COVID-19's Impact on ASD - Replication Package

Pernilla Ågren & Eli Knoph

July 2021

1 Introduction

This is a replication package created based on the data used in COVID-19's Impact on Agile Software Development.

1.1 Loading Data

The data used is available at Zenodo[1]. In this guide, we we use the excel data called RawData_excel.xlsx, but there is also a csv version available.

The statistical models were built and fitted using the brms (Bayesian regression models using 'Stan') package [1] in R. Besides the brms package, there are a couple of other libraries that need to be imported as well before loading the data.

```
library(brms)
library(rstan)
library(openxlsx)
library(bayesplot)
library(ggplot2)
library(here)
```

The data is loaded into variable *d*, and the NA fields are coded as N/A.

```
d <- read.xlsx(here("data/RawData\_excel.xlsx"), na.strings = "N/A")
```

The command - dim(d) - is run to see how many rows and columns we have in the data table d. If everything went well we should have 96 rows and 112 columns.

```
dim(d)
```

[1] https://doi.org/10.5281/zenodo.4916225

There might be a need to deal with missing data, therefore, we performed a complete case analysis to see how many rows that have complete answers. Out of these 96 rows, only six entries have complete answers, which means that some respondents skipped some questions or dropped out in the middle of the questionnaire. This means that we have to deal with missing data in the upcoming steps.

```
nrow( d[complete.cases(d), ] )
```

To check the data types, we run the command str(d).

```
str(d)
```

If everything went well, all rows should be of a numeric type, labelled as *num*. An extract of the generated output is illustrated below.

```
'data.frame':        96 obs. of  112 variables:
 $ Number       : num  1 2 3 4 5 6 7 8 9 10 ...
 $ P_Q1_3C      : num  1 2 2 1 2 2 2 2 2 3 ...
 $ P_Q2         : num  26 7 3 8 7 8 5 6 5 8 ...
 $ P_Q3_5C      : num  3 4 2 4 2 2 2 3 2 4 ...
 $ P_Q4_3C      : num  3 3 3 3 1 1 3 3 3 1 ...
 $ P_Q5_3C      : num  1 1 1 1 2 1 1 1 1 1 ...
 $ P_Q6_3C      : num  3 1 3 3 2 1 1 1 1 2 ...
 $ P_Q7         : num  5 3 3 7 20 4 2 1 2 1 ...
 $ P_Q9_1_5L    : num  2 1 1 2 1 1 1 1 1 1 ...
 $ P_Q10_1_5L   : num  5 4 5 5 5 5 1 5 1 5 ...
 $ P_Q11_7C     : num  3 3 3 3 4 3 NA 4 NA 3 ...
 $ P_Q12_1_7L   : num  2 2 2 6 2 4 NA 7 NA 5 ...
```

1.2 Data Cleaning

Out of a total of 111 questions (excluding open questions) from the questionnaire, 11 of the demographic questions were set as predictors (Q1-Q7, Q9-Q12), and the rest as outcomes. Question Q8 (Which development methodology do you mainly follow in your team/work now?) were removed from the data analysis, as mentioned in the thesis.

The predictors Q2 and Q7 were continuous variables, and were therefore standardized using scale() and recoded into two new variable names as shown below.

```
d\$P_Q2_s <- scale(d\$P_Q2)
d\$P_Q7_s <- scale(d\$P_Q7)
```

1.3 Descriptive Statistics

Out of the 96 rows in the cleaned data set, there are 96 answers to the demographic questions Q1, Q3, Q4, Q5, Q6, Q9, and Q10. For Q2 and Q11, there are 92 answers; for Q7 there are 93 answers; and for Q12, there are 91 answers.

