

Demographics: descriptive statistics

Overview

Let's look at survey participation rates across various groups (demographics). These are mostly just basic descriptive statistics, though there are a couple plots and a z-test relating to aspiring vs. experienced contributors.

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

```
other_quant <- load_qualtrics_data("clean_data/other_quant.tsv")
status <- load_qualtrics_data("clean_data/contributor_status_Q3.tsv")
qual <- load_qualtrics_data("qual_responses.tsv")

raw_data <- cbind(status, other_quant)
nrow(raw_data)
```

```
[1] 332
```

```
head(raw_data)
```

	Past	Future	campus	favorite_solution	field_of_study
1	True	True	UC Santa Barbara	Sustainability grants	Math and CS
2	True	True	UC Santa Barbara	Containerization	Life sciences
3	True	True	UC Santa Barbara	Computing environments	Humanities
4	True	True	UC Santa Barbara	Sustainability grants	Math and CS
5	True	True	UC Santa Barbara	Documentation help	Life sciences
6	False	True	UC Santa Barbara		Math and CS

	job_category	staff_categories
1	Faculty	
2	Post-Doc	
3	Other research staff	
4	Faculty	
5	Faculty	
6	Other research staff	

Filter out people who were neither past nor future contributors. We'll use this filtered data frame moving forward.

```
# Filter duds
data <- raw_data %>%
  filter(!(Past == "" | Future == "")) %>%
  filter(!(Past == "False" & Future == "False"))
```

Experienced vs aspiring

First, let's see how many experienced and aspiring open source contributors took the survey.

Experienced:

```
total_expd <- nrow(subset(raw_data, Past=="True"))
total_expd
```

```
[1] 233
```

Aspiring:

```
total_asp <- nrow(subset(raw_data, Past=="False" & Future=="True"))
total_asp
```

```
[1] 61
```

```
status_data <- data %>%
  mutate(status = if_else(Past == "True", "Experienced", "Aspiring")) %>%
  select(job_category, status)

stat_sum <- data.frame(
  ftable(xtabs(~ job_category + status, data = status_data))
)
subset(stat_sum, status == "Aspiring") %>% arrange(desc(Freq))
```

	job_category	status	Freq
1	Non-research Staff	Aspiring	20
2	Grad Student	Aspiring	17
3	Other research staff	Aspiring	9
4	Faculty	Aspiring	7
5	Undergraduate	Aspiring	6
6	Post-Doc	Aspiring	2

```
subset(stat_sum, status == "Experienced") %>% arrange(desc(Freq))
```

	job_category	status	Freq
1	Non-research Staff	Experienced	86
2	Faculty	Experienced	59
3	Other research staff	Experienced	40
4	Grad Student	Experienced	26
5	Post-Doc	Experienced	15
6	Undergraduate	Experienced	7

Here we see that we have only 7 experienced undergraduate contributors and only 15 experienced postdocs.

Experienced vs. aspiring by job: plot

Prepare data for plotting

```

sj_counts <- status_data %>% group_by(job_category, status) %>% count()

# Reorder factor levels by the highest proportion of experienced contributors
ordered_jobs <- sj_counts %>%
  group_by(job_category) %>%
  summarise(
    Aspiring = n[status=="Aspiring"],
    Experienced = n[status=="Experienced"],
    .groups = "drop"
  ) %>%
  mutate(exp_to_asp = Experienced / Aspiring) %>%
  arrange(desc(exp_to_asp)) %>%
  pull(job_category)

sj_counts$job_category <- factor(sj_counts$job_category, levels = ordered_jobs)

