

# Project sizes: statistics

## Overview

This notebook explores Q5: “How frequently have you contributed to projects of the following size?”. I am following up on some plots and counts from `sizes_plots.qmd` with statistics.

## 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

```
sizes_raw <- load_qualtrics_data("clean_data/project_size_Q5.tsv")
other_quant <- load_qualtrics_data("clean_data/other_quant.tsv")
```

## Wrangle data

```
sizes_job <- cbind(sizes_raw, other_quant$job_category)
# Rename column
names(sizes_job)[ncol(sizes_job)] <- "job_category"
# Filter out people who didn't answer either question
sizes_job <- exclude_empty_rows(sizes_job, strict = TRUE)
sizes_job$participantID <- seq(nrow(sizes_job))
```

```
sizes_job_long <- sizes_job %>%
  pivot_longer(
    cols = -c(job_category, participantID),
    names_to = "size",
    values_to = "frequency"
  )

head(sizes_job_long)
```

```
# A tibble: 6 x 4
  job_category participantID size    frequency
  <chr>          <int> <chr>    <chr>
1 Faculty              1 Small  Relatively frequently
2 Faculty              1 Medium Occasionally
3 Faculty              1 Large  Relatively infrequently
4 Post-Doc             2 Small  Occasionally
5 Post-Doc             2 Medium Relatively infrequently
6 Post-Doc             2 Large  Never
```

```
# three way cross tabs (xtabs) and flatten the table
# code from: https://ladal.edu.au/tutorials/regression/regression.html
ftable(xtabs(~ job_category + size + frequency, data = sizes_job_long))
```

job_category	size	frequency Never	Occasionally	Relatively frequently	Relatively infrequently
Faculty	Large	26	6		8
	Medium	13	17		10
	Small	6	17		28
Grad Student	Large	11	7		1
	Medium	8	10		2
	Small	0	7		14
Non-research Staff	Large	15	17		20
	Medium	11	28		22

	Small	10	25	33
Other research staff	Large	17	5	8
	Medium	6	8	14
	Small	0	11	22
Post-Doc	Large	8	3	1
	Medium	1	4	4
	Small	0	6	8
Undergraduate	Large	5	1	0
	Medium	4	1	1
	Small	0	1	4

## Create different job category labels

Let's fold in the smaller job categories, like we did with the other regressions. Acutally, let's try it two ways: first, with 4 groups (nr staff, students, postdocs, and faculty), and then with two: non-research staff vs. academics. We'll see which model looks better.

```
combined4 <- sizes_job_long %>%
  mutate(
    job_category = recode(
      job_category,
      "Post-Doc" = "Postdocs and Staff Researchers",
      "Other research staff" = "Postdocs and Staff Researchers"
    )
  )

combined4 <- combined4 %>%
  mutate(
    job_category = recode(
      job_category,
      "Grad Student" = "Students",
      "Undergraduate" = "Students"
    )
  )
```

```
combined2 <- sizes_job_long %>%
  mutate(
    job_category = recode(
      job_category,
      "Grad Student" = "Academic",
      "Undergraduate" = "Academic",
      "Other research staff" = "Academic",
    )
  )
```

```

    "Post-Doc" = "Academic",
    "Faculty" = "Academic"
  )
)

```

```

# example
combined2

```

```

# A tibble: 699 x 4
  job_category participantID size    frequency
  <chr>          <int> <chr>    <chr>
1 Academic             1 Small  Relatively frequently
2 Academic             1 Medium Occasionally
3 Academic             1 Large  Relatively infrequently
4 Academic             2 Small  Occasionally
5 Academic             2 Medium Relatively infrequently
6 Academic             2 Large  Never
7 Academic             3 Small  Occasionally
8 Academic             3 Medium Relatively infrequently
9 Academic             3 Large  Never
10 Academic            4 Small  Relatively frequently
# i 689 more rows

```

Reorder factor levels.

```

ordered_sizes <- c(
  "Small",
  "Medium",
  "Large"
)

ordered_freqs <- c(
  "Never",
  "Relatively infrequently",
  "Occasionally",
  "Relatively frequently"
)

ordered_jobs4 <- c(
  "Students",
  "Postdocs and Staff Researchers",

