(M & !TS),
    "M, T"  = sum(M & TS)
  )
})

two_way_counts
```

```
M, !T  M, T
   66    68
```

Eh, not that interesting. I thought more maintainers would also be tech supporters.

What's the ratio of maintainers to other contributors?

```
nrow(subset(roles, Maintainer == 1))
```

```
[1] 134
```

```
nrow(subset(roles, Maintainer == 0))
```

```
[1] 99
```

That's somewhat interesting, and I guess it's consistent with the "three-way venn-diagram" above. 134 people identified as maintainers, and only 99 people identified as contributors of some sort, but not maintainers. So $134/233 = 57.5\%$ of experienced contributors in this survey are maintainers. I'm getting the sense, basically, that there are a lot of maintainers in this survey.

Bring in field of study

Are Math/CS people more likely to be maintainers than biologists?

```
roles_field <- cbind(roles_raw, other_quant$field_of_study)
# Rename column
names(roles_field)[ncol(roles_field)] <- "field_of_study"
# Filter out non-academics (people who didn't answer the field of study question)
roles_field <- roles_field %>%
    filter(field_of_study != "")

nrow(roles_field)
```

```
[1] 188
```

```
# Remove people who did not select any role
roles_field_clean <- roles_field[rowSums(roles_field[, -ncol(roles_field)]) != 0,]

nrow(roles_field_clean)
```

```
[1] 147
```

```
# Sanity check: make sure none of the rows sums to 0
unname(rowSums(roles_field_clean %>% select(-field_of_study)))
```

```
  [1]  7  1  5  5  5  4  5  3  6  4  3  2  3  3  5  2  3  4  2  4  8  3  5  3  3
 [26]  4  1  4  2  3  3  3  8  4  2  3  6  1  8 10  4  5  2  3  5  4  3  3  3  4
 [51]  4  7  2  6  6  5  4  2  7  7  3  4  3  5  2  2  2  5  9  5  6  2  7  6  1
 [76]  2  8  2  5  4  7  4  3  4  3  3  4  2  5  1  2  3  3  4  4  4 10  4  6  4
[101]  4  6  5  4  3  1  1  2  6  7  1  1  7  8  5  4  2  4  6  2  6  5  8  4  8
[126]  7  4  7  7  3  1  2  6  4  4  1  4  5  3  5  6  2  8  4  3  6  2
```

Okay, so we have a total of 188 academics. It looks like 147 of those are experienced contributors–people who selected at least one role.

```
maintainer_props <- roles_field_clean %>%
  group_by(field_of_study) %>%
  summarise(
    n_people          = n(),
```

```
    n_maintainers     = sum(Maintainer == 1),
    prop_maintainers  = mean(Maintainer == 1)
  ) %>%
  arrange(desc(prop_maintainers))

maintainer_props
```

```
# A tibble: 5 x 4
  field_of_study    n_people n_maintainers prop_maintainers
  <chr>                <int>         <int>            <dbl>
1 Math and CS             72            53            0.736
2 Physical sciences       27            18            0.667
3 Life sciences           34            18            0.529
4 Social sciences         10             4            0.4
5 Humanities               4             1            0.25
```

Meh, mildly interesting. 50% or more of our survey respondents from STEM fields are maintainers. There's really not sufficient respondents from the non-STEM fields to draw any conclusions there.

Campus

What proportion of respondents are maintainers, per campus?

