Solutions

Overview

This script makes some plots from Q10, which is about what solutions participants would find most useful.

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

```
solutions_raw <- load_qualtrics_data("clean_data/solutions_Q10.tsv")
other_quant <- load_qualtrics_data("clean_data/other_quant.tsv")
```

Wrangle data

First, 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.

```r
nrow(solutions_raw)
```

```
[1] 332
```

```r
solutions <- exclude_empty_rows(solutions_raw) # from scripts/utils.R
nrow(solutions)
```

```
[1] 233
```

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

```r
long_data <- solutions %>%
  pivot_longer(
    cols = everything(),
    names_to = "solution",
    values_to = "utility"
  )

long_data <- long_data %>%
  mutate(
    utility_score = recode(
      utility,
      "Non-applicable" = 0L,
      "Not very useful" = 0L,
      "Useful" = 1L,
      "Very useful" = 2L
    )
  )
# Using interger literals 0L, 1L, etc., ensures that
# the new column will be integers, not doubles.

long_data
```

```
# A tibble: 2,796 x 3
   solution               utility     utility_score
   <chr>                  <chr>               <int>
 1 Computing environments Very useful             2
 2 Publicity              Very useful             2
 3 Containerization       Very useful             2
 4 Documentation help     Very useful             2
```

```
 5 A learning community   Very useful             2
 6 Event planning         Very useful             2
 7 Mentoring programs     Very useful             2
 8 Education              Very useful             2
 9 Legal support          Very useful             2
10 Industry partnerships  Very useful             2
# i 2,786 more rows
```

Descriptive statistics

Next, let's calculate some simple descriptive statistics. I will choose:

- The total "score", that is, the total number of "points" a solution received (see scoring scheme in previous code chunk)
- The mean (which might be misleading if 0s drag it down, and also, who's to say what a 1.5 really means? Are the distances between the Likert points equal? We don't know.)
- The median
- The mode
- The standard deviation

```r
# 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(solution) %>%
  summarise(
    total  = sum(utility_score),
    mean   = mean(utility_score, na.rm = TRUE),
    median = median(utility_score),
    mode   = get_mode(utility_score),
    st_dev = sd(utility_score, na.rm = TRUE)
  ) %>%
  ungroup()

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

summary_df
```

```
# A tibble: 12 x 6
   solution               total  mean median  mode st_dev
   <chr>                  <int> <dbl>  <int> <int>  <dbl>
 1 Sustainability grants    353 1.52       2     2  0.732
 2 Help finding funding     316 1.36       2     2  0.764
 3 Computing environments   301 1.29       1     2  0.783
 4 A learning community     251 1.08       1     1  0.733
 5 Documentation help       248 1.06       1     1  0.788
 6 Legal support            242 1.04       1     1  0.762
 7 Education                236 1.01       1     1  0.801
 8 Industry partnerships    232 0.996      1     0  0.838
 9 Publicity                232 0.996      1     1  0.817
10 Mentoring programs       216 0.927      1     1  0.776
11 Containerization         203 0.871      1     0  0.820
12 Event planning           190 0.815      1     0  0.807
```

Cool. It looks like sustainability grants are by far the most popular, with assistance identifying funding sources and free computing environments in second and third place. These were the only three solutions that had a mode of 2.

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

```
summary_df %>%
  arrange(desc(st_dev))
```

```
# A tibble: 12 x 6
   solution               total  mean median  mode st_dev
   <chr>                  <int> <dbl>  <int> <int>  <dbl>
 1 Industry partnerships    232 0.996      1     0  0.838
 2 Containerization         203 0.871      1     0  0.820
 3 Publicity                232 0.996      1     1  0.817
 4 Event planning           190 0.815      1     0  0.807
 5 Education                236 1.01       1     1  0.801
 6 Documentation help       248 1.06       1     1  0.788
 7 Computing environments   301 1.29       1     2  0.783
 8 Mentoring programs       216 0.927      1     1  0.776
 9 Help finding funding     316 1.36       2     2  0.764
10 Legal support            242 1.04       1     1  0.762
11 A learning community     251 1.08       1     1  0.733
12 Sustainability grants    353 1.52       2     2  0.732
```

This analysis doesn't seem as interesting as it was for the challenges. Industy partnerships, Containerization, and Publicity all show high variance/stdev. These were also somewhat less popular. Come to think of it, the std devs don't vary much.

```
max(summary_df$st_dev)
```

```
[1] 0.8381931
```

```
min(summary_df$st_dev)
```

```
[1] 0.731665
```

```
max(summary_df$st_dev)-min(summary_df$st_dev)
```

```
[1] 0.1065281
```

Yeah, the std devs only range from 0.73 to 0.84 (~0.11)–substantially less than the differences in std devs between challenges (~0.66). We could probably demonstrate this with a significance test later if it feels interesting.

Out of curiosity, how many people said they would all be very useful?

```
nrow(
  solutions %>%
    filter(if_all(.cols = everything(), ~ . == "Very useful"))
)
```

```
[1] 14
```

Ah, ok. Not that many.

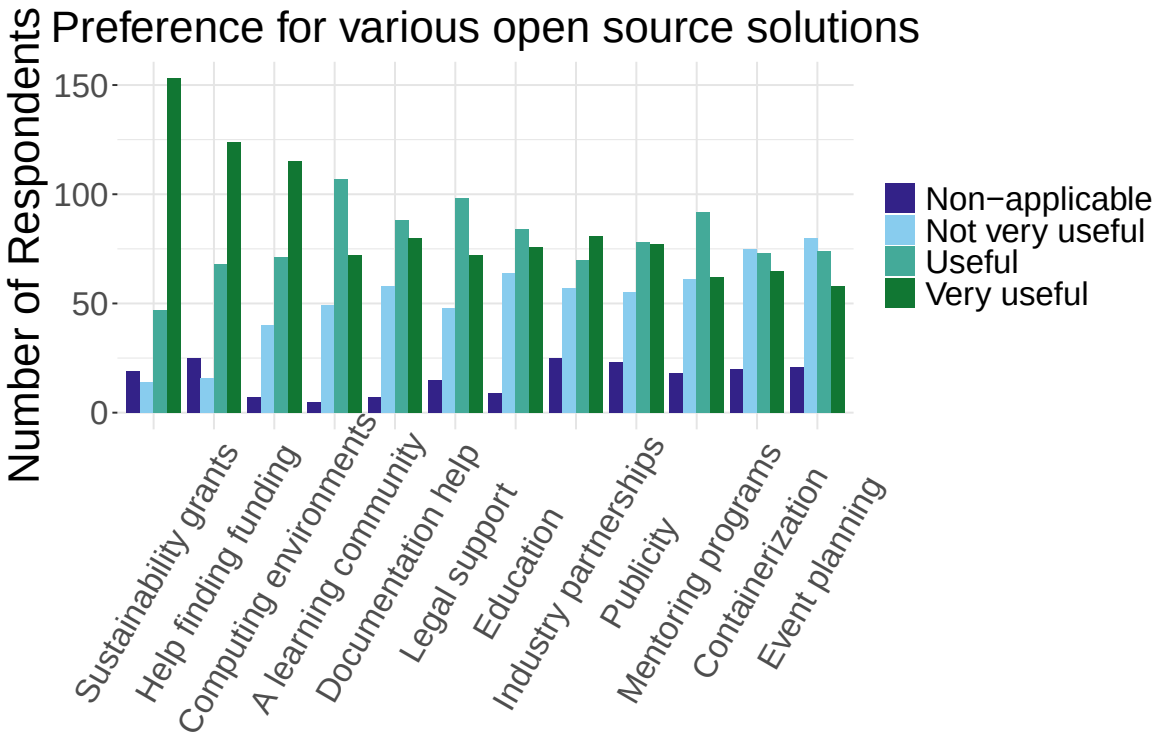
Plot the distributions

Prepare data for plotting.

```
ordered_levels <- (summary_df %>%
    arrange(desc(total)))$solution

long_data$solution <- factor(long_data$solution, levels = ordered_levels)
```