```

Plot

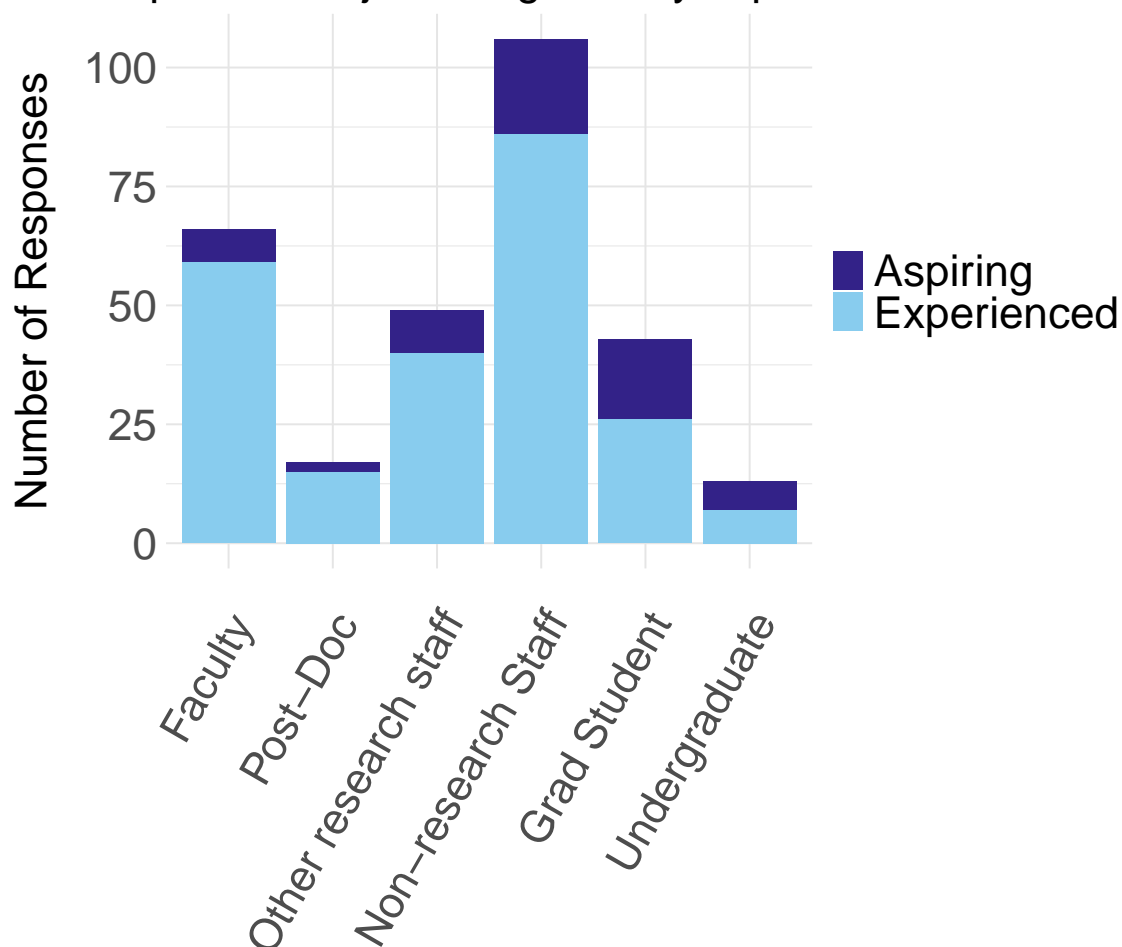
```

stack <- stacked_bar_chart(
  df = sj_counts,
  x_var = "job_category",
  y_var = "n",
  fill = "status",
  title = "Composition of job categories by experience")

stack

