Motivations for contributing to OS: statistical analysis

Overview

I'm redoing an earlier analysis of 6, which is about participants' reasons for contributing to open source. I was trying to incorporate all the binary response variables (yes/no to each possible motivation) in one model, but I think it just ended up being obtuse and hard to understand. I also don't feel good about using mvabund(), which did exactly what I needed, but it's an ecology tool, and basically 100% of the papers that cite it are ecology papers, so it just didn't feel right.

I'm just going to use more popular functions, and I'll use multiple small models instead of one big complicated one.

The old analysis is in notebooks/defunct/motivations_stats.qmd.

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

```
motivations <- load_qualtrics_data("clean_data/motivations_Q6.tsv")
other_quant <- load_qualtrics_data("clean_data/other_quant.tsv")
```

Wrangle data

```
motivations_job <- cbind(motivations, other_quant$job_category)
# Rename last col
names(motivations_job)[length(names(motivations_job))] <- "job_category"
# Remove any rows where the job_category is missing
motivations_job_clean <- exclude_empty_rows(motivations_job, strict = TRUE)
# Remove rows of all 0s
motivations_job_clean <- motivations_job_clean %>%
  filter(!if_all(Job:Other, ~ .x == 0))

# drop the "Other" column
motivations_job_clean <- motivations_job_clean %>%
  select(-c("Other"))

head(motivations_job_clean)
```

```
  Job Improve Tools Customize Network Give back Skills Fun         job_category
1   1             1         1       1         1      1   1              Faculty
2   0             1         1       1         0      1   0             Post-Doc
3   0             1         1       0         0      1   1 Other research staff
4   1             1         1       0         1      0   0              Faculty
5   0             1         1       0         1      1   1              Faculty
6   0             1         1       0         0      0   0              Faculty
```

```
dim(motivations_job_clean)
```

```
[1] 233   8
```

```
# This will also come in handy later.
motivation_cols <- names(motivations_job_clean)[
  -length(names(motivations_job_clean))
]
motivation_cols
```

```
[1] "Job"           "Improve Tools" "Customize"     "Network"
[5] "Give back"     "Skills"        "Fun"
```

Since other models elsewhere in this study have had trouble converging, I'm just going to combine some of the groups a priori so we can have larger sample sizes per group. This is, in a gut-sense kind of way, consistent with the power analysis I did in the earlier version of this script–the comparisons I was interested in that involved undergrads and postdocs didn't have enough statistical power for hypothesis testing. The initial group labels on the survey (faculty, postdoc, grad student, undergrad, staff researcher, non-research staff) were somewhat arbitrary, so I have no qualms about combining them now into a different set of somewhat arbitrary groups.

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

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

head(combined)
```

```
  Job Improve Tools Customize Network Give back Skills Fun
1   1             1         1       1         1      1   1
2   0             1         1       1         0      1   0
3   0             1         1       0         0      1   1
4   1             1         1       0         1      0   0
5   0             1         1       0         1      1   1
6   0             1         1       0         0      0   0
                    job_category
1                        Faculty
2 Postdocs and Staff Researchers
```

```
3 Postdocs and Staff Researchers
4                        Faculty
5                        Faculty
6                        Faculty
```

Regression on job categories

Let's make a simple logistic regression model for each motivation. The only independent variable is job_category.

```
# run a separate model for each outcome (motivation)
models <- lapply(motivation_cols, function(x) {
  # wrap the column name in backticks so "My Column" becomes `My Column`
  f_text <- paste0("`", x, "`", " ~ job_category")
  f <- as.formula(f_text)
  stats::glm(f, family = "binomial", data = combined)
})

# example
models[[1]]
```

```

Call:  stats::glm(formula = f, family = "binomial", data = combined)

Coefficients:
                               (Intercept)
                                   -0.4480
            job_categoryNon-research Staff
                                    0.1671
job_categoryPostdocs and Staff Researchers
                                    0.9299
                      job_categoryStudents
                                    0.2657

Degrees of Freedom: 232 Total (i.e. Null);  229 Residual
Null Deviance:      322
Residual Deviance: 315.1    AIC: 323.1
```

Quick AIC check just because it's easy.