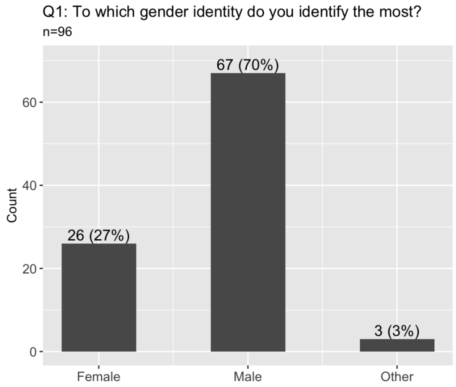

Figure 1: Predictor: Gender

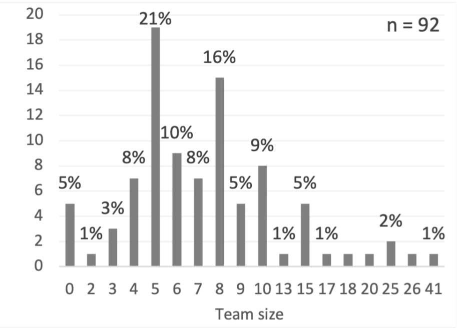

Figure 2: Predictor: Team Size

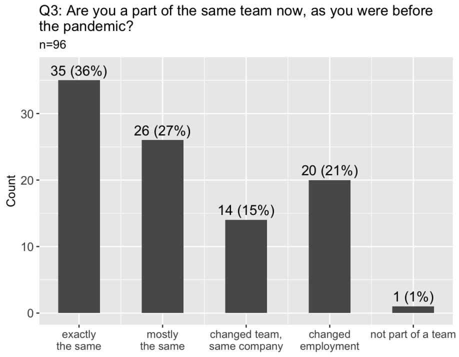

Figure 3: Predictor: Team Constellation

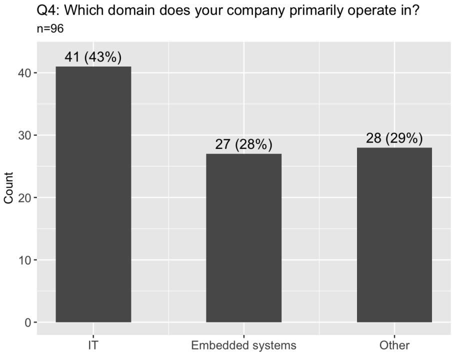

Figure 4: Predictor: Domain

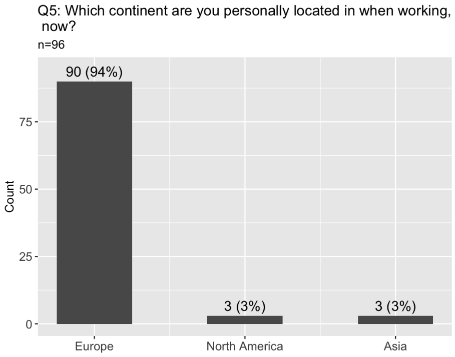

Figure 5: Predictor: Continent

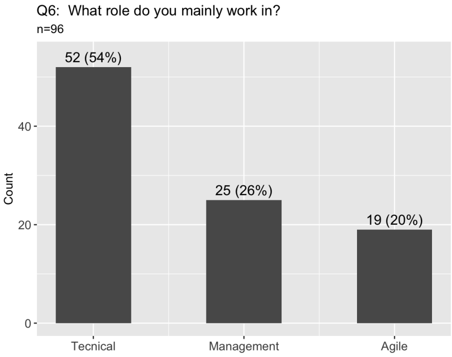

Figure 6: Predictor: Role

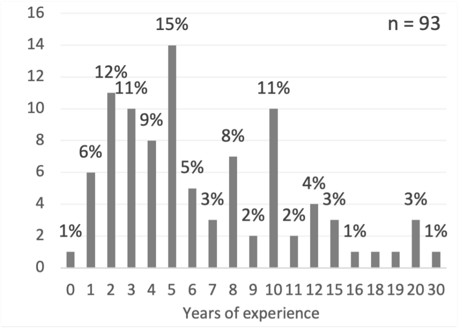

Figure 7: Predictor: Years of Experience with Agile Software Development

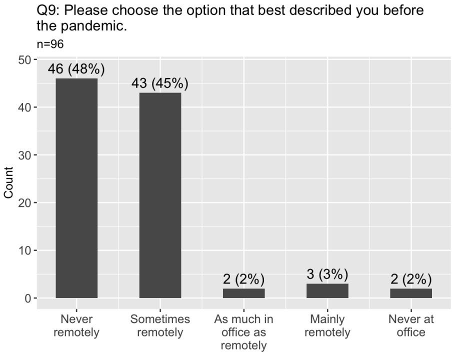

Figure 8: Predictor: How the practitioner worked before the pandemic

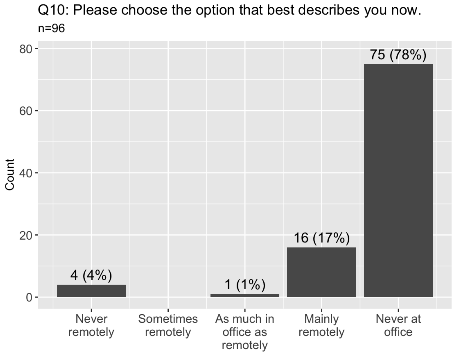

Figure 9: Predictor: How the respondent work "now" (at time of filling out the questionnaire)

