

Supplementary Document: Cognitive and cultural models in psychological science

Theiss Bendixen^a & Benjamin Grant Purzycki^a

^a *Department of the Study of Religion, Aarhus University, Denmark*

Overview

In Supplementary Section A, we implement an alternative likelihood to the zero-one inflated beta, namely the *ordered beta* distribution. For technical details and background, see Kubinec (2022) and for practical implementation, see the R package `ordbetareg` (<https://github.com/saudiwin/ordbetareg>) and source code for the present supplementary document.

In Supplementary Section B, we illustrate and implement alternative likelihood models that do not require transforming free-list data to salience scores, as discussed briefly in the main manuscript. We integrate key code chunks in the text but refer to the source code for full reproducible code. For simplicity, we restrict the examples to the Model 1 data and exclude field-site specific ‘random effects’ in all models.

In Supplementary Section C, we first present plots for Models 3-8 (Figures 10 and 11). Then, we present results for the reduced models, where only non-zero free-list responses are retained (Figures 12–14). Finally, we present results from leave-one-group-out cross-validation (Table 3), an alternative model comparison to the approximate leave-one-out cross-validation as presented in the main manuscript.

In Supplementary Section D, we list R packages, their dependencies, and version number used for this project.

Table 1

Focal parameter estimates for all ordered beta models and model comparison metrics.

	Intercept	β^{Scale}	Akaike Weights	δ ELPD [SE]
m1_ordbeta	0.16 [-0.33, 0.64]	0.07 [-0.49, 0.67]	0.10	-1.68 [0.74]
m1_ordbeta_null	0.21 [-0.20, 0.62]	—	0.90	*
m2_ordbeta	-0.85 [-1.77, 0.18]	1.47 [0.48, 2.46]	0.76	*
m2_ordbeta_null	-0.23 [-1.32, 0.81]	—	0.24	-3.68 [4.03]
m3_ordbeta	-0.16 [-0.70, 0.34]	0.08 [-0.05, 0.19]	0.02	*
m3_ordbeta_null	0.08 [-0.34, 0.46]	—	0.98	-0.47 [1.78]
m4_ordbeta	-0.03 [-0.70, 0.59]	0.04 [-0.12, 0.19]	0.01	-0.69 [0.98]
m4_ordbeta_null	0.10 [-0.32, 0.48]	—	0.99	*
m5_ordbeta	-0.30 [-0.83, 0.20]	0.04 [-0.11, 0.20]	1.00	*
m5_ordbeta_null	-0.16 [-0.72, 0.37]	—	0.00	-0.20 [1.62]
m6_ordbeta	-0.46 [-1.46, 0.40]	0.05 [-0.16, 0.25]	0.10	*
m6_ordbeta_null	-0.21 [-1.14, 0.49]	—	0.90	-0.08 [1.13]
m7_ordbeta	-0.66 [-1.86, 0.42]	0.07 [-0.16, 0.28]	0.16	-0.30 [0.74]
m7_ordbeta_null	-0.31 [-1.37, 0.54]	—	0.84	*
m8_ordbeta	-1.07 [-2.12, -0.18]	0.03 [-0.18, 0.23]	0.01	-1.51 [0.84]
m8_ordbeta_null	-0.90 [-1.78, -0.20]	—	0.99	*

Note. Point estimates of the β coefficients are posterior means with 95% credible intervals in brackets. δ ELPD = differences in expected log posterior density between model pairs; SE = standard error of ELPD differences. The best performing model of the eight pairs is denoted by an asterisk.

Supplementary Section A: The ordered beta distribution

Here's an excerpt from Kubinec (2022) outlining the gist of the ordered beta:

[the ordered beta] employ[s] ordered cut points, similar in spirit to an ordered logit model, to estimate the joint probability of 0s (the lower bound), continuous proportions, and 1s (the upper bound) in bounded continuous data. As only one predictive model is used for all of the outcomes, the effect of covariates is identified across degenerate and continuous observations without resulting in overfitting. The use of cut points permits the model to fit distributions with mostly degenerate observations or no degenerate observations at all, which makes it a general solution to this problem. [...] Each cut point represents the point at which the bounds of the observed scale become more likely than the continuous values. (p. 2, 4)

In the context of our case study, the ZOIB and the ordered beta yield similar inferences and posterior predictions (Figures 1 – 3 and Table 1).

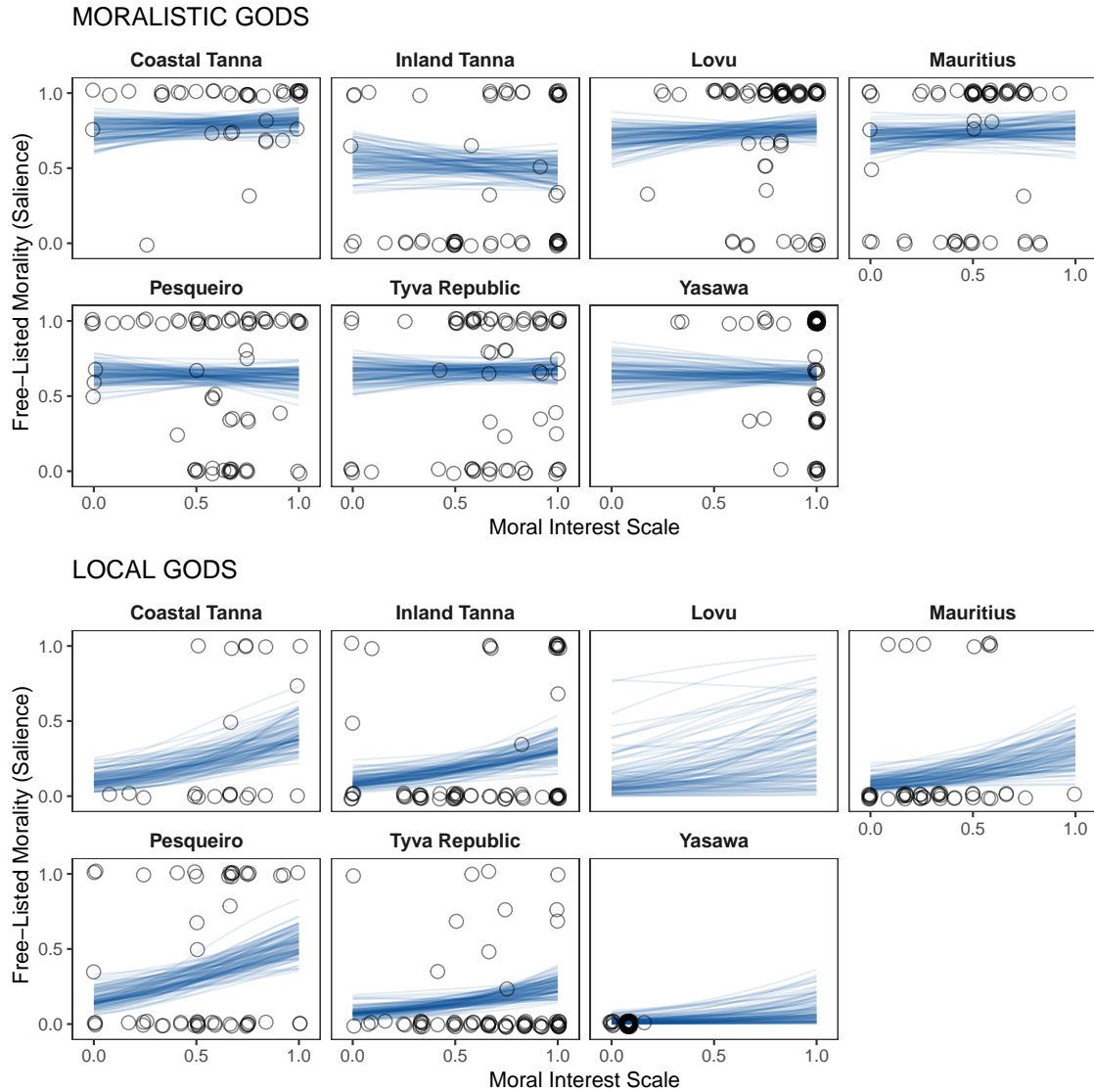


Figure 1. Ordered beta models. Posterior predictions for each field site and each deity from the ordered beta Model 1 and 2 on the relationship between the moral interest scale (x -axis) and saliency of the free-listed general Morality code (y -axis). Lines are draws of expected values from the posterior predictive distribution. Raw data points are slightly jittered.