```r
for (i in seq_along(motivation_cols)) {
  cat(
    sprintf(
      "%s  %.3f\n",
      motivation_cols[i],
      stats::AIC(models[[i]])   # AIC rounded to 3 decimals
    )
  )
}
```

```
Job  323.064
Improve Tools  201.450
Customize  292.599
Network  303.564
Give back  296.556
Skills  296.765
Fun  322.957
```

Hmm. Some pretty big differences here. Improve Tools is by far the best fit. Fun and Job are a pretty poor fit.

Let's makes some null models with an intercept only, and no predictor (job_category).

```r
null_models <- lapply(motivation_cols, function(x) {
  # wrap the column name in backticks so "My Column" becomes `My Column`
  f_text <- paste0("`", x, "`", " ~ 1")
  f <- as.formula(f_text)
  stats::glm(f, family = "binomial", data = combined)
})
```

And let's do ANOVA to compare the null models vs. full models. (Printing an example)

```r
anova_results <- mapply(
  FUN = function(null_m, full_m) {
    stats::anova(null_m, full_m)
  },
  null_models,
  models,
  SIMPLIFY = FALSE
)

anova_results[[1]]
```

```
Analysis of Deviance Table

Model 1: Job ~ 1
Model 2: Job ~ job_category
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       232     322.04
2       229     315.06  3   6.9767  0.07264 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Let's look at p-values for all the ANOVAs.

```r
p_vals <- c()
for (i in seq_along(motivation_cols)) {
  p_val <- anova_results[[i]]$`Pr(>Chi)`[2]
  p_vals[i] <- p_val
  cat(
    sprintf(
      "%s  %.3f\n",
      motivation_cols[i],
      p_val # p-value rounded to 3 decimals
    )
  )
}
```

```
Job  0.073
Improve Tools  0.295
Customize  0.317
Network  0.680
Give back  0.068
Skills  0.000
Fun  0.147
```

Skills has a super low p-value, as we'd expect based on the earlier analysis with mvabund, which found a similar result. Maybe we should do a multiple test correction?

```r
# Choosing BH pretty arbitrarily. I don't feel a
# need to be super conservative, and I hear BH
# is more forgiving than holm.
p_fdr  <- p.adjust(p_vals, method = "BH")
for (i in seq_along(p_fdr)) {
```

```
  cat(
    sprintf(
      "%s  %.3f\n",
      motivation_cols[i],
      p_fdr[i] # p-value rounded to 3 decimals
    )
  )
}
```

```
Job  0.170
Improve Tools  0.369
Customize  0.369
Network  0.680
Give back  0.170
Skills  0.000
Fun  0.257
```

Yup. Give back is no longer significant. This too, aligns with what I saw in the previous analysis with mvabund. I didn't do quite the same procedure this time, but I noticed the coefficients for both Skills and Give back were big, and really different from the other variables, but only Skills turned out to have a significant (p-adjusted) ANOVA from univariate ANOVAs.

So, as we saw previously, Skills is the most interesting one. It's the only case where the model fit is significantly improved by inclusion of the job_category variable. It was sort of middling in terms of AIC, and I'm okay with that. Let's take a closer look at the model output.

```
skills_model <- models[[which(motivation_cols=="Skills")]]
skills_model
```

```
Call:  stats::glm(formula = f, family = "binomial", data = combined)

Coefficients:
                              (Intercept)
                                  -0.4480
            job_categoryNon-research Staff
                                   1.2297
job_categoryPostdocs and Staff Researchers
                                   0.7783
                      job_categoryStudents
                                   2.1708
```

```
Degrees of Freedom: 232 Total (i.e. Null);  229 Residual
Null Deviance:      311.8
Residual Deviance: 288.8    AIC: 296.8
```

```
p_fdr[[which(motivation_cols=="Skills")]]
```

```
[1] 0.0002843586
```