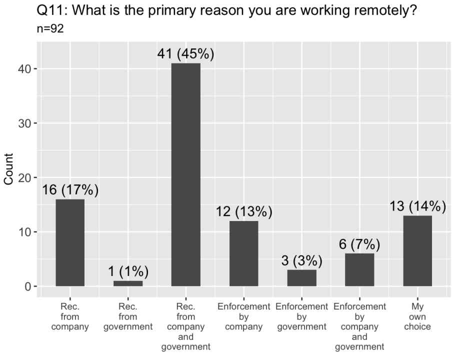

Figure 10: Predictor: Reason for working remotely

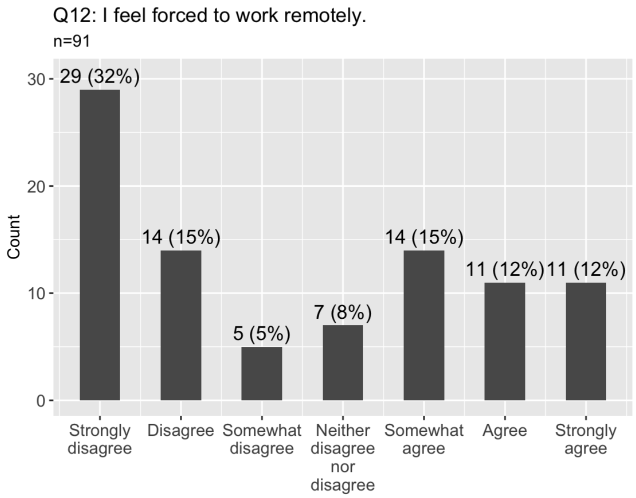

Figure 11: Predictor: If the respondent feels forced to work remotely or not

2 Model Design

Models were created for all 100 outcomes, Q13-Q26, including their sub questions. Throughout this guide, we will describe the procedure of the Bayesian analysis with the example of one model. The model we use is Q16 as an outcome variable, and eleven demographic questions as predictors. The same process is done for all outcome models.

We begin by filtering out the outcomes we do not need for the creation of the models, and store the data in a new data table called d16. This command leaves us with 13 columns in the data table, namely Q1, Q3, Q4, Q5, Q6, Q7, Q9, Q10, Q11, Q12, Q16 and the scaled variables Q2_s and Q7_s.

```
d16 <- d[-c(3,8,13:31,33:112)]
```

We then perform a complete case analysis of the data to avoid having to deal with missing data. This means that we only keep the rows with values in each one of the 13 columns for the analysis.

```
d16 <- d16[ complete.cases(d16), ]
```

Thereafter, we check the number of rows and columns, which is 76 and 13 respectively. This means that we have reduced the data set from 96 rows to 76 rows.

```
dim(d16)
```

2.1 Prior Predictive Checks

The command - get_prior() - is run to see what kind of predictors the model needs, and the list of the priors needed are stored in the variable p16.

```
p16 <- get_prior(
    O_Q16_1_7L ~ 1 + P_Q1_3C + P_Q2_s +
    P_Q3_5C + P_Q4_3C + P_Q5_3C + P_Q6_3C +
    P_Q7_s + mo(P_Q9_1_5L) +
    mo(P_Q10_1_5L) + P_Q11_7C + mo(P_Q12_1_7L),
    data = d16,
    family = cumulative)
p16
```

The predictors Q9, Q10 and Q12 were ordered categorical predictors, and were therefore modeled with monoticity (which is done by enclosing the predictor within mo()). We perform a sensitivity analysis of our priors, and choose weakly informative priors to affect the model as little as possible. For most of the predictors, the priors were set to 'normal(0, 0.25)' [2], and for the ordered categorical (monotonic) predictors, the priors were set to 'dirichlet(2)'.