```

Composition of job categories by experience

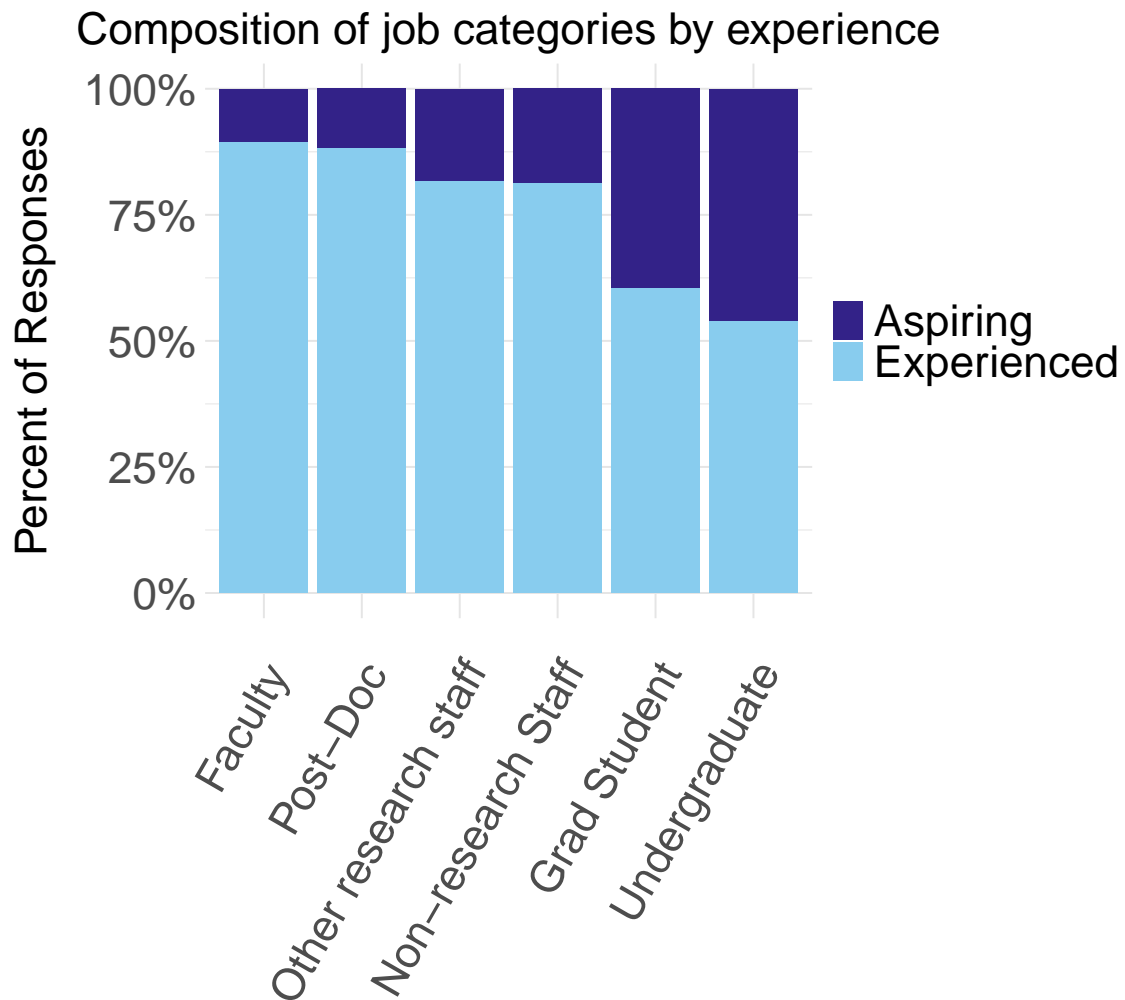


```
save_plot("future_contribs_stack.tiff", 12, 9, p=stack)
```

```
stack_prop <- stacked_bar_chart(  
  df = sj_counts,  
  x_var = "job_category",  
  y_var = "n",  
  ylabel = "Percent of Responses",  
  fill = "status",  
  title = "Composition of job categories by experience",  
  proportional = TRUE)  
  
stack_prop <- stack_prop +
```

```
scale_y_continuous(labels = scales::percent)

stack_prop
```



```
save_plot("future_contribs_stack_prop.tiff", 12, 9, p=stack_prop)
```

