Solutions Stats Part 2: Aspiring Contributors

Now let's look at Q15, where we asked aspiring contributors what would make them more likely to participate in open source projects.

This is the second time now that I am flummoxed with how to analyze a "select all that apply" question. While it seems strange to me that statisticians haven't come up with a better solution, from my Googlings, it seems like it's pretty common to just make a bunch of little models with a single binary outcome variable, rather than trying to model a bunch of binary outcomes at once. So let's try that approach.

https://www.statalist.org/forums/forum/general-stata-discussion/general/1394776-multivariate-analysis-on-a-multiple-selection-survey-question

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

```
future_solns <- load_qualtrics_data("clean_data/future_contributors_Q15.tsv")
other_quant <- load_qualtrics_data("clean_data/other_quant.tsv")
```

Wrangle data

```r
# build a logical index: TRUE for rows that have at least one non-zero entry
# rowSums(future_solns != 0) tallies the total for each row
keep <- rowSums(future_solns != 0) > 0

# subset both data.frames
future_solns_clean <- future_solns[keep, ]

other_quant_clean <- other_quant[keep, "job_category", drop = FALSE]

future_solns_clean <- cbind(future_solns_clean, other_quant_clean)
nrow(future_solns_clean)
```

```
[1] 61
```

Good. We know we had 61 survey respondents who identified as aspiring contributors.

Let's remove the "Other" option.

```r
future_solns_clean <- future_solns_clean %>%
    select(-c("Other"))
```

Since we have much less data than we did from experienced contributors, I'm just going to assume a priori that we'll want to relabel the groups to have a smaller number of job categories, as we did with the experienced contributors. This way, each category will have more observations.

```r
future_solns_clean <- future_solns_clean %>%
  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
    )
  )
```

Create models

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

```r
# remove "job_category" from this list of strings
solution_names <- names(future_solns_clean)[-ncol(future_solns_clean)]

# run a separate model for each outcome (solution)
models <- lapply(solution_names, 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 = future_solns_clean)
})

# example
models[[1]]
```

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

Coefficients:
                                (Intercept)
                                    -0.9163
            job_categoryNon-research Staff
                                     0.7156
job_categoryPost-docs and staff researchers
                                     0.3567
                       job_categoryStudents
                                     0.6539

Degrees of Freedom: 60 Total (i.e. Null);  57 Residual
Null Deviance:      82.57
Residual Deviance: 81.81    AIC: 89.81
```

Quick AIC check just because it's easy.

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

```
Conferences and hackathons  89.814
Computing environments  86.070
Educational materials  90.165
Learning community  86.874
Dedicated grants  76.477
Industry networking  86.754
Academic job opportunities  80.100
Help finding funding  82.309
Legal support  83.083
Mentoring programs  87.152
```

Hmm. The model for "Dedicated grants" looks a bit better than the others in terms of AIC, and "Educational materials" is the worst. Good to know.

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

```
null_models <- lapply(solution_names, 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 = future_solns_clean)
})
```

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

```
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: `Conferences and hackathons` ~ 1
Model 2: `Conferences and hackathons` ~ job_category
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        60     82.569
2        57     81.814  3  0.75518   0.8602
```

Womp womp. At least for that solution, "Conferences and hackathons", including the predictor variable, job, does not improve model fit. Let's look at p-values for all the ANOVAs.

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

```
Conferences and hackathons  0.860
Computing environments  0.924
Educational materials  0.785
Learning community  0.408
Dedicated grants  0.010
Industry networking  0.129
Academic job opportunities  0.054
Help finding funding  0.237
Legal support  0.197
Mentoring programs  0.153
```

Hmm. Only for "Dedicated grants" does including the job predictor make a difference ($p<0.05$). This seems reasonable, considering that this model had the best fit, and only in this case was the predictor variable necessary.

Let's do pairwise contrasts to dig deeper.

```r
names(models) <- solution_names

#I'm doing this in a functionalized way
```

```
# (across a list containing only one model)
# because I originally ran this script with the
# un-consolidated group labels, which resulted in
# more than one solution being significantly improved
# by including job_category.
# After consolidating the group labels, only "grants"
# was worth probing futher, but I didn't feel like
# bothering to refactor the code.

sig_models <- models[
  sapply(anova_results, function(a) a$`Pr(>Chi)`[2] < 0.05)
]

sig_models
```

```
$`Dedicated grants`

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

Coefficients:
                                (Intercept)
                                    -0.2877
             job_categoryNon-research Staff
                                    -1.9095
job_categoryPost-docs and staff researchers
                                    -0.2719
                       job_categoryStudents
                                     0.5500

Degrees of Freedom: 60 Total (i.e. Null);  57 Residual
Null Deviance:      79.76
Residual Deviance: 68.48    AIC: 76.48
```

Here we get estimated marginal means on a "response" scale (For our purposes, I believe this just means we get odds ratios instead of log odds.)

