Importance of open source

Overview

This script creates bar plots from question 2 on the survey, which is about the perceived importance of open source for different job categories and different tasks. My favorite plot is the one in the final section, “Percent more than moderately important”.

Import packages and utilities

```r
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_percent_more_than_moderate

- Arguments:
  - `df`: A data frame with a column named `importance_level`. Should contain only rows that you want to count. Extra columns are okay.
- Details:
  - A simple function to count what percent of rows in a data frame have an `importance_level` of more than `Moderately important`. Checks that there are no extra rows with an unexpected value in the importance_level column.

- Outputs:
    - A vector with three items: the number of participants who selected an `importance_level` of more than `Moderately important`, the total number of eligible respondents, and a percentage: the first number / the second number *100 and rounded to 2 decimal places.

```
get_percent_more_than_moderate <- function(df) {
  # check that df has the required column
  if (!"importance_level" %in% names(df)) {
    stop("`df` must have a column named 'importance_level'.")
  }

  high_importance_cats <- c(
    "Very important",
    "Important"
  )
  low_importance_cats <- c(
    "Moderately important",
    "Slightly important",
    "Not at all important"
  )

  n_high_rows <- nrow(df %>% filter(importance_level %in% high_importance_cats))
  n_low_rows <- nrow(df %>% filter(importance_level %in% low_importance_cats))
  total <- nrow(df)
  if (n_high_rows + n_low_rows != total) {
    stop("`df` has at least one unexpected value in 'importance_level'.")
  }
  pct <- round(n_high_rows / total * 100, 2)
  return(c(round(n_high_rows), round(total), pct))
}
```

Load data

```
importance <- load_qualtrics_data("clean_data/importance_Q2.tsv")
other_quant <- load_qualtrics_data("clean_data/other_quant.tsv")
```

Wrangle data

Let's extract the columns we care about.

```
importance_and_job <- cbind(importance, other_quant$job)
names(importance_and_job)[length(names(importance_and_job))] <- "job_category"
```

Remove all rows that contain an empty string in any column. Since both questions were mandatory, I'm actually only removing people who never saw the demographic questions.

```
importance_and_job <- exclude_empty_rows(importance_and_job, strict = TRUE)
```

Let's reshape the data from wide to long format.

```
long_data <- importance_and_job %>%
  pivot_longer(
    cols = -job_category,
    names_to = "importance_area",
    values_to = "importance_level"
  )
```

Reorder factor levels for plotting.

```
long_data$importance_level <- factor(
  long_data$importance_level,
  levels = c(
    "Very important",
    "Important",
    "Moderately important",
    "Slightly important",
    "Not at all important",
    "Non-applicable"
  ),
  ordered = TRUE
)
long_data
```

```
# A tibble: 1,470 x 3
   job_category importance_area          importance_level
   <chr>        <chr>                    <ord>
```