Experienced vs. aspiring by job: stats

I think this might be easier to get a handle on if we combine some of these groups.

```

sj_counts_relabeled <- sj_counts %>%
  mutate(
    job_category = case_when(
      job_category %in% c("Other research staff", "Post-Doc") ~
        "Post-docs and staff researchers",
      job_category %in% c("Grad Student", "Undergraduate") ~ "Students",
      TRUE ~ job_category
    )
  ) %>%
  group_by(job_category, status) %>%
  summarise(n = sum(n, na.rm = TRUE), .groups = "drop")

asp <- subset(sj_counts_relabeled, status == "Aspiring") %>% arrange(desc(n))
expd <- subset(sj_counts_relabeled, status == "Experienced") %>% arrange(desc(n))

asp

```

```

# A tibble: 4 x 3
  job_category      status      n
  <chr>           <chr>   <int>
1 Students       Aspiring    23
2 Non-research Staff Aspiring    20
3 Post-docs and staff researchers Aspiring    11
4 Faculty        Aspiring     7

```

```
expd
```

```

# A tibble: 4 x 3
  job_category      status      n
  <chr>           <chr>   <int>
1 Non-research Staff Experienced    86
2 Faculty         Experienced    59
3 Post-docs and staff researchers Experienced    55
4 Students       Experienced    33

```

Let's look at the proportions, which will make this even easier to see.

```

sj_counts_prop <- sj_counts %>%
  ungroup() %>% # drop existing grouping
  group_by(job_category) %>% # group only by job_category
  mutate(
    prop = n / sum(n), # proportion, for statistics
    pct = round(prop * 100, 1) # percent, easier to read
  ) %>%
  ungroup()

subset(sj_counts_prop, status == "Aspiring")

```

```

# A tibble: 6 x 5
  job_category      status      n prop  pct
  <fct>           <chr>   <int> <dbl> <dbl>
1 Faculty         Aspiring     7 0.106  10.6
2 Grad Student    Aspiring    17 0.395  39.5
3 Non-research Staff Aspiring    20 0.189  18.9
4 Other research staff Aspiring     9 0.184  18.4
5 Post-Doc        Aspiring     2 0.118  11.8
6 Undergraduate   Aspiring     6 0.462  46.2

```

Undergrads and grad students both have a lot of aspiring contributors—40ish%, twice as much the next highest proportion which is staff.

Quick 2-proportion z-test

Can we do a quick z-test to check whether aspiring contributors make up a higher proportion of students than they do of nr staff, the next-highest proportion?

First, combine students for more statistical power. (Copying the code from previous cell, just on a relabeled data frame.)

```

sj_counts_prop2 <- sj_counts_relabeled %>%
  ungroup() %>%
  group_by(job_category) %>%
  mutate(
    prop = n / sum(n),
    pct = round(prop * 100, 1)
  ) %>%
  ungroup()

```



```
subset(sj_counts_prop2, status == "Aspiring")
```

```
# A tibble: 4 x 5
```

	job_category	status	n	prop	pct
	<chr>	<chr>	<int>	<dbl>	<dbl>
1	Faculty	Aspiring	7	0.106	10.6
2	Non-research Staff	Aspiring	20	0.189	18.9
3	Post-docs and staff researchers	Aspiring	11	0.167	16.7
4	Students	Aspiring	23	0.411	41.1

Let's start with a power analysis to see whether we have an adequate sample size. I could make this code more concise, but I'm sort of copy-pasting bits from other notebooks here.

```
n_stud <- sum(subset(sj_counts_prop2, job_category == "Students")$n)
n_stud_asp <- subset(
  sj_counts_prop2,
  job_category == "Students" & status == "Aspiring"
)$n

n_staff <- sum(subset(sj_counts_prop2, job_category == "Non-research Staff")$n)
n_staff_asp <- subset(
  sj_counts_prop2,
  job_category == "Non-research Staff" & status == "Aspiring"
)$n

# Sanity check
n_stud
```

```
[1] 56
```

```
n_stud_asp
```

```
[1] 23
```

```
n_staff
```

```
[1] 106
```

```
n_staff_asp
```

```
[1] 20
```

```
p_stud_asp <- n_stud_asp / n_stud  
p_staff_asp <- n_staff_asp / n_staff  
  
p_stud_asp
```

```
[1] 0.4107143
```

```
p_staff_asp
```

```
[1] 0.1886792
```