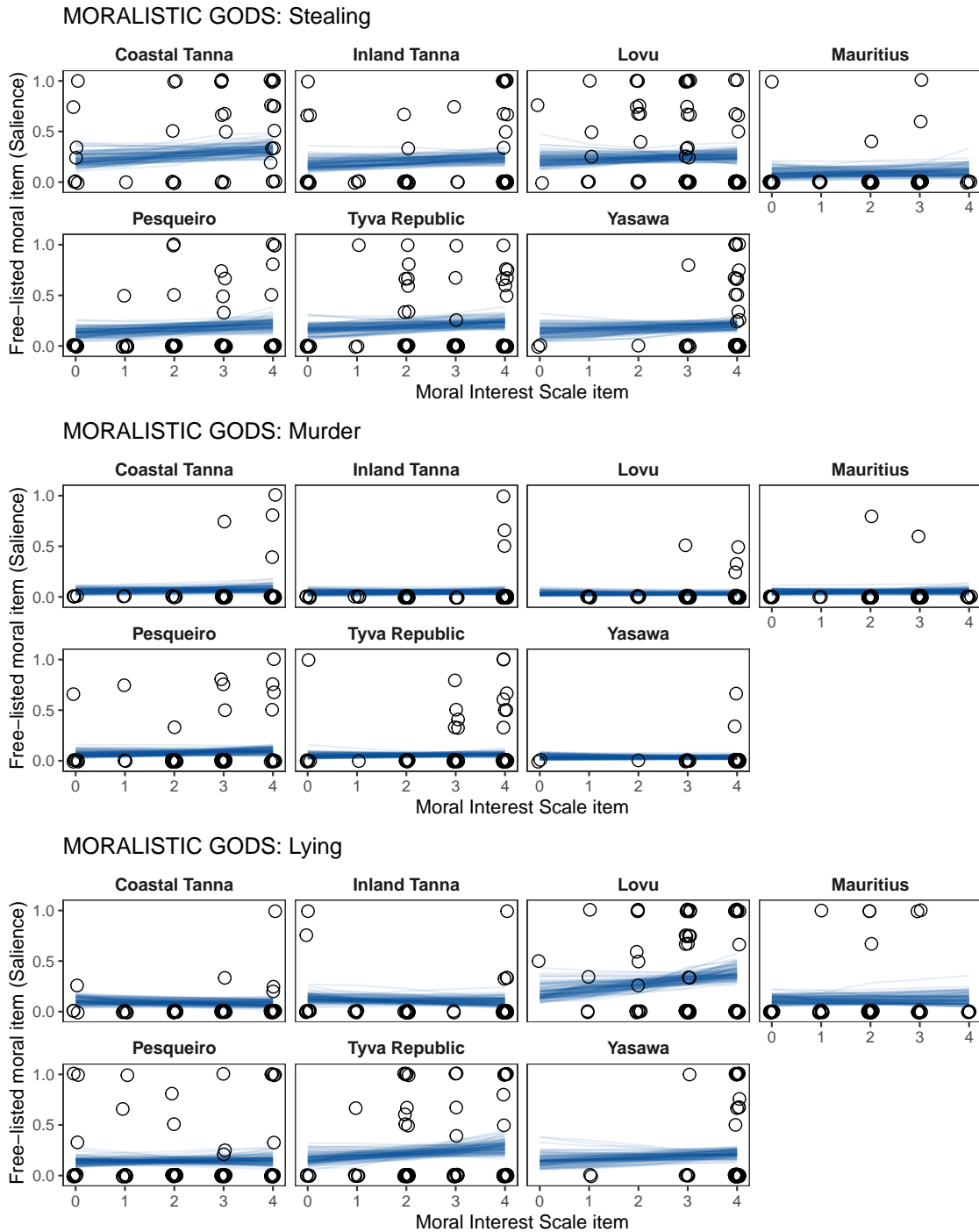


Figure 2. Ordered beta models. Posterior predictions for each field site from the ordered beta Model 3 through 5 on the relationship between the moral interest scale items (x -axis) and salience of the corresponding free-listed code (y -axis). Lines are draws of expected values from the posterior predictive distributions. Raw data points are slightly jittered.

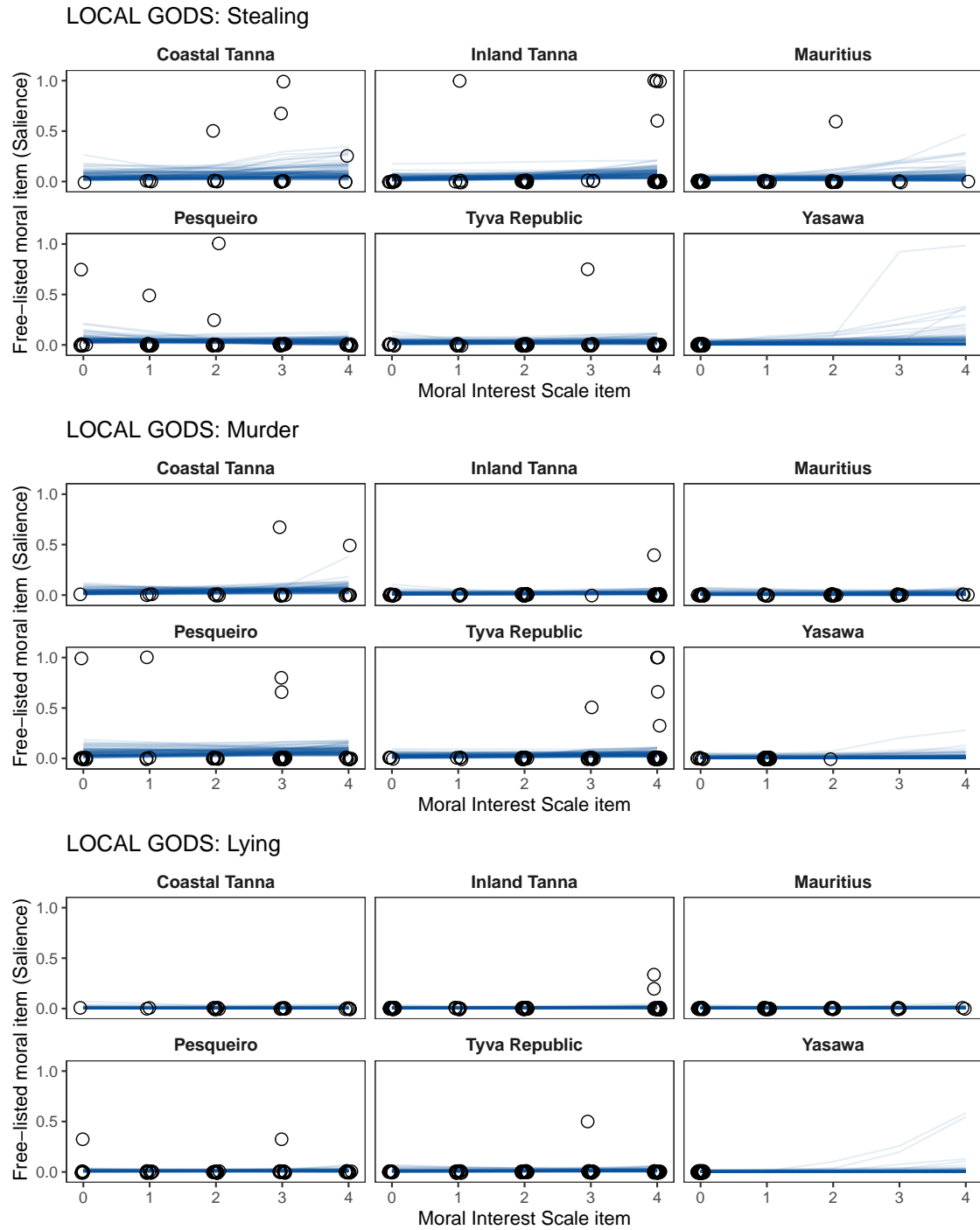


Figure 3. Ordered beta models. Posterior predictions for each field site from ordered beta Model 6 through 8 on the relationship between the moral interest scale items (x -axis) and saliency of the corresponding free-listed code (y -axis). Lines are draws of expected values from the posterior predictive distributions. Raw data points are slightly jittered.

37 **Supplementary Section B: Alternative likelihood models for free-list data**

38 **Presence/absence**

39 An analyst could be interested in simply modeling the presence and absence of some
40 target item using a logistic (Bernoulli) regression. With `tableType = 'PRESENCE'` in
41 `AnthroTools::FreeListTable()` we get a data frame with the presence (= 1) and absence
42 (= 0) of all items listed across participants in our free-list data. We then extract the
43 general ‘Morality’ code, combine this data with the scale responses, store it in the object
44 `bgd_gen_data_bern`, and then fit a logistic regression with `brms`. A higher coefficient for
45 `scale` is interpreted as a higher probability of listing ‘Morality’ as predicted by the moral
46 interest scale. For practical applications, see e.g. Purzycki and Bendixen (2020) and Purzycki,
47 Stagnaro, and Sasaki (2020).

```
m1_bern <- brm(  
  y ~ 1 + scale,  
  data = bgd_gen_data_bern,  
  family = bernoulli(),  
  cores = 4, chains = 4,  
  iter = 1000, control = list(adapt_delta = 0.99),  
  seed = 2021)
```

48 Number of ‘successes’

49 With a binomial model, we can instead model the number of mentions of a target item
 50 given the total number of items listed per participant. With `tableType = 'FREQUENCY'` in
 51 `AnthroTools::FreeListTable()` we get a data frame with the number of times each item
 52 was listed across participants in our free-list data. Using the `rowSums()` command on this
 53 data frame, we can get the total number of items listed per participant; this number will
 54 serve as the number of ‘trials’ `n` in our binomial model. We extract the general ‘Morality’
 55 code (i.e., the number of times each participant listed ‘Morality’), which is the ‘successes’ `y`
 56 in the binomial model, combine the successes and trials with the scale response, store the
 57 resulting data in `bgd_gen_data_bin`, and fit a binomial model to this data.

```
m1_bin <- brm(bf(
  y | trials(n) ~ 1 + scale),
  data = bgd_gen_data_bin,
  family = binomial(),
  cores = 4, chains = 4,
  iter = 1000, control = list(adapt_delta = 0.99),
  seed = 2021)
```

58 We can check the fit of the model with a convenient posterior predictive check of
 59 the marginal distribution of outcome values plotted against model predictions across total
 60 number of items listed (Figure 4). The fit is decent but not perfect; the model seems to
 61 consistently over-shoot the number of ones and twos, for instance. In an applied case, we
 62 would likely work on achieving a closer fit or, failing to do so, abandon this particular
 63 likelihood model.

