Effects on projectivity ratings by Embedding Operator and Trigger — Data Analysis

Lisa Hofmann

November 16, 2022

Contents

1 Introducing the dataset

```
> str(data_all)

'data.frame': 57160 obs. of  9 variables:
 $ workerid     : Factor w/ 2682 levels "1","3","4","5",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ content      : Factor w/ 20 levels "charley","danny",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ short_trigger: chr  "acknowledge" "hear" "reveal" "discover" ...
 $ ai_block     : chr  "block1" "block1" "block1" "block1" ...
 $ ai           : num  0.98 0.99 0.99 0.99 0.98 0.98 1 0.99 0.99 0.99 ...
 $ projective   : num  0.3 0.98 0.01 0.99 0.98 0.99 0.01 0.01 0.27 0.01 ...
 $ verb         : Factor w/ 20 levels "acknowledge",..: 1 11 16 9 18 13 17 7 14 8 ...
 $ op           : Factor w/ 4 levels "q","n","m","c": 1 1 1 1 1 1 1 1 1 1 ...
 $ exp_block    : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...

> length(levels(data_all$workerid))

[1] 2682
```

The dataset consists of 57160 observations from 2682 participants (recruited on the online platforms Prolific and Amazon Mechanical Turk), across 12 experiments.

We are interested in how highly participants rate speaker commitment to the truth of an embedded complement clause, coded as `projective` on a real-numbered sliding scale between $0 - 1$.

The complement clause was embedded under an attitude verb, which in turn was embedded under an entailment-cancelling operator. Our fixed effects factors manipulate the following:

1. The choice of attitude verb (coded as `verb`)
2. The entailment-cancelling operator (coded as `op`)

The levels for our fixed effects factors are the following:

```
> levels(data_all$verb)

 [1] "acknowledge" "admit"       "announce"    "be_annoyed"  "be_right"
 [6] "confess"     "confirm"     "demonstrate" "discover"    "establish"
[11] "hear"        "inform"      "know"        "pretend"     "prove"
[16] "reveal"      "say"         "see"         "suggest"     "think"

> length(levels(data_all$verb))

[1] 20

> levels(data_all$op)

[1] "q" "n" "m" "c"

> length(levels(data_all$op))

[1] 4
```

We are interested in the effect on `projective` of `verb` and `op`, as well as their interaction, corresponding to a 20×4 factorial design, yielding

```
> length(levels(data_all$verb))*length(levels(data_all$op))

[1] 80
```

conditions.

We have 20 items, corresponding to the content of the complement clause.

```
> levels(data_all$content)

 [1] "charley"  "danny"    "emily"    "emma"     "frank"    "grace"
 [7] "isabella" "jackson"  "jayden"   "jon"      "josh"     "josie"
[13] "julian"   "mary"     "mia"      "olivia"   "owen"     "sophia"
[19] "tony"     "zoe"

> length(levels(data_all$content))

[1] 20
```