```
 1 Faculty      Research                 Very important
 2 Faculty      Teaching                 Very important
 3 Faculty      Learning                 Very important
 4 Faculty      Professional Development Very important
 5 Faculty      Job                      Very important
 6 Post-Doc     Research                 Very important
 7 Post-Doc     Teaching                 Moderately important
 8 Post-Doc     Learning                 Important
 9 Post-Doc     Professional Development Important
10 Post-Doc     Job                      Non-applicable
# i 1,460 more rows
```

Bar plots

Simple bar plot for teachers

Now let's start making some bar plots. Let's start by making a bar plot showing how teachers rate the importance of open source for their teaching. Since we didn't ask people "Do you teach?", and since there was a "Non-applicable" option, we will simply assume that if they gave an answer for the "Teaching" option, they must be a teacher.

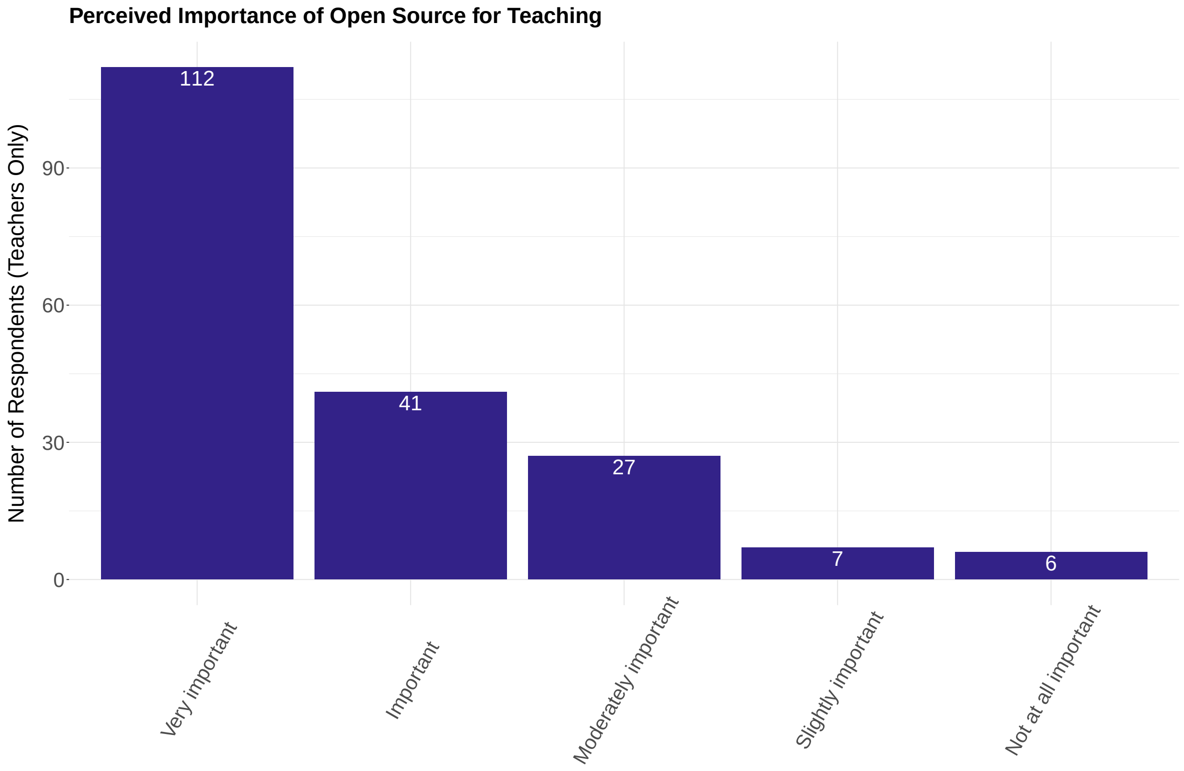
```r
teaching <- long_data %>%
  filter(
    importance_area == "Teaching"
  ) %>%
  filter(
    importance_level != "Non-applicable"
  )

# For our bar plot, we only care about how many times each 'importance level' was selected.
teaching_to_plot <- teaching %>% select(-c(job_category, importance_area))

teaching_to_plot <- teaching_to_plot %>%
  count(importance_level, name = "Counts")

# By sheer luck, the columns are already ordered by response rates.
teaching_to_plot
```

```
# A tibble: 5 x 2
  importance_level     Counts
  <ord>                 <int>
```

```
1 Very important           112
2 Important                 41
3 Moderately important      27
4 Slightly important         7
5 Not at all important       6
```

Now let's make that bar chart using a function that lives in my utilities script (`scripts/utils.R`).

```
basic_bar_chart(
  teaching_to_plot,
  x_var = "importance_level",
  y_var = "Counts",
  title = "Perceived Importance of Open Source for Teaching",
  ylabel = "Number of Respondents (Teachers Only)",
  show_bar_labels = TRUE
)
```



Save the plot using a function that lives in my utilities script (`scripts/utils.R`).