```
emm_grids_resp <- sapply(seq_along(sig_models), function(i) {
  return(
    emmeans(sig_models[[i]], ~ job_category, type = "response")
  )
})
```

```r
pair_results_resp <- sapply(seq_along(emm_grids_resp), function(i) {
  return(
    pairs(emm_grids_resp[[i]])
  )
})

names(pair_results_resp) <- names(sig_models)

for (i in seq_along(pair_results_resp)) {
  res <- summary(pair_results_resp[[i]], infer=TRUE)
  if (any(res$p.value < 0.05)) {
    print(names(pair_results_resp)[i])
    print(res)
    }
}
```

```
[1] "Dedicated grants"
 contrast                                                 odds.ratio     SE  df
 Faculty / (Non-research Staff)                               6.7500 7.2000 Inf
 Faculty / (Post-docs and staff researchers)                  1.3125 1.3000 Inf
 Faculty / Students                                           0.5769 0.5030 Inf
 (Non-research Staff) / (Post-docs and staff researchers)     0.1944 0.1890 Inf
 (Non-research Staff) / Students                              0.0855 0.0731 Inf
 (Post-docs and staff researchers) / Students                 0.4396 0.3320 Inf
 asymp.LCL asymp.UCL null z.ratio p.value
   0.43518    104.70    1   1.789  0.2784
   0.10369     16.61    1   0.275  0.9927
   0.06142      5.42    1  -0.631  0.9221
   0.01593      2.37    1  -1.682  0.3334
   0.00948      0.77    1  -2.874  0.0211
   0.06322      3.06    1  -1.089  0.6963

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
```

This analysis indicates that non-research staff are much less likely that students to select this solution. Let's do a sanity check on the raw data.

```r
tally_df <- aggregate(
  . ~ job_category,
  data = future_solns_clean,
  FUN  = function(x) sum(x, na.rm = TRUE)
)

head(tally_df)
```

```
                     job_category Conferences and hackathons
1                         Faculty                          2
2             Non-research Staff                           9
3 Post-docs and staff researchers                          4
4                        Students                         10
  Computing environments Educational materials Learning community
1                      4                     4                  3
2                     13                    10                 14
3                      8                     6                  5
4                     15                    15                 15
  Dedicated grants Industry networking Academic job opportunities
1                3                   3                          1
2                2                  12                          6
3                4                   2                          2
4               13                  12                         13
  Help finding funding Legal support Mentoring programs
1                    3             2                  2
2                    4             6                  7
3                    3             2                  5
4                   11            12                 15
```

K, looks good. Out of curiosity, does running emmeans without type="response", i.e., keeping it on a latent scale, make a difference?

```r
emm_grids <- sapply(seq_along(sig_models), function(i) {
  return(
    emmeans(sig_models[[i]], ~ job_category)
  )
})

pair_results <- sapply(seq_along(emm_grids), function(i) {
  return(
    pairs(emm_grids[[i]])
  )
})

names(pair_results) <- names(sig_models)

for (i in seq_along(pair_results)) {
  res <- summary(pair_results[[i]], infer=TRUE)
  if (any(res$p.value < 0.05)) {
    print(names(pair_results)[i])
    print(res)
    }
}
```

```
[1] "Dedicated grants"
 contrast                                                 estimate    SE  df
 Faculty - (Non-research Staff)                              1.910 1.070 Inf
 Faculty - (Post-docs and staff researchers)                 0.272 0.988 Inf
 Faculty - Students                                         -0.550 0.872 Inf
 (Non-research Staff) - (Post-docs and staff researchers)   -1.638 0.974 Inf
 (Non-research Staff) - Students                            -2.460 0.856 Inf
 (Post-docs and staff researchers) - Students               -0.822 0.755 Inf
 asymp.LCL asymp.UCL z.ratio p.value
    -0.832     4.651   1.789  0.2784
    -2.266     2.810   0.275  0.9927
    -2.790     1.690  -0.631  0.9221
    -4.139     0.864  -1.682  0.3334
    -4.658    -0.261  -2.874  0.0211
    -2.761     1.117  -1.089  0.6963

Results are given on the log odds ratio (not the response) scale.
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
```

Okay, so it changed the estimates, but not the p-values. Good. That makes sense.

Since we're here and it's interesting, let's add a row with the totals for the numeric columns.