64 We can then plot the expected number of times that ‘Morality’ was listed as a function
 65 of the moral interest scale and the total number of items listed per participant (Figure 5).
 66 For instance, the model predicts that a participant that lists only one item (panel 1) lists
 67 ‘Morality’ as that one item regardless of their score on the moral interest scale. Generally,
 68 the moral interest scale does not appear to predict an increased probability of listing more
 69 ‘Morality’ items across various number of trials. A possible exception is for those who lists
 70 five items (panel 5), where the model’s best guesses trend upwards from two ‘Morality’ items
 71 at the lowest end of the scale to three ‘Morality’ items at the higher end of the scale.

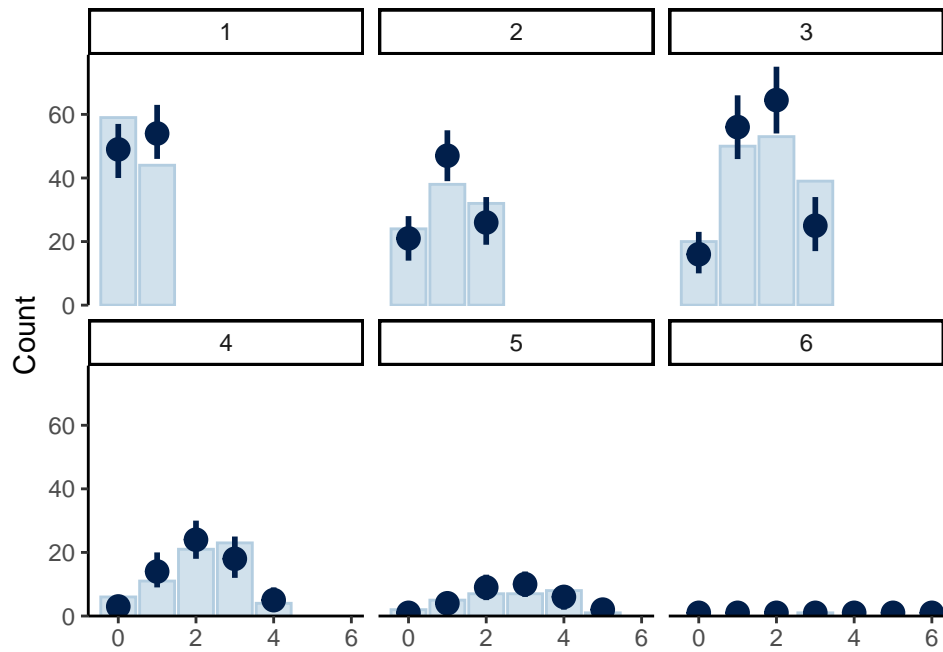


Figure 4. Posterior predictive check for the binomial model. Light blue bars are frequency of each observed outcome while points and intervals are posterior predictions. Panel numbers refer to number of ‘trials’ n , i.e. total number of items listed per participant.

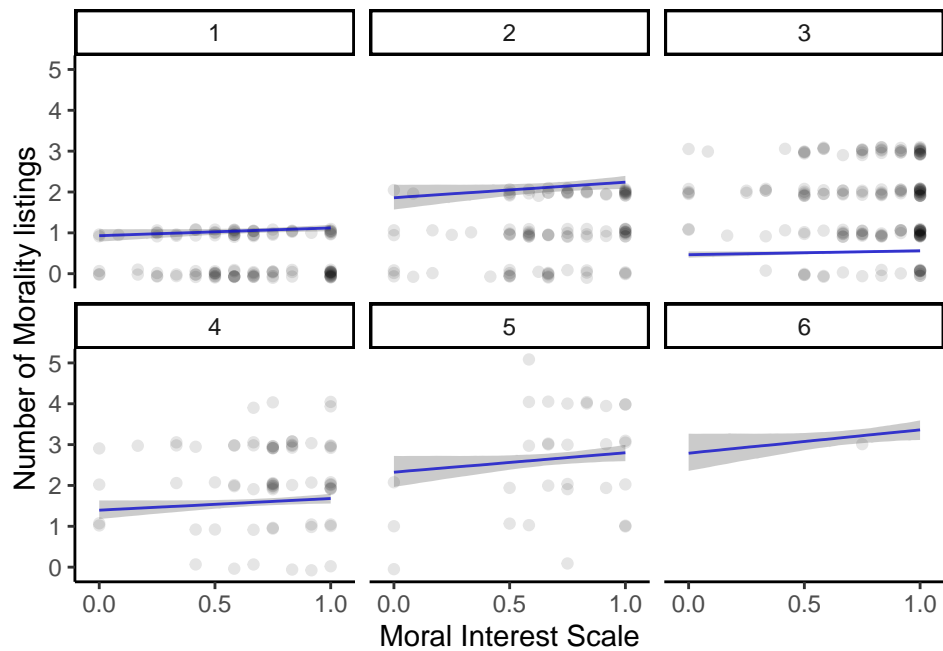


Figure 5. Binomial model. Lines are posterior medians of expected values with 95% interval bands. Raw data points are slightly jittered vertically. Panel numbers refer to number of ‘trials’ n , i.e. total number of items listed per participant.

72 Average list position

73 We can also model the expected list position and how this expectation change as a func-
 74 tion of some predictor(s), using a zero-inflated negative binomial. Using `tableType =`
 75 `'HIGHEST_RANK'` in `AnthroTools::FreeListTable()`, we get the highest rank (i.e., list po-
 76 sition) for all items listed across participants. As above, we extract the general ‘Morality’ code,
 77 combine this data with the scale responses, and store it in the object `bgd_gen_data_rank`.
 78 Now, `y` is the order at which ‘Morality’ was listed, i.e. 1 means first, 2 means second, etc.,
 79 while 0 means that ‘Morality’ was not listed.

```
str(bgd_gen_data_rank)
```

```
80 ## 'data.frame':   455 obs. of  4 variables:
81 ## $ id      : chr  "CERC1" "CERC10" "CERC11" "CERC113" ...
82 ## $ y       : num   1 1 1 0 1 0 2 0 1 0 ...
83 ## $ culture: chr   "Coastal Tanna" "Coastal Tanna" "Coastal Tanna" "Inland Tanna" ...
84 ## $ scale  : num   0.167 0.333 0.417 0.25 0.333 ...
```

85 We then fit a zero-inflated negative binomial model to this data, using `scale` as a
 86 predictor for both the zero-inflation $z_i \sim \dots$ and the negative binomial $y \sim \dots$. The
 87 implementation is straightforward as this likelihood model is native to `brms`, specified as
 88 `family = zero_inflated_negbinomial()`. Note that a larger regression coefficient in the
 89 negative binomial part means that the target item moves *down* on participants free-lists
 90 (i.e., becomes less salient) as a function of the predictor.

```
m1_zinb <- brm(bf(
  y ~ 1 + scale,
  zi ~ 1 + scale),
  data = bgd_gen_data_rank,
  family = zero_inflated_negbinomial(),
  cores = 4, chains = 4,
  iter = 1000, control = list(adapt_delta = 0.99),
  seed = 2021)
```

91 We can again assess the fit of model with a quick posterior predictive check (Figure
 92 6). The fit looks far from perfect, as the model fails to capture key aspects of the observed
 93 marginal outcome. In an applied case, we would have to look further into this mis-fit, perhaps
 94 resulting in the inference that the zero-inflated negative binomial is, after all, not a good
 95 model for these particular data. One possible reason for this is that the negative binomial is
 96 an uncapped count model, and so it is not prevented from returning predictions that are
 97 substantially outside the range of observed values, which in this case means predicting a
 98 target item appearing in much later list positions than observed.

99 For illustrative purposes, however, let’s assess the expected list position of ‘Morality’
 100 as predicted by the moral interest scale (Figure 7). There is not a lot of change, although if
 101 anything it seems that the expected position at which ‘Morality’ is listed trends upwards
 102 (e.g., becomes less salient) with higher scores on the moral interest scale.