Calculate Cohen's h, the effect size.

```
h <- pwr::ES.h(p_stud_asp, p_staff_asp)
```

Now, what ratio of students to nr staff is needed to achieve 80% power? This one-sided test allows us to specify our unequal group sizes.

```
pwr::pwr.2p2n.test(  
  h = h,  
  n1 = n_stud,  
  sig.level = 0.05,  
  power = 0.8,  
  alternative = "greater"  
)
```

difference of proportion power calculation for binomial distribution (arcsine transform)

```
      h = 0.4925795  
      n1 = 56  
      n2 = 46.75545  
sig.level = 0.05  
power = 0.8  
alternative = greater
```

NOTE: different sample sizes

So we would need 46 nr staff to achieve 80% power.

```
n_staff
```

```
[1] 106
```

We have 106!

Now proceed with the z-test.

```
# Perform the one-sided prop test (testing if group1 > group2)
stats::prop.test(
  x = c(n_stud_asp, n_staff_asp),
  n = c(n_stud, n_staff),
  alternative = "greater",
)
```

2-sample test for equality of proportions with continuity correction

```
data:  c(n_stud_asp, n_staff_asp) out of c(n_stud, n_staff)
X-squared = 8.161, df = 1, p-value = 0.00214
alternative hypothesis: greater
95 percent confidence interval:
 0.08348806 1.00000000
sample estimates:
   prop 1    prop 2 
0.4107143 0.1886792
```

Sweet. The difference in proportions is statistically significant, according to a simple z-test.

Campus

I already learned while plotting the data that UCSB and UCLA are overrepresented. What proportion of respondents came from these two schools?

```
unique(data$campus)
```

```
[1] "UC Santa Barbara" "UC San Diego"      "UC Los Angeles"    "UC Davis"
[5] "UC Santa Cruz"    "UC San Francisco"  "UC Berkeley"       "Other UC"
[9] "UC Irvine"        "UC Merced"
```

First, a quick glance at the raw data to see how many non-UC respondents we got.

```
nrow(raw_data)
```

```
[1] 332
```

```
nrow(
  subset(raw_data, campus != "I'm not affiliated with UC")
)
```

```
[1] 330
```

Only 2 respondents were not UC affiliates.

```
campus_count <- data.frame(table(data$campus))
names(campus_count) <- c("Campus", "Count")
total <- sum(campus_count$Count)
ucsb <- subset(campus_count, Campus=="UC Santa Barbara")["Count"]
ucla <- subset(campus_count, Campus=="UC Los Angeles")["Count"]
ucsb + ucla
```

```
[1] 139
```

```
total
```

```
[1] 294
```

```
round((ucsb + ucla) / total * 100, digits = 1)
```

```
[1] 47.3
```

So 47% of respondents came from these two campuses.

Field of study

How many respondents were from STEM, social science, and humanities?

```
# Remove people who didn't answer this question--non-research staff
tmp <- data$field_of_study[nzchar(data$field_of_study)]
field_count <- data.frame(table(tmp))
names(field_count) <- c("Field", "Count")
total <- sum(field_count$Count)
stem <- sum(
  subset(
    field_count,
    Field == "Life sciences" |
    Field == "Math and CS" |
    Field == "Physical sciences"
  )[, "Count"]
)
sosc_hum <- sum(
  subset(
    field_count,
    Field == "Humanities" |
    Field == "Social sciences"
  )[, "Count"]
)
# sanity check
total == stem + sosc_hum
```

```
[1] TRUE
```

```
total
```

```
[1] 188
```

```
field_count
```

	Field	Count
1	Humanities	11
2	Life sciences	43
3	Math and CS	86
4	Physical sciences	33
5	Social sciences	15

```
round(stem / total * 100, digits = 1)
```

```
[1] 86.2
```

```
round(sosc_hum / total * 100, digits = 1)
```

```
[1] 13.8
```

So 86% of respondents are from stem, and 14% are from social sciences/humanities.