Grouped bar chart showing the distributions of answers.

```
grouped_plot <- grouped_bar_chart(
  df = long_data,
  x_var = "solution",
  fill_var = "utility",
  title = "Preference for various open source solutions"
)

grouped_plot
```



Save the plot if you wish.

```
save_plot("fave_solutions.tiff", 10, 6, p=grouped_plot)
```

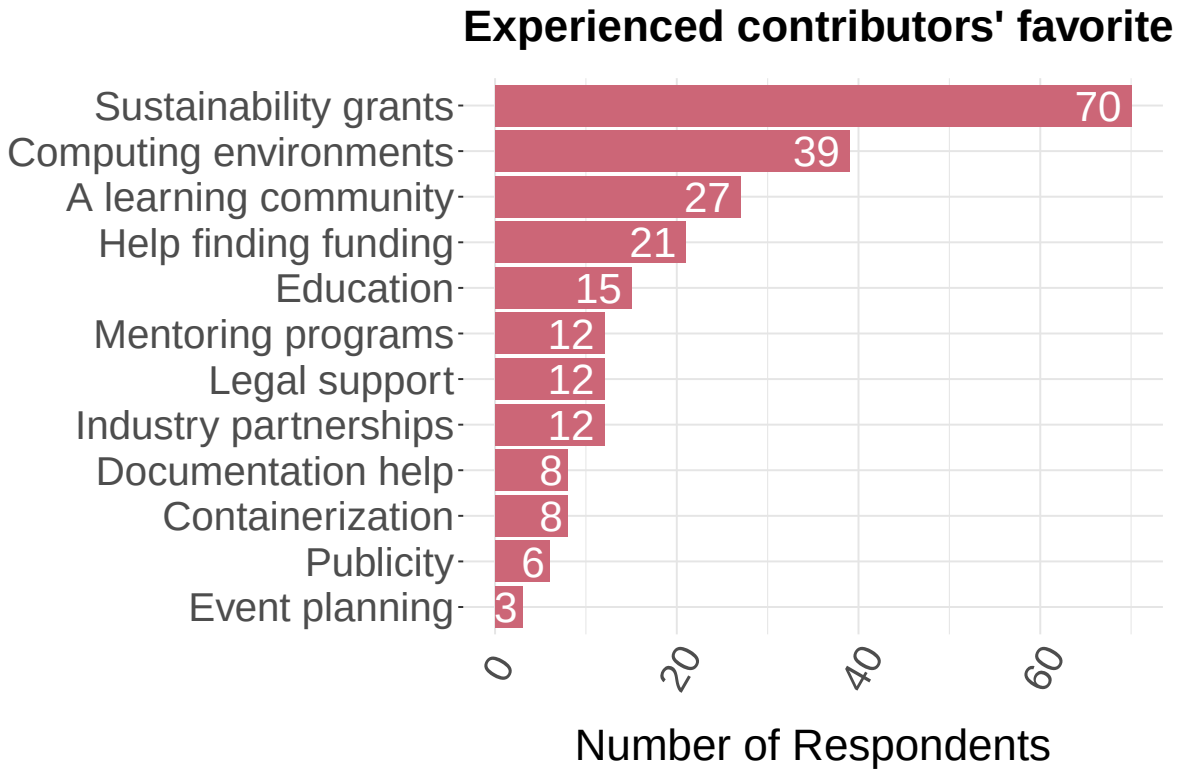
Simple bar plot

Now let's make a simpler bar plot from the next question, which asked participants to choose their favorite solution.

```r
favorites <- data.frame(other_quant$favorite_solution)
favorites <- exclude_empty_rows(favorites) # from scripts/utils.R

fav_to_plot <- data.frame(table(favorites[, 1]))
# from scripts/utils.R
fav_to_plot <- reorder_factor_by_column(
  df = fav_to_plot,
  factor_col = Var1,
  value_col = Freq,
  descending = FALSE
)
head(fav_to_plot)
```

```
                   Var1 Freq
1   A learning community   27
2 Computing environments   39
3       Containerization    8
4     Documentation help    8
5              Education   15
6         Event planning    3
```

```r
faves_plot <- basic_bar_chart(
  df = fav_to_plot,
  x_var = "Var1",
  y_var = "Freq",
  title = "Experienced contributors' favorite solutions",
  show_axis_title_x = TRUE,
  show_axis_title_y = FALSE,
  ylabel = "Number of Respondents",
  show_bar_labels = TRUE,
  color_index = 7,
  horizontal = TRUE
)

faves_plot
```



The top solutions are not exactly the same in this question compared to tallying up the totals from the previous one, though they are close.

Save the plot if you wish.

```
save_plot("fave_solutions_simple.tiff", 12, 6, p=faves_plot)
```

Incorporating demographics

Plots

Who are these people who want access to computing environments? Don't all the UCs already offer this?

Let's focus on job category.

```
campus_job_fave <- other_quant %>%
  select(campus, job_category, favorite_solution)
campus_job_fave <- exclude_empty_rows(campus_job_fave, strict = TRUE)
```

```r
# For visual clarity, let's combine postdocs and other staff researchers.
campus_job_fave <- campus_job_fave %>%
  mutate(
    job_category = recode(
      job_category,
      "Post-Doc" = "Postdocs and\nStaff Researchers",
      "Other research staff" = "Postdocs and\nStaff Researchers"
    )
  )

head(campus_job_fave)
```

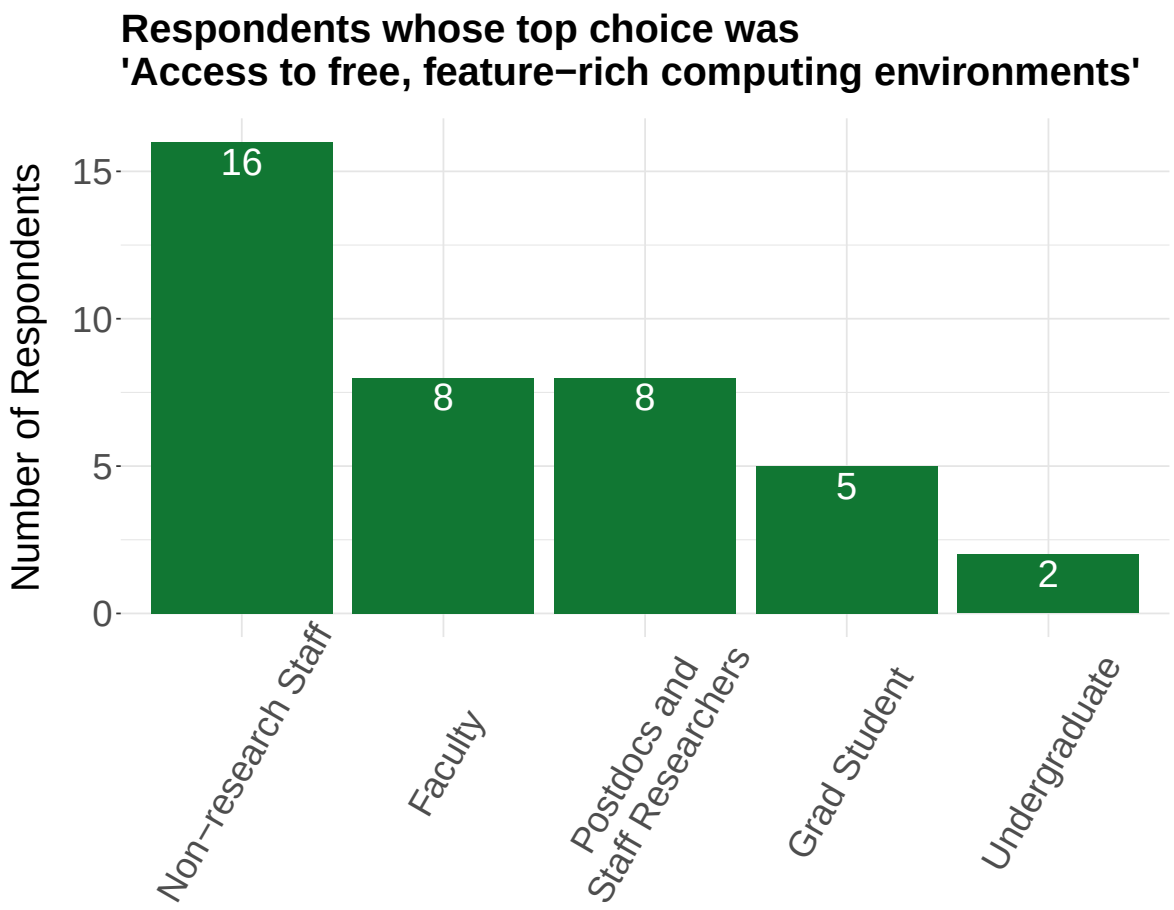
```
            campus                     job_category      favorite_solution
1 UC Santa Barbara                          Faculty  Sustainability grants
2 UC Santa Barbara Postdocs and\nStaff Researchers       Containerization
3 UC Santa Barbara Postdocs and\nStaff Researchers Computing environments
4 UC Santa Barbara                          Faculty  Sustainability grants
5 UC Santa Barbara                          Faculty     Documentation help
7 UC Santa Barbara                          Faculty          Legal support
```