```

```

    "Faculty",
    "Non-research Staff"
  )

ordered_jobs2 <- c(
  "Academic",
  "Non-research Staff"
)

combined4$size <- factor(combined4$size, levels = ordered_sizes)
combined4$frequency <- factor(combined4$frequency, levels = ordered_freqs)
combined4$job_category <- factor(combined4$job_category, levels = ordered_jobs4)

combined2$size <- factor(combined2$size, levels = ordered_sizes)
combined2$frequency <- factor(combined2$frequency, levels = ordered_freqs)
combined2$job_category <- factor(combined2$job_category, levels = ordered_jobs2)

```

## Model selection part 1: comparing non-mixed models

To start with, I'm going to create and compare a variety of non-mixed models to see which one looks best. `clm()`, which is for non-mixed models, has a couple of diagnostic capabilities that `clmm()` (mixed models) does not. Once I've chosen a non-mixed model that looks good, I'll add in the random effects term.

Here are the models I want to inspect:

```

freq ~ size
freq ~ combined2 + size
freq ~ combined2 * size
freq ~ combined4 + size
freq ~ combined4 * size

```

### Model 1: no job data

```

# Since we're ignoring the job_category column,
# it doesn't matter which data frame we use
fit1 <- ordinal::clm(frequency ~ size,
  data = combined2, link = "logit", Hess = TRUE)

```

### Model 2: 2 job categories, no interaction

```
fit2 <- ordinal::clm(frequency ~ job_category + size,  
  data = combined2, link = "logit", Hess = TRUE)
```

### Model 3: 4 job categories, no interaction

```
fit3 <- ordinal::clm(frequency ~ job_category + size,  
  data = combined4, link = "logit", Hess = TRUE)
```

### Model 4: 2 job categories, with interaction

```
fit4 <- ordinal::clm(frequency ~ job_category * size,  
  data = combined2, link = "logit", Hess = TRUE)
```

### Model 5: 4 job categories, with interaction

```
fit5 <- ordinal::clm(frequency ~ job_category * size,  
  data = combined4, link = "logit", Hess = TRUE)
```

## Goodness-of-fit

### AICs

```
models <- list(  
  "fit1"=fit1,  
  "fit2"=fit2,  
  "fit3"=fit3,  
  "fit4"=fit4,  
  "fit5"=fit5  
)
```

First, let's get a general sense of goodness-of-fit by looking at the AICs. You're not supposed to compare AICs for models fit to different data sets, but since I've only changed the `job_category` labels, not the observations or the number of observations, I think this is ok.

```
sapply(models, function(x) round(stats::AIC(x)))
```

```
fit1 fit2 fit3 fit4 fit5  
1831 1829 1828 1812 1815
```

AICs are very similar across the board. The last two models look a teensy bit better.

### “Condition number of the Hessian”

Let's check the condition number of the Hessian. I don't really understand what this is, but the `clmm2` tutorial says that high numbers, say larger than say  $10^4$  or  $10^6$ , indicate poor fit.

```
sapply(models, function(x)  
summary(x)$info["cond.H"]  
)
```

```
$fit1.cond.H  
[1] "3.2e+01"
```

```
$fit2.cond.H  
[1] "3.4e+01"
```

```
$fit3.cond.H  
[1] "8.3e+01"
```

```
$fit4.cond.H  
[1] "1.1e+02"
```

```
$fit5.cond.H  
[1] "6.8e+02"
```

All look fine.

## ANOVAs

Let's use some anovas to compare nested models.

```
stats::anova(fit1, fit2)
```

Likelihood ratio tests of cumulative link models:

```
      formula:                      link: threshold:
fit1 frequency ~ size                logit flexible
fit2 frequency ~ job_category + size logit flexible

      no.par    AIC  logLik LR.stat df Pr(>Chisq)
fit1      5 1831.3 -910.64
fit2      6 1829.4 -908.71  3.8758  1    0.04899 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hmm. Including two job categories (no interaction) is just barely significant.