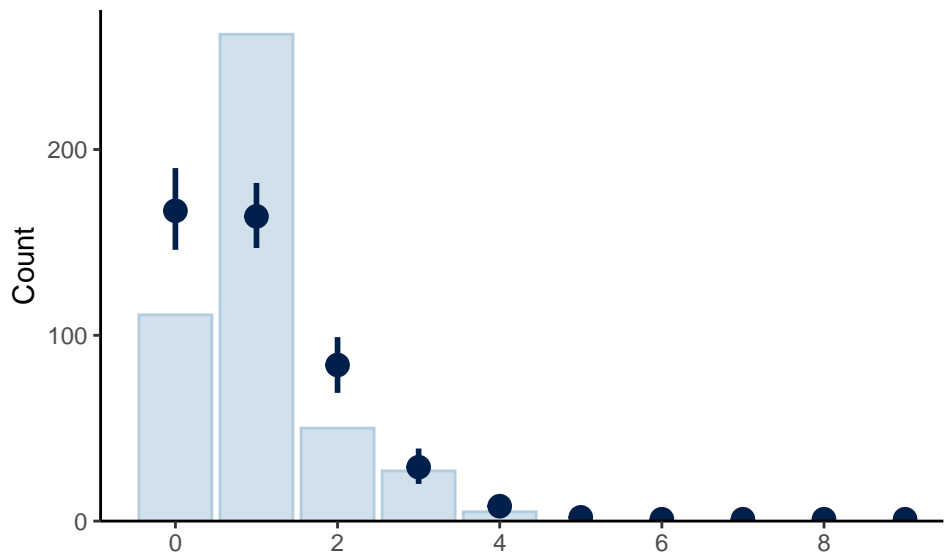


Figure 6. Posterior predictive check for zero-inflated negative binomial. Light blue bars are frequency of each observed outcome while points and intervals are posterior predictions.

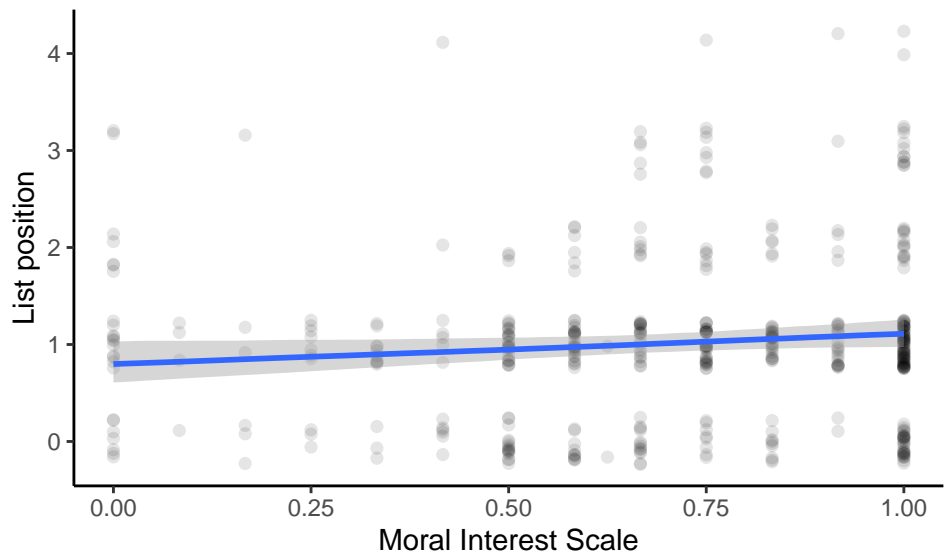


Figure 7. Expected list position according to the zero-inflated negative binomial model. Raw data points are slightly jittered vertically.

Cumulative probability of list positions

Finally, we can also model the cumulative probabilities of a target item appearing in each of the positions in which the item appears across the sample. This could be achieved with a zero-inflated ordinal regression. This likelihood model is not native to **brms**, and hence requires some custom code (stored in **stanvars** in the model code chunk). See the source code, which adopts and modifies code after <https://github.com/octmedina/zi-ordinal>. We also set some weakly regularizing priors for computational efficiency.

The $y \sim 1 + \text{scale}$ part models the zero-inflation (i.e., the probability of not listing the target item), while $\text{eta} \sim 0 + \text{scale}$ is the ordinal regression of the non-zero outcomes. Note that as with the zero-inflated negative binomial, a larger regression coefficient in the ordinal part means that the target item moves down on participants' free-lists (i.e., becomes less salient) as a function of the predictor. We again use the data frame **bgd_gen_data_rank**

```
ziord_priors <- set_prior("normal(0,1)", class = "b", dpar = "eta") +
  set_prior("normal(0,1)", class = "b") +
  set_prior("normal(0,2)", class = "Intercept")

m1_zi_ord <- brm(bf(
  y ~ 1 + scale,
  eta ~ 0 + scale),
  data = bgd_gen_data_rank,
  prior = ziord_priors,
  family = zi_ordinal, stanvars = stanvars,
  chains = 4, cores = 4, control = list(adapt_delta = 0.99),
  seed = 2021, iter = 10000)
```

With this data and model, we experienced convergence issues, which would have to be fixed in an actual applied case. We get reasonable convergence diagnostics with a relatively high number of iterations (10,000); however, increasing computational costs is not the ideal strategy for alleviating problematic chain behavior. In the source code, we illustrate an implementation in raw **Stan** that samples much more efficiently.

We can run a posterior predictive check to assess the fit of the model against observed data (Figure 8). The fit looks very reasonable, in that the observed data are well inside the intervals of the posterior predictions. Lastly, we can plot the cumulative probabilities of each list position as predicted by the moral interest scale. That is, how does the probability of listing 'Morality' change as a function of higher or lower scores on the three-item scale? This is Figure 9; lines are medians of posterior predictive draws and bands are 80% intervals to make the trends stand out more clearly. Note that the probability of not listing the target item (0; red line/band) goes down as participants score higher on the scale, whereas the probability of listing 'Morality' as the first item (1; yellow line/band) seems stable across scale responses. Conversely, the probability of 'Morality' appearing at list position two (green) and three (turquoise) seem to increase slightly with higher scores on the scale. The probability of listing 'Morality' as the fourth (blue) and fifth (purple) item is practically zero across the span of the scale.

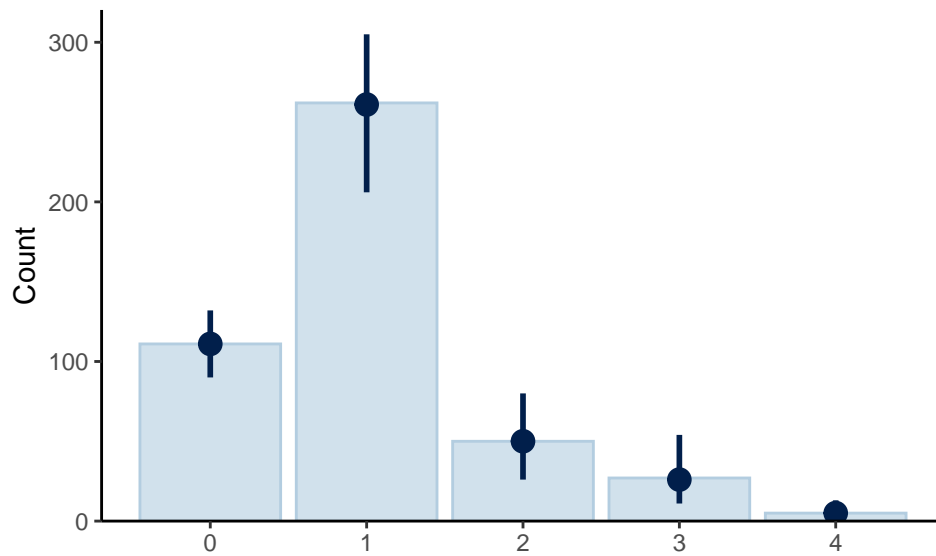


Figure 8. Posterior predictive check for zero-inflated ordinal regression. Light blue bars are frequency of each observed outcome while points and intervals are posterior predictions.

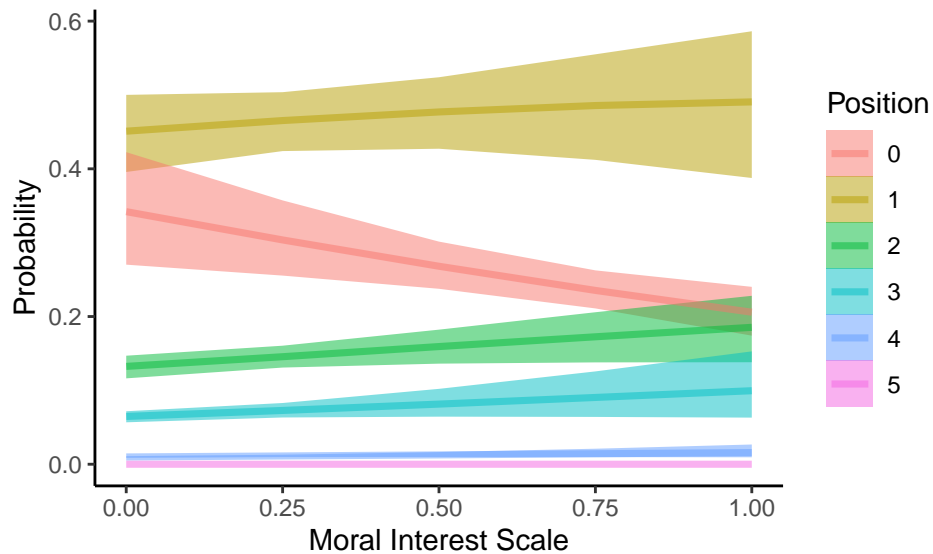


Figure 9. Cumulative probabilities of each list position. 0 = item not listed. Lines are posterior medians and bands are 80% credible intervals.