```
#save_plot("importance_teachers.tiff", 8, 5)
```

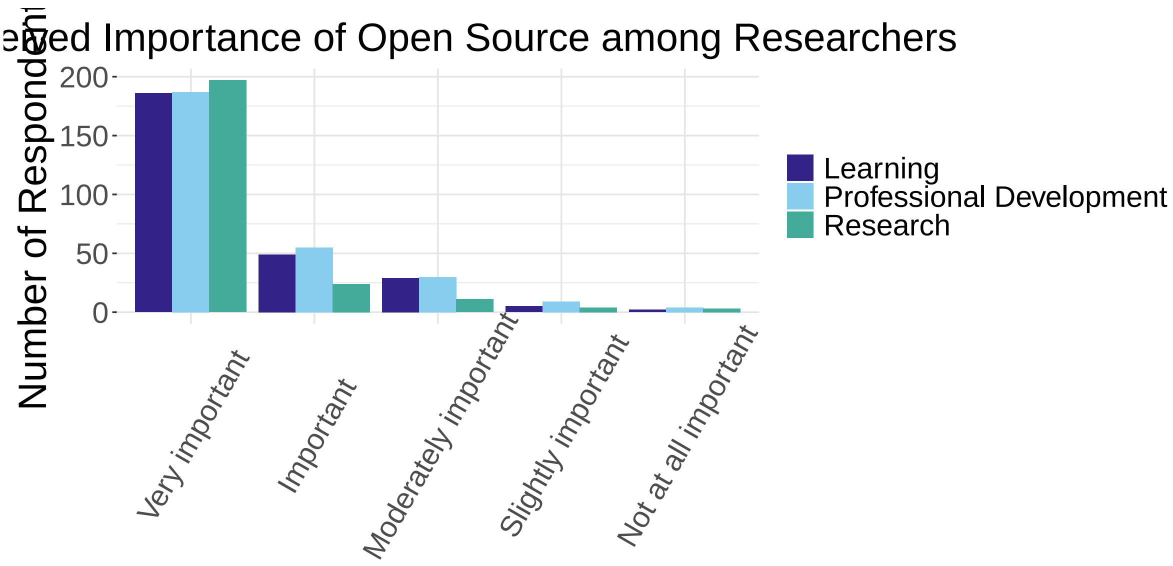
Grouped bar plot for researchers

Now let's look at researchers, and the importance categories that apply to all researchers. The importance categories again are:
Research
Teaching –> Does not apply
Learning
Professional Development
Job (For non-research staff) –> Does not apply

So we'll make a bar plot with just those three categories that apply to all researchers. As with teachers above, we will assume that if they didn't select "Non-applicable", they must be a researcher.

The `grouped_bar_chart` function, like the `basic_bar_chart` function, lives in my utility script.

```
research_learning_pd <- long_data %>%
  filter(
    importance_area == "Research" |
      importance_area == "Learning" |
      importance_area == "Professional Development"
  ) %>%
  filter(importance_level != "Non-applicable")

grouped_bar_chart(
  df = research_learning_pd,
  x_var = "importance_level",
  fill_var = "importance_area",
  title = "Perceived Importance of Open Source among Researchers"
)
```

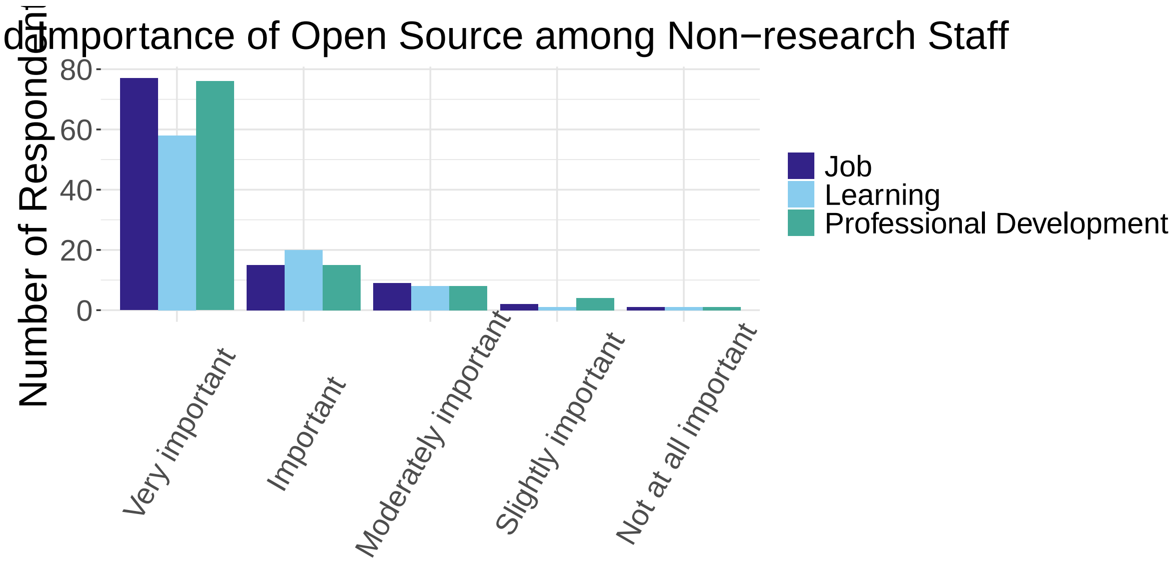


```
#save_plot("importance_researchers.tiff", 10, 5)
```

Grouped bar plot for non-research staff

This is very similar to what I did above, except the three applicable categories have changed.

