

# Motivations for contributing to OS: statistical analysis

## Overview

This script runs some statistical tests on data from Q6, which is about participants' reasons for contributing to open source.

## 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

Here, we use some functions in my utility script (scripts/utilities.R) to clean up the data for Q6. We'll call the resulting data frame `motivations_raw`. This data frame also has a `Role` column indicating the participant's job category.



```

4          Faculty
5          Faculty
6          Faculty

```

Here, we combine postdocs and other research staff into one category. We'll call the resulting data frame `motivations_processed`. We will use this for most of our statistical analysis. It gives us more statistical power, and I think it is reasonable in terms of interpretability.

```

motivations_processed <- motivations_raw %>%
  mutate(
    Role = recode(
      Role,
      "Post-Doc" = "Postdocs and Staff Researchers",
      "Other research staff" = "Postdocs and Staff Researchers"
    )
  )

```

## Create the regression model

I'm interested in the whether a person's job category affects how they will answer this question. In other words, can we predict their profile of motivations significantly better when taking job category into account? I am doing a multivariate logistic regression predicting a vector of binary responses. It's multivariate because instead of doing  $Y \sim X$ , we are now doing  $[Y_1, Y_2, Y_3...] \sim X$ . It's logistic because all response variables are binary. I'm using the `mvabund()` package, which is designed for non-continuous data: counts and binary outcomes (they call these "abundance data" in the package documentation). Base R's `lm()` is for continuous data, and I didn't see a function in base R for this type of analysis.

First, we just split our data frame into two. For each observation,  $X$  is a single categorical outcome, and we have seven  $Y$ s which are binary outcomes.

```

Y <- as.matrix(motivations_processed[, motivation_cols])
X <- motivations_processed$Role
head(Y)

```

	Job	Improve	Tools	Customize	Network	Give back	Skills	Fun	Other
[1,]	1		1	1	1	1	1	1	0
[2,]	0		1	1	1	0	1	0	0
[3,]	0		1	1	0	0	1	1	0
[4,]	1		1	1	0	1	0	0	0
[5,]	0		1	1	0	1	1	1	0
[6,]	0		1	1	0	0	0	0	1

```
head(X)
```

```
[1] "Faculty" "Postdocs and Staff Researchers"
[3] "Postdocs and Staff Researchers" "Faculty"
[5] "Faculty" "Faculty"
```

Create the model. I'm mostly using the default settings.

```
fit <- mvabund::manyglm(Y ~ X, family = "binomial", show.coef=TRUE)
fit
```

Call: mvabund::manyglm(formula = Y ~ X, family = "binomial", show.coef = TRUE)

```
[1] "binomial(link=logit)"
```

Coefficients:

	Job	Improve Tools	Customize	Network
(Intercept)	-0.448	2.380	1.168	-0.989
XGrad Student	0.448	-0.343	-0.169	0.353
XNon-research Staff	0.167	-0.979	-0.543	0.415
XPostdocs and Staff Researchers	0.930	-0.609	-0.187	0.350
XUndergraduate	-0.468	-1.463	-2.959	0.701

	Give back	Skills	Fun	Other
(Intercept)	0.593	-0.448	0.170	-1.264
XGrad Student	0.405	1.883	0.641	-0.773
XNon-research Staff	0.356	1.230	0.207	-1.014
XPostdocs and Staff Researchers	-0.411	0.778	-0.426	-1.282
XUndergraduate	14.222	15.264	0.118	0.347

Degrees of Freedom: 232 Total (i.e. Null); 228 Residual

	Job	Improve Tools	Customize	Network	Give back
2*log-likelihood:	-314.0	-192.4	-276.4	-295.4	-284.7
Residual Deviance:	314.0	192.4	276.4	295.4	284.7
AIC:	324.0	202.4	286.4	305.4	294.7

	Skills	Fun	Other
2*log-likelihood:	-286.1	-314.6	-171.1
Residual Deviance:	286.1	314.6	171.1
AIC:	296.1	324.6	181.1

Immediately, we notice that the coefficients for Undergraduates on “Skills” and “Give back” are very different from all other coefficients. This is presumably because all 7 undergraduates who answered this question selected both those options.