Of the people who selected "Computing environments", what is the distribution of job categories?

```r
compute <- campus_job_fave %>%
  filter(favorite_solution == "Computing environments")
compute_counts <- data.frame(table(compute$job_category))

compute_counts <- compute_counts %>% rename(job_category = Var1, compute = Freq)

compute_counts <- reorder_factor_by_column(
  df = compute_counts,
  factor_col = job_category,
  value_col = compute
)
```

```r
compute_bar <- basic_bar_chart(
  df = compute_counts,
  x_var = "job_category",
  y_var = "compute",
  title = "Respondents whose top choice was\n'Access to free, feature-rich computing environ
```

```
  color_index = 4,
  show_bar_labels = TRUE
)
compute_bar
```



Save the plot if you wish.

```
save_plot("compute_job.tiff", 10, 10, p=compute_bar)
```

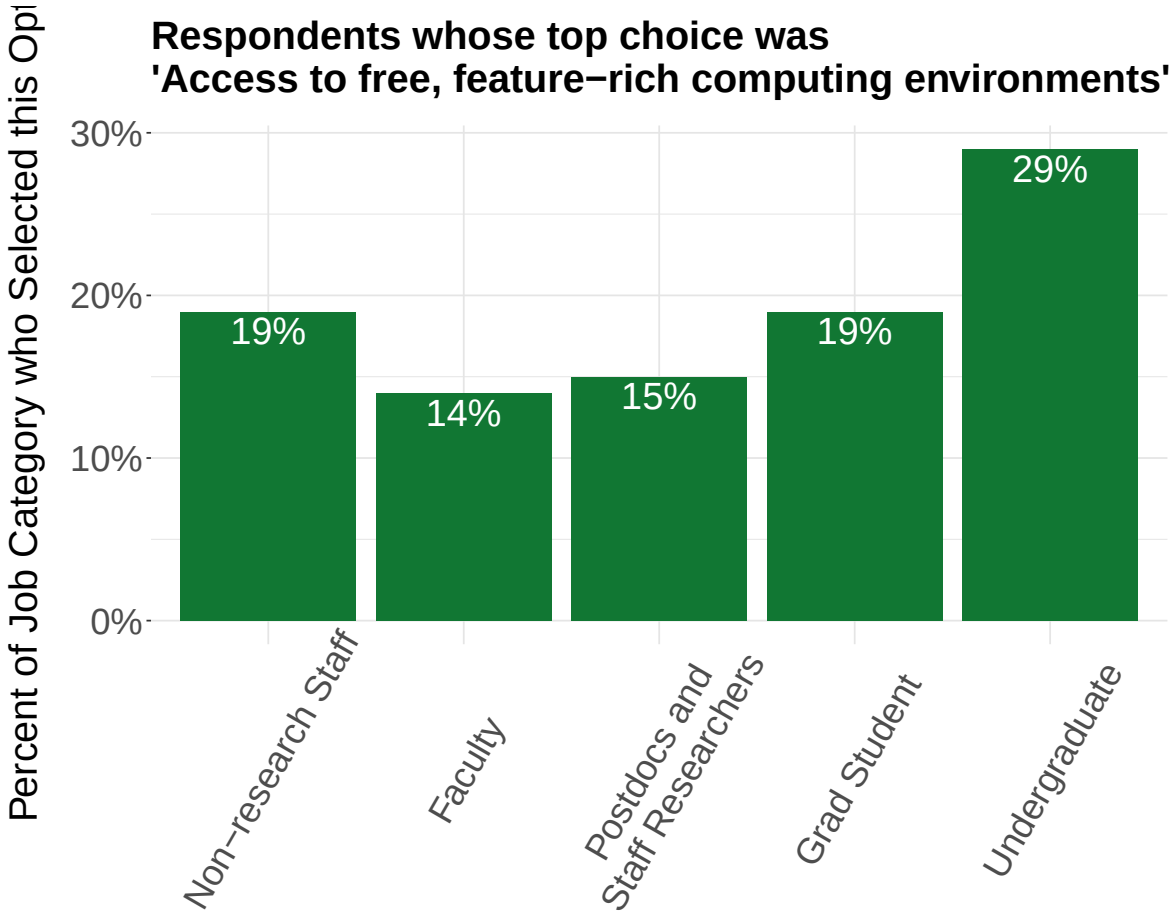
So those are the absolute numbers, but they don't normalize for the sample sizes of the different job categories. The number of non-research staff who voted for computing environments might be high because there are simply a lot of non-research staff in our survey.

```r
total_counts <- data.frame(table(campus_job_fave$job_category))

total_counts <- total_counts %>% rename(job_category = Var1, total = Freq)

proportion_df <- compute_counts %>%
  left_join(total_counts, by = "job_category") %>%
  mutate(proportion = round(compute / total, 2))

proportion_df
```

```
                     job_category compute total proportion
1                         Faculty       8    59       0.14
2                     Grad Student       5    26       0.19
3               Non-research Staff      16    86       0.19
4 Postdocs and\nStaff Researchers       8    55       0.15
5                     Undergraduate       2     7       0.29
```

The previous plot suggested the demand was mostly coming from non-research staff, but that was deceiving, because we do indeed have a lot of non-research staff in our sample. Let's make a plot that is, I think, more informative. This plot shows the percent of people in that job category who selected computing environments as their favorite solution.

```r
compute_bar_prop <- basic_bar_chart(
  df = proportion_df,
  x_var = "job_category",
  y_var = "proportion",
  ylabel = "Percent of Job Category who Selected this Option",
  title = "Respondents whose top choice was\n'Access to free, feature-rich computing environm
  color_index = 4,
  show_bar_labels = TRUE,
  percent = TRUE
)
compute_bar_prop
```



Save the plot if you wish.