Okay, apparently Faculty are our reference level. All the coefficients are positive, so everyone else is more likely to choose "Skills" than faculty.

```
emm <- emmeans(skills_model, ~ job_category, type="response")
pairs(emm, type="response", infer = TRUE)
```

```
 contrast                                           odds.ratio     SE  df
 Faculty / (Non-research Staff)                          0.292 0.1030 Inf
 Faculty / Postdocs and Staff Researchers                0.459 0.1750 Inf
 Faculty / Students                                      0.114 0.0632 Inf
 (Non-research Staff) / Postdocs and Staff Researchers   1.571 0.5630 Inf
 (Non-research Staff) / Students                         0.390 0.2100 Inf
 Postdocs and Staff Researchers / Students               0.248 0.1380 Inf
 asymp.LCL asymp.UCL null z.ratio p.value
    0.1178     0.726    1  -3.475  0.0029
    0.1721     1.225    1  -2.037  0.1744
    0.0275     0.474    1  -3.918  0.0005
    0.6249     3.948    1   1.258  0.5896
    0.0979     1.555    1  -1.748  0.2986
    0.0594     1.040    1  -2.499  0.0600

Confidence level used: 0.95
Conf-level adjustment: tukey method for comparing a family of 4 estimates
Intervals are back-transformed from the log odds ratio scale
P value adjustment: tukey method for comparing a family of 4 estimates
Tests are performed on the log odds ratio scale
```

Meh, not sure if these contrasts are interesting enough to be worth reporting.

Test for trend in "skills"

In my other script, motivations_plots, we have one plot where we apparently see a trend: the probability of a respondent choosing "skills" as a motivator appears to decrease as they advance in their academic career. We will use a Cochrane-Armitage test for trend to evaluate whether this trend is real. More precisely, I believe we are evaluating whether the order "P(Yes | Undergrad) > P(Yes | Grad) > P(Yes | Postdoc) > P(Yes | Faculty)" is highly unlikely (<95% chance) given the null hypothesis that all four categories have the same probability of a "yes" response.

Full disclosure: I'm being a little p-hacky here, because I'm only trying this after I tried a series of pairwise z-tests to see whether the proportion of "yes" for "skills" was significantly different from undergrads vs. grads, grads vs. postdocs, etc. That analysis is in the old notebook. In all seriousness, I don't actually feel that I am p-hacking because I'm not just using a new test to try and make the same claim; this is a different test and we will interpret it appropriately. I'm not claiming that undergrads are more likely than grads to select skills; I'm just claiming that there is a trend across the 4 categories.

```r
# Here, I haven't combined post-docs and other research staff
n_postdoc <- sum(motivations_job_clean$job_category == "Post-Doc")
n_postdoc_yes <- sum(
  motivations_job_clean$job_category == "Post-Doc" &
    motivations_job_clean$Skills == 1
)
# For the other groups, it doesn't matter if we use the raw or processed data
n_faculty <- sum(motivations_job_clean$job_category == "Faculty")
n_faculty_yes <- sum(
  motivations_job_clean$job_category == "Faculty" &
    motivations_job_clean$Skills == 1
)

n_grad <- sum(motivations_job_clean$job_category == "Grad Student")
n_grad_yes <- sum(
  motivations_job_clean$job_category == "Grad Student" &
    motivations_job_clean$Skills == 1
)

n_undergrad <- sum(motivations_job_clean$job_category == "Undergraduate")
n_undergrad_yes <- sum(
  motivations_job_clean$job_category == "Undergraduate" &
    motivations_job_clean$Skills == 1
)
```

```r
n_yes <- c(
  n_undergrad_yes,
  n_grad_yes,
  n_postdoc_yes,
  n_faculty_yes
)

n_tot <- c(
  n_undergrad,
  n_grad,
  n_postdoc,
  n_faculty
)

# Assign scores 1,2,3,4 for Undergrad --> Faculty
# To indicate the ordering
scores <- 1:4

stats::prop.trend.test(
  x = n_yes,
  n = n_tot,
  score = scores
)
```

```
    Chi-squared Test for Trend in Proportions

data:  n_yes out of n_tot ,
 using scores: 1 2 3 4
X-squared = 19.818, df = 1, p-value = 8.518e-06
```

I'm honestly not sure whether this is a one-tailed or two-tailed test… I would assume one-tailed, but the documentation is terse. Anyway, even if we divide that p-value by two it's still well under p=0.05. So, yes, there is a trend of skills declining as a motivator.