The residual deviance statistic is close to the degrees of freedom, which I think suggests a good fit? <https://online.stat.psu.edu/stat504/lesson/2/2.5>

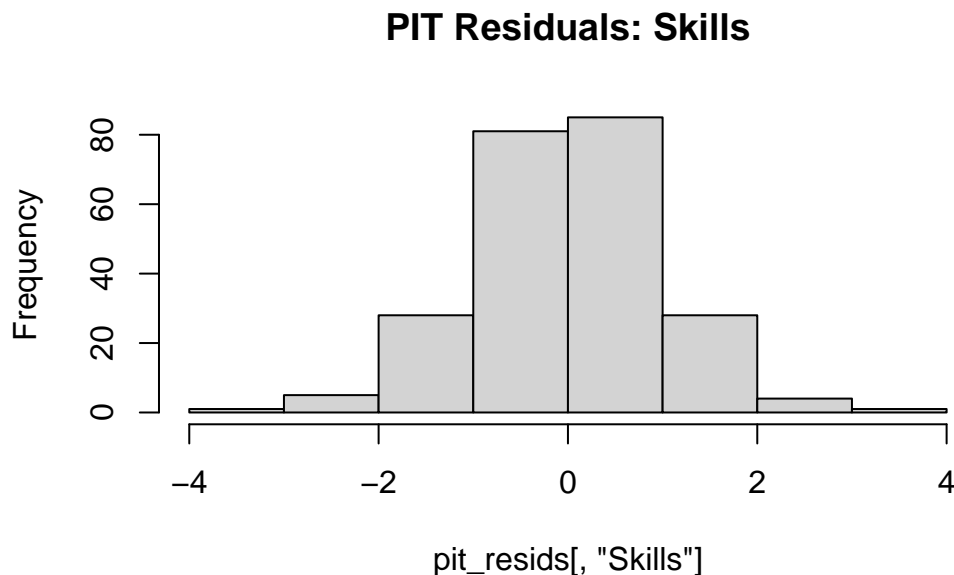
This is a really dumb way to assess goodness of fit, but I also like to look at the AIC, and if it's in the thousands, I start to get nervous. Ours are in the hundreds, so that seems promising?

## Assess goodness of fit

Next, I'm eyeballing some plots to assess goodness of fit.

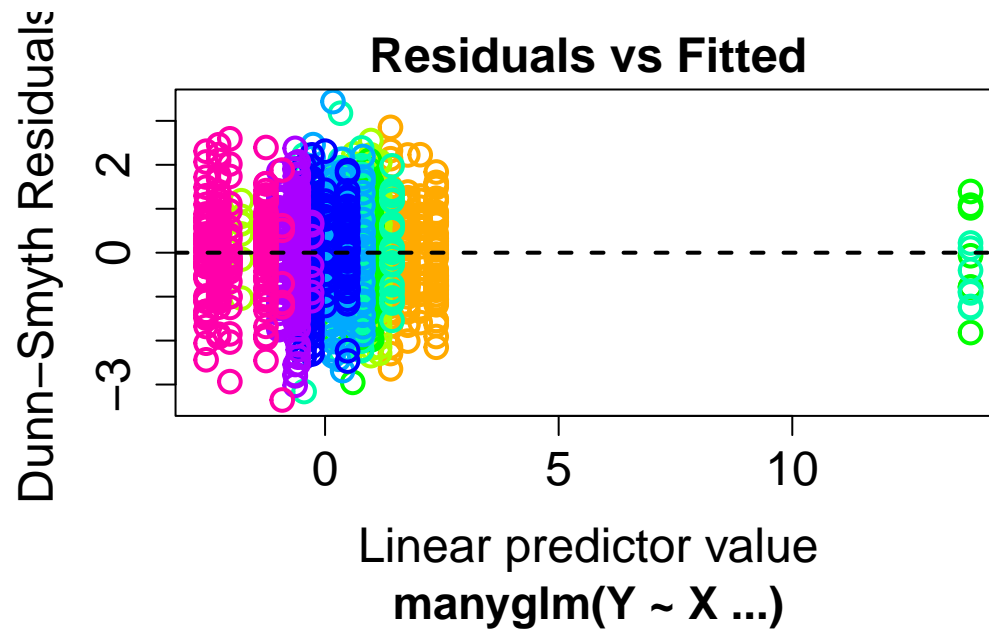
The distribution of the residuals seems normal-ish.

```
pit_resids <- residuals(fit, type = "pit.trap")
hist(pit_resids[, "Skills"], main = "PIT Residuals: Skills")
```



