

# 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

```
data <- load_qualtrics_data("deidentified_no_qual.tsv")
```

## Wrangle data

```
solutions <- data %>%
  select(
    starts_with("solution_offerings")
  )
```

```
head(solutions)
```

```
  solution_offerings_1 solution_offerings_2 solution_offerings_3
1      Very useful      Very useful      Very useful
2           Useful      Very useful      Very useful
3      Very useful      Very useful      Very useful
4    Not very useful           Useful      Useful
5           Useful    Not very useful      Useful
6
  solution_offerings_4 solution_offerings_5 solution_offerings_6
1      Very useful      Very useful      Very useful
2    Not very useful           Useful    Non-applicable
3      Very useful           Useful      Useful
4      Very useful    Not very useful      Useful
5      Very useful    Not very useful    Not very useful
6
  solution_offerings_7 solution_offerings_8 solution_offerings_9
1      Very useful      Very useful      Very useful
2      Very useful      Very useful      Very useful
3           Useful    Not very useful      Very useful
4    Not very useful    Not very useful      Useful
5           Useful      Very useful      Useful
6
  solution_offerings_10 solution_offerings_11 solution_offerings_12
1      Very useful      Very useful      Very useful
2           Useful      Very useful      Useful
3      Very useful      Very useful      Very useful
4    Not very useful      Very useful      Very useful
5           Useful      Very useful      Useful
6
```

**STOP!!** Presumably, “solution\_offerings\_1” corresponds to the first option, “solution\_offerings\_2” corresponds to the second option, etc., but we still need to check. I am manually comparing the answers in this data frame to those in the Qualtrics interface, which shows the whole response, i.e. “Access to free, feature-rich computing environments”, not just “solution\_offerings\_1”. To be extra confident that I am comparing the same rows between the two tables, I am looking at responses associated with a particular email. After this code chunk, I go back to using the data frame that doesn’t contain the emails.

Since this code only needed to be run once, I’ve commented it out.

```
# pii <- load_qualtrics_data("pii.tsv")
# emails <- pii %>%
#   select(starts_with("stay_in_touch_email"))

# t <- cbind(emails, solutions)

# # Next, I run this line repeatedly with different emails,
# # to make sure that this person's response to "solution_offerings_1"
# # matches their response to "Access to free, feature-rich computing environments", etc.
# subset(t, startsWith(stay_in_touch_email, "PERSON_NAME_HERE"))
```

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

```
codes <- c(
  "Computing environments" = "solution_offerings_1",
  "Publicity" = "solution_offerings_2",
  "Containerization" = "solution_offerings_3",
  "Documentation help" = "solution_offerings_4",
  "A learning community" = "solution_offerings_5",
  "Event planning" = "solution_offerings_6",
  "Mentoring programs" = "solution_offerings_7",
  "Education" = "solution_offerings_8",
  "Legal support" = "solution_offerings_9",
  "Industry partnerships" = "solution_offerings_10",
  "Sustainability grants" = "solution_offerings_11",
  "Help finding funding" = "solution_offerings_12"
)
solutions <- rename(solutions, any_of(codes))
```

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

```
nrow(solutions)
```

```
[1] 332
```

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

[1] 233

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

```
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
```

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 mode
- The standard deviation