[2] normal(0, 0.25) corresponds to a normal distribution with a mean of 0 and a standard deviation of 0.25

```
p16$prior[1] <- "normal(0, 0.25)"
p16$prior[13] <- "normal(0, 2)"
p16$prior[c(20:22)] <- "dirichlet(2)"
```

Then we create the actual model, and start by only sampling from priors.

```
m_full16 <- brm(
    O_Q16_1_7L ~ 1 + P_Q1_3C + P_Q2_s + P_Q3_5C + P_Q4_3C +
    P_Q5_3C + P_Q6_3C + P_Q7_s + mo(P_Q9_1_5L) + mo(P_Q10_1_5L) +
    P_Q11_7C + mo(P_Q12_1_7L),
    family = cumulative(),
    prior = p16,
    sample_prior = "only",
    control = list(adapt_delta = 0.95),
    data = d16,
    backend = "cmdstanr"
)
```

The output will look similar to the text below.

```
Compiling Stan program...
Start sampling
Running MCMC with 4 parallel chains...
[...]
Chain 1 finished in 1.1 seconds.
Chain 2 finished in 1.1 seconds.
Chain 3 finished in 1.1 seconds.
Chain 4 finished in 1.1 seconds.
[...]
All 4 chains finished successfully.
Mean chain execution time: 1.1 seconds.
Total execution time: 1.3 seconds.
```

When we have samples from only the priors, we do a prior predictive (pp) check to make sure that our priors affect the empirical data as little as possible. The result can be seen in Figure 12 and shows that means are approximately uniform and that the intervals are of approximately the same size.

```
pp_check(m_full16, type = "bars", nsamples = 500)
```

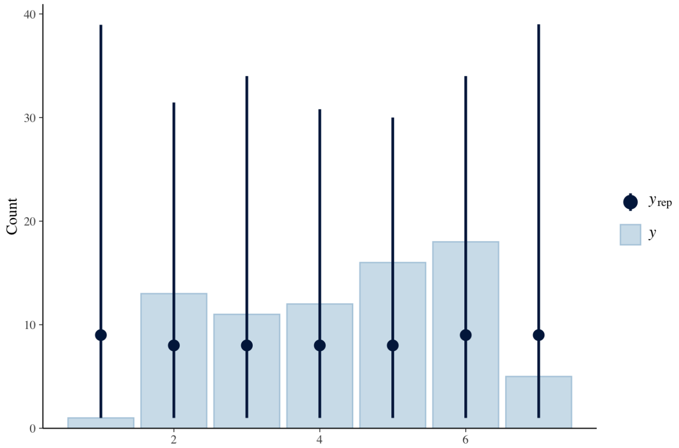

Figure 12: Plot from pp_check() (prior predictive check).

2.2 Sample With Data

After setting the priors we can create the final model and sample with real data.

```
m_full16 <- brm(
    O_Q16_1_7L ˜ 1 + P_Q1_3C + P_Q2_s + P_Q3_5C +
    P_Q4_3C + P_Q5_3C + P_Q6_3C + P_Q7_s + mo(P_Q9_1_5L) +
    mo(P_Q10_1_5L) + P_Q11_7C + mo(P_Q12_1_7L),
    family = cumulative(),
    prior = p16,
    data = d16,
    backend = "cmdstanr"
)
```

The output from the sampling will look similar to the text below, and as seen, this step takes longer time then the previous sampling.

```
Compiling Stan program...
Start sampling
Running MCMC with 4 parallel chains...
[...]
Chain 4 finished in 16.9 seconds.
Chain 2 finished in 17.4 seconds.
Chain 3 finished in 17.4 seconds.
Chain 1 finished in 17.7 seconds.
[...]
All 4 chains finished successfully.
Mean chain execution time: 17.3 seconds.
Total execution time: 17.9 seconds.
```

2.3 Diagnostics

After having fitted the model and sampled from the observed data, we check for some of the model's diagnostics: divergence, energy, tree depth, effective sample size (ESS), and Rhat.

First, we check the Hamiltonian Monte Carlo (HMC) diagnostics.

```
check_hmc_diagnostics(m_full16\$fit)
```

The output is shown below, and illustrates a desired result, that is, no divergences, no saturation of the maximum tree depth, and no pathological behavior.

```
Divergences:
0 of 4000 iterations ended with a divergence.

Tree depth:
0 of 4000 iterations saturated the maximum tree depth of 10.

Energy:
E-BFMI indicated no pathological behavior.
```

Thereafter, we check the ESS, which should not be below 0.1. In this case ESS is approximately 0.41, which is fine.

```
min(neff_ratio(m_full16), na.rm=TRUE)
```

Rhat should go towards 1.00, and if it exceeds 1.01, it is an indication of bias. In this case, Rhat = 1.001462, which is fine.

```
max(rhat(m_full16), na.rm=TRUE)
```

One additional way to check the diagnostics is to plot a trace plot using mcmc_trace().

```
mcmc_trace(
    m_full16,
    pars = c("b_P_Q1_3C", "b_P_Q2_s", "b_P_Q3_5C", "b_P_Q4_3C", "b_P_Q5_3C", "b_P_Q6_3
    )
```

The plots should look like fat hairy caterpillars as seen in Figure 13, which means that the chains are well-mixed.