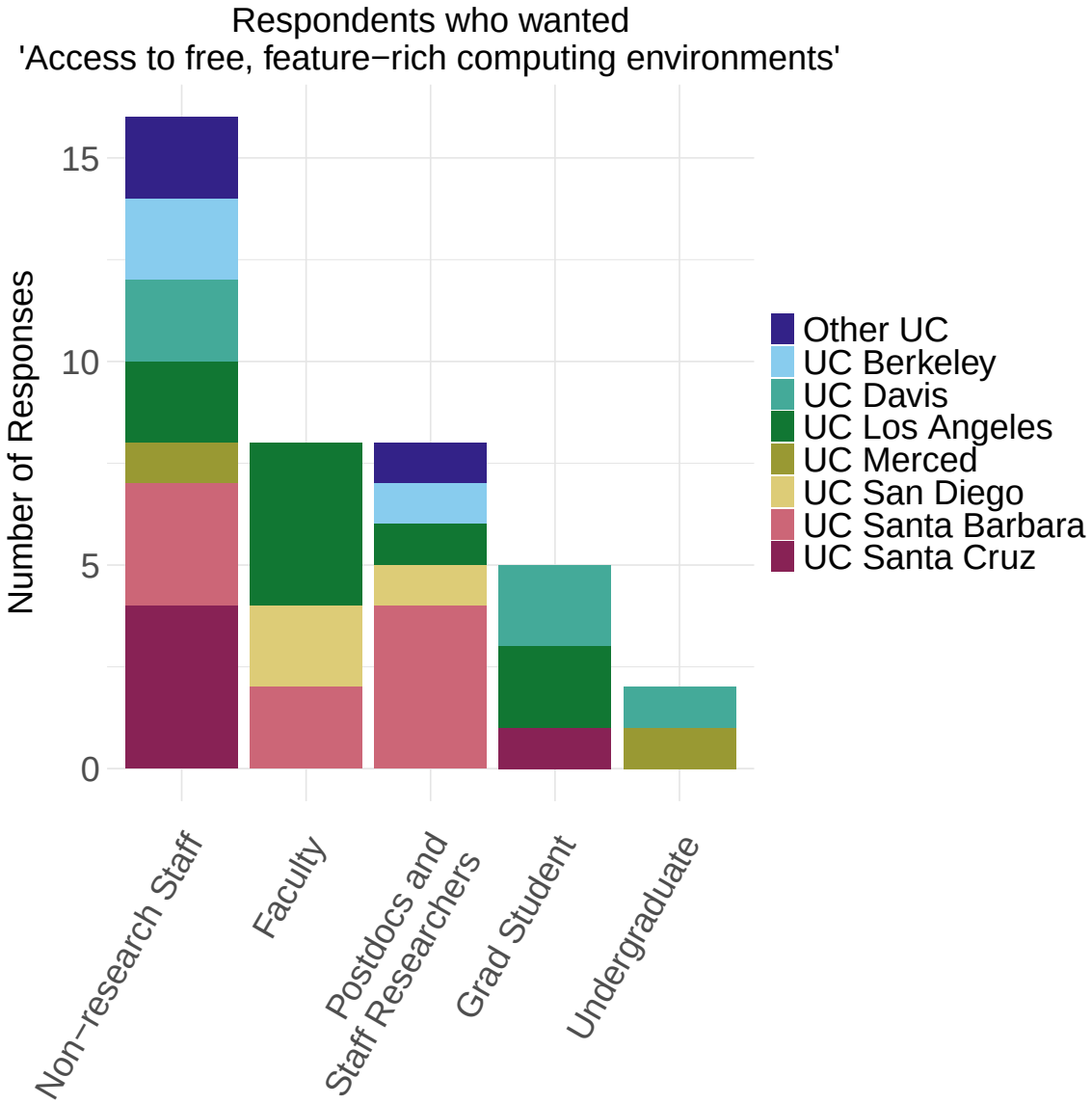
```
save_plot("compute_job_prop.tiff", 10, 10, p=compute_bar_prop)
```

Let's make the same plot, but this time with campus info.

```
compute_counts2 <- compute %>%
  select(-favorite_solution) %>%
  count(
    campus,
    job_category,
    name = "count"
  )

compute_counts2$job_category <- factor(
```

```
  compute_counts2$job_category,
  levels = levels(compute_counts$job_category)
)
```

```
compute_campus_bar <- stacked_bar_chart(
  df = compute_counts2,
  x_var = "job_category",
  y_var = "count",
  fill = "campus",
  title = "Respondents who wanted\n'Access to free, feature-rich computing environments'",
  ylabel = NULL,
  proportional = FALSE
)

compute_campus_bar
```



This one is a bit harder to interpret, because it's a busy plot and the sample sizes are small. Anyway, save the plot if you wish.

```
save_plot("compute_job_campus.tiff", 14, 14, p=compute_campus_bar)
```

Response rates by campus, for "Compute environments"

I'm wondering if there's one or two campuses in particular where compute environments are lacking.

```
compute_counts_campus <- campus_job_fave %>%
  filter(favorite_solution == "Computing environments") %>%
  count(campus, name = "compute_n")

# a scalar
total_compute_votes <- nrow(campus_job_fave %>%
  filter(favorite_solution == "Computing environments"))
```

```
campus_totals <- campus_job_fave %>%
  count(campus, name = "campus_total")

campus_totals <- left_join(campus_totals, compute_counts_campus, by="campus")
campus_totals <- exclude_empty_rows(campus_totals, strict=TRUE)

campus_totals %>% mutate( compute_perc = 100*compute_n / campus_total)
```

```
             campus campus_total compute_n compute_perc
1          Other UC           19         3     15.78947
2       UC Berkeley           26         3     11.53846
3          UC Davis           29         5     17.24138
5    UC Los Angeles           40         9     22.50000
6         UC Merced            8         2     25.00000
7      UC San Diego            9         3     33.33333
9  UC Santa Barbara           61         9     14.75410
10    UC Santa Cruz           32         5     15.62500
```

So, anywhere from 12% to 33% of respondents selected this as their favorite solution, when we break it down by campus. The numbers from UCSD (33%) and UC Merced (25%) should probably be taken with a grain of salt, since those campuses had really low participation rates.

For each job category, what are the top 5 favorite solutions?

From the "choose one" question

First, calculate the result based on Q11, the "choose one" question, where participants had to choose their favorite solution.

```r
job_fave <- campus_job_fave %>% select(-campus)
#Reorder factor levels for plotting
job_fave$job_category <- factor(job_fave$job_category, levels = c(
  "Faculty",
  "Postdocs and\nStaff Researchers",
  "Grad Student",
  "Undergraduate",
  "Non-research Staff"
))

job_fave_counts <- job_fave %>%
  count(
    job_category,
    favorite_solution,
    name = "count"
  )

# 2) For each job_category, keep only the top 5 solutions by count
top_solutions <- job_fave_counts %>%
  group_by(job_category) %>%
  # slice_max() picks the rows with the highest `count`
  slice_max(order_by = count, n = 5, with_ties = TRUE) %>%
  ungroup()

top_solutions
```

```
# A tibble: 32 x 3
   job_category                     favorite_solution      count
   <fct>                            <chr>                  <int>
 1 "Faculty"                        Sustainability grants     24
 2 "Faculty"                        Computing environments     8
 3 "Faculty"                        Help finding funding       6
 4 "Faculty"                        Industry partnerships      5
 5 "Faculty"                        Containerization           3
 6 "Faculty"                        Documentation help         3
 7 "Faculty"                        Education                  3
 8 "Postdocs and\nStaff Researchers" Sustainability grants     16
 9 "Postdocs and\nStaff Researchers" Help finding funding       9
10 "Postdocs and\nStaff Researchers" Computing environments     8
# i 22 more rows
```

This looks like it's worth plotting. Let's go back to the big data frame, since my