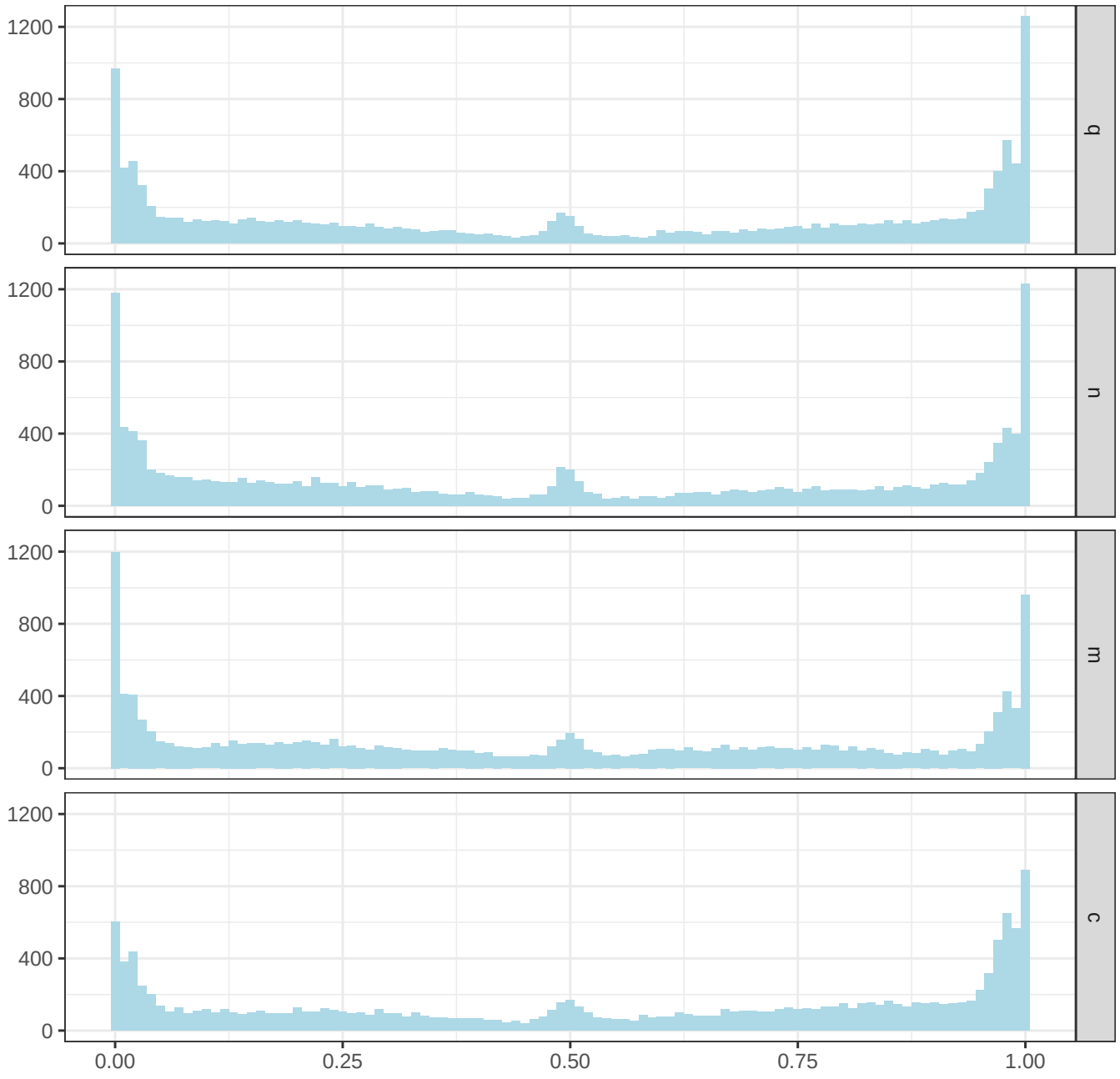
We have roughly 36 observations by item and condition. This is an approximate number, because the `op` manipulation is a between-studies manipulation, and the number of participants differs by experiment:

```
> # n observations
> length(data_all[,1])

[1] 57160

> # observations by item
> length(data_all[,1])/length(levels(data_all$content))

[1] 2858

> table(data_all$content)


 charley    danny    emily     emma    frank    grace isabella  jackson
    2858     2858     2858     2858     2858     2858     2858     2858
  jayden      jon     josh    josie   julian     mary      mia   olivia
    2858     2858     2858     2858     2858     2858     2858     2858
    owen   sophia     tony      zoe
    2858     2858     2858     2858
```

```
> # observations by verb
> length(data_all[,1])/length(levels(data_all$verb))

[1] 2858

> table(data_all$verb)


acknowledge       admit    announce  be_annoyed    be_right     confess
       2858        2858        2858        2858        2858        2858
    confirm demonstrate    discover   establish        hear      inform
       2858        2858        2858        2858        2858        2858
       know     pretend       prove      reveal         say         see
       2858        2858        2858        2858        2858        2858
    suggest       think
       2858        2858

> # observations by operator
> length(data_all[,1])/length(levels(data_all$op))

[1] 14290

> table(data_all$op)


    q     n     m     c
13740 14340 14680 14400

> # observations by item and condition
> length(data_all[,1])/length(levels(data_all$content))/
+   (length(levels(data_all$verb))*length(levels(data_all$op)))

[1] 35.725
```