```
job_learning_pd <- long_data %>%
  filter(
    importance_area == "Job" |
      importance_area == "Learning" |
      importance_area == "Professional Development"
  ) %>%
  filter(job_category == "Non-research Staff") %>%
  filter(importance_level != "Non-applicable")

grouped_bar_chart(
  df = job_learning_pd,
  x_var = "importance_level",
  fill_var = "importance_area",
  title = "Perceived Importance of Open Source among Non-research Staff"
)
```



```
#save_plot("importance_nrstaff.tiff", 10, 5)
```

Percent more than moderately important

Renata suggested I try to combine all these data into one figure that summarizes the question at a glance. Here's my attempt.

I think a useful "statistic" is the percent of a particular group that said OS is more than moderately important for a particular area of work. Let's make a dataframe with those percentages. I'd ultimately like to turn this into a bar plot where the color or design of the bars corresponds to the five job categories, and the x-axis shows four groups that I think are most relevant: teachers, researchers, non-research staff, and students. However, not all these groups were explicit survey categories, and not all 5 importance areas apply to all groups, so we'll need to do a fair amount of data wrangling.

To start, let's get the percent of teachers who said teaching was more than moderately important.

```
teaching <- long_data %>%
  filter(
    importance_area == "Teaching"
  ) %>%
  filter(
    importance_level != "Non-applicable"
  )

# This will be useful later
res_teachers_teaching <- get_percent_more_than_moderate(teaching)

more_than_mod <- data.frame(
  job_category = "Teachers",
  importance_area = "Teaching",
  pct = res_teachers_teaching[3]
)

more_than_mod
```

```
  job_category importance_area   pct
1     Teachers        Teaching 79.27
```

The code is basically the same for researchers and researching: as with teaching, we will assume that anyone who gave an answer for research (i.e, didn't select "N/A") is a researcher.

```
research <- long_data %>%
  filter(
    importance_area == "Research"
  ) %>%
  filter(
    importance_level != "Non-applicable"
  )

res_researchers_research <- get_percent_more_than_moderate(research)

more_than_mod <- rbind(
  more_than_mod,
  list(
    "Researchers",
    "Research",
    res_researchers_research[3]
  )
)
```

The code for non-research staff and "Job" is slightly different. Our intention was that only non-research staff would answer this question, but there were some people who answered this but did not select "non-research staff" as their job category. So let's just ensure that we're only looking at responses from non-research staff by filtering for non-research staff using the job_category column.

```
nrstaff <- long_data %>%
  filter(
    job_category == "Non-research Staff"
  ) %>%
  filter(
    importance_area == "Job"
  ) %>%
  filter(
    importance_level != "Non-applicable"
  )

res_staff_job <- get_percent_more_than_moderate(nrstaff)

more_than_mod <- rbind(
  more_than_mod,
  list(
    "Non-research Staff",
    "Job",
    res_staff_job[3]
  )
)
```

Next, I want to look at two importance areas, Learning and Professional Development, for all four job categories: Teachers, Researchers, Non-research staff, and Students. We'll have to determine teachers and researchers based on who answered the teaching question and who answered the research question, respectively. Meanwhile, for "Student", we'll have to combine grad students and undergrads into one group.

Let's keep rows from teachers, but keep columns for Learning and Professional Development. Then we change the job_category column to "Teacher".

```
teachers_learn_pd <- importance_and_job %>%
  filter(Teaching != "Non-applicable") %>%
  select(Learning, `Professional Development`, job_category)

teachers_learn_pd$job_category <- "Teacher"
head(teachers_learn_pd)
```

```
              Learning Professional Development job_category
```

```
1       Very important           Very important      Teacher
2            Important                Important      Teacher
3       Very important           Very important      Teacher
4            Important                Important      Teacher
5       Very important           Very important      Teacher
6 Moderately important     Moderately important      Teacher
```

Now we can add two more rows to `more_than_mod`.