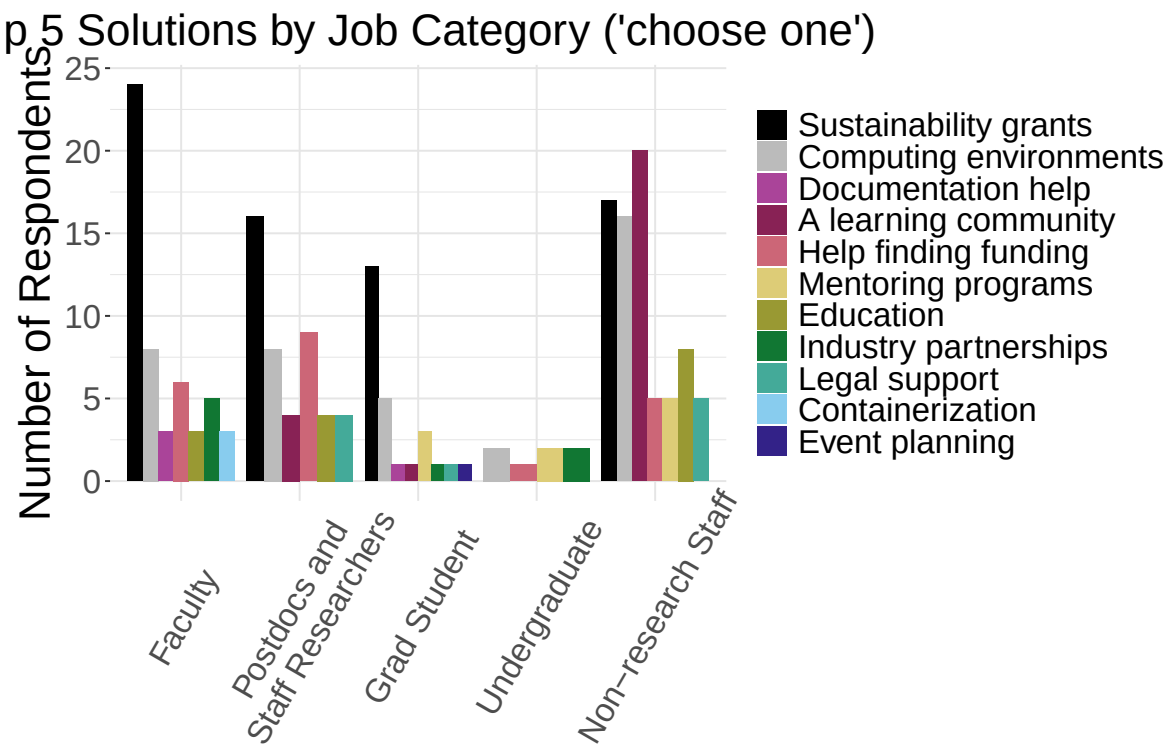
grouped_bar_chart function doesn't want counts (it will count rows itself); drop all job/solution combinations except those that appear in the top_solutions data frame.

```
job_fave_top <- job_fave %>%
  semi_join(
    top_solutions,
    by = c("job_category", "favorite_solution")
  )

head(job_fave_top)
```

```
                    job_category      favorite_solution
1                        Faculty  Sustainability grants
2 Postdocs and\nStaff Researchers Computing environments
3                        Faculty  Sustainability grants
4                        Faculty     Documentation help
5 Postdocs and\nStaff Researchers Computing environments
6                        Faculty Computing environments
```

```
# Reorder factor levels so legend items are in order of appearance
job_fave_top <- job_fave_top %>%
  mutate(favorite_solution = fct_inorder(favorite_solution))
```

```
top_plot <- grouped_bar_chart(
  df = job_fave_top,
  x_var = "job_category",
  fill_var = "favorite_solution",
  title = "Top 5 Solutions by Job Category ('choose one')",
  color_palette = rev(c(COLORS, "#000000")) #from utils.R
)
top_plot
```




```
save_plot("top_solutions_by_job.tiff", 12, 10, p=top_plot)
```

Likert points

Now we ask the same question (top 5 per job category), but we get the result by tallying up the Likert scale "points" from Q10.

```
solutions_job_raw <- cbind(solutions_raw, other_quant$job_category)
# Rename last column
names(solutions_job_raw)[ncol(solutions_job_raw)] <- "job_category"
```

Remove rows that contain any empty entries.

```
nrow(solutions_job_raw)
```

```
[1] 332
```

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

```
[1] 233
```

For visual clarity in our plots, let's combine postdocs and other staff researchers, as well as undergrads and grad students.

```
solutions_job <- solutions_job %>%
  mutate(
    job_category = recode(
      job_category,
      "Post-Doc" = "Postdocs and Staff Researchers",
      "Other research staff" = "Postdocs and Staff Researchers"
    )
  )

solutions_job <- solutions_job %>%
  mutate(
    job_category = recode(
      job_category,
      "Grad Student" = "Students",
      "Undergraduate" = "Students"
    )
  )

solutions_job$participantID <- row.names(solutions_job)

head(solutions_job)
```

```
  Computing environments       Publicity Containerization Documentation help
1            Very useful     Very useful      Very useful        Very useful
2                 Useful     Very useful      Very useful    Not very useful
3            Very useful     Very useful      Very useful        Very useful
4        Not very useful          Useful           Useful        Very useful
5                 Useful Not very useful           Useful        Very useful
7        Not very useful Not very useful      Very useful    Not very useful
  A learning community  Event planning Mentoring programs       Education
1          Very useful     Very useful        Very useful     Very useful
2               Useful  Non-applicable        Very useful     Very useful
3               Useful          Useful             Useful Not very useful
```

```
4      Not very useful          Useful    Not very useful Not very useful
5      Not very useful Not very useful              Useful     Very useful
7      Not very useful Not very useful    Not very useful Not very useful
  Legal support Industry partnerships Sustainability grants
1   Very useful           Very useful           Very useful
2   Very useful                Useful           Very useful
3   Very useful           Very useful           Very useful
4        Useful       Not very useful           Very useful
5        Useful                Useful           Very useful
7   Very useful       Not very useful           Very useful
  Help finding funding                   job_category participantID
1          Very useful                        Faculty             1
2               Useful Postdocs and Staff Researchers             2
3          Very useful Postdocs and Staff Researchers             3
4          Very useful                        Faculty             4
5               Useful                        Faculty             5
7          Very useful                        Faculty             7
```

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

```r
long_data_job <- solutions_job %>%
  pivot_longer(
    cols = -c(participantID, job_category),
    names_to = "solution",
    values_to = "utility"
  )

long_data_job <- long_data_job %>%
  mutate(
    utility_score = recode(
      utility,
      "Non-applicable" = 0L,
      "Not very useful" = 0L,
      "Useful" = 1L,
      "Very useful" = 2L
    )
  )

long_data_job
```

```
# A tibble: 2,796 x 5
   job_category participantID solution              utility     utility_score
```

```
   <chr>        <chr>         <chr>                  <chr>               <int>
 1 Faculty      1             Computing environments Very useful             2
 2 Faculty      1             Publicity              Very useful             2
 3 Faculty      1             Containerization       Very useful             2
 4 Faculty      1             Documentation help     Very useful             2
 5 Faculty      1             A learning community   Very useful             2
 6 Faculty      1             Event planning         Very useful             2
 7 Faculty      1             Mentoring programs     Very useful             2
 8 Faculty      1             Education              Very useful             2
 9 Faculty      1             Legal support          Very useful             2
10 Faculty      1             Industry partnerships  Very useful             2
# i 2,786 more rows
```