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

summary_df <- long_data %>%
  group_by(solution) %>%
  summarise(
    total = sum(utility_score),
    mean = mean(utility_score, na.rm = TRUE),
    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 5
  solution          total mean mode st_dev
  <chr>          <int> <dbl> <int> <dbl>
1 Sustainability grants    353 1.52     2 0.732
2 Help finding funding     316 1.36     2 0.764
3 Computing environments   301 1.29     2 0.783
4 A learning community     251 1.08     1 0.733
5 Documentation help       248 1.06     1 0.788
6 Legal support            242 1.04     1 0.762
7 Education               236 1.01     1 0.801
8 Industry partnerships    232 0.996     0 0.838
9 Publicity                232 0.996     1 0.817
10 Mentoring programs      216 0.927     1 0.776
```

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

This analysis doesn't seem as interesting as it was for the challenges. Industry partnerships, Containerization, and Publicity all show high variance/stdev. These were also somewhat less popular.

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.

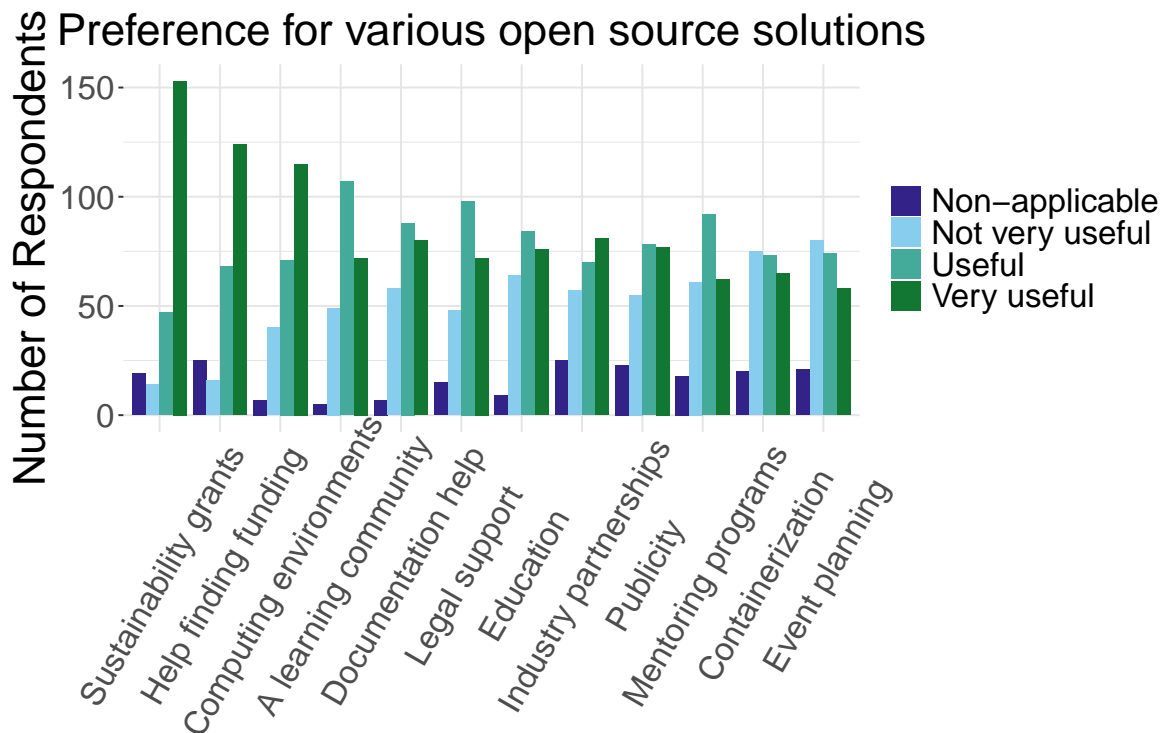
## Plot the distributions

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.

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

codes2 <- c(
  "Access to free" = "Computing environments",
  "Assistance promoting your" = "Publicity",
  "Assistance creating" = "Containerization",
  "Assistance writing" = "Documentation help",
  "An open source discussion" = "A learning community",
  "Assistance with event" = "Event planning",
  "A mentor" = "Mentoring programs",
  "Educational materials" = "Education",
  "Legal and licensing" = "Legal support",
  "Assistance building industry" = "Industry partnerships",
  "Dedicated grants" = "Sustainability grants",
  "Assistance identifying potential" = "Help finding funding"
)

favorites <- shorten_long_responses(favorites, codes2)

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
)

faves_plot <- basic_bar_chart(
  df = fav_to_plot,
  x_var = "Var1",
```



```

y_var = "Freq",
title = "Participant's favorite solution",
show_axis_title_y = FALSE,
ylabel = "Number of Respondents",
show_bar_labels = TRUE,
color_index = 7,
horizontal = TRUE
)

faves_plot