Table 2
Example data structure for random effects modeling.

ID	Saliency	Scale	Specific free-list item	General free-list code
ID1	0.25	4	steal	Morality
ID1	0.75	2	murder	Morality
ID1	0.75	1	lie	Morality
ID1	0.50	3	hard-working	Virtue
ID1	1.00	4	kind	Virtue
ID1	0.00	1	bad conscience	Virtue
ID2	0.75	2	steal	Morality
ID2	0.50	3	murder	Morality
ID2	0.75	4	lie	Morality
ID2	0.25	4	hard-working	Virtue
ID2	0.50	0	kind	Virtue
ID2	0.75	1	bad conscience	Virtue

Note. Hypothetical data.

Random effects modeling

Above and beyond the choice of likelihood and inspired by item response theory, it's possible to jointly model separate free-list items that nonetheless share some conceptual characteristic using random item effects. For instance, in our case study, we conceptualized “stealing”, “murder”, and “lying” as collectively constituting an overarching category, namely “Morality”. Imagine we were also interested in jointly modeling the saliency of another set of free-list items, which we might conceptualize as “Virtue” (for context, see Bendixen et al., forthcoming). Examples could include responses such as “hard-working”, “kind”, and “bad conscience”.

With the data structured as illustrated in Table 2, we can then treat each specific item as nested within the general code category. This would allow us to estimate the average association between the outcome (here, saliency scores of various free-list items) and a predictor (i.e., some multiple choice scale) within each general code while simultaneously accounting for variability in the constituting specific items (see e.g., Donnellan, Usami, & Murayama, 2022).

In `brms`, the syntax could look like the following, where `saliency` is the outcome, `scale` is our focal predictor, and the specification `(1 + scale | gen_flitem/spec_flitem)` encodes the nested structure of the free-list specific items and general codes. Lastly, we'd likely want to account for the fact that we now have repeated observations for each individual, which we achieve by including a random intercept by participant `(1 | id)`.

```
brm(y ~ 1 + scale + (1 + scale | gen_flitem/spec_flitem) + (1 | id),
    data = dat_random,
    family = zero_one_inflated_beta())
```

Note that we are only showing the hypothetical data and model structure for illustration here. We are not fitting an actual model to our case study data for a few reasons: First, our scale items (i.e., how important it is for a given deity to punish stealing, murder, lying,

155 respectively) correspond directly to a target free-list item (whether a given deity is angered
156 by stealing, murder, and lying, respectively), and so, it's not meaningful to regress the
157 salience of all three free-list items on all three scale items. Second, for current theoretical
158 purposes at least we have no reason suspect a statistical relationship between the scale items
159 and the "Virtue" free-list items and code.

Supplementary Section C: Supplementary plots and tables

Table 3
Results from leave-one-group-out cross-validation (LOGO-CV).

	Akaike Weights	δ ELPD [SE]
m1	0.24	-1.13 [2.11]
m1_null	0.76	*
m2	1.00	*
m2_null	0.00	-8.76 [3.8]
m3	0.00	-2.63 [1.26]
m3_null	1.00	*
m4	0.00	-1.68 [1.59]
m4_null	1.00	*
m5	0.58	*
m5_null	0.42	-9.59 [10.85]
m6	0.00	-5.14 [1.01]
m6_null	1.00	*
m7	0.00	-2.03 [1.98]
m7_null	1.00	*
m8	0.00	-16.52 [10.79]
m8_null	1.00	*

Note. δ ELPD = differences in expected log posterior density between model pairs; SE = standard error of ELPD differences. Asterisks denote the best performing model in each model pair.

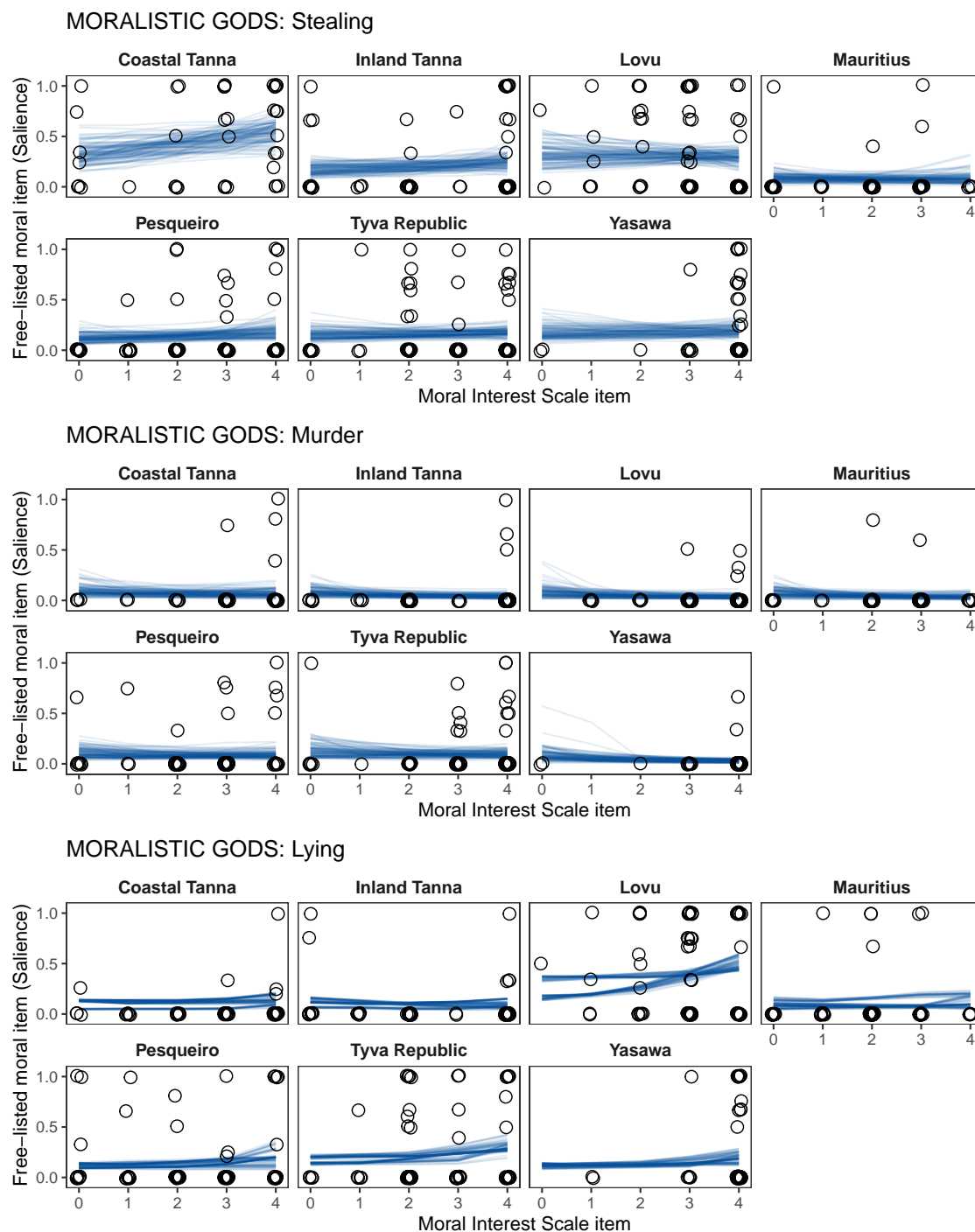


Figure 10. Posterior predictions for each field site from Model 3 through 5 on the relationship between the moral interest scale items (x -axis) and salience of the corresponding free-listed code (y -axis). Lines are draws of expected values from the posterior predictive distributions. Raw data points are slightly jittered. Note the convergence issues in Model 5.