Modifying some code I copied from challenges_plus_job.qmd.

```
get_summary_df <- function(job_str, df) {
  res <- df %>%
    filter(job_category == job_str) %>%
    group_by(solution) %>%
    summarise(
      total = sum(utility_score),
      mean = mean(utility_score, na.rm = TRUE)
    ) %>%
    ungroup() %>%
    arrange(desc(total))
  return(res)
}
```

```
jobs_ordered <- c(
  "Faculty",
  "Postdocs and Staff Researchers",
  "Students",
  "Non-research Staff"
)

summary_tables <- lapply(jobs_ordered, function(j) get_summary_df(j, long_data_job))
names(summary_tables) <- jobs_ordered
summary_tables
```

```
$Faculty
# A tibble: 12 x 3
   solution               total  mean
```

```
   <chr>                  <int> <dbl>
 1 Sustainability grants     98 1.66
 2 Help finding funding      85 1.44
 3 Computing environments    75 1.27
 4 Industry partnerships     61 1.03
 5 Publicity                 60 1.02
 6 Containerization          53 0.898
 7 Documentation help        52 0.881
 8 Legal support             52 0.881
 9 A learning community      46 0.780
10 Education                 45 0.763
11 Event planning            41 0.695
12 Mentoring programs        39 0.661

$`Postdocs and Staff Researchers`
# A tibble: 12 x 3
   solution               total  mean
   <chr>                  <int> <dbl>
 1 Sustainability grants     90 1.64
 2 Help finding funding      81 1.47
 3 Computing environments    68 1.24
 4 Publicity                 67 1.22
 5 Documentation help        62 1.13
 6 A learning community      58 1.05
 7 Education                 57 1.04
 8 Industry partnerships     57 1.04
 9 Legal support             56 1.02
10 Event planning            51 0.927
11 Mentoring programs        50 0.909
12 Containerization          49 0.891

$Students
# A tibble: 12 x 3
   solution               total  mean
   <chr>                  <int> <dbl>
 1 Sustainability grants     62  1.88
 2 Computing environments    56  1.70
 3 Help finding funding      55  1.67
 4 Industry partnerships     49  1.48
 5 Mentoring programs        44  1.33
 6 Documentation help        43  1.30
 7 Education                 43  1.30
 8 A learning community      42  1.27
```

```
 9 Publicity                 42  1.27
10 Legal support             41  1.24
11 Containerization          37  1.12
12 Event planning            36  1.09

$`Non-research Staff`
# A tibble: 12 x 3
   solution               total  mean
   <chr>                  <int> <dbl>
 1 A learning community     105 1.22
 2 Sustainability grants    103 1.20
 3 Computing environments   102 1.19
 4 Help finding funding      95 1.10
 5 Legal support             93 1.08
 6 Documentation help        91 1.06
 7 Education                 91 1.06
 8 Mentoring programs        83 0.965
 9 Industry partnerships     65 0.756
10 Containerization          64 0.744
11 Publicity                 63 0.733
12 Event planning            62 0.721
```

Let's plot the top 5. First, a little data wrangling.

```
all_tbl <- bind_rows(summary_tables, .id = "job_category")

top_by_job_points <- all_tbl %>%
  group_by(job_category) %>%
  slice_max(mean, n = 5, with_ties = FALSE) %>%
  ungroup() %>%
  select(job_category, solution, mean) %>%
  mutate(job_category = factor(job_category, levels = jobs_ordered))

# Reorder factor levels for visual clarity

ordered_solns_top_by_points <- c(
  "Sustainability grants",
  "Help finding funding",
  "Computing environments",
  "Industry partnerships",
  "Publicity",
  "Documentation help",
```

```
  "Legal support",
  "A learning community",
  "Mentoring programs"
)

top_by_job_points$solution <- factor(
  top_by_job_points$solution,
  levels = ordered_solns_top_by_points
)
```

Let's add a whitespace in this long job category name

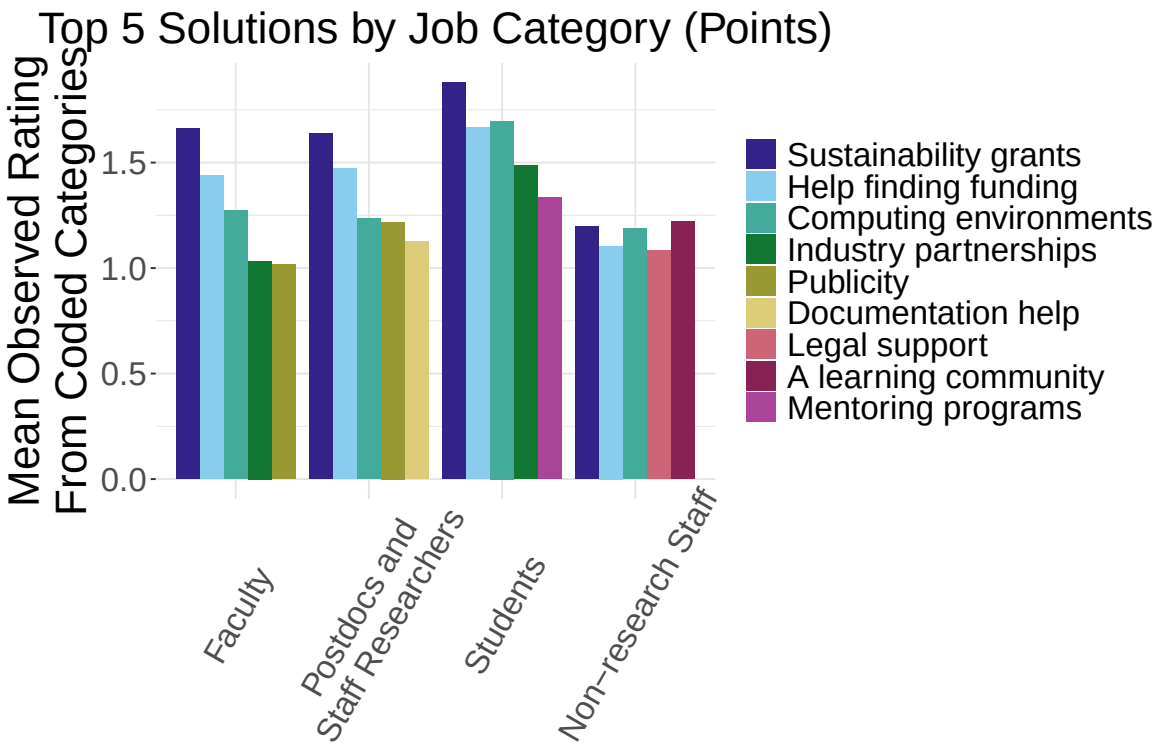
```
top_by_job_points <- top_by_job_points %>%
  mutate(
    job_category = recode(
      job_category,
      "Postdocs and Staff Researchers" = "Postdocs and\nStaff Researchers",
    )
  )
```

Let's hard-code a color palette that is tailored to these data. This will be useful in the next section, when we plot almost the same set of challenges, and we'll want the challenges to correspond to the same colors in the legend.

```
# I'm just including the names here for my own reference,
# but they're not actually used in the code.
# chall_colors <- list(
#   # modified from https://sronpersonalpages.nl/~pault/
#   "Documentation time" = "#332288",
#   "Coding time" = "#88CCEE",
#   "Education time" = "#44AA99",
#   "Educational resources" = "#117733",
#   "Securing funding" = "#999933",
#   "Finding funding" = "#DDCC77",
#   "Managing issues" = "#CC6677",
#   "Attracting users" = "#882255"
# )
```

```
top_points_plot <- ggplot(
  top_by_job_points,
  aes(
```

```
    x = job_category,
    y = mean,
    fill = solution
  )
) +
  geom_col(position = position_dodge()) +
  scale_fill_manual(values = COLORS) +
  labs(
    x = "Job Category",
    y = "Mean Observed Rating\nFrom Coded Categories",
    fill = "Solution",
    title = "Top 5 Solutions by Job Category (Points)"
  ) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_text(size = 24),
    axis.text.x = element_text(angle = 60, vjust = 0.6, size = 18),
    axis.text.y = element_text(size = 18),
    axis.ticks.x = element_blank(),
    legend.title = element_blank(),
    legend.text = element_text(size = 18),
    panel.background = element_blank(),
    panel.grid = element_line(linetype = "solid", color = "gray90"),
    plot.title = element_text(hjust = 0.5, size = 24),
    plot.margin = unit(c(0.3, 0.3, 0.3, 0.3), "cm")
  )
top_points_plot
```




```
save_plot("top5_solutions_by_job_points.tiff", 12, 10, p=top_points_plot)
```