```
teachers_learning <- teachers_learn_pd %>%
  select(Learning, job_category) %>%
  #unlikely but you never know
  filter(Learning != "Non-applicable") %>%
  # Change the column name because our function expects it
  rename(importance_level = Learning)

more_than_mod <- rbind(
  more_than_mod,
  list(
    "Teachers",
    "Learning",
    get_percent_more_than_moderate(teachers_learning)[3]
  )
)

teachers_pd <- teachers_learn_pd %>%
  select(`Professional Development`, job_category) %>%
  #unlikely but you never know
  filter(`Professional Development` != "Non-applicable") %>%
  # Change the column name because our function expects it
  rename(importance_level = `Professional Development`)

more_than_mod <- rbind(
  more_than_mod,
  list(
    "Teachers",
    "Professional Development",
    get_percent_more_than_moderate(teachers_pd)[3]
  )
)
more_than_mod
```

```
        job_category          importance_area   pct
```

```
1           Teachers                 Teaching 79.27
2        Researchers                 Research 92.47
3 Non-research Staff                      Job 88.46
4           Teachers                 Learning 84.29
5           Teachers Professional Development 81.48
```

And let's do the same for researchers.

```r
researchers_learn_pd <- importance_and_job %>%
  filter(Research != "Non-applicable") %>%
  select(Learning, `Professional Development`, job_category)

researchers_learn_pd$job_category <- "Researcher"

researchers_learning <- researchers_learn_pd %>%
  select(Learning, job_category) %>%
  #unlikely but you never know
  filter(Learning != "Non-applicable") %>%
  # Change the column name because our function expects it
  rename(importance_level = Learning)

more_than_mod <- rbind(
  more_than_mod,
  list(
    "Researchers",
    "Learning",
    get_percent_more_than_moderate(researchers_learning)[3]
  )
)

researchers_pd <- researchers_learn_pd %>%
  select(`Professional Development`, job_category) %>%
  #unlikely but you never know
  filter(`Professional Development` != "Non-applicable") %>%
  # Change the column name because our function expects it
  rename(importance_level = `Professional Development`)

more_than_mod <- rbind(
  more_than_mod,
  list(
    "Researchers",
    "Professional Development",
    get_percent_more_than_moderate(researchers_pd)[3]
  )
)
more_than_mod
```

```
        job_category          importance_area   pct
1           Teachers                 Teaching 79.27
2        Researchers                 Research 92.47
3 Non-research Staff                      Job 88.46
4           Teachers                 Learning 84.29
5           Teachers Professional Development 81.48
6        Researchers                 Learning 85.90
7        Researchers Professional Development 83.98
```

Now get percentages for non-research staff. This is straightforward since it was a survey category.

```
nrstaff_learn_pd <- importance_and_job %>%
  filter(job_category == "Non-research Staff") %>%
  select(Learning, `Professional Development`, job_category)

nrstaff_learning <- nrstaff_learn_pd %>%
  select(Learning, job_category) %>%
  #unlikely but you never know
  filter(Learning != "Non-applicable") %>%
  # Change the column name because our function expects it
  rename(importance_level = Learning)

more_than_mod <- rbind(
  more_than_mod,
  list(
    "Non-research Staff",
    "Learning",
    get_percent_more_than_moderate(nrstaff_learning)[3]
  )
)

nrstaff_pd <- nrstaff_learn_pd %>%
  select(`Professional Development`, job_category) %>%
  #unlikely but you never know
  filter(`Professional Development` != "Non-applicable") %>%
  # Change the column name because our function expects it
  rename(importance_level = `Professional Development`)

more_than_mod <- rbind(
  more_than_mod,
  list(
    "Non-research Staff",
    "Professional Development",
    get_percent_more_than_moderate(nrstaff_pd)[3]
  )
)
```

Finally, let’s get students.