```



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

Save the plot if you wish.

```

#save_plot("fave_solutions_simple.tiff", 10, 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 <- data %>%
  select(
    starts_with("campus") | starts_with("job_category"), favorite_solution
  )

campus_job_fave <- exclude_empty_rows(campus_job_fave)

# Clean up this one long job name:
# "Other research staff (e.g., research scientist, research software engineer)"
campus_job_fave$job_category <- gsub(
  "^Other.*",
  "Other research staff",
  campus_job_fave$job_category
)

campus_job_fave <- exclude_empty_rows(campus_job_fave, strict = TRUE)
campus_job_fave <- shorten_long_responses(campus_job_fave, codes2)

# 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	Staff Researchers	Containerization
3 UC Santa Barbara Postdocs and	Staff Researchers	Computing environments
4 UC Santa Barbara	Faculty	Sustainability grants
5 UC Santa Barbara	Faculty	Documentation help
7 UC Santa Barbara	Faculty	Legal support

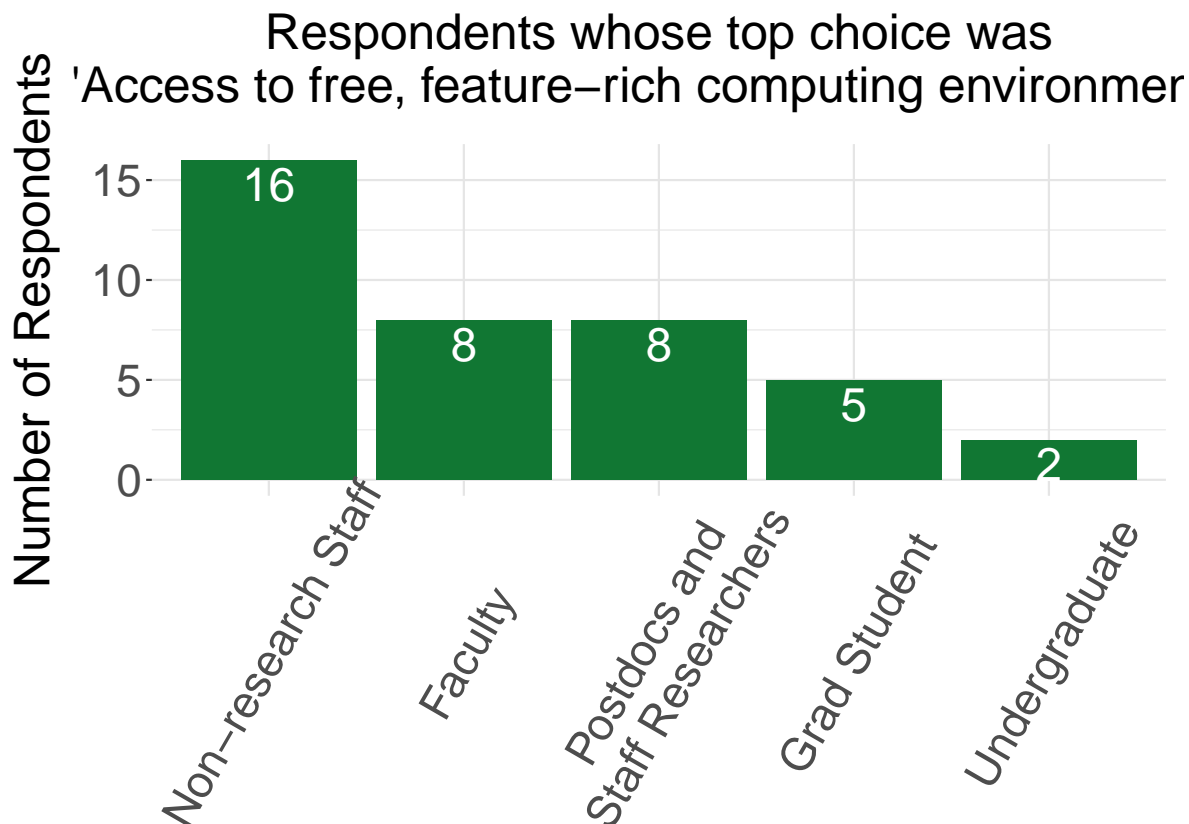
Of the people who selected “Computing environments”, what is the distribution of job categories?