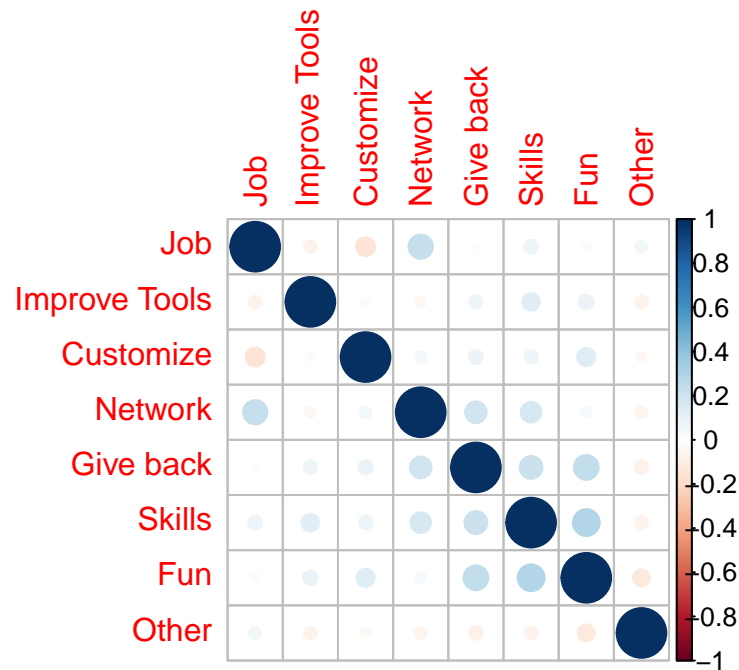
Admittedly, I don't fully understand this plot, but the paper emphasizes it. (<https://doi.org/10.1111/j.2041-210X.2012.00190.x>) I think the point is that the residuals should be distributed around zero. What you don't want is a "fan shape", where as the predictions get more extreme, the residuals do, too. This tutorial from the package author shows an example of this. <https://cran.r-project.org/web/packages/ecostats/vignettes/Chapter14Solutions.html> I think my plot look as good as the plots he approves of in that tutorial.

```
plot(fit)
```



I think a correlation matrix is also applicable here. The correlations between variables are near zero (ish), suggesting that the model captures the important relationships.

```
corrplot(cor(pit_resids), method = "circle")
```



## Hypothesis testing

Now we want to know whether this model is significantly better than a model where the probability of a particular set of motivations is the same for all job categories. `anova()` summarizes the statistical significance of the fitted model. `test="LR"` is the default, and specifies a likelihood ratio test. So I guess we are using the likelihood ratio test statistic instead of the standard anova F-statistic, but I think this might be the kind of situation where those two statistics are basically the same? (nested models for hypothesis vs. null) The `resamp="pit.trap"` ("probability integral transform" residuals) argument is the default resampling method. I think the function resamples the data to get a null distribution.

## Global model fit + univariate tests

```
anova_result <- anova(fit, resamp = "pit.trap", test = "LR", p.uni = "adjusted")
```

Time elapsed: 0 hr 0 min 0 sec

## anova\_result

### Analysis of Deviance Table

Model: Y ~ X

Multivariate test:

	Res.Df	Df.diff	Dev	Pr(>Dev)
(Intercept)	232			
X	228	4	76.37	0.001 ***

---

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

Univariate Tests:

	Job		Improve Tools		Customize		Network
	Dev	Pr(>Dev)		Dev	Pr(>Dev)		Dev
(Intercept)							
X	8.032	0.426		4.805	0.573	11.753	0.136
		Give back		Skills		Fun	Other
		Pr(>Dev)		Dev	Pr(>Dev)		Dev
(Intercept)							
X	0.805	10.943	0.156	25.603	0.001	5.715	0.556

	Pr(>Dev)
(Intercept)	
X	0.426

Arguments:

Test statistics calculated assuming uncorrelated response (for faster computation)  
P-value calculated using 999 iterations via PIT-trap resampling.

The anova output starts with a table of the multivariate test statistics. This tests for the global effect of Role, by resampling the whole response vector.

The next part of the table is the univariate test statistics, which are separate logistic regressions for each response variable, ignoring the other variables.

Our Pr(>Dev) is a statistically significant p-value, indicating that Role significantly predicts motivation profile. Basically, the model including role better explains the data than the null model.