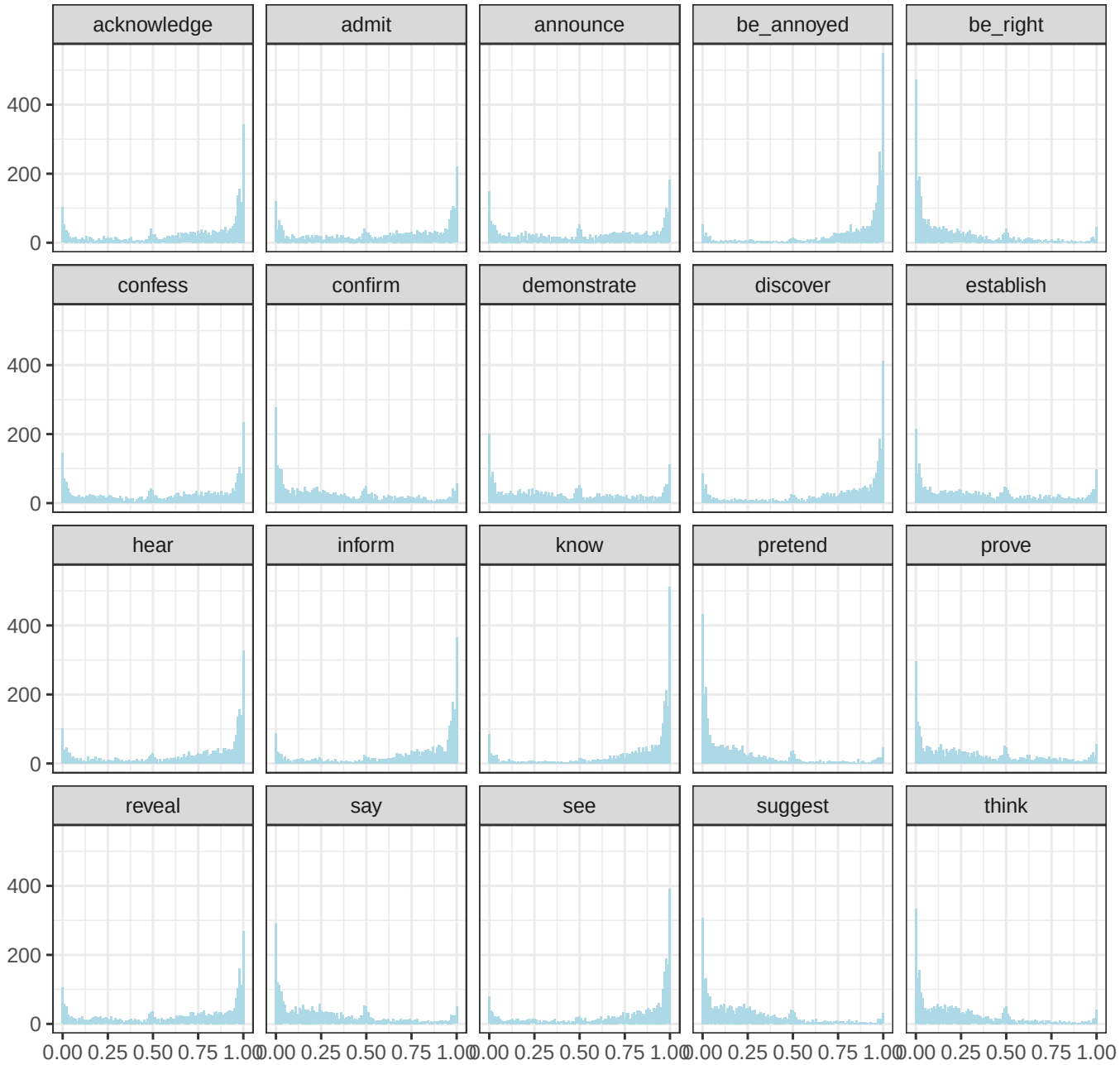
2 Data Overview and Statistical Summaries