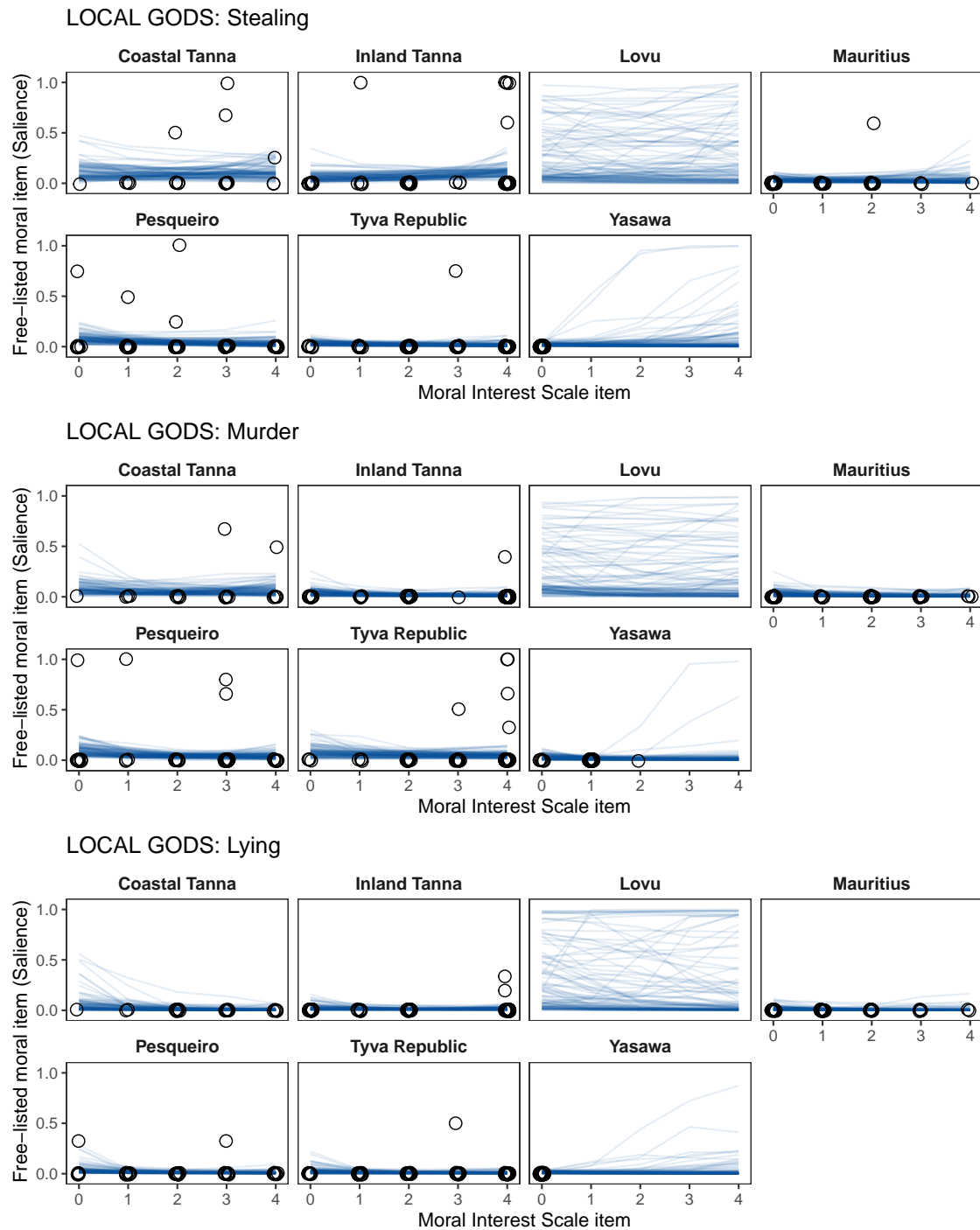


Figure 11. Posterior predictions for each field site from Model 6 through 8 on the relationship between the moral interest scale items (x -axis) and saliency of the corresponding free-listed code (y -axis). Lines are draws of expected values from the posterior predictive distributions. Raw data points are slightly jittered.

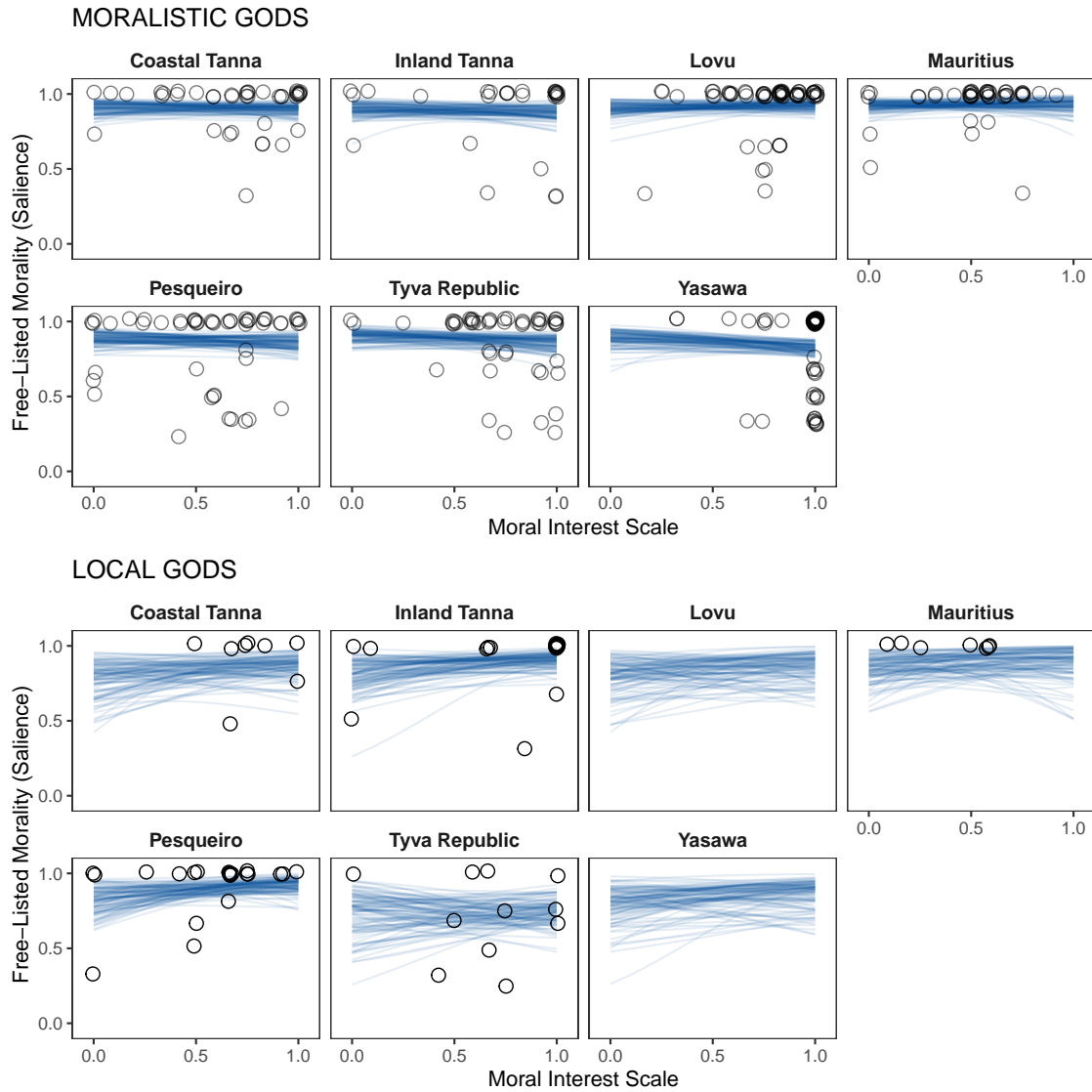


Figure 12. Reduced models. Posterior predictions for each field site from Model 1 and 2 on the relationship between the moral interest scale (x -axis) and saliency of free-listed Morality (y -axis) with only non-zero free-list responses. Lines are draws of expected values from the posterior predictive distributions. Raw data points are slightly jittered.