By popular demand: IT vs. Academics

Greg raised an interesting question: what about IT staff vs. academics? Let's play around with this.

I plotted the data (see motivations_plots.qmd), and it appears that these groups are somewhat different. The "Job" motivation looks to be the most different, just by eyeballing it. But let's see what the statistics say.

Data Wrangling

```
motivations_job_staff <- cbind(motivations, other_quant$job_category)
# Rename columns
names(motivations_job_staff)[length(names(
  motivations_job_staff
))] <- "job_category"
motivations_job_staff <- cbind(
  motivations_job_staff,
  other_quant$staff_categories
)
names(motivations_job_staff)[length(names(
  motivations_job_staff
))] <- "staff_category"
# Remove any rows where the job_category or staff_category are missing
motivations_job_staff_clean <- exclude_empty_rows(
  motivations_job_staff,
  strict = TRUE
)
# Remove rows of all 0s
motivations_job_staff_clean <- motivations_job_staff_clean %>%
  filter(!if_all(Job:Other, ~ .x == 0))

# drop the "Other" column
motivations_job_staff_clean <- motivations_job_staff_clean %>%
  select(-c("Other"))

head(motivations_job_staff_clean)
```

```
  Job Improve Tools Customize Network Give back Skills Fun        job_category
1   0             1         1       0         0      1   1 Non-research Staff
2   1             0         0       0         1      0   0 Non-research Staff
3   0             1         1       1         0      0   0 Non-research Staff
4   1             0         0       1         1      0   0 Non-research Staff
5   0             1         1       0         1      1   1 Non-research Staff
6   1             1         1       1         1      1   1 Non-research Staff
```

```
                   staff_category
1                           Other
2 DevOps or System Administration
3 DevOps or System Administration
4     Information Technology (IT)
5 DevOps or System Administration
6                           Other
```

```
it <- motivations_job_staff_clean %>%
  filter(staff_category == "Information Technology (IT)") %>%
  select(-c(job_category, staff_category))
it$Role <- "IT"
head(it)
```

```
  Job Improve Tools Customize Network Give back Skills Fun Role
1   1             0         0       1         1      0   0   IT
2   1             1         1       1         1      1   1   IT
3   0             1         1       1         1      1   1   IT
4   0             1         0       0         0      0   0   IT
5   0             1         0       0         1      1   0   IT
6   0             1         1       0         1      1   1   IT
```

```
dim(it)
```

```
[1] 33  8
```

```
# Everyone except non-research staff
academics <- combined %>%
  filter(
    job_category == "Faculty" |
    job_category == "Students" |
    job_category == "Postdocs and Staff Researchers"
  ) %>%
    select(-job_category)
academics$Role <- "Academic"
head(academics)
```

```
  Job Improve Tools Customize Network Give back Skills Fun     Role
1   1             1         1       1         1      1   1 Academic
2   0             1         1       1         0      1   0 Academic
```

```
3    0                1          1        0          0       1    1 Academic
4    1                1          1        0          1       0    0 Academic
5    0                1          1        0          1       1    1 Academic
6    0                1          1        0          0       0    0 Academic
```

```r
dim(academics)
```

```
[1] 147   8
```

```r
it_acad <- rbind(it, academics)
it_acad$Role <- as.factor(it_acad$Role)
dim(it_acad)
```

```
[1] 180   8
```

Great. Now we have a data frame with the responses of IT staff and academics.