Proportion useful or very useful

As another way of confirming/exploring these trends, let's look at the proportion of each group who said "useful" or "very useful".

```
# Calculate proportion of TRUEs by taking the mean of a logical vector,
# created by %in%.
proportions <- long_data_job %>%
  group_by(job_category, solution) %>%
  summarize(proportion = mean(utility %in% c("Useful", "Very useful"))) %>%
  ungroup()
```

```
`summarise()` has grouped output by 'job_category'. You can override using the
`.groups` argument.
```

```
proportions
```

```
# A tibble: 48 x 3
   job_category solution                proportion
   <chr>        <chr>                        <dbl>
 1 Faculty      A learning community         0.610
 2 Faculty      Computing environments       0.780
 3 Faculty      Containerization             0.593
 4 Faculty      Documentation help           0.593
 5 Faculty      Education                    0.559
 6 Faculty      Event planning               0.508
 7 Faculty      Help finding funding         0.831
 8 Faculty      Industry partnerships        0.644
 9 Faculty      Legal support                0.678
10 Faculty      Mentoring programs           0.525
# i 38 more rows
```

```
top_by_prop <- proportions %>%
  group_by(job_category) %>%
  slice_max(order_by = proportion, n = 5)
```

```
# Filter to include only challenges present in the top n dataframe
filtered_props <- proportions %>%
  semi_join(top_by_prop, by = c("job_category", "solution"))
```

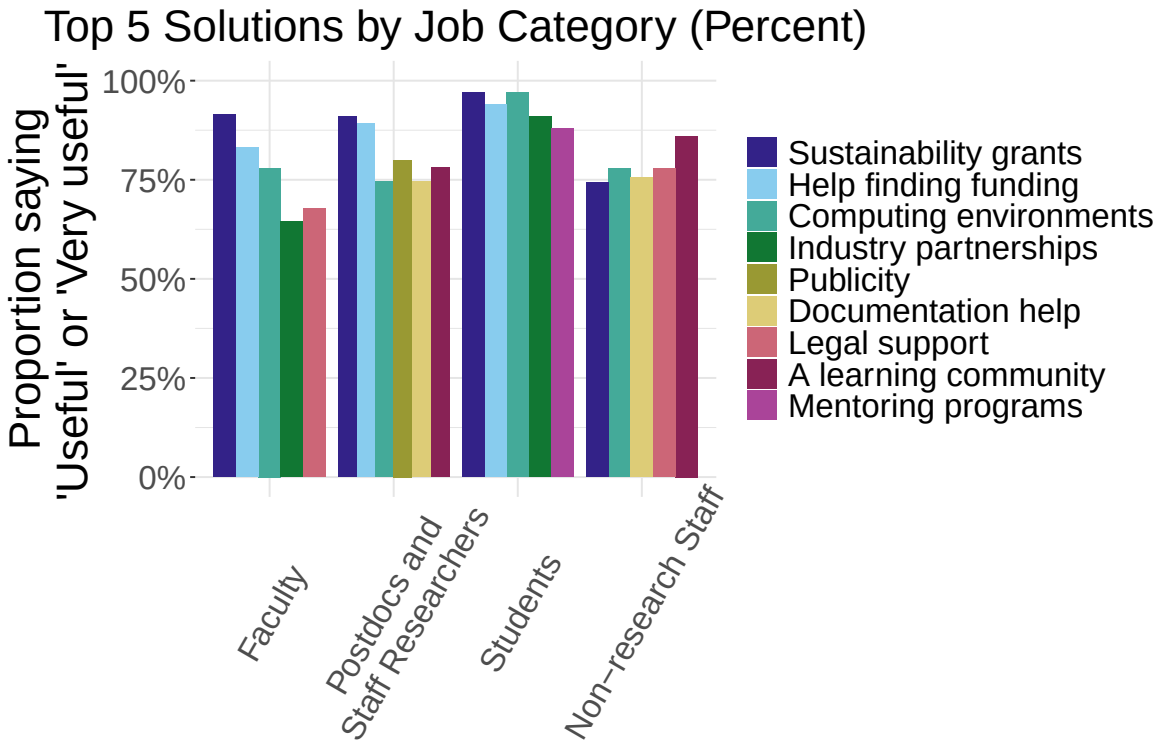
```
# Reorder factor levels
filtered_props$solution <- factor(
  filtered_props$solution,
  levels = ordered_solns_top_by_points
)

filtered_props$job_category <- factor(
  filtered_props$job_category,
  levels = jobs_ordered
)
```

Let's add a whitespace in this long job category name

```
filtered_props <- filtered_props %>%
  mutate(
```

```
    job_category = recode(
      job_category,
      "Postdocs and Staff Researchers" = "Postdocs and\nStaff Researchers",
    )
  )
```

```
top_perc_plot <- ggplot(
  filtered_props,
  aes(
    x = job_category,
    y = proportion,
    fill = solution
  )
) +
  geom_col(position = position_dodge()) +
  scale_y_continuous(labels = scales::percent, limits = c(0, 1)) +
  scale_fill_manual(values = COLORS) +
  labs(
    x = "Job Category",
    y = "Proportion saying\n'Useful' or 'Very useful'",
    fill = "Challenge",
    title = "Top 5 Solutions by Job Category (Percent)"
  ) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_text(size = 24),
    axis.text.x = element_text(angle = 60, vjust = 0.6, size = 18),
    axis.text.y = element_text(size = 18),
    axis.ticks.x = element_blank(),
    legend.title = element_blank(),
    legend.text = element_text(size = 18),
    panel.background = element_blank(),
    panel.grid = element_line(linetype = "solid", color = "gray90"),
    plot.title = element_text(hjust = 0.5, size = 24),
    plot.margin = unit(c(0.3, 0.3, 0.3, 0.3), "cm")
  )
top_perc_plot
```




```
save_plot("top5_solns_by_job_percent.tiff", 12, 10, p=top_perc_plot)
```

A Learning Community

Let's make a plot similar to the one before, where we plotted the percent of folks in each job category who selected "computing environments" as their favorite solution. This time, though, we'll look at "A learning community".