```
stats::anova(fit1, fit3)
```

Likelihood ratio tests of cumulative link models:

```
      formula:                      link: threshold:
fit1 frequency ~ size                logit flexible
fit3 frequency ~ job_category + size logit flexible

      no.par    AIC  logLik LR.stat df Pr(>Chisq)
fit1      5 1831.3 -910.64
fit3      8 1828.2 -906.09  9.1026  3    0.02796 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Including four job categories (no interaction) is slightly more significant. Let's proceed with fit3 as the one to beat. We'll compare it to fit5, which uses the same job labels.

```
stats::anova(fit3, fit5)
```



Likelihood ratio tests of cumulative link models:

```
      formula:                      link: threshold:
fit3 frequency ~ job_category + size logit flexible
fit5 frequency ~ job_category * size logit flexible

      no.par    AIC  logLik LR.stat df Pr(>Chisq)
fit3      8 1828.2 -906.09
fit5     14 1815.1 -893.55   25.08  6  0.0003299 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Yes, inclusion of the interaction term is helpful.

## Nominal and scale tests for clm

[https://www.rdocumentation.org/packages/ordinal/versions/2023.12-4.1/topics/nominal\\_test](https://www.rdocumentation.org/packages/ordinal/versions/2023.12-4.1/topics/nominal_test)  
`nominal_test()`, as I understand it, tests for violations of the proportional odds assumption, which is the assumption that the effect of the explanatory variables are the same across all levels of the outcome variable (remember, we're assuming the outcome categories are cut-offs of an underlying continuous variable). It does ANOVA/LRT on models where a predictor is allowed to have different effects on the different factor levels (cut-off regions), and tests if a version of the model where this assumption is relaxed is a significantly better fit than the one where the assumption is required.

`scale_test()` does the same sort of thing, but instead of testing for non-proportional odds, it's testing for heteroskedasticity.

```
nominal_test(fit5)
```

Tests of nominal effects

```
formula: frequency ~ job_category * size
      Df  logLik    AIC    LRT Pr(>Chi)
<none>      -893.55 1815.1
job_category    6 -890.83 1821.7  5.439  0.48885
size           4 -891.58 1819.2  3.944  0.41362
job_category:size 22 -875.69 1823.4 35.721  0.03253 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
scale_test(fit5)
```

Tests of scale effects

```
formula: frequency ~ job_category * size
      Df logLik   AIC    LRT Pr(>Chi)
<none>      -893.55 1815.1
job_category    3 -893.03 1820.1  1.0382   0.7920
size            2 -892.27 1816.5  2.5567   0.2785
job_category:size 11 -887.31 1824.6 12.4788   0.3287
```

Hmm. Looks like the interaction term violates the non-proportional odds assumption. In other words, the effect of the interaction varies at different frequency levels. No bueno. But the model is improved by the interaction term, according to the LRTs, so we don't want to get rid of it. I will proceed cautiously, and maybe just report it? I really don't want to get into the weeds with this.

## Mixed models

### Model 6: size as fixed effect only

```
fit6 <- ordinal::clmm(frequency ~ job_category * size +
  (1 | participantID),
  data = combined4, link = "logit", Hess = TRUE)
```

### Model 7: size as both a fixed and random effect

```
fit7 <- ordinal::clmm(frequency ~ job_category * size +
  (1 + size | participantID),
  data = combined4, link = "logit", Hess = TRUE)
```

Warning: no. random effects (=699) >= no. observations (=699) for term: (1 + size | participantID)

Warning: no. random effects (=699) >= no. observations (=699)

Hm. I think the 4-job-category data are too sparse for this model to converge. I'm kind of curious about the 2-job label.

## Model 8: size as fixed + random, 2 job cats

```
fit8 <- ordinal::clmm(frequency ~ job_category * size +  
  (1 + size | participantID),  
  data = combined2, link = "logit", Hess = TRUE)
```

Warning: no. random effects (=699) >= no. observations (=699) for term: (1 + size | participantID)

Warning: no. random effects (=699) >= no. observations (=699)

Hm. This one is having the same problem. I guess we'll proceed with the one that converged.

## Goodness-of-fit: mixed

```
stats::anova(fit5, fit6)
```

Likelihood ratio tests of cumulative link models:

	formula:	link:	threshold:
fit5	frequency ~ job_category * size	logit	flexible
fit6	frequency ~ job_category * size + (1   participantID)	logit	flexible

	no.par	AIC	logLik	LR.stat	df	Pr(>Chisq)
fit5	14	1815.1	-893.55			
fit6	15	1794.7	-882.36	22.377	1	2.24e-06 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

It appears that the model is very much improved by including the participant ID as a random effect (each participant has their own intercept (“baseline”). Since `clmm()` doesn't support relaxing the non-proportional odds requirement with `nominal=`, and since I really don't want to switch to a new package/methodology that does, I'm just going to report this and move forward with `fit6`.

## Hypothesis testing (emmeans)