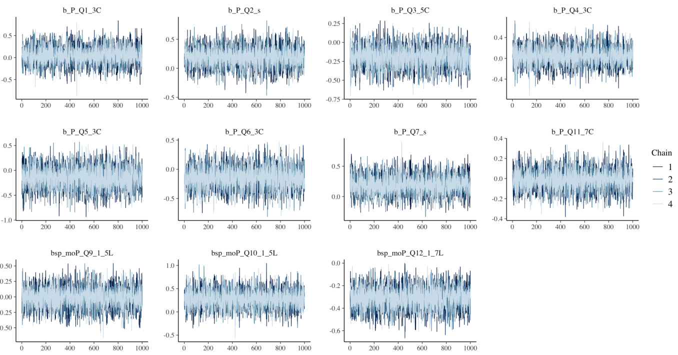

Figure 13: A trace plot for some of the estimated parameters, where fat hairy caterpillar plots indicate a good result.

2.4 Posterior Predictive Checks

Similar to how we performed prior predictive checks, we also performed posterior predictive checks in order to check how well the model estimated the different levels of the outcome.

```
    pp_check(m_full16, type = "bars", nsamples = 500)
```

In this case we have seven levels, and as seen in Figure 14, the model estimated perfectly on four levels, and slightly above or below on the other levels.

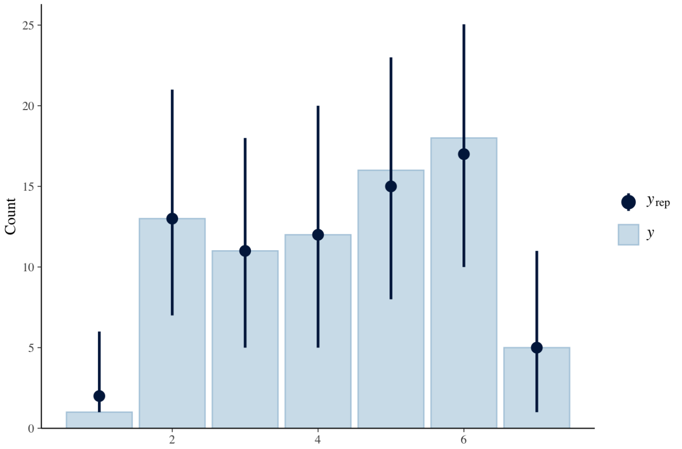

Figure 14: A posterior predictive check of how well the model estimated the different levels of outcome Q16.

3 Model Comparison

As all of the outcome variables were of an ordinal categorical type, there were at least three different types of models to choose among: cumulative, sequential, or adjacent-category [1]. To decide which likelihood to use for the model, we built one model of each of the three aforementioned types, and then performed a model comparison. The following example illustrates three different models with only one predictor.

```
m_c <- brm(
  O_Q16_1_7L ~ 1 + P_Q2_s,
  family = cumulative(),
  data = d,
  backend = "cmdstanr"
)

m_ac <- brm(
  O_Q16_1_7L ~ 1 + P_Q2_s,
  family = acat(),
  data = d,
  backend = "cmdstanr"
)

m_sr <- brm(
  O_Q16_1_7L ~ 1 + P_Q2_s,
  family = sratio(),
  data = d,
  backend = "cmdstanr"
)
```

The model comparison was performed using loo(), as seen below.

```
loo(m_c, m_ac, m_sr)
```

As seen in Table 1, the model comparison did not show any significant difference between the different types of models, and we therefore settled on using the cumulative family (model m_c in the example below) since it is rooted in mathematical theory [2] and usually suitable for Likert-item data [1].