2.1 Distribution of projectivity ratings by operator:



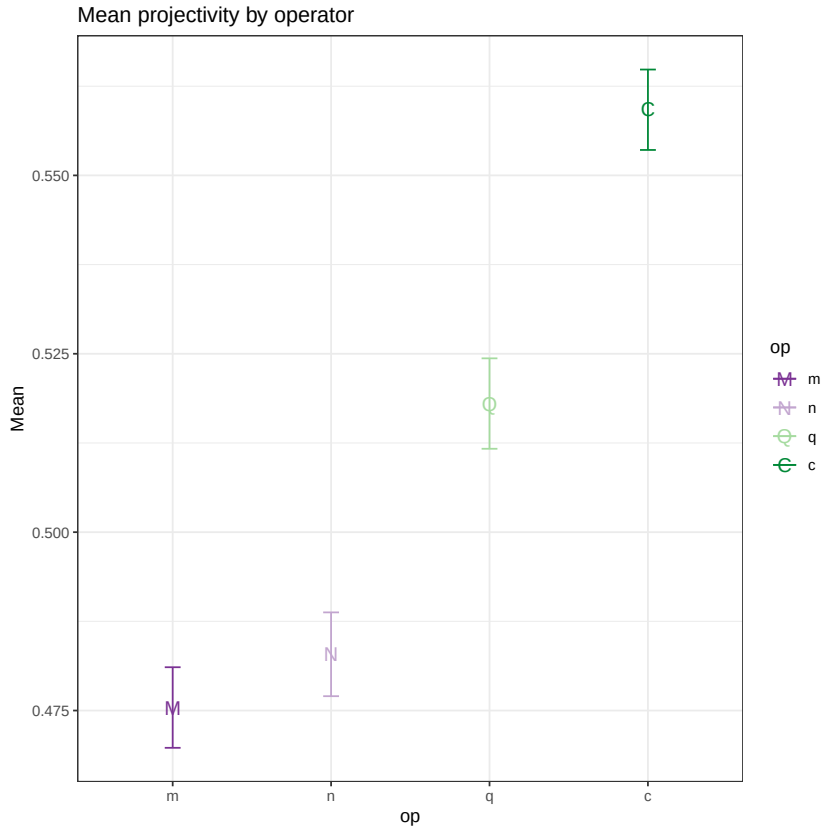
- These definitely do not look normal
- Maybe a beta-regression would be useful?
- But even that would be relying on some simplifying assumptions, since we might be ignoring the little bump in the middle

2.2 Distribution of projectivity ratings by verb:



- Some of these also show a higher mass around the middle of the scale
- but it looks the beta-distribution could be a useful approximation