```
summary(fit6)
```

Cumulative Link Mixed Model fitted with the Laplace approximation

formula: frequency ~ job\_category \* size + (1 | participantID)

data: combined4

link	threshold	nobs	logLik	AIC	niter	max.grad	cond.H
logit	flexible	699	-882.36	1794.73	1371(4115)	7.60e-04	5.9e+02

Random effects:

Groups	Name	Variance	Std.Dev.
participantID	(Intercept)	0.7988	0.8938

Number of groups: participantID 233

Coefficients:

	Estimate	Std. Error	
job_categoryPostdocs and Staff Researchers	0.1496	0.4840	
job_categoryFaculty	-0.3564	0.4772	
job_categoryNon-research Staff	-0.8317	0.4472	
sizeMedium	-2.4019	0.4875	
sizeLarge	-3.1399	0.5066	
job_categoryPostdocs and Staff Researchers:sizeMedium	1.0664	0.6045	
job_categoryFaculty:sizeMedium	0.8673	0.5932	
job_categoryNon-research Staff:sizeMedium	1.9299	0.5613	
job_categoryPostdocs and Staff Researchers:sizeLarge	0.1753	0.6210	
job_categoryFaculty:sizeLarge	0.5136	0.6102	
job_categoryNon-research Staff:sizeLarge	2.2159	0.5748	
	z value	Pr(> z )	
job_categoryPostdocs and Staff Researchers	0.309	0.757160	
job_categoryFaculty	-0.747	0.455136	
job_categoryNon-research Staff	-1.860	0.062952	.
sizeMedium	-4.927	8.37e-07	***
sizeLarge	-6.197	5.74e-10	***
job_categoryPostdocs and Staff Researchers:sizeMedium	1.764	0.077741	.
job_categoryFaculty:sizeMedium	1.462	0.143741	
job_categoryNon-research Staff:sizeMedium	3.438	0.000586	***
job_categoryPostdocs and Staff Researchers:sizeLarge	0.282	0.777782	
job_categoryFaculty:sizeLarge	0.842	0.399954	

```
job_categoryNon-research Staff:sizeLarge          3.855 0.000116 ***
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Threshold coefficients:

	Estimate	Std. Error	z value
Never Relatively infrequently	-3.2971	0.4143	-7.957
Relatively infrequently Occasionally	-1.6379	0.3921	-4.177
Occasionally Relatively frequently	-0.2328	0.3845	-0.605

That's a lot of parameters to interpret. emmeans to the rescue.

I'm not going to attempt to average the results across job category, so there's no weighting scheme needed.

```
emm <- emmeans(fit6, ~ size * job_category, mode = "mean.class")
```

```
summary(emm) %>%  
  arrange(desc(mean.class))
```

size	job_category	mean.class	SE	df	asympt.LCL	asympt.UCL
Small	Postdocs and Staff Researchers	3.42	0.118	Inf	3.19	3.65
Small	Students	3.36	0.160	Inf	3.05	3.67
Small	Faculty	3.20	0.133	Inf	2.94	3.46
Small	Non-research Staff	2.97	0.117	Inf	2.74	3.20
Medium	Postdocs and Staff Researchers	2.78	0.149	Inf	2.49	3.07
Medium	Non-research Staff	2.72	0.117	Inf	2.49	2.95
Large	Non-research Staff	2.47	0.120	Inf	2.24	2.71
Medium	Faculty	2.40	0.141	Inf	2.12	2.68
Medium	Students	2.13	0.185	Inf	1.77	2.49
Large	Postdocs and Staff Researchers	1.92	0.143	Inf	1.64	2.20
Large	Faculty	1.84	0.131	Inf	1.59	2.10
Large	Students	1.77	0.167	Inf	1.45	2.10