```
students_learn_pd <- importance_and_job %>%
  filter(job_category == "Undergraduate" | job_category == "Grad Student") %>%
  select(Learning, `Professional Development`, job_category)

students_learn_pd$job_category <- "Student"

students_learning <- students_learn_pd %>%
  select(Learning, job_category) %>%
  #unlikely but you never know
  filter(Learning != "Non-applicable") %>%
  # Change the column name because our function expects it
  rename(importance_level = Learning)

more_than_mod <- rbind(
  more_than_mod,
  list(
    "Students",
    "Learning",
    get_percent_more_than_moderate(students_learning)[3]
  )
)

students_pd <- students_learn_pd %>%
  select(`Professional Development`, job_category) %>%
  #unlikely but you never know
  filter(`Professional Development` != "Non-applicable") %>%
  # Change the column name because our function expects it
  rename(importance_level = `Professional Development`)
```

```r
more_than_mod <- rbind(
  more_than_mod,
  list(
    "Students",
    "Professional Development",
    get_percent_more_than_moderate(students_pd)[3]
  )
)

more_than_mod
```

```
         job_category          importance_area   pct
1            Teachers                 Teaching 79.27
2         Researchers                 Research 92.47
3  Non-research Staff                      Job 88.46
4            Teachers                 Learning 84.29
5            Teachers Professional Development 81.48
6         Researchers                 Learning 85.90
7         Researchers Professional Development 83.98
8  Non-research Staff                 Learning 88.64
9  Non-research Staff Professional Development 87.50
10           Students                 Learning 92.86
11           Students Professional Development 87.04
```

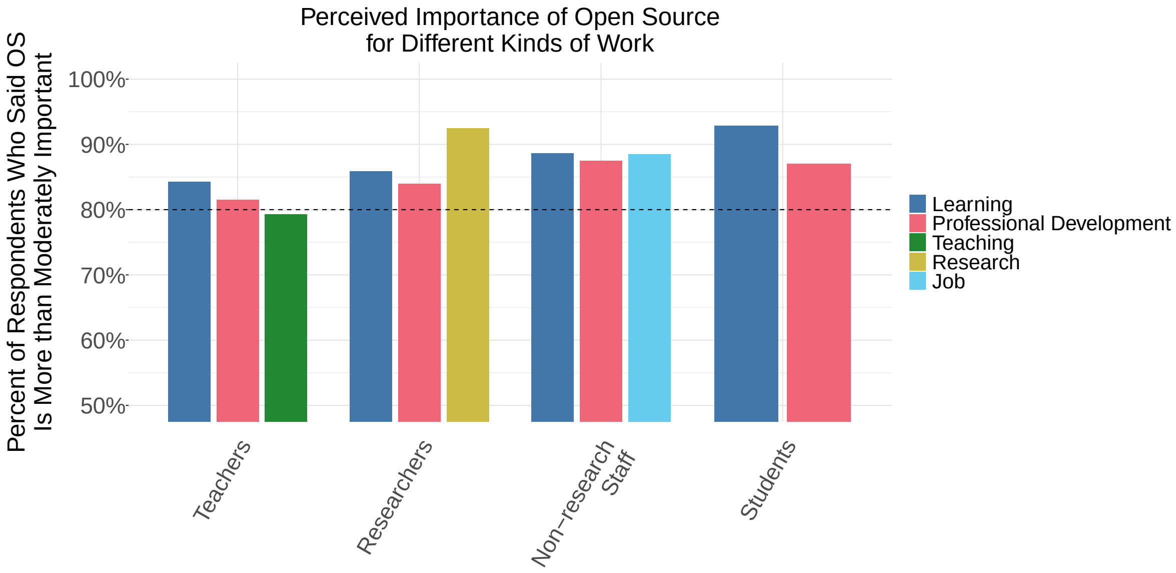
FINALLY, let's plot it!

Reorder factor levels

```r
more_than_mod$job_category <- gsub(
  "Non-research Staff",
  "Non-research\nStaff",
  more_than_mod$job_category
)

more_than_mod$job_category <- factor(
  more_than_mod$job_category,
  levels = c(
    "Teachers",
    "Researchers",
    "Non-research\nStaff",
    "Students"
  )
)

more_than_mod$importance_area <- factor(
  more_than_mod$importance_area,
  levels = c(
    "Learning",
    "Professional Development",
    "Teaching",
    "Research",
    "Job"
  )
)
```

I'm not using my `grouped_bar_chart` function in `scripts/utils.R` because I have pre-computed the bar heights, and that function counts rows. Since I'm currently only creating this kind of bar chart once, I'm not bothering to create a new function (or incorporate this option into the `grouped_bar_chart` function).