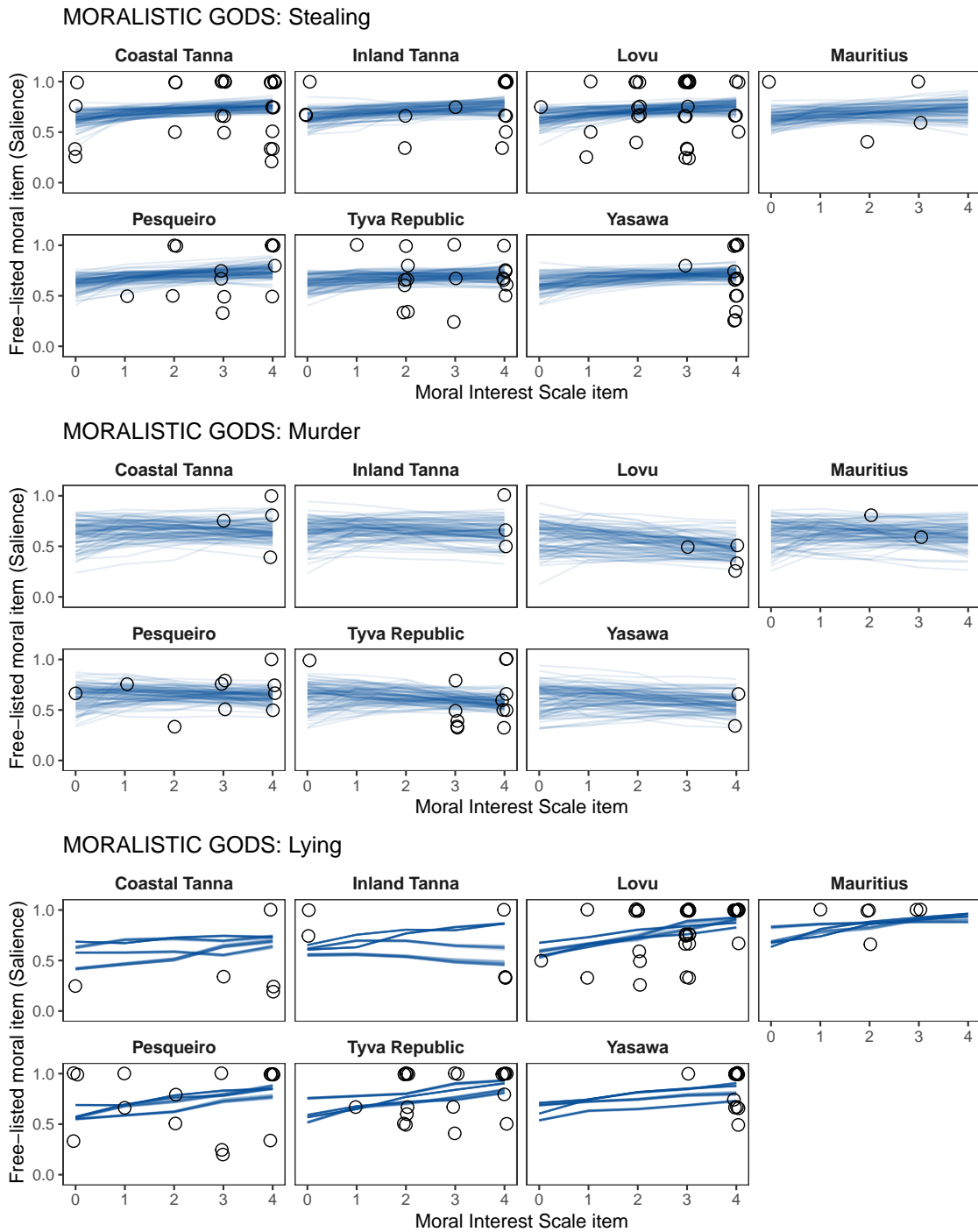


Figure 13. Reduced models. Posterior predictions for each field site from Model 3 through 5 on the relationship between the moral interest scale (x -axis) and saliency of the corresponding free-listed code (y -axis) with only non-zero free-list responses. Lines are draws of expected values from the posterior predictive distributions. Raw data points are slightly jittered. Note the convergence issues in Model 5.

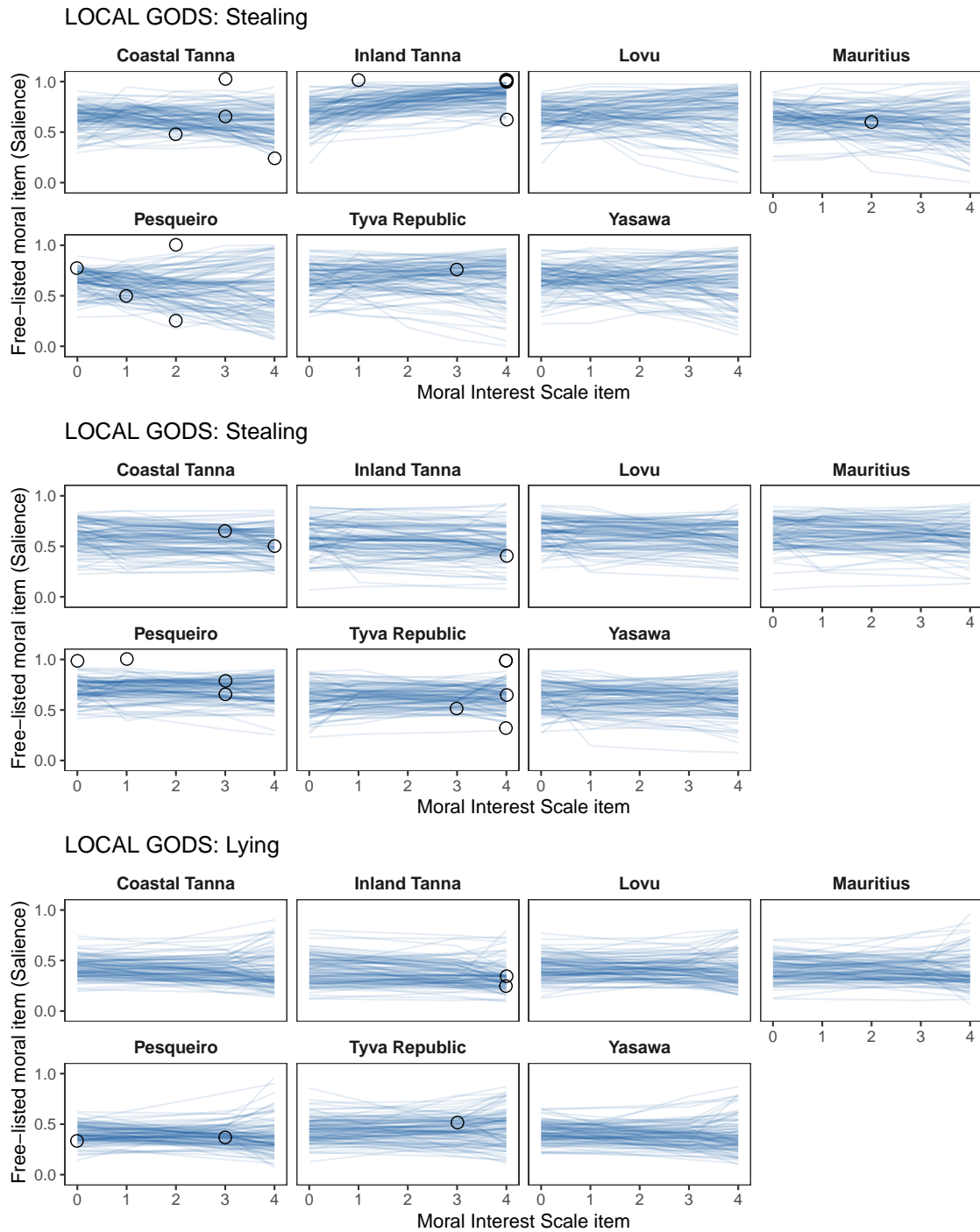


Figure 14. Reduced models. Posterior predictions for each field site from Model 6 through 8 on the relationship between the moral interest scale (x -axis) and saliency of the corresponding free-listed code (y -axis) with only non-zero free-list responses. Lines are draws of expected values from the posterior predictive distributions. Raw data points are slightly jittered.

Supplementary Section D

We used R version 4.1.2 (R Core Team, 2021) and the following R packages: abind v. 1.4.5 (Plate & Heiberger, 2016), AnthroTools v. 0.8 (Lane & Purzycki., 2016), arrayhelpers v. 1.1.0 (Beleites, 2020), askpass v. 1.1 (Ooms, 2019), assertthat v. 0.2.1 (H. Wickham, 2019a), backports v. 1.4.1 (Lang & R Core Team, 2021), base64enc v. 0.1.3 (Urbanek, 2015), bayesplot v. 1.8.1 (Gabry & Mahr, 2021; Gabry, Simpson, Vehtari, Betancourt, & Gelman, 2019), bdsmatrix v. 1.3.4 (Therneau, 2020), BH v. 1.75.0.0 (Eddelbuettel, Emerson, & Kane, 2021), bit v. 4.0.4 (Oehlschlägel & Ripley, 2020), bit64 v. 4.0.5 (Oehlschlägel & Silvestri, 2020), blob v. 1.2.2 (H. Wickham, 2021a), bookdown v. 0.25 (Xie, 2016, 2022a), bridgesampling v. 1.1.2 (Gronau, Singmann, & Wagenmakers, 2020), brio v. 1.1.3 (Hester & Csárdi, 2021), brms v. 2.16.3 (Bürkner, 2017, 2018, 2021), Brobdingnag v. 1.2.6 (Hankin, 2007), bslib v. 0.3.1 (Sievert & Cheng, 2021a), cachem v. 1.0.6 (Chang, 2021a), callr v. 3.7.0 (Csárdi & Chang, 2021a), cellranger v. 1.1.0 (Bryan, 2016), checkmate v. 2.0.0 (Lang, 2017), clipr v. 0.7.1 (Lincoln, 2020), coda v. 0.19.4 (Plummer, Best, Cowles, & Vines, 2006), colorspace v. 2.0.2 (Stauffer, Mayr, Dabernig, & Zeileis, 2009; Zeileis et al., 2020; Zeileis, Hornik, & Murrell, 2009), colourpicker v. 1.1.1 (Attali, 2021a), commonmark v. 1.7 (Ooms, 2018), cpp11 v. 0.4.2 (Hester & François, 2021), crosstalk v. 1.2.0 (Cheng & Sievert, 2021), curl v. 4.3.2 (Ooms, 2021a), data.table v. 1.14.2 (Dowle & Srinivasan, 2021), DBI v. 1.1.2 (R Special Interest Group on Databases (R-SIG-DB), Wickham, & Müller, 2021), desc v. 1.4.0 (Csárdi, Müller, & Hester, 2021), diffobj v. 0.3.5 (Gaslam, 2021), digest v. 0.6.29 (Antoine Lucas et al., 2021), distributional v. 0.3.0 (O'Hara-Wild, Kay, & Hayes, 2022), DT v. 0.20 (Xie, Cheng, & Tan, 2021), dygraphs v. 1.1.1.6 (Vanderkam, Allaire, Owen, Gromer, & Thieurmél, 2018), ellipsis v. 0.3.2 (H. Wickham, 2021b), evaluate v. 0.15 (H. Wickham & Xie, 2022), extraDistr v. 1.9.1 (Wolodzko, 2020), fansi v. 1.0.2 (Gaslam, 2022), farver v. 2.1.0 (Pedersen, Nicolae, & François, 2021), fastmap v. 1.1.0 (Chang, 2021b), finalfit v. 1.0.4 (Harrison, Drake, & Ots, 2021), fontawesome v. 0.2.2 (Iannone, 2021), fs v. 1.5.2 (Hester, Wickham, & Csárdi, 2021), future v. 1.23.0 (Bengtsson, 2021a), gargle v. 1.2.0 (Bryan, Citro, & Wickham, 2021), generics v. 0.1.2 (H. Wickham, Kuhn, & Vaughan, 2022), GGally v. 2.1.2 (Schloerke et al., 2021), ggdist v. 3.0.1 (Kay, 2021), ggExtra v. 0.9 (Attali & Baker, 2019), ggridges v. 0.5.3 (Wilke, 2021), globals v. 0.14.0 (Bengtsson, 2020), glue v. 1.6.1 (Hester & Bryan, 2022), grateful v. 0.1.11 (Rodríguez-Sánchez, Jackson, & Hutchins, 2022), gridExtra v. 2.3 (Auguie, 2017), gtable v. 0.3.0 (H. Wickham & Pedersen, 2019), gtools v. 3.9.2 (Warnes, Bolker, & Lumley, 2021), HDInterval v. 0.2.2 (Meredith & Kruschke, 2020), highr v. 0.9 (Xie & Qiu, 2021), htmltools v. 0.5.2 (Cheng et al., 2021), htmlwidgets v. 1.5.4 (Vaidyanathan et al., 2021), httpuv v. 1.6.5 (Cheng & Chang, 2022), ids v. 1.0.1 (FitzJohn, 2017), igraph v. 1.2.11 (Csardi & Nepusz, 2006), inline v. 0.3.19 (Sklyar et al., 2021), isoband v. 0.2.5 (Wilke & Pedersen, 2021), jquerylib v. 0.1.4 (Sievert & Cheng, 2021b), knitr v. 1.39 (Xie, 2014, 2015, 2022b), labeling v. 0.4.2 (Justin Talbot, 2020), later v. 1.3.0 (Chang & Cheng, 2021), lazyeval v. 0.2.2 (H. Wickham, 2019b), lifecycle v. 1.0.1 (Henry & Wickham, 2021a), listen v. 0.8.0 (Bengtsson, 2019), lme4 v. 1.1.27.1 (Bates, Mächler, Bolker, & Walker, 2015), loo v. 2.4.1 (Vehtari et al., 2020; Vehtari, Gelman, & Gabry, 2017; Yao, Vehtari, Simpson, & Gelman, 2017), markdown v. 1.1 (Allaire, Horner, Xie, Marti, & Porte, 2019), matrixStats v. 0.61.0 (Bengtsson, 2021b), memoise v. 2.0.1 (H. Wickham, Hester, Chang, Müller, & Cook, 2021), mice v. 3.14.0 (van Buuren & Groothuis-Oudshoorn,