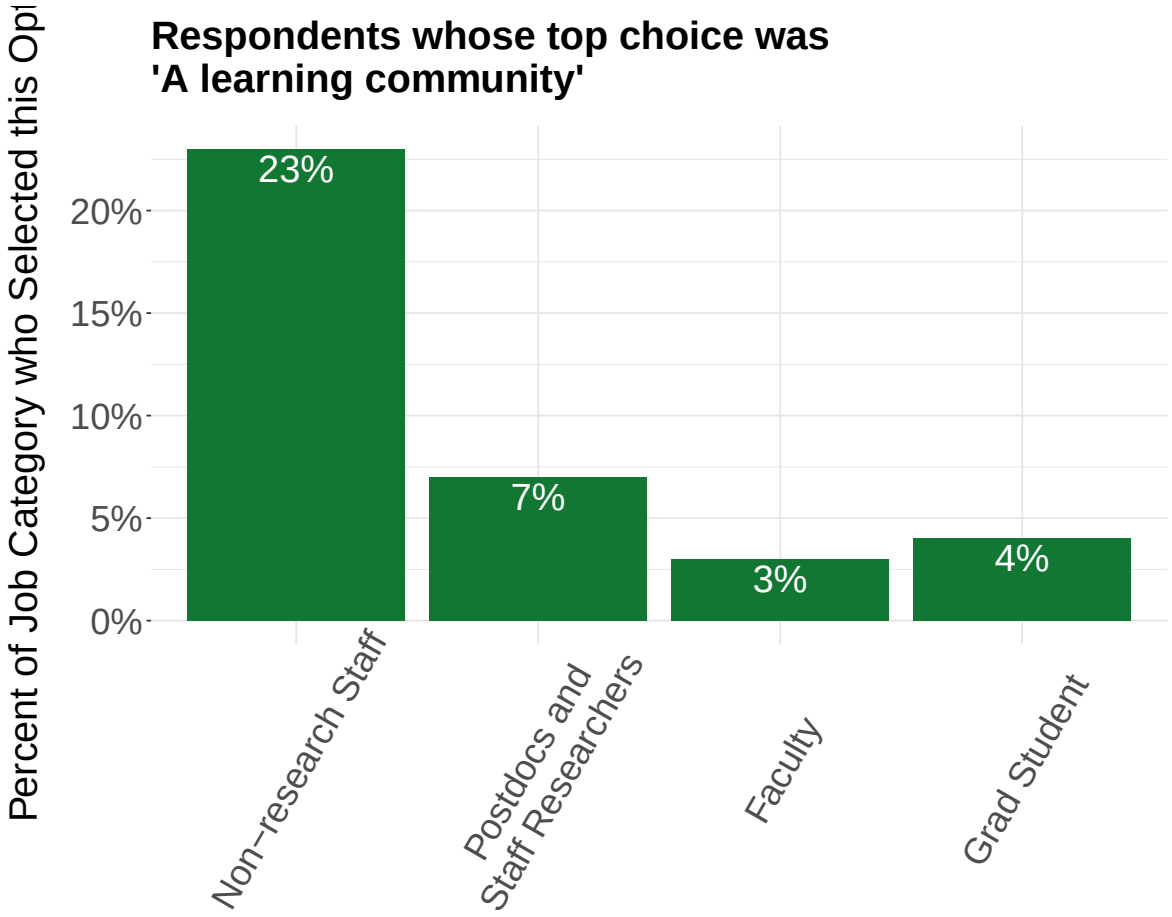
```
learn <- campus_job_fave %>%
  filter(favorite_solution == "A learning community")
learn_counts <- data.frame(table(learn$job_category))

learn_counts <- learn_counts %>% rename(job_category = Var1, learning = Freq)

learn_counts <- reorder_factor_by_column(
  df = learn_counts,
  factor_col = job_category,
```

```
  value_col = learning
)
```

```
total_counts <- data.frame(table(campus_job_fave$job_category))

total_counts <- total_counts %>% rename(job_category = Var1, total = Freq)

proportion_df2 <- learn_counts %>%
  left_join(total_counts, by = "job_category") %>%
  mutate(proportion = round(learning / total, 2))

proportion_df2
```

```
                    job_category learning total proportion
1                        Faculty        2    59       0.03
2                    Grad Student        1    26       0.04
3              Non-research Staff       20    86       0.23
4 Postdocs and\nStaff Researchers        4    55       0.07
```

```
learn_bar_prop <- basic_bar_chart(
  df = proportion_df2,
  x_var = "job_category",
  y_var = "proportion",
  ylabel = "Percent of Job Category who Selected this Option",
  title = "Respondents whose top choice was\n'A learning community'",
  color_index = 4,
  show_bar_labels = TRUE,
  percent = TRUE
)
learn_bar_prop
```




```
save_plot("learn_job_prop.tiff", 12, 10, p=learn_bar_prop)
```

Sustainability grants

Same thing, but now with sustainability grants.

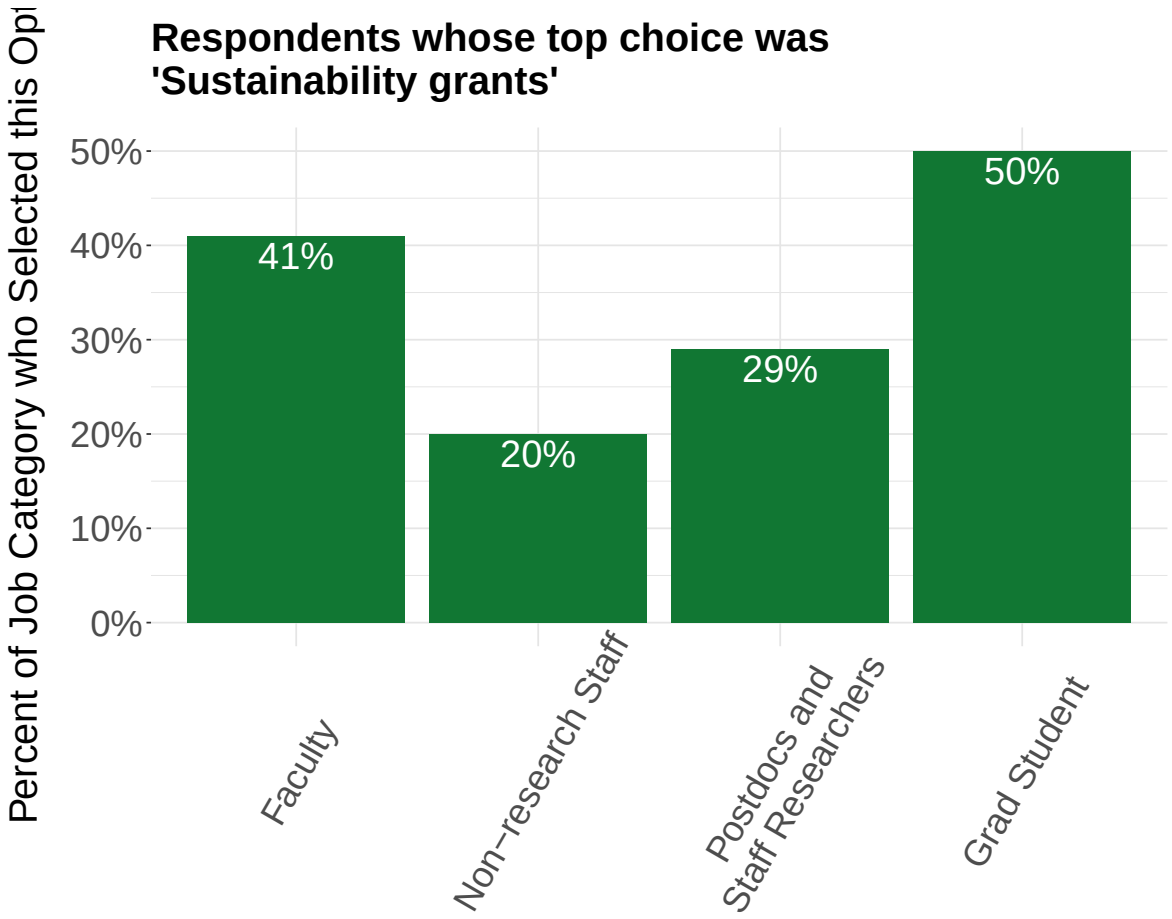
```
grants <- campus_job_fave %>%
  filter(favorite_solution == "Sustainability grants")
grants_counts <- data.frame(table(grants$job_category))

grants_counts <- grants_counts %>% rename(job_category = Var1, grants = Freq)

grants_counts <- reorder_factor_by_column(
```

```
  df = grants_counts,
  factor_col = job_category,
  value_col = grants
)
```

```
total_counts <- data.frame(table(campus_job_fave$job_category))

total_counts <- total_counts %>% rename(job_category = Var1, total = Freq)

proportion_df3 <- grants_counts %>%
  left_join(total_counts, by = "job_category") %>%
  mutate(proportion = round(grants / total, 2))

proportion_df3
```

```
                    job_category grants total proportion
1                        Faculty     24    59       0.41
2                    Grad Student     13    26       0.50
3              Non-research Staff     17    86       0.20
4 Postdocs and\nStaff Researchers     16    55       0.29
```

```
grants_bar_prop <- basic_bar_chart(
  df = proportion_df3,
  x_var = "job_category",
  y_var = "proportion",
  ylabel = "Percent of Job Category who Selected this Option",
  title = "Respondents whose top choice was\n'Sustainability grants'",
  color_index = 4,
  show_bar_labels = TRUE,
  percent = TRUE
)
grants_bar_prop
```




```
save_plot("grants_job_prop.tiff", 12, 10, p=grants_bar_prop)
```

Session Info

```
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;

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
```