Confidence level used: 0.95

Hmm. Mildly interesting. As suggested by the exploratory plots, the results are fairly clean for small and large projects, but more muddled for medium projects.

```
by_job <- summary(
  pairs(emm, by = "job_category"),
  infer = TRUE # infer CIs
)
by_job
```

job\_category = Students:

contrast	estimate	SE	df	asympt.LCL	asympt.UCL	z.ratio	p.value
Small - Medium	1.229	0.223	Inf	0.7056	1.753	5.502	<.0001
Small - Large	1.586	0.213	Inf	1.0873	2.085	7.450	<.0001
Medium - Large	0.357	0.225	Inf	-0.1702	0.884	1.587	0.2511

job\_category = Postdocs and Staff Researchers:

contrast	estimate	SE	df	asympt.LCL	asympt.UCL	z.ratio	p.value
Small - Medium	0.639	0.172	Inf	0.2348	1.043	3.706	0.0006
Small - Large	1.496	0.171	Inf	1.0958	1.896	8.761	<.0001
Medium - Large	0.857	0.188	Inf	0.4175	1.297	4.569	<.0001

job\_category = Faculty:

contrast	estimate	SE	df	asympt.LCL	asympt.UCL	z.ratio	p.value
Small - Medium	0.801	0.176	Inf	0.3884	1.214	4.547	<.0001
Small - Large	1.357	0.172	Inf	0.9548	1.759	7.908	<.0001
Medium - Large	0.555	0.174	Inf	0.1472	0.964	3.189	0.0041

job\_category = Non-research Staff:

contrast	estimate	SE	df	asympt.LCL	asympt.UCL	z.ratio	p.value
Small - Medium	0.249	0.149	Inf	-0.0999	0.599	1.673	0.2155
Small - Large	0.494	0.152	Inf	0.1382	0.850	3.253	0.0033
Medium - Large	0.245	0.151	Inf	-0.1082	0.598	1.625	0.2348

Confidence level used: 0.95

Conf-level adjustment: tukey method for comparing a family of 3 estimates

P value adjustment: tukey method for comparing a family of 3 estimates

Wow, okay, still a lot of parameters to interpret.

After staring at this for a while, I think the main conclusion is that everyone is more likely to contribute to small projects than large projects.

```
by_size <- summary(
  pairs(emm, by = "size"),
  infer = TRUE # infer CIs
)
```

```
)
by_size
```

```
size = Small:
```

contrast	estimate	SE	df
Students - Postdocs and Staff Researchers	-0.0607	0.198	Inf
Students - Faculty	0.1579	0.208	Inf
Students - (Non-research Staff)	0.3919	0.198	Inf
Postdocs and Staff Researchers - Faculty	0.2186	0.177	Inf
Postdocs and Staff Researchers - (Non-research Staff)	0.4526	0.166	Inf
Faculty - (Non-research Staff)	0.2341	0.177	Inf

  

asympt.LCL	asympt.UCL	z.ratio	p.value
-0.5687	0.4473	-0.307	0.9900
-0.3760	0.6918	0.760	0.8725
-0.1179	0.9017	1.975	0.1975
-0.2355	0.6726	1.237	0.6034
0.0267	0.8785	2.730	0.0321
-0.2215	0.6896	1.320	0.5502

```
size = Medium:
```

contrast	estimate	SE	df
Students - Postdocs and Staff Researchers	-0.6511	0.238	Inf
Students - Faculty	-0.2699	0.233	Inf
Students - (Non-research Staff)	-0.5878	0.219	Inf
Postdocs and Staff Researchers - Faculty	0.3812	0.206	Inf
Postdocs and Staff Researchers - (Non-research Staff)	0.0632	0.190	Inf
Faculty - (Non-research Staff)	-0.3179	0.183	Inf

  

asympt.LCL	asympt.UCL	z.ratio	p.value
-1.2622	-0.0400	-2.737	0.0315
-0.8674	0.3276	-1.160	0.6518
-1.1503	-0.0254	-2.685	0.0365
-0.1469	0.9093	1.854	0.2481
-0.4239	0.5504	0.333	0.9872
-0.7892	0.1533	-1.733	0.3063

```
size = Large:
```

contrast	estimate	SE	df
Students - Postdocs and Staff Researchers	-0.1510	0.219	Inf
Students - Faculty	-0.0716	0.212	Inf
Students - (Non-research Staff)	-0.7002	0.206	Inf
Postdocs and Staff Researchers - Faculty	0.0794	0.193	Inf
Postdocs and Staff Researchers - (Non-research Staff)	-0.5492	0.186	Inf