Regression

Let's do a quick regression to see whether the IT/Academic groups improve the model fit.

```r
# run a separate model for each outcome (motivation)
models_it_acad <- lapply(motivation_cols, function(x) {
  # wrap the column name in backticks so "My Column" becomes `My Column`
  f_text <- paste0("`", x, "`", " ~ Role")
  f <- as.formula(f_text)
  stats::glm(f, family = "binomial", data = it_acad)
})

null_models_it_acad <- lapply(motivation_cols, function(x) {
  # wrap the column name in backticks so "My Column" becomes `My Column`
  f_text <- paste0("`", x, "`", " ~ 1")
  f <- as.formula(f_text)
  stats::glm(f, family = "binomial", data = it_acad)
})
```

If I wanted to (borderline) p-hack, I could just test "Job" and avoid multiple test correction, since the plot suggested that was the biggest difference. But I want to report robust differences that can withstand correction. So let's test them all and do the correction.

```r
anova_results_it_acad <- mapply(
  FUN = function(null_m, full_m) {
    stats::anova(null_m, full_m)
  },
  null_models_it_acad,
  models_it_acad,
  SIMPLIFY = FALSE
)

p_vals_it_acad <- c()
for (i in seq_along(motivation_cols)) {
  p_vals_it_acad[i] <- anova_results_it_acad[[i]]$`Pr(>Chi)`[2]
}

p_fdr_it_acad  <- p.adjust(p_vals_it_acad, method = "BH")
for (i in seq_along(p_fdr_it_acad)) {
  cat(
    sprintf(
      "%s  %.3f\n",
      motivation_cols[i],
      p_fdr_it_acad[i] # p-value rounded to 3 decimals
    )
  )
}
```

```
Job  0.019
Improve Tools  0.464
Customize  0.852
Network  0.852
Give back  0.138
Skills  0.627
Fun  0.627
```

Great. Once again, the results are in accordance with the earlier mvabund analysis. Only Job is significant. Let's look at the model.

```r
it_acad_job_model <- models_it_acad[[which(motivation_cols=="Job")]]
it_acad_job_model
```

```
Call:  stats::glm(formula = f, family = "binomial", data = it_acad)
```

```
Coefficients:
(Intercept)       RoleIT
   -0.04082     -1.27136

Degrees of Freedom: 179 Total (i.e. Null);  178 Residual
Null Deviance:      246.8
Residual Deviance: 237.8    AIC: 241.8
```

I believe the negative coefficient indicates that the IT are less likely to select "yes".

And here's the full p-value from the ANOVA:

```r
p_fdr_it_acad[[which(motivation_cols=="Job")]]
```

```
[1] 0.01882033
```

```r
emm_it_acad <- emmeans(it_acad_job_model, ~ Role, type="response")
pairs(emm_it_acad, type="response", infer = TRUE)
```

```
 contrast      odds.ratio   SE  df asymp.LCL asymp.UCL null z.ratio p.value
 Academic / IT       3.57 1.63 Inf      1.46      8.73    1   2.784  0.0054

Confidence level used: 0.95
Intervals are back-transformed from the log odds ratio scale
Tests are performed on the log odds ratio scale
```

We can interpret that odds ratio as: The odds that an academic selects 'Yes' are 3.57× the odds for IT staff.

Actually, this is just the inverse of the exponentiated coefficient from the model: $\exp(-1.27136) = 0.280$, and $1/0.280 = 3.57$, which makes sense. So we didn't need emmeans. The only difference is that emmeans is just using Academic as the numerator in the odds ratio (reference level), while the model is using IT. I think integer odds are easier to understand than fractional odds (0.280), so I'll report that one. I also want to get confidence intervals. To minimize me exponentiating things by hand, I'll just redo the model with the factor level order switched.

```r
levels(it_acad$Role)
```

```
[1] "Academic" "IT"
```

```r
it_acad$Role <- relevel(it_acad$Role, ref = "IT")

rev_model <- stats::glm(Job ~ Role, family = "binomial", data = it_acad)
rev_model
```

```
Call:  stats::glm(formula = Job ~ Role, family = "binomial", data = it_acad)

Coefficients:
 (Intercept)  RoleAcademic
      -1.312         1.271

Degrees of Freedom: 179 Total (i.e. Null);  178 Residual
Null Deviance:      246.8
Residual Deviance: 237.8    AIC: 241.8
```

```r
exp(stats::confint(rev_model)) # exp to get it on the odds-ratio scale
```

```
Waiting for profiling to be done...
```

```
                 2.5 %    97.5 %
(Intercept)  0.1077063 0.5873741
RoleAcademic 1.5273847 9.3756444
```