```
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
)

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 environm",
  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.

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

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

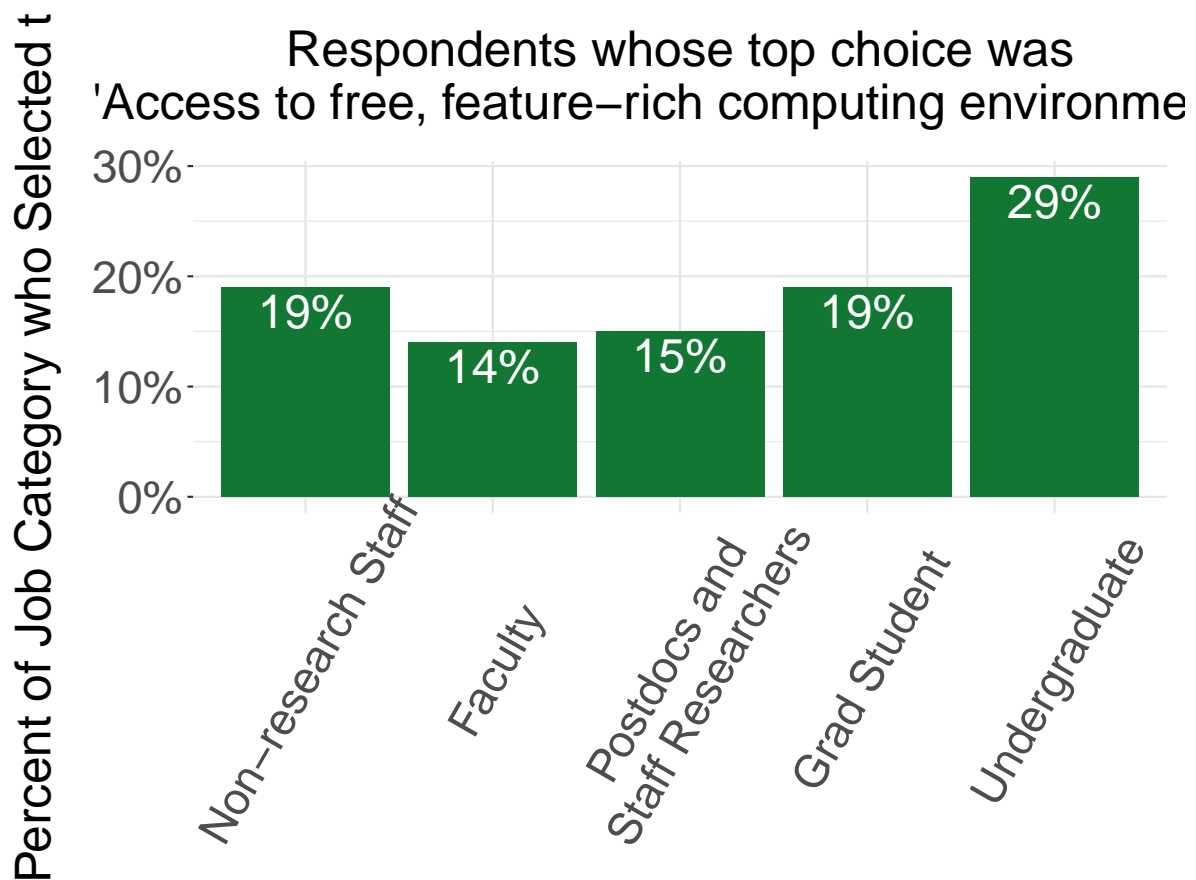
proportion_df <- compute_counts %>%
  left_join(total_counts, by = "job_category") %>%
  mutate(proportion = round(compute / total, 2))