```
roles_campus <- cbind(roles_raw, other_quant$campus)
# Rename column
names(roles_campus)[ncol(roles_campus)] <- "campus"
# Filter out non-UC
roles_campus <- roles_campus %>%
    filter(campus != "I'm not affiliated with UC")
# Remove people who did not select any role
roles_campus <- roles_campus[rowSums(roles_campus[, -ncol(roles_campus)]) != 0,]

nrow(roles_campus)
```

```
[1] 233
```

```
maint_by_campus <- roles_campus %>%
  group_by(campus) %>%
  summarise(
    n_rows = n(),
    n_maintainer = sum(Maintainer == 1),
    prop_maintainer = mean(Maintainer == 1)
  ) %>%
  arrange(desc(prop_maintainer))


maint_by_campus <- maint_by_campus[
  order(
    maint_by_campus$prop_maintainer,
    decreasing = TRUE
  ),
]
maint_by_campus
```

```
# A tibble: 10 x 4
   campus           n_rows n_maintainer prop_maintainer
   <chr>             <int>        <int>            <dbl>
 1 Other UC             19           15            0.789
 2 UC Santa Cruz        32           22            0.688
 3 UC Berkeley          26           16            0.615
 4 UC San Francisco      7            4            0.571
 5 UC Santa Barbara     61           34            0.557
 6 UC Los Angeles       40           21            0.525
 7 UC Merced             8            4            0.5
 8 UC Davis             29           14            0.483
 9 UC San Diego          9            4            0.444
10 UC Irvine             2            0            0
```

Meh, again, only mildly interesting. I suppose it's an interesting add-on to the stat that about half of our respondents are maintainers. Almost like error basr, where the variability comes from the campuses. The campuses with fewer than ten respondents just feel like noise to me, since we can't really draw conclusions. Let's filter those out.

```
subset(maint_by_campus, n_rows > 10)
```

```
# A tibble: 6 x 4
  campus           n_rows n_maintainer prop_maintainer
```

```
  <chr>             <int>        <int>           <dbl>
1 Other UC             19           15           0.789
2 UC Santa Cruz        32           22           0.688
3 UC Berkeley          26           16           0.615
4 UC Santa Barbara     61           34           0.557
5 UC Los Angeles       40           21           0.525
6 UC Davis             29           14           0.483
```

That's a little clearer.

Job category

There are many ways we could slice and dice these data, but mainly I'd like to know whether the % of maintainers is similar between academics and non-research staff.

```
roles_job_raw <- cbind(roles_raw, other_quant$job_category)
# Rename column
names(roles_job_raw)[ncol(roles_job_raw)] <- "job_category"
# Filter out people who didn't answer the job_category question
roles_job_raw <- roles_job_raw %>%
    filter(job_category != "")

nrow(roles_job_raw)
```

```
[1] 294
```

```
# Remove people who did not select any role
roles_job <- roles_job_raw[rowSums(roles_job_raw[, -ncol(roles_job_raw)]) != 0,]

nrow(roles_job)
```

```
[1] 233
```

```
# Sanity check: make sure none of the rows sums to 0
unname(rowSums(roles_job %>% select(-job_category)))
```

```
  [1]  7  1  5  5  5  4  5  3  6  4  3  2  3  3  5  2  3  4  4  2  4  4  8  3  3
 [26]  5  3  3  5  4  1  4  2  3  3  3  8  4  2  3  6  1  8  5 10  4  5  2  3  9
 [51]  5  5  1  4  3  2  5  5  6  5  8  7  3  9  3  7  4  4  2  4  7  2  6  6  5
```

```
 [76]  4  4  2  7  7  3  4  3  2  2  5  2  2  2  5  9  5  3  3  1  6  5  4  2  2
[101]  2  6  2  7  5  5  6  1  2  8  4  2  3  4  5  4  7  3  3  4  3  3  4  7  3
[126]  3  4  2  4  5  3  1  2  7  3  3  4  4  6  4  6 10  4  5  5  3  3  6  4  4
[151]  6  7  4  4  1  1  2  9  7  2  2  6  8  7  7  5  8  4  6  5  4  2  3  1  4
[176]  1  2  6  6  7  1  1  7  4  8  5  4  8  5  4  2  4  6  2  6  5  4  5  1  8
[201]  4  2  8  7  4  7  6  7  2  3  1  2  6  4  4  1  4  5  3  5  3  5  9  5  6
[226]  4  2  8  5  4  3  6  2
```