Only “Skills” shows a significant univariate effect of Role after multiple testing correction (p.uni = “adjusted”). In other words, when considering whether Role can predict a single motivation, it can only predict Skills.



I think this also means that we have enough undergraduates, because if we didn't, we wouldn't have enough statistical power to reject the null hypothesis, right?

## Pairwise tests for job categories

I believe we can use the `pairwise.comp` argument to test whether pairs of categories in our explanatory variable are significantly different from each other.

```
anova_pw <- anova(  
  fit,  
  resamp = "pit.trap",  
  test = "LR",  
  p.uni = "adjusted",  
  pairwise.comp = X  
)
```

Time elapsed: 0 hr 0 min 0 sec

```
anova_pw
```

### Analysis of Deviance Table

Model: Y ~ X

Multivariate test:

	Res.Df	Df.diff	Dev	Pr(>Dev)
(Intercept)	232			
X	228	4	76.37	0.001 ***

---

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

Pairwise comparison results:

	Observed statistic
Faculty vs Undergraduate	31.555
Postdocs and Staff Researchers vs Undergraduate	30.628
Faculty vs Non-research Staff	25.727
Faculty vs Postdocs and Staff Researchers	19.755
Non-research Staff vs Undergraduate	19.450
Faculty vs Grad Student	18.768
Grad Student vs Undergraduate	18.417
Non-research Staff vs Postdocs and Staff Researchers	15.943

Grad Student vs Postdocs and Staff Researchers	13.104
Grad Student vs Non-research Staff	4.494
	Free Stepdown Adjusted P-Value
Faculty vs Undergraduate	0.015
Postdocs and Staff Researchers vs Undergraduate	0.018
Faculty vs Non-research Staff	0.052
Faculty vs Postdocs and Staff Researchers	0.193
Non-research Staff vs Undergraduate	0.193
Faculty vs Grad Student	0.193
Grad Student vs Undergraduate	0.193
Non-research Staff vs Postdocs and Staff Researchers	0.209
Grad Student vs Postdocs and Staff Researchers	0.285
Grad Student vs Non-research Staff	0.794

Faculty vs Undergraduate	*
Postdocs and Staff Researchers vs Undergraduate	*
Faculty vs Non-research Staff	.
Faculty vs Postdocs and Staff Researchers	
Non-research Staff vs Undergraduate	
Faculty vs Grad Student	
Grad Student vs Undergraduate	
Non-research Staff vs Postdocs and Staff Researchers	
Grad Student vs Postdocs and Staff Researchers	
Grad Student vs Non-research Staff	

---

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

#### Univariate Tests:

	Job		Improve Tools		Customize		Network	
	Dev	Pr(>Dev)	Dev	Pr(>Dev)	Dev	Pr(>Dev)	Dev	
(Intercept)								
X	8.032	0.421	4.805	0.592	11.753	0.153	1.671	
		Give back		Skills		Fun		Other
	Pr(>Dev)	Dev	Pr(>Dev)	Dev	Pr(>Dev)	Dev	Pr(>Dev)	Dev
(Intercept)								
X	0.815	10.943	0.186	25.603	0.001	5.715	0.573	7.852

Pr(>Dev)

(Intercept)

X 0.421

#### Arguments:

Test statistics calculated assuming uncorrelated response (for faster computation)

P-value calculated using 999 iterations via PIT-trap resampling.

The results indicate three significant pairwise comparisons:

- \* Faculty vs Undergraduate
- \* Postdocs and Staff Researchers vs Undergraduate
- \* Faculty vs Non-research Staff

## 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 Cochran-Armitage test for trend to evaluate whether this trend is real. More precisely, I believe we are evaluating whether the order “ $P(\text{Yes} \mid \text{Undergrad}) > P(\text{Yes} \mid \text{Grad}) > P(\text{Yes} \mid \text{Postdoc}) > P(\text{Yes} \mid \text{Faculty})$ ” 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. I can show those data later. 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 the way a Cochran-Armitage test is meant to be interpreted. 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.

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

n_grad <- sum(motivations_processed$Role == "Grad Student")
n_grad_yes <- sum(
  motivations_processed$Role == "Grad Student" &
  motivations_processed$Skills == 1
)
```

```

)

n_undergrad <- sum(motivations_processed$Role == "Undergraduate")
n_undergrad_yes <- sum(
  motivations_processed$Role == "Undergraduate" &
  motivations_processed$Skills == 1
)

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  $p=0.05$ .