2.3 Means and confidence intervals for projectivity rating by operator

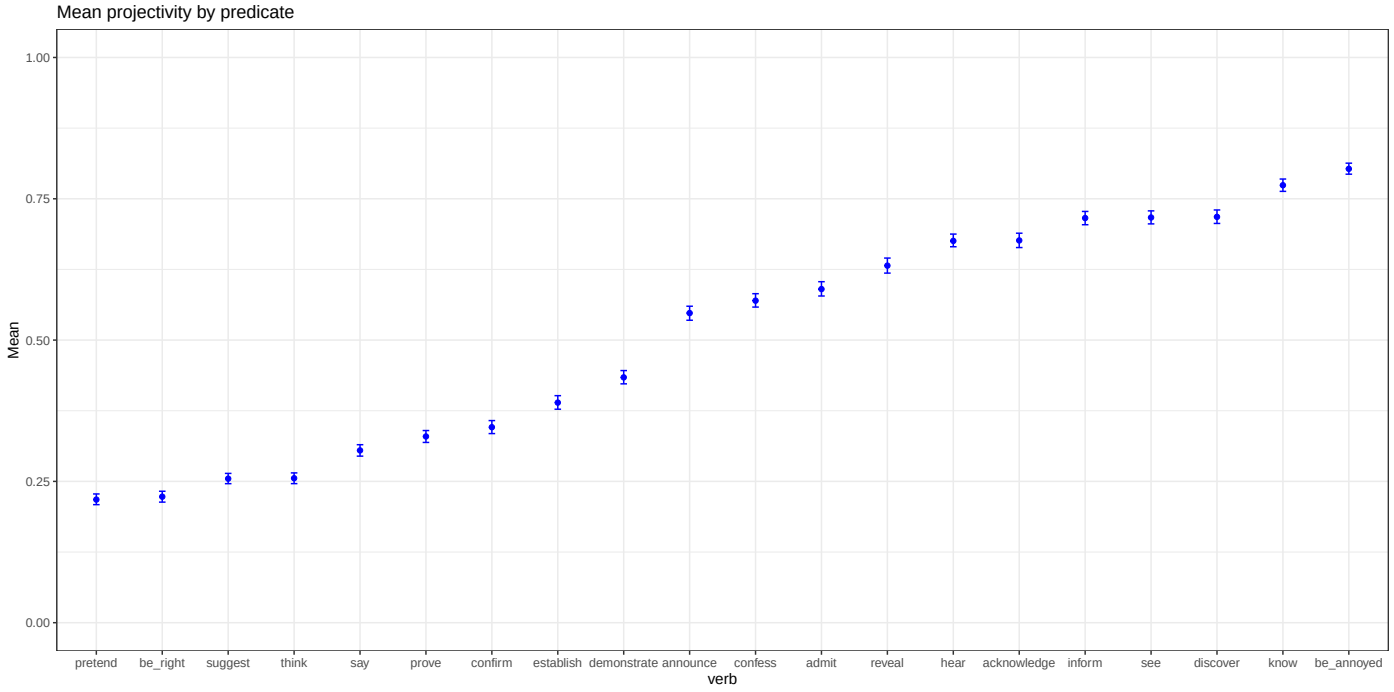


The following generalizations emerge:

- Conditionals have the highest projectivity ratings
- Projectivity ratings for questions are higher than those for modals and negation, but lower than those for conditionals
- Modals and negation have the lowest projectivity ratings
- The ratings for negation look a little higher than for modals, but error bars overlap