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 Staff 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.

```
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 environ",
  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)
)

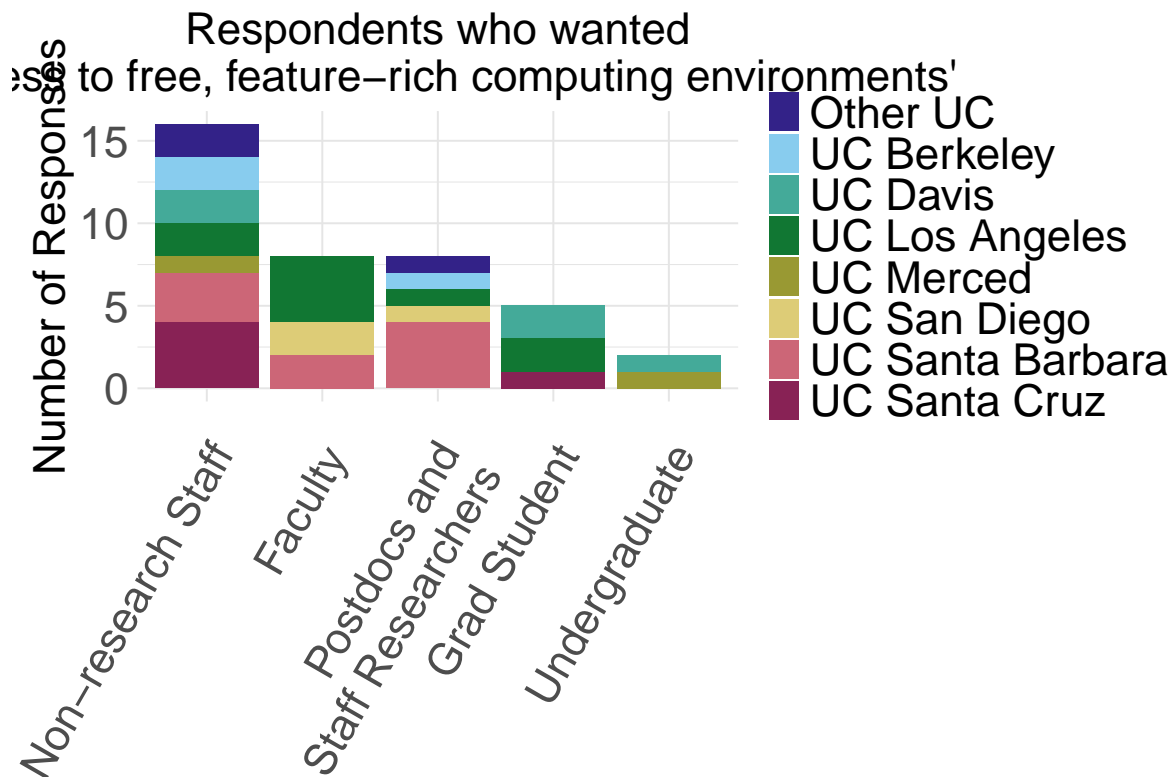
```

```

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
```



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 3 favorite solutions?**



```

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 3 solutions by count
top3_solutions <- job_fave_counts %>%
  group_by(job_category) %>%
  # slice_max() picks the rows with the highest `count`
  slice_max(order_by = count, n = 3, with_ties = TRUE) %>%
  ungroup()

top3_solutions