```r
total_row <- tally_df %>%
  summarise(
    across(
      where(is.numeric),
      \(x) sum(x, na.rm = TRUE)
  )) %>%
  # add the "Total" label as an entry in the job_category column
  mutate(job_category = "Total") %>%
  select(job_category, everything())  # put job_category first

tally_df_with_total <- bind_rows(tally_df, total_row)

tally_df_with_total
```

```
                     job_category Conferences and hackathons
1                         Faculty                          2
2              Non-research Staff                          9
3 Post-docs and staff researchers                          4
4                        Students                         10
5                           Total                         25
  Computing environments Educational materials Learning community
1                      4                     4                  3
2                     13                    10                 14
3                      8                     6                  5
4                     15                    15                 15
5                     40                    35                 37
  Dedicated grants Industry networking Academic job opportunities
1                3                   3                          1
2                2                  12                          6
3                4                   2                          2
4               13                  12                         13
5               22                  29                         22
  Help finding funding Legal support Mentoring programs
1                    3             2                  2
2                    4             6                  7
3                    3             2                  5
4                   11            12                 15
5                   21            22                 29
```

So, here's my hot take.
- We don't have enough data to conclude whether the most popular solutions, in terms of total votes, are "significantly different" from each other. What does that even mean when we only have one observation?
- I relabeled some of the groups to make them larger, adding more statistical power.
- Only the "Dedicated grants" model was improved by adding job_category as a factor.

- The only significant pairwise difference was that Students are significantly more likely than Non-research staff to select Dedicated grants.

Compare aspiring vs. experienced?

Can we bring in a comparison to experienced contributors? It will have to be rudimentary since the questions are not really comparable. The question for experienced contributors was a likert scale Q, while the question for aspiring contributors was a "select all that apply". The options presented were also not exactly the same, though this was to some extent unavoidable, since some solutions only apply to experienced folks.

Maybe we can look at the ordered solutions from each and say whether the rankings are different.

Let's get the aspiring contributors' solutions, ordered by total number of votes.

```r
totals <- colSums(tally_df[ , -1], na.rm = TRUE)

ordered_solutions <- names(sort(totals, decreasing = TRUE))

print(ordered_solutions)
```

```
 [1] "Computing environments"     "Learning community"
 [3] "Educational materials"      "Industry networking"
 [5] "Mentoring programs"         "Conferences and hackathons"
 [7] "Dedicated grants"           "Academic job opportunities"
 [9] "Legal support"              "Help finding funding"
```

Lifting some code from solutions_plots.qmd to get the list of solutions from aspiring contributors.

```r
solutions <- load_qualtrics_data("clean_data/solutions_Q10.tsv")
other_quant <- load_qualtrics_data("clean_data/other_quant.tsv")

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

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
    )
  )

# 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),
    median = median(utility_score),
    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$solution
```

```
 [1] "Sustainability grants"  "Help finding funding"   "Computing environments"
 [4] "A learning community"   "Documentation help"     "Legal support"
 [7] "Education"              "Industry partnerships"  "Publicity"
[10] "Mentoring programs"     "Containerization"       "Event planning"
```

So, here are the two lists again, in an easier-to-read format:

Aspiring	Experienced
Computing environments	Sustainability grants
A learning community	Help finding funding
Educational materials	Computing environments
Industry networking	A learning community
Mentoring programs	Documentation help
Conferences and hackathons	Legal support
Legal support/ Academic job opportunities/ Dedicated grants	Educational materials
Help finding funding	Industry partnerships
	Publicity
	Mentoring programs
	Containerization
	Event planning

Hm. I don't like this. It feels icky to be comparing two questions that are really quite different. I think this is all I'm comfortable concluding:
- The two funding-related solutions are in the bottom quartile among aspiring contributors, but they are in the top quartile among experienced contributors.
- Learning community and Computing environments are in the top quartile for both groups.

We could maybe argue that education-related solutions are higher up the list for aspiring contributors, but in the absence of statistical testing (which we could really only do if the questions were more comparable), I'd rather not draw any conclusions.