```
head(roles_job)
```

```
  Maintainer Contributor Bug/Issue Reporter Community Manager Educator Other
1          1           1                  1                 1         1     0
2          0           1                  0                 0         0     0
3          1           1                  1                 0         1     0
4          1           1                  1                 0         1     0
5          1           1                  1                 0         1     0
7          1           1                  0                 0         0     1
  Supervisor IT/Systems administrator UI/UX Designer Technical support
1          1                        0              0                 1
2          0                        0              0                 0
3          0                        0              1                 0
4          1                        0              0                 0
5          1                        0              0                 0
7          1                        0              0                 0
         job_category
1             Faculty
2            Post-Doc
3 Other research staff
4             Faculty
5             Faculty
7             Faculty
```

Table

First, let's explore the % maintainers for all 5 job categories.

```
by_job <- roles_job %>%
  group_by(job_category) %>%
  summarise(
    n_yes   = sum(Maintainer == 1),
    n_total = n(),
```

```
    pct_yes = round(100 * n_yes/n_total, 2)
  ) %>%
  ungroup()

by_job
```

```
# A tibble: 6 x 4
  job_category         n_yes n_total pct_yes
  <chr>                <int>   <int>   <dbl>
1 Faculty                 40      59    67.8
2 Grad Student            13      26    50
3 Non-research Staff      40      86    46.5
4 Other research staff    29      40    72.5
5 Post-Doc                 9      15    60
6 Undergraduate            3       7    42.9
```

Plots

```
job_maint_to_plot <- roles_job %>%
  group_by(job_category) %>%
  summarise(
    maintainer = sum(Maintainer == 1),
    not_maintainer = sum(Maintainer == 0)
  ) %>%
  ungroup()

job_maint_to_plot <- job_maint_to_plot %>%
  pivot_longer(
    cols = -c(job_category),
    names_to = "Maintainer",
    values_to = "n"
  )
```