```
p <- ggplot(
  more_than_mod,
  aes(
    x = job_category,
    y = pct,
    fill = importance_area
  )
) +
  geom_col(position = position_dodge(width = 0.8), width = 0.7) +
  geom_hline(yintercept = 80, linetype = "dashed", color = "black") +
  ggtitle("Perceived Importance of Open Source\nfor Different Kinds of Work") +
  labs(
    y = "Percent of Respondents Who Said OS\nIs More than Moderately Important"
  ) +
  scale_y_continuous(labels = function(x) paste0(x, "%")) +
  coord_cartesian(ylim = c(50, 100)) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_text(size = 24),
    axis.text.x = element_text(
      angle = 60,
      vjust = 1,
      hjust = 1,
      size = 22,
      margin = margin(t = 12)
    ),
    axis.text.y = element_text(
      size = 22,
      margin = margin(l = 8)
    ),
    axis.ticks.x = element_blank(),
    legend.title = element_blank(),
    legend.text = element_text(size = 20),
    panel.background = element_blank(),
    plot.title = element_text(hjust = 0.5, size = 24),
    plot.margin = unit(c(0.3, 0.3, 0.3, 0.3), "cm"),
    panel.grid = element_line(linetype = "solid", color = "gray90")
  ) +
  #https://sronpersonalpages.nl/~pault/
  scale_fill_manual(
    values = c(
      '#4477AA',
      '#EE6677',
      '#228833',
      '#CCBB44',
      '#66CCEE'
    )
  )

p
```



```
save_plot("importance_all_pct.tiff", 12, 8, p=p)
```

Get numbers, not just %

A last-minute add-on for the manuscript: let's get the raw numbers, not just the percentages, for those categories we chose to highlight.

```
res_teachers_teaching
```

```
[1] 153.00 193.00  79.27
```

```
res_researchers_research
```

```
[1] 221.00 239.00  92.47
```

```
res_staff_job
```

```
[1]  92.00 104.00  88.46
```

Session Info

```r
sessionInfo()
```

```
R version 4.4.2 (2024-10-31)
Platform: aarch64-apple-darwin20
Running under: macOS Sequoia 15.6.1

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] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.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       labeling_0.4.3      utf8_1.2.6
[19] rmarkdown_2.29      ggfittext_0.10.2    tzdb_0.5.0
```

```
[22] ps_1.9.1            nloptr_2.2.1        purrr_1.1.0
[25] xfun_0.53           modeltools_0.2-24   jsonlite_2.0.0
[28] tweenr_2.0.3        parallel_4.4.2      prabclus_2.3-4
[31] R6_2.6.1            stringi_1.8.7       RColorBrewer_1.1-3
[34] boot_1.3-31         diptest_0.77-2      numDeriv_2016.8-1.1
[37] estimability_1.5.1  Rcpp_1.1.0          iterators_1.0.14
[40] knitr_1.50          IRanges_2.40.1      splines_4.4.2
[43] nnet_7.3-19         tidyselect_1.2.1    yaml_2.3.10
[46] doParallel_1.0.17   codetools_0.2-20    processx_3.8.6
[49] lattice_0.22-6      tibble_3.3.0        withr_3.0.2
[52] evaluate_1.0.4      polyclip_1.10-7     xml2_1.4.0
[55] circlize_0.4.16     mclust_6.1.1        kernlab_0.9-33
[58] pillar_1.11.0       renv_1.1.5          foreach_1.5.2
[61] stats4_4.4.2        reformulas_0.4.1    generics_0.1.4
[64] rprojroot_2.1.1     S4Vectors_0.44.0    hms_1.1.3
[67] minqa_1.2.8         xtable_1.8-4        class_7.3-22
[70] glue_1.8.0          robustbase_0.99-4-1 mvtnorm_1.3-3
[73] rbibutils_2.3       colorspace_2.1-1    nlme_3.1-166
[76] cli_3.6.5           textshaping_1.0.1   gtable_0.3.6
[79] DEoptimR_1.1-4      digest_0.6.37       BiocGenerics_0.52.0
[82] ucminf_1.2.2        ggrepel_0.9.6       rjson_0.2.23
[85] farver_2.1.2        htmltools_0.5.8.1   lifecycle_1.0.4
[88] GlobalOptions_0.1.2 MASS_7.3-61
```