205 2011), mime v. 0.12 (Xie, 2021a), miniUI v. 0.1.1.1 (Cheng, 2018), minqa v. 1.2.4 (Bates,
 206 Mullen, Nash, & Varadhan, 2014), munsell v. 0.5.0 (C. Wickham, 2018), mvtnorm v. 1.1.3
 207 (Genz & Bretz, 2009; Genz et al., 2021), nleqslv v. 3.3.2 (Hasselman, 2018), nloptr v. 1.2.2.2
 208 (Johnson, ?), numDeriv v. 2016.8.1.1 (Gilbert & Varadhan, 2019), openssl v. 1.4.5 (Ooms,
 209 2021b), papaja v. 0.1.1 (Aust & Barth, 2022), parallelly v. 1.29.0 (Bengtsson, 2021c),
 210 patchwork v. 1.1.1 (Pedersen, 2020), pkgbuild v. 1.3.1 (H. Wickham, Hester, & Csárdi,
 211 2021), pkgconfig v. 2.0.3 (Csárdi, 2019), pkgload v. 1.2.4 (H. Wickham, Chang, Hester,
 212 & Henry, 2021), plyr v. 1.8.6 (H. Wickham, 2011b), posterior v. 1.2.0 (Bürkner, Gabry,
 213 Kay, & Vehtari, 2022; Vehtari, Gelman, Simpson, Carpenter, & Bürkner, 2021), praise
 214 v. 1.0.0 (Csardi & Sorhus, 2015), prettyunits v. 1.1.1 (Csardi, 2020), pROC v. 1.18.0
 215 (Robin et al., 2011), processx v. 3.5.2 (Csárdi & Chang, 2021b), progress v. 1.2.2 (Csárdi &
 216 FitzJohn, 2019), promises v. 1.2.0.1 (Cheng, 2021), ps v. 1.6.0 (Loden, Daeschler, Rodola',
 217 & Csárdi, 2021), R6 v. 2.5.1 (Chang, 2021c), rappdirs v. 0.3.3 (Ratnakumar, Mick, &
 218 Davis, 2021), RColorBrewer v. 1.1.2 (Neuwirth, 2014), Rcpp v. 1.0.8 (Eddelbuettel, 2013;
 219 Eddelbuettel & Balamuta, 2018; Eddelbuettel & François, 2011), RcppEigen v. 0.3.3.9.1
 220 (Bates & Eddelbuettel, 2013), RcppParallel v. 5.1.5 (Allaire, Francois, et al., 2022), rematch
 221 v. 1.0.1 (Csardi, 2016), rematch2 v. 2.1.2 (Csárdi, 2020), renv v. 0.15.4 (Ushey, 2022),
 222 reshape v. 0.8.8 (H. Wickham, 2007a), reshape2 v. 1.4.4 (H. Wickham, 2007b), rmarkdown v.
 223 2.14 (Allaire, Xie, et al., 2022; Xie, Allaire, & Golemund, 2018; Xie, Dervieux, & Riederer,
 224 2020), rmdfilt v. 0.1.3 (Aust, 2020), rprojroot v. 2.0.2 (Müller, 2020), rstan v. 2.21.3 (Stan
 225 Development Team, 2021), rstantools v. 2.1.1 (Gabry, Goodrich, & Lysy, 2020), sass v.
 226 0.4.0 (Cheng, Mastny, Iannone, Schloerke, & Sievert, 2021), scales v. 1.1.1 (H. Wickham
 227 & Seidel, 2020), selectr v. 0.4.2 (Potter, 2012), shiny v. 1.7.1 (Chang et al., 2021), shinyjs
 228 v. 2.1.0 (Attali, 2021b), shinystan v. 2.5.0 (Gabry, 2018), shinythemes v. 1.2.0 (Chang,
 229 2021d), sourcetools v. 0.1.7 (Ushey, 2018), StanHeaders v. 2.21.0.7 (Stan Development
 230 Team, 2020), stringi v. 1.7.6 (Gagolewski, 2021a, 2021b), svUnit v. 1.0.6 (Grosjean, 2021),
 231 sys v. 3.4 (Ooms, 2020), tensorA v. 0.36.2 (van den Boogaart, 2020), testthat v. 3.1.2 (H.
 232 Wickham, 2011a), threejs v. 0.3.3 (Lewis, 2020), tidybayes v. 3.0.2 (Kay, 2022), tidyselect v.
 233 1.1.1 (Henry & Wickham, 2021b), tidyverse v. 1.3.1 (H. Wickham et al., 2019), tinylab v.
 234 0.2.3 (Barth, 2022), tinytex v. 0.36 (Xie, 2019, 2021b), tzdb v. 0.2.0 (Vaughan, 2021), utf8
 235 v. 1.2.2 (Perry, 2021), uuid v. 1.0.3 (Urbanek & Ts'o, 2021), vctrs v. 0.3.8 (H. Wickham,
 236 Henry, & Vaughan, 2021), viridisLite v. 0.4.0 (Garnier et al., 2021), vroom v. 1.5.7 (Hester,
 237 Wickham, & Bryan, 2021), waldo v. 0.3.1 (H. Wickham, 2021c), withr v. 2.4.3 (Hester et
 238 al., 2021), xfun v. 0.29 (Xie, 2021c), xtable v. 1.8.4 (Dahl, Scott, Roosen, Magnusson, &
 239 Swinton, 2019), xts v. 0.12.1 (Ryan & Ulrich, 2020), yaml v. 2.3.5 (Garbett et al., 2022), zip
 240 v. 2.2.0 (Csárdi, Podgórski, & Geldreich, 2021), zoo v. 1.8.9 (Zeileis & Grothendieck, 2005).

Data Availability

Data and code to reproduce this manuscript is available at: <https://github.com/tbendixen/freelist-tutorial>.

References

- Allaire, J., Francois, R., Ushey, K., Vandenbrouck, G., Geelnard, M., & Intel. (2022). *RcppParallel: Parallel programming tools for 'rcpp'*. Retrieved from <https://CRAN.R-project.org/package=RcppParallel>
- Allaire, J., Horner, J., Xie, Y., Marti, V., & Porte, N. (2019). *Markdown: Render markdown with the c library 'sundown'*. Retrieved from <https://CRAN.R-project.org/package=markdown>
- Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., ... Iannone, R