Last-minute add-on: NR staff vs academics

So, the previous analysis of IT vs. academics is pretty different from the rest of the analyses in this paper, in that I normally look at non-research staff, not just IT alone. If the trend still holds, I'd rather report the trend for academics vs. nr staff, for consistency with the rest of the paper. Let's see if academics are more likely to select "it's part of my job" than non-research staff.

I see two possible approaches. We could look at nr staff vs. each academic group in turn, and see if all p-values are below 0.05; OR we could lump all academics into one group, and compare the two groups, as we did for IT vs. academics. The first approach is presumably more stringent, since we'll have less power to detect subtle differences. We kind of already know that it won't be significant, since we saw above that including job category as a fixed effect didn't significantly improve model fit, when job category had 4 levels. Let's check, just to be sure.

Approach #1: 4 job categories

```
job_model <- models[[which(motivation_cols=="Job")]]
job_model
```

```
Call:  stats::glm(formula = f, family = "binomial", data = combined)

Coefficients:
                               (Intercept)
                                   -0.4480
            job_categoryNon-research Staff
                                    0.1671
job_categoryPostdocs and Staff Researchers
                                    0.9299
                      job_categoryStudents
                                    0.2657

Degrees of Freedom: 232 Total (i.e. Null);  229 Residual
Null Deviance:      322
Residual Deviance: 315.1    AIC: 323.1
```

```
p_fdr[[which(motivation_cols=="Job")]]
```

```
[1] 0.1695023
```

```
emm_job <- emmeans(job_model, ~ job_category, type="response")
pairs(emm_job, type="response", infer = TRUE)
```

```
 contrast                                              odds.ratio    SE  df
 Faculty / (Non-research Staff)                             0.846 0.291 Inf
 Faculty / Postdocs and Staff Researchers                   0.395 0.152 Inf
 Faculty / Students                                         0.767 0.337 Inf
 (Non-research Staff) / Postdocs and Staff Researchers      0.466 0.165 Inf
 (Non-research Staff) / Students                            0.906 0.373 Inf
 Postdocs and Staff Researchers / Students                  1.943 0.867 Inf
 asymp.LCL asymp.UCL null z.ratio p.value
     0.349      2.05    1  -0.485  0.9624
     0.147      1.06    1  -2.415  0.0743
```

```
     0.248      2.37    1  -0.604  0.9308
     0.188      1.15    1  -2.162  0.1340
     0.315      2.61    1  -0.239  0.9952
     0.617      6.12    1   1.488  0.4448

Confidence level used: 0.95
Conf-level adjustment: tukey method for comparing a family of 4 estimates
Intervals are back-transformed from the log odds ratio scale
P value adjustment: tukey method for comparing a family of 4 estimates
Tests are performed on the log odds ratio scale
```

As expected, none of the differences between these smaller groups are significant. Let's try the larger groups (approach #1). (We'll report both results!)

Approach #2: 2 job categories

```
combined_relabeled <- combined %>%
    mutate(
    job_category = recode(
      job_category,
      "Students" = "Academic",
      "Postdocs and Staff Researchers" = "Academic",
      "Faculty" = "Academic"
    )
  )

unique(combined_relabeled$job_category)
```

```
[1] "Academic"           "Non-research Staff"
```

Copying code from above

```
# run a separate model for each outcome (motivation)
models_nr_acad <- lapply(motivation_cols, function(x) {
  # wrap the column name in backticks so "My Column" becomes `My Column`
  f_text <- paste0("`", x, "`", " ~ job_category")
  f <- as.formula(f_text)
  stats::glm(f, family = "binomial", data = combined_relabeled)
})
```

```
null_models_nr_acad <- lapply(motivation_cols, function(x) {
  # wrap the column name in backticks so "My Column" becomes `My Column`
  f_text <- paste0("`", x, "`", " ~ 1")
  f <- as.formula(f_text)
  stats::glm(f, family = "binomial", data = combined_relabeled)
})
```