How many of the STEM respondents are from math or CS?

```
math_cs <- sum(
  subset(
    field_count,
    Field == "Math and CS"
  )[, "Count"]
)
round(math_cs / stem * 100, digits = 1)
```

```
[1] 53.1
```

53% of STEM respondents are from math or CS.

How many experienced contributors were from humanities or social sciences?

```
nrow(subset(data, field_of_study=="Humanities" & Past == "True"))
```

```
[1] 4
```

```
nrow(subset(data, field_of_study=="Social sciences" & Past == "True"))
```

```
[1] 10
```

We had 4 experienced contributors from the humanities, and 10 from the social sciences.

Job category

```
# Remove people who didn't answer this question--
# neither future nor past contributors
tmp <- data$job_category[nzchar(data$job_category)]
job_count <- data.frame(table(tmp))
names(job_count) <- c("Job", "Count")
total <- sum(job_count$Count)

nr_staff <- subset(job_count, Job == "Non-research Staff")[, "Count"]
academics <- sum(subset(job_count, Job != "Non-research Staff")[, "Count"])

job_count
```

	Job	Count
1	Faculty	66
2	Grad Student	43
3	Non-research Staff	106
4	Other research staff	49
5	Post-Doc	17
6	Undergraduate	13

```
round(nr_staff / total * 100, digits = 1)
```

```
[1] 36.1
```

```
round(academics / total * 100, digits = 1)
```

```
[1] 63.9
```

36% of survey respondents are non-research staff, while 64% are academics.

Staff categories

What about the job areas of the non-research staff?

```
# Remove everybody except non-research staff
tmp <- data$staff_categories[nzchar(data$staff_categories)]
staff_count <- data.frame(table(tmp))
names(staff_count) <- c("Area", "Count")
```

```
total <- sum(staff_count$Count)
```

```
staff_count
```

	Area	Count
1	Academic and Research Support	27
2	Administration and General Operations	4
3	Admissions and Enrollment Services	2
4	DevOps or System Administration	8
5	Finance	2
6	Human Resources	1
7	Information Technology (IT)	44
8	Marketing and Communications	2
9	Other	15
10	Student Affairs and Services	1

```
rs <- subset(staff_count, Area == "Academic and Research Support")[, "Count"]
```

```
it <- subset(staff_count, Area == "Information Technology (IT)")[, "Count"]
```

```
round( (rs + it) / total * 100, digits = 1)
```

```
[1] 67
```

67% of the non-research staff respondents were from either IT or Academic and Research Support, which we told participants “includes research administration, libraries, and instructional design”.

Qualitative responses: staff categories

Let’s look at the free-response field for staff job categories. These are the non-research staff who selected “other” and wrote in their job area.

```
qual_staff <- qual$staff_categories_13_TEXT[nzchar(qual$staff_categories_13_TEXT)]
```

I looked at these manually, but for the sake of data privacy, I am not printing the free responses here. I can see that the word “Library” and the abbreviation “IT” each occur multiple times.


```
length(qual_staff)
```

```
[1] 13
```

```
sum(str_count(qual_staff, pattern = "Library"))
```

```
[1] 4
```

```
sum(str_count(qual_staff, pattern = "IT"))
```

```
[1] 3
```

So, seven of the free-text responses contained either the word “Library” or “IT” or both. I am looking manually, and I can see that these came from 6 people. (One person put “Library IT”.)

```
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	