```
# Reorder factor levels by the highest proportion of maintainers
ordered_jobs <- job_maint_to_plot %>%
  group_by(job_category) %>%
  summarise(
    maintainer = n[Maintainer=="maintainer"],
    not_maintainer = n[Maintainer=="not_maintainer"],
    .groups = "drop"
```

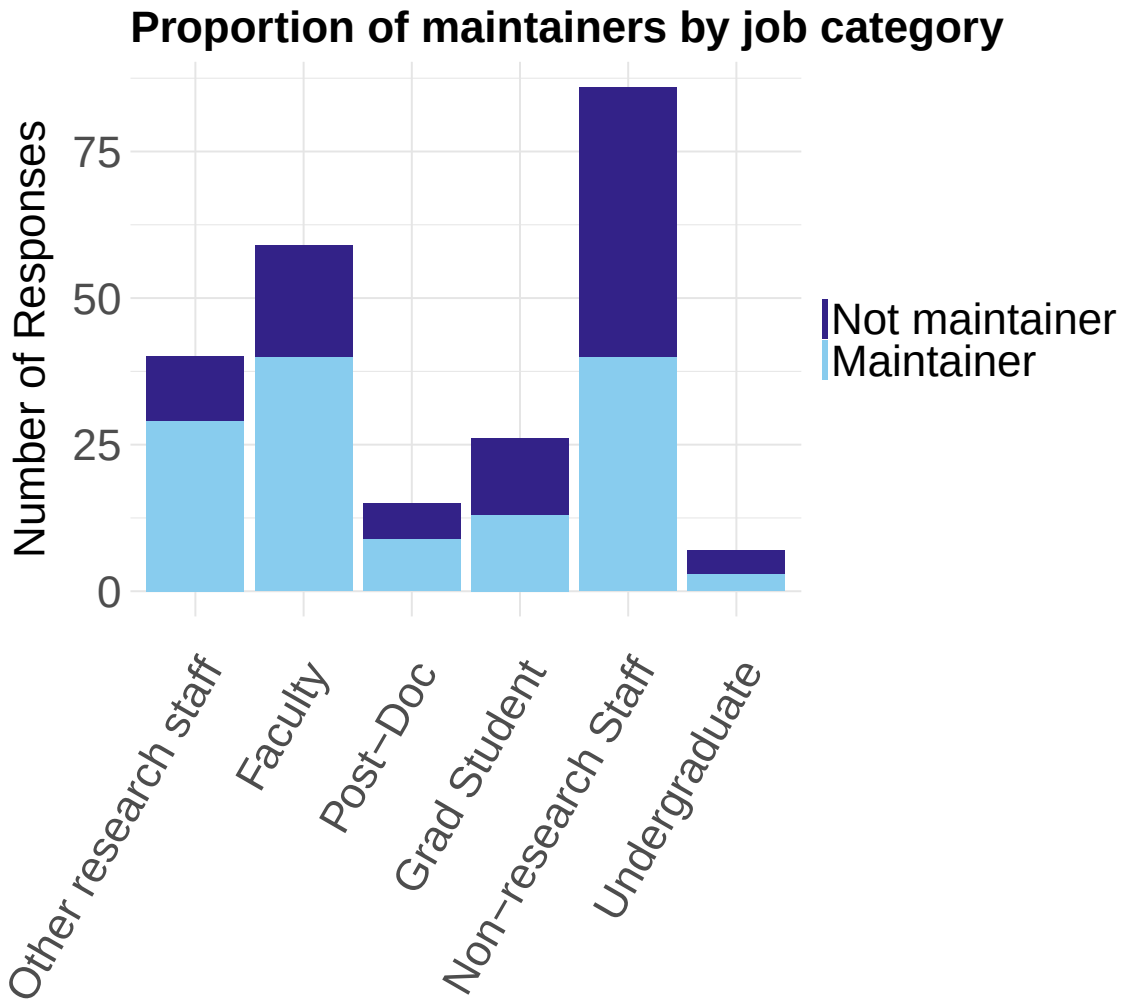
```
  ) %>%
  mutate(ratio = maintainer / not_maintainer) %>%
  arrange(desc(ratio)) %>%
  pull(job_category)

job_maint_to_plot$job_category <- factor(job_maint_to_plot$job_category, levels = ordered_jo

job_maint_to_plot$Maintainer <- factor(job_maint_to_plot$Maintainer, levels = c("not_maintai

# Make the labels prettier for legend
legend_labs <- stringr::str_to_sentence(gsub("_", " ", levels(job_maint_to_plot$Maintainer))
```

Plot

```
stack <- stacked_bar_chart(
  df = job_maint_to_plot,
  x_var = "job_category",
  y_var = "n",
  fill = "Maintainer",
  title = "Proportion of maintainers by job category")

# COLORS from utils.R
stack <- stack +
  scale_fill_manual(labels = legend_labs, values = COLORS)
```

```
Scale for fill is already present.
Adding another scale for fill, which will replace the existing scale.
```