Faculty - (Non-research Staff)				-0.6286	0.178	Inf
asympt.LCL	asympt.UCL	z.ratio	p.value			
-0.7143	0.4124	-0.688	0.9015			
-0.6163	0.4731	-0.338	0.9867			
-1.2290	-0.1714	-3.402	0.0037			
-0.4171	0.5758	0.411	0.9766			
-1.0277	-0.0707	-2.949	0.0168			
-1.0855	-0.1716	-3.534	0.0023			

Confidence level used: 0.95

Conf-level adjustment: tukey method for comparing a family of 4 estimates

P value adjustment: tukey method for comparing a family of 4 estimates

After staring at this for a while, I think we can conclude that academics contribute to large projects less frequently than non-research staff, but there is very little evidence to support the reverse—that academics are more likely to contribute to small projects than NR staff. (Only true for postdocs and staff researchers.)

So, w.r.t. conclusion #2, I think we've mitigated the danger of the non-proportional odds violation by just looking at a single level, i.e. Large projects for different job\_categories, so it doesn't matter that the effect of the interaction is different at large vs. medium, for example, because we're only looking at large. I don't think this is true for #1, where we were saying that the frequency of small is higher than the frequency of large for all groups.

## Wilcoxon test

Let's use a Wilcoxon test to confirm/deny the claim that all groups contribute to small projects more than they contribute to large projects. I'm subsetting the data to just small/large, removing medium, since I'm not making claims about contributions to medium projects. Since I'm just looking at two categories, small vs. large (for each of the 4 jobs independently), I'll use a Wilcoxon test instead of a Kruskal-Wallis test.

```
# Note, no need to worry about missing data as all options were mandatory
paired <- combined4 %>%
  mutate(
    freq_score = recode(
      frequency,
      "Never" = 0L,
      "Relatively infrequently" = 1L,
      "Occasionally" = 2L,
      "Relatively frequently" = 3L
    )
  )
```



```

)
) %>%
  select(job_category, participantID, size, freq_score) %>%
  filter(size != "Medium") %>%
  mutate(size = forcats::fct_relevel(size, "Small", "Large")) %>%
  pivot_wider(names_from = size, values_from = freq_score) %>%
  mutate(diff = Small - Large)

paired

```

```

# A tibble: 233 x 5
  job_category      participantID Small Large diff
  <fct>              <int> <int> <int> <int>
1 Faculty              1      3      1      2
2 Postdocs and Staff Researchers  2      2      0      2
3 Postdocs and Staff Researchers  3      2      0      2
4 Faculty              4      3      0      3
5 Faculty              5      3      1      2
6 Faculty              6      3      0      3
7 Postdocs and Staff Researchers  7      2      0      2
8 Faculty              8      3      0      3
9 Postdocs and Staff Researchers  9      2      2      0
10 Students           10      3      2      1
# i 223 more rows

```

```

# Wilcoxon test per job_category (one-sided: Small > Large)

wilc_results <- lapply(split(paired, paired$job_category), function(df) {
  wilcox.test(
    df$Small,
    df$Large,
    paired = TRUE,
    alternative = "greater",
    conf.int = TRUE,
    conf.level = 0.95
  )
})

```

```

Warning in wilcox.test.default(df$Small, df$Large, paired = TRUE, alternative =
"greater", : cannot compute exact p-value with ties

```

```
Warning in wilcox.test.default(df$Small, df$Large, paired = TRUE, alternative =  
"greater", : cannot compute exact confidence interval with ties
```

```
Warning in wilcox.test.default(df$Small, df$Large, paired = TRUE, alternative =  
"greater", : cannot compute exact p-value with zeroes
```

```
Warning in wilcox.test.default(df$Small, df$Large, paired = TRUE, alternative =  
"greater", : cannot compute exact confidence interval with zeroes
```

```
Warning in wilcox.test.default(df$Small, df$Large, paired = TRUE, alternative =  
"greater", : cannot compute exact p-value with ties
```

```
Warning in wilcox.test.default(df$Small, df$Large, paired = TRUE, alternative =  
"greater", : cannot compute exact confidence interval with ties
```

```
Warning in wilcox.test.default(df$Small, df$Large, paired = TRUE, alternative =  
"greater", : cannot compute exact p-value with zeroes
```

```
Warning in wilcox.test.default(df$Small, df$Large, paired = TRUE, alternative =  
"greater", : cannot compute exact confidence interval with zeroes
```

```
# Example  
wilc_results[[1]]
```