```

# A tibble: 15 x 3

	job_category <fct>	favorite_solution <chr>	count <int>
1	"Faculty"	Sustainability grants	24
2	"Faculty"	Computing environments	8
3	"Faculty"	Help finding funding	6
4	"Postdocs and\nStaff Researchers"	Sustainability grants	16
5	"Postdocs and\nStaff Researchers"	Help finding funding	9
6	"Postdocs and\nStaff Researchers"	Computing environments	8
7	"Grad Student"	Sustainability grants	13
8	"Grad Student"	Computing environments	5
9	"Grad Student"	Mentoring programs	3
10	"Undergraduate"	Computing environments	2
11	"Undergraduate"	Industry partnerships	2
12	"Undergraduate"	Mentoring programs	2
13	"Non-research Staff"	A learning community	20

14 "Non-research Staff"	Sustainability grants	17
15 "Non-research Staff"	Computing environments	16

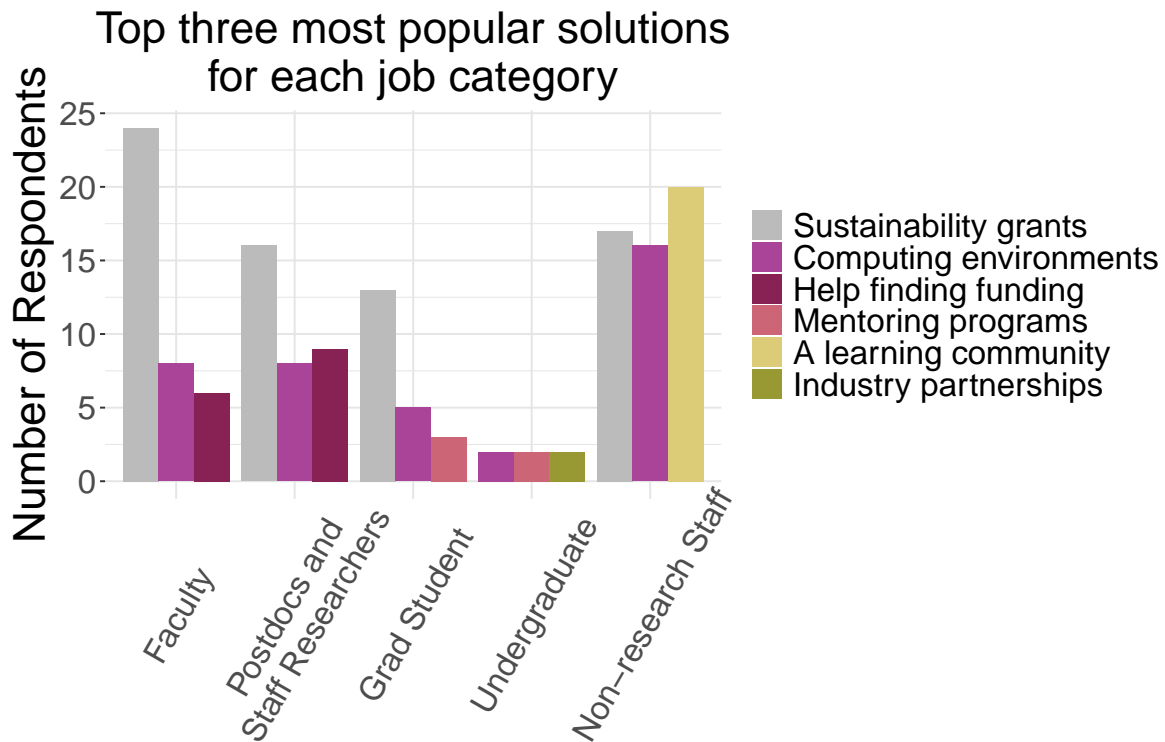
This looks like it's worth plotting. Let's go back to the big data frame, since my `grouped_bar_chart` function doesn't want counts (it will count rows itself); drop all job/solution combinations except those that appear in the `top3_solutions` data frame.

```
job_fave_top3 <- job_fave %>%
  semi_join(
    top3_solutions,
    by = c("job_category", "favorite_solution")
  )
head(job_fave_top3)
```