```
stack
```

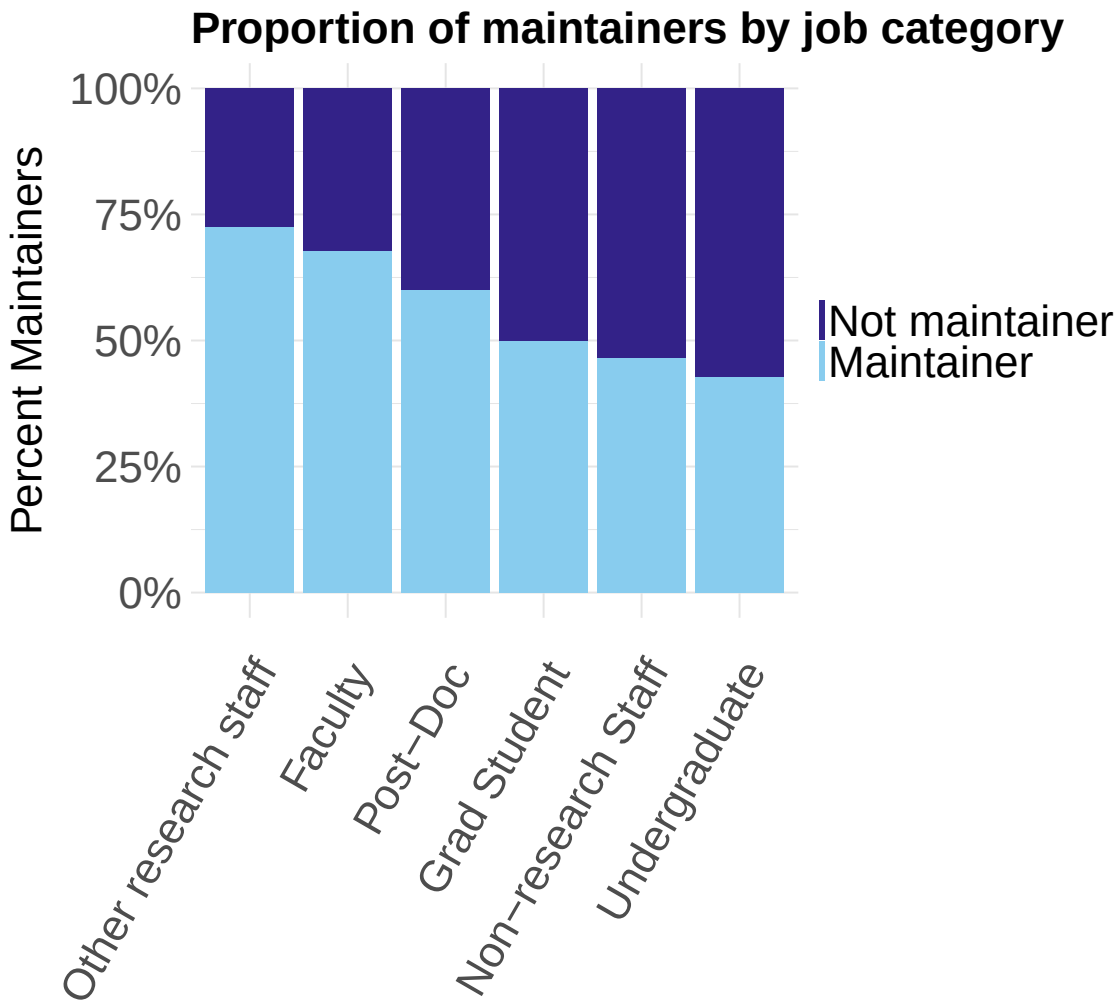



```
stack_prop <- stacked_bar_chart(
  df = job_maint_to_plot,
  x_var = "job_category",
  y_var = "n",
  ylabel = "Percent Maintainers",
  fill = "Maintainer",
  title = "Proportion of maintainers by job category",
  proportional = TRUE
)

stack_prop <- stack_prop +
  scale_y_continuous(labels = scales::percent)
```

```r
# COLORS from utils.R
stack_prop <- stack_prop +
  scale_fill_manual(labels = legend_labs, values = COLORS)
```

```
Scale for fill is already present.
Adding another scale for fill, which will replace the existing scale.
```

```r
stack_prop
```



```
p_combined <- stack + stack_prop & theme(plot.margin = unit(c(1, 1, 1, 1), "cm"))
```

```
svglite::svglite(file.path(FIGURE_PATH, "figure4.svg"), width = 16, height = 8); print(p_comb
```

```
pdf
  2
```

Also saving the data and making the final figure for submission in a separate script.

```
# see utils.R
write_df_to_file(
  job_maint_to_plot,
  file.path("data_for_plots/maintainers_bar.tsv")
)
```