Table 1: Model comparison using loo(), which shows that there is no significant difference between the different types of models.

model	elpd_diff	se_diff	elpd_loo estimate	elpd_loo SE
m_c	0.0	0.0	-153.7	4.6
m_ac	-0.1	0.5	-153.9	4.6
m_sr	-0.6	0.7	-154.3	4.5

4 Estimates and Effects

Significant effects are identified by either comparing the values of the lower and upper credible interval (CI)[3] values of the population level effects for the given outcome and predictors, as seen in Table 2, or by plotting probability densities, as seen in Figure 15. An effect is significant if the CI values do not include zero, that is, both the lower and upper CI are negative or positive [3].

4.1 Parameter estimates

To determine what effects that are significant, we run the command summary() to get the population level effects.

```
summary(m_full16)
```

An extract of the population level effects are listed in Table 2. As described, an effect is significant if both the lower and upper CI values are located on the same side of zero, and as seen in Table 2, only one of the predictors is significant, namely moP_Q12_1_7L, which affects the outcome negatively.

Table 2: Population level effects on Logit-scale. Significant effects are marked in bold.

Variable	Estimate	Est. Error	l-95% CI	u-95% CI
P_Q1_3C	0.05	0.21	-0.35	0.46
P_Q2_s	0.19	0.17	-0.15	0.53
P_Q3_5C	-0.19	0.14	-0.48	0.09
P_Q4_3C	0.05	0.18	-0.30	0.40
P_Q5_3C	-0.15	0.23	-0.60	0.29
P_Q6_3C	-0.15	0.19	-0.52	0.22
P_Q7_s	0.20	0.16	-0.11	0.52
P_Q11_7C	-0.01	0.11	-0.22	0.20
moP_Q9_1_5L	-0.01	0.17	-0.35	0.32
moP_Q10_1_5L	0.28	0.20	-0.11	0.68
moP_Q12_1_7L	-0.31	0.10	**-0.51**	**-0.12**

[3] The CI reflects the uncertainty interval and is referred to as confidence intervals in frequentist statistics [3]

As an illustration on how to visually determine what effects that are significant, we plot the posterior probability effects using the command mcmc_areas(). The result is illustrated in Figure 15).

```
mcmc_areas_ridges(
    m_full16,
    regex_pars = c("^b_P", "^bsp_mo"),
    prob = 0.5,
    prob_outer = 0.95
    )
```

Figure 15 illustrates probability densities for each predictor of outcome Q16 (self-reported productivity). Similarly to the CI values, the density plots that do not cross zero represent a significant effect. By examining the figure, only one of the density plots do not cross zero, namely the density plot for Q12 in the bottom of Figure 15. In other words, Q12 is the only significant predictor for this model, and this significance means that on the 95%-level, Q12 significantly negatively affects the outcome.

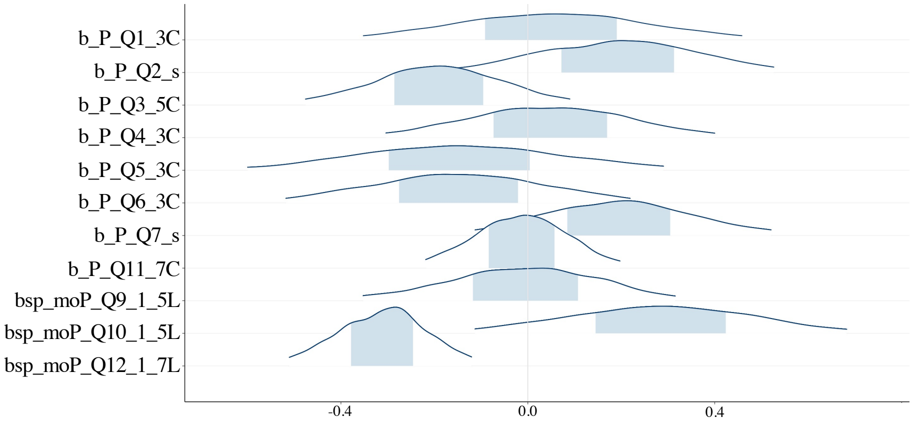

Figure 15: Posterior probability densities plotted on a logit scale for outcome Q16. The curves correspond to 95% probability and the shaded area corresponds to 50% of the probability. The only significant predictor was Q12, which is located at the bottom of the figure. Predictor Q2, Q7, and Q10 showed some tendencies toward positive significance, whereas Q3 and Q6 showed tendencies toward negative significance. Predictor Q1, Q4, Q11 and Q9 were clearly not significant.