If I wanted to (borderline) p-hack, I could just test "Job" and avoid multiple test correction, since the plot suggested that was the biggest difference. But I want to report robust differences that can withstand correction. So let's test them all and do the correction.

```
anova_results_nr_acad <- mapply(
  FUN = function(null_m, full_m) {
    stats::anova(null_m, full_m)
  },
  null_models_nr_acad,
  models_nr_acad,
  SIMPLIFY = FALSE
)

p_vals_nr_acad <- c()
for (i in seq_along(motivation_cols)) {
  p_vals_nr_acad[i] <- anova_results_nr_acad[[i]]$`Pr(>Chi)`[2]
}

# Just to get a sense of the range pre-correction
p_vals_nr_acad
```

```
[1] 0.37873923 0.12594150 0.31633710 0.52584751 0.19902643 0.06492533 0.35431301
```

```
p_fdr_nr_acad  <- p.adjust(p_vals_nr_acad, method = "BH")
for (i in seq_along(p_fdr_nr_acad)) {
  cat(
    sprintf(
      "%s  %.3f\n",
      motivation_cols[i],
      p_fdr_nr_acad[i] # p-value rounded to 3 decimals
    )
  )
}
```

```
Job  0.442
Improve Tools  0.441
Customize  0.442
Network  0.526
Give back  0.442
Skills  0.441
Fun  0.442
```

Okay, so even with just two job categories, and thus bigger sample sizes, including job_category as a fixed effect does not improve the model for the "it's part of my job" question. So it's only significant for IT vs. academics, not nr staff vs. academics.

Out of curiosity, what proportion of each group selected yes?

```r
combined_job <- combined %>%
  select(Job, job_category)

job_summary <- combined_job %>%
  group_by(job_category) %>%
  summarise(
    n_yes = sum(Job),
    total = length(Job),
    pct_yes = round(100 * sum(Job)/length(Job), 2)
  ) %>%
    ungroup()

job_summary
```

```
# A tibble: 4 x 4
  job_category                    n_yes total pct_yes
  <chr>                           <int> <int>   <dbl>
1 Faculty                            23    59    39.0
2 Non-research Staff                 37    86    43.0
3 Postdocs and Staff Researchers     34    55    61.8
4 Students                           15    33    45.4
```

Hm. Well, that's probably at least part of the reason why it's only significant when we group all the academics into one category. The percent of faculty who said yes is actually lower than the percent of NR staff who said yes.

We can get the same stats for just IT.

```
job_summary <- rbind(
  job_summary,
  c(
    "IT",
    sum(it$Job),
    length(it$Job),
    round(sum(it$Job) / length(it$Job) * 100, 2)
  )
)

job_summary
```

```
# A tibble: 5 x 4
  job_category                     n_yes total pct_yes
  <chr>                            <chr> <chr> <chr>
1 Faculty                          23    59    38.98
2 Non-research Staff               37    86    43.02
3 Postdocs and Staff Researchers 34    55    61.82
4 Students                         15    33    45.45
5 IT                               7     33    21.21
```

Okay, so most of the discrepancy is indeed driven by IT.

Why not round it out with a row for academics, too.

```
job_summary <- rbind(
  job_summary,
  c(
    "Academics",
    sum(academics$Job),
    length(academics$Job),
    round(sum(academics$Job) / length(academics$Job) * 100, 2)
  )
)

job_summary
```

```
# A tibble: 6 x 4
  job_category                     n_yes total pct_yes
  <chr>                            <chr> <chr> <chr>
1 Faculty                          23    59    38.98
```

2	Non-research Staff	37	86	43.02
3	Postdocs and Staff Researchers	34	55	61.82
4	Students	15	33	45.45
5	IT	7	33	21.21
6	Academics	72	147	48.98