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

```
# Reorder factor levels so legend items are in order of appearance
job_fave_top3 <- job_fave_top3 %>%
  mutate(favorite_solution = fct_inorder(favorite_solution))

top3_plot <- grouped_bar_chart(
  df = job_fave_top3,
  x_var = "job_category",
  fill_var = "favorite_solution",
  title = "Top three most popular solutions\nfor each job category",
  color_palette = rev(colors)
)
top3_plot
```



```
#save_plot("top3_solutions_by_job.tiff", 12, 10, p=top3_plot)
```

So, I think these are the takeaways:

- Dedicated grants for OS project sustainability is the most popular solution. This solution was in the top3 for all but undergrads.
- The other top solutions depend on how you look at the data. For non-research staff, the most popular solution is a learning community, though grants and access to free, feature-rich computing environments are close behind.
- I was surprised that access to computing environments was in second place. Upon inspection, this seems to be because this choice is popular among non-research staff, and we had a lot of non-research staff in our participant pool. About 12-33% of respondents said this was their top choice, depending on the campus.
- Undergraduates were the only group in which nobody selected grants as their top choice.
- Grad students and undergraduates were the only groups for whom a mentoring program was in their top 3.
- Researchers and non-research staff have very distinct needs.

## Session Info

```
sessionInfo()
```

```
R version 4.4.2 (2024-10-31)
Platform: aarch64-apple-darwin20
Running under: macOS Sequoia 15.4.1
```

```
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; I
```

```
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      stats      graphics  grDevices datasets  utils      methods
```

```
[8] base
```

```
other attached packages:
```

```
[1] treemap_2.4-4      tidyr_1.3.1        stringr_1.5.1
[4] scales_1.4.0       readr_2.1.5        pwr_1.3-0
[7] patchwork_1.3.0    mvabund_4.2.1      languageserver_0.3.16
[10] here_1.0.1         gtools_3.9.5       fpc_2.2-13
[13] forcats_1.0.0      factoextra_1.0.7   ggplot2_3.5.2
[16] dplyr_1.1.4        corrplot_0.95      cluster_2.1.8.1
```

```
loaded via a namespace (and not attached):
```

```
[1] gtable_0.3.6      xfun_0.52          ggrepel_0.9.6
[4] processx_3.8.6    lattice_0.22-6     callr_3.7.6
[7] tzdb_0.5.0        vctrs_0.6.5        ps_1.9.1
[10] generics_0.1.4    stats4_4.4.2       parallel_4.4.2
[13] flexmix_2.3-20    tibble_3.2.1       DEoptimR_1.1-3-1
[16] pkgconfig_2.0.3   data.table_1.17.6  RColorBrewer_1.1-3
[19] lifecycle_1.0.4   compiler_4.4.2     farver_2.1.2
[22] statmod_1.5.0     httpuv_1.6.16      htmltools_0.5.8.1
[25] class_7.3-22      yaml_2.3.10        later_1.4.2
[28] pillar_1.10.2     prabclus_2.3-4     MASS_7.3-61
```

[31]	diptest_0.77-1	mclust_6.1.1	mime_0.13
[34]	robustbase_0.99-4-1	tidyselect_1.2.1	digest_0.6.37
[37]	stringi_1.8.7	purrr_1.0.4	kernlab_0.9-33
[40]	labeling_0.4.3	rprojroot_2.0.4	fastmap_1.2.0
[43]	grid_4.4.2	colorspace_2.1-1	cli_3.6.5
[46]	magrittr_2.0.3	utf8_1.2.5	withr_3.0.2
[49]	promises_1.3.3	tweedie_2.3.5	rmarkdown_2.29
[52]	igraph_2.1.4	nnet_7.3-19	modeltools_0.2-24
[55]	hms_1.1.3	shiny_1.11.0	evaluate_1.0.3
[58]	knitr_1.50	rlang_1.1.6	Rcpp_1.0.14
[61]	xtable_1.8-4	gridBase_0.4-7	glue_1.8.0
[64]	xml2_1.3.8	renv_1.1.4	jsonlite_2.0.0
[67]	R6_2.6.1		