By examining Figure 15, it is also possible to identify potential tendencies of significance. In this case, we interpret an effect as having a tendency of significance if 50% of the density is located on one side of zero. This means that predictor Q2, Q7, and Q10 show some tendencies toward positive significance, whereas Q3 and Q6 show tendencies toward negative significance. Predictor Q1, Q4, Q11 and Q9 are clearly not significant.

4.2 Conditional Effects

A significant effect can be further investigated by plotting the conditional effects for a given estimate. Since we know that Q12 is a significant predictor for Q16, we Q12 in this example. When plotting conditional effects, the default is set to treat predictions as continuous variables, but by setting categorical to TRUE, it is possible to plot conditional effects per category. Below, the first block of code generates the graph in Figure 16, and the second block of code generates the graph in Figure 17.

```
conditional_effects(
    m_full16,
    effects = "P_Q12_1_7L"
)

conditional_effects(
    m_full16,
    effects = "P_Q12_1_7L",
    categorical = TRUE
    )
```

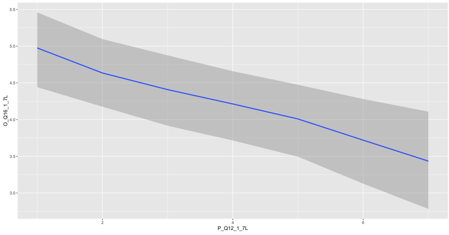

Figure 16: Conditional effect for outcome Q16 (on y-axis) and predictor Q12 (on x-axis). Outcome Q16 has seven levels, ranging from much less productive (1) to much more productive (7), and with same as before in-between (4). Feeling forced to work remotely ranges from strongly disagree (1) to strongly agree (7), with neither disagree nor agree (4) in the middle.

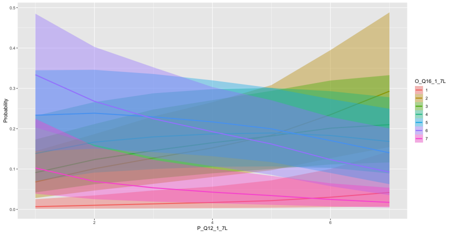

Figure 17: Categorical conditional effects for outcome Q16 (on y-axis) and predictor Q12 (on x-axis). Each color represents a specific level of the ordered categories of Q16, and the shaded areas represents uncertainty.

In Figure 17, each color represents a specific level of the ordered categories of Q16. The shaded areas represents uncertainty. Outcome Q16 has seven levels, ranging from much less productive (1) to much more productive (7), where 1 is red and 7 is pink. The purple slope corresponds to category 6 (more productive) and shows how the probability mass for category 6 is negatively affected by higher values of predictor Q12. Similarly, but in the opposite direction, for category 2 (less productive), the probability mass increases the higher value of Q12. This means that the more forced a practitioner feels to work remotely, the less probability mass is set for feeling more productive and more probability mass is set for feeling less productive. In other words, the more forced a practitioner feels to work remotely, the less productive he/she feels. The same conclusion is also found by examining Figure 16, in which the negative slope shows that not feeling forced to work remotely (low value on the x-axis) is associated with higher productivity, whereas feeling forced to work remotely (high value on the x-axis) is associated with lower productivity.

To plot all of the effects from the model at once, we leave out the argument 'effects' in the command.

```
conditional_effects(m_full16)
```

This way, the command will generate a graph for each effect, and ask you to hit 'Return' to plot the next one, as shown below.

```
Hit <Return> to see next plot:
```

As the model contains eleven predictors, eleven graphs will be generated. Since the graph for Q12 is already shown in Figure 16, we only show the other ten effects below.

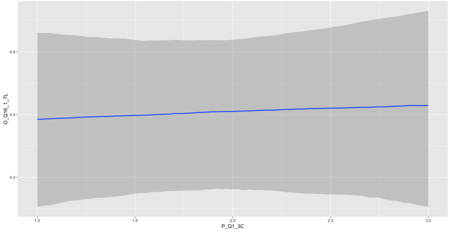

Figure 18: Predictor: Gender

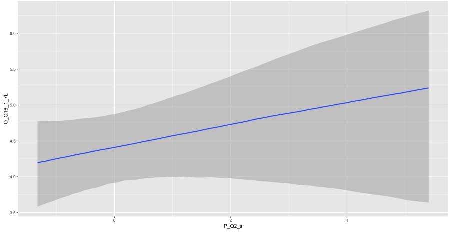

Figure 19: Predictor: Team size

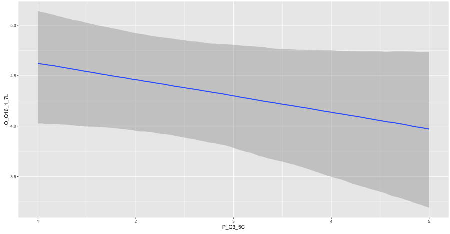

Figure 20: Predictor: Team constellation

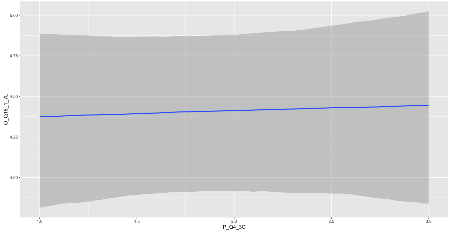

Figure 21: Predictor: Domain

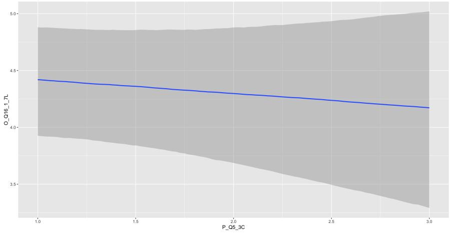

Figure 22: Predictor: Continent

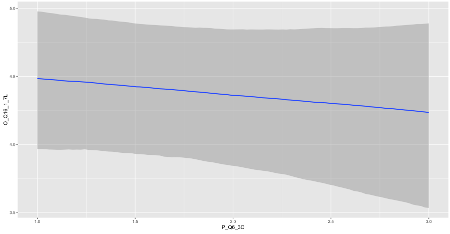

Figure 23: Predictor: Role

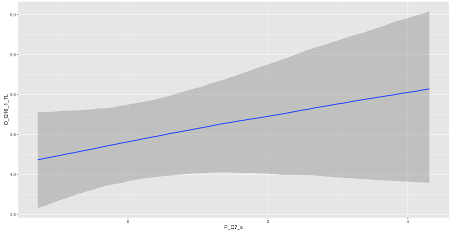

Figure 24: Predictor: Years of experience

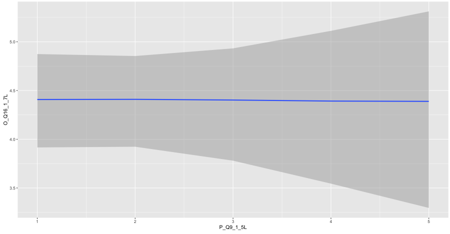

Figure 25: Predictor: Work setting before the pandemic

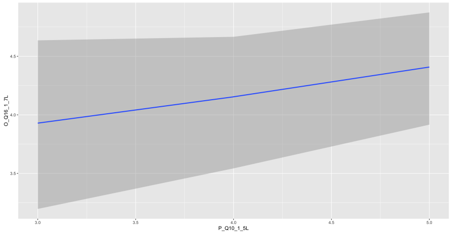

Figure 26: Predictor: Work setting during pandemic

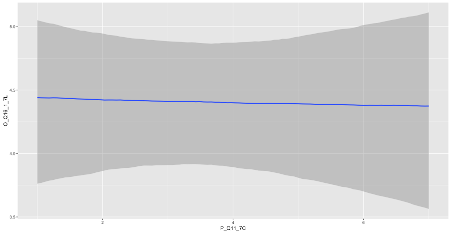

Figure 27: Predictor: Reason to work remotely

References

[1] Paul-Christian Bürkner and Matti Vuorre. “Ordinal Regression Models in Psychology: A Tutorial”. In: *Advances in Methods and Practices in Psychological Science* 2.1 (2019), pp. 77–101. ISSN: 2515-2459. DOI: 10.1177/2515245918823199.

[2] Fumiko Samejima. “Graded Response Model”. In: *Handbook of Modern Item Response Theory*. Ed. by Wim J van der Linden and Ronald K Hambleton. New York, NY: Springer New York, 1997, pp. 85–100. ISBN: 978-1-4757-2691-6. DOI: 10.1007/978-1-4757-2691-6_5. URL: https://doi.org/10.1007/978-1-4757-2691-6_5.

[3] Carlo Alberto Furia, Robert Feldt, and Richard Torkar. “Bayesian Data Analysis in Empirical Software Engineering Research”. In: *IEEE Transactions on Software Engineering* (2019). ISSN: 19393520. DOI: 10.1109/TSE.2019.2935974. eprint: 1811.05422.