Wilcoxon signed rank test with continuity correction

```
data: df$Small and df$Large  
V = 325, p-value = 4.72e-06  
alternative hypothesis: true location shift is greater than 0  
95 percent confidence interval:  
 1.50001      Inf  
sample estimates:  
(pseudo)median  
      2
```

Let's look at the adjusted p-values.

```

job_cats <- names(split(paired, paired$job_category))
wilc_results_pvals <- sapply(seq(length(job_cats)), function(i) wilc_results[[i]]$p.value)
names(wilc_results_pvals) <- job_cats

p.adjust(wilc_results_pvals, method = "holm")

```

Students	Postdocs	and Staff	Researchers
1.415879e-05			1.866168e-07
Faculty		Non-research	Staff
1.968021e-05			6.147576e-03

Great. This gives us more confidence in claiming that all groups contribute to small projects more than large projects.

### Wilcoxon test: focusing on large projects

Well, since we're here, we might as well also do a wilcoxon test our other finding that non-research staff contribute to large projects more frequently than academics do. These data are not paired, obviously, since we don't have repeated measures per person for just large projects. Now we're comparing between job categories instead of between project sizes.

```

large_projs <- combined4 %>%
  mutate(
    freq_score = recode(
      frequency,
      "Never" = 0L,
      "Relatively infrequently" = 1L,
      "Occasionally" = 2L,
      "Relatively frequently" = 3L
    )
  ) %>%
  filter(size == "Large") %>%
  mutate(
    group = if_else(
      job_category == "Non-research Staff",
      "nrstaff",
      "academic"
    )
  ) %>%
  select(group, freq_score)

```

```
large_projs$group <- factor(large_projs$group, levels = c("nrstaff", "academic"))
```

```
# one-sided: NRS > Academics
wilc_results2 <- wilcox.test(
  freq_score ~ group,
  data = large_projs,
  alternative = "greater",
  conf.int = TRUE
)

wilc_results2
```

Wilcoxon rank sum test with continuity correction

```
data:  freq_score by group
W = 8226.5, p-value = 3.033e-05
alternative hypothesis: true location shift is greater than 0
95 percent confidence interval:
 2.496483e-05      Inf
sample estimates:
difference in location
      0.9999661
```

```
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           corrrplot_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         utf8_1.2.6            rmarkdown_2.29
[19] ggfittext_0.10.2      tzdb_0.5.0            ps_1.9.1
[22] nloptr_2.2.1          purrr_1.1.0           xfun_0.53
[25] modeltools_0.2-24     jsonlite_2.0.0        tweenr_2.0.3
[28] parallel_4.4.2        prabclus_2.3-4        R6_2.6.1
[31] stringi_1.8.7         RColorBrewer_1.1-3    boot_1.3-31
[34] diptest_0.77-2        numDeriv_2016.8-1.1  estimability_1.5.1
[37] Rcpp_1.1.0            iterators_1.0.14      knitr_1.50
[40] IRanges_2.40.1        splines_4.4.2         nnet_7.3-19
[43] tidyselect_1.2.1      yaml_2.3.10           doParallel_1.0.17
[46] codetools_0.2-20      processx_3.8.6        lattice_0.22-6
[49] tibble_3.3.0          withr_3.0.2           evaluate_1.0.4
[52] polyclip_1.10-7       xml2_1.4.0            circlize_0.4.16
[55] mclust_6.1.1          kernlab_0.9-33        pillar_1.11.0
[58] renv_1.1.5            foreach_1.5.2         stats4_4.4.2
[61] reformulas_0.4.1      generics_0.1.4        rprojroot_2.1.1
[64] S4Vectors_0.44.0      hms_1.1.3             minqa_1.2.8
[67] xtable_1.8-4          class_7.3-22          glue_1.8.0
[70] robustbase_0.99-4-1  mvtnorm_1.3-3         rbibutils_2.3
[73] colorspace_2.1-1      nlme_3.1-166          cli_3.6.5
[76] textshaping_1.0.1     gtable_0.3.6          DEoptimR_1.1-4
[79] digest_0.6.37         BiocGenerics_0.52.0  ucminf_1.2.2
```

[82]	ggrepel_0.9.6	rjson_0.2.23	farver_2.1.2
[85]	htmltools_0.5.8.1	lifecycle_1.0.4	GlobalOptions_0.1.2
[88]	MASS_7.3-61		