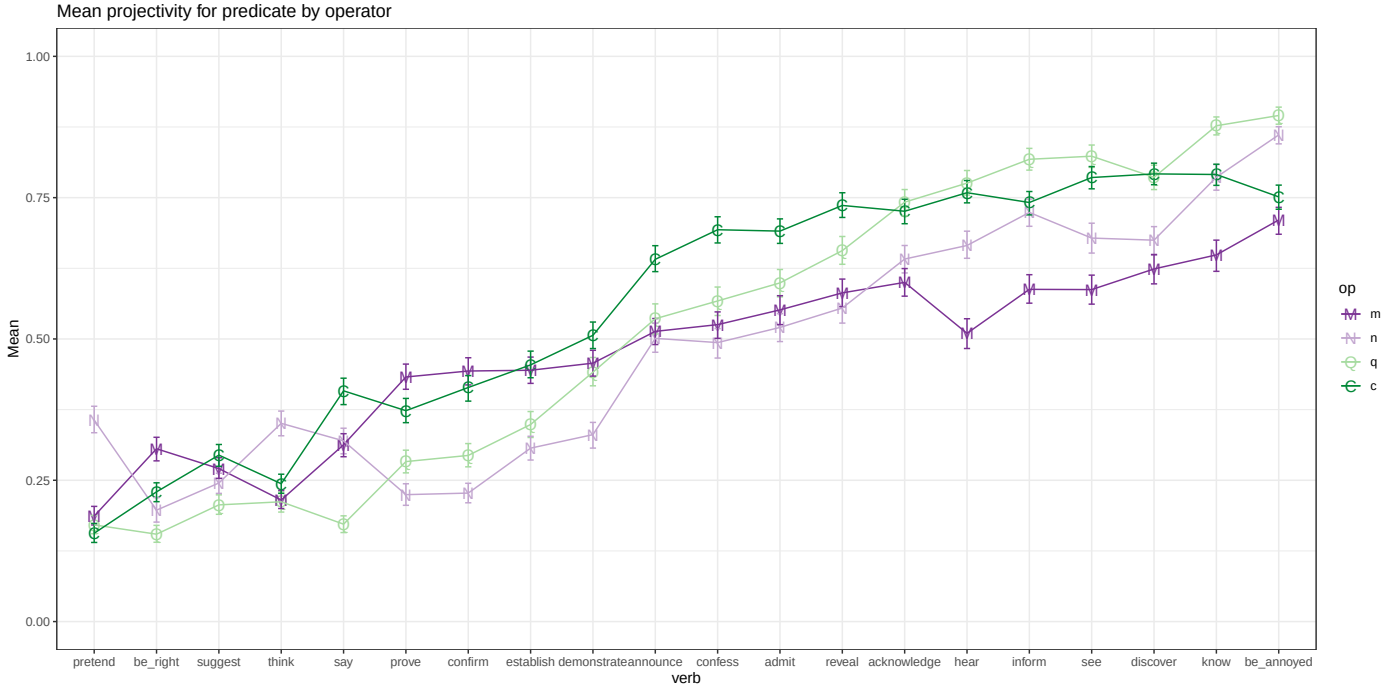
Although these differences appear to be significant, they are quite small.

2.4 Means and confidence intervals for projectivity rating by verb:



- We see gradual differences in projectivity between verbs

2.5 Means and confidence intervals for projectivity rating by verb and operator:



- We see interactions between verb and operator
- However, we do not see any group of verbs that could be characterized as 'semi-factive' in the sense of Karttunen. Specifically, *discover* does not follow the predicted pattern: It is not more projective under negation, but most projective in conditionals and questions.
- Two verbs show highest projectivity under negation: the anti-veridical *pretend*, and the non-veridical *think*. These are verb with relatively low overall projectivity.

- What else...?

3 Analysis

We are interested in how the response (projectivity ratings) depends on the verb and embedding operator.

```
> formula <- response ~ op * verb + (1 | workerid) + (1 | item)
> # formula <- response ~ op * verb + (1 + op + verb | workerid) + (1 + op + verb | item)
```

Trying a beta-regression. Our response variable has observations from a sliding scale between $(0, 1)$. For beta regression, we rescale it to $[0, 1]$, using method used in Degen & Tonhauser (2022), from Smithson & Verkuilen (2006).

```
> data_all$response = (data_all$projective*(nrow(data_all)-1) + .5)/nrow(data_all)
```

```
> library(brms)

Loading required package: Rcpp
Loading 'brms' package (version 2.16.3). Useful instructions
can be found by typing help('brms'). A more detailed introduction
to the package is available through vignette('brms_overview').

Attaching package: 'brms'
The following object is masked from 'package:stats':
      ar

> model1 <- brm(formula = response ~ op * verb + (1 | workerid) + (1 | item), family = )

Error: Data must be specified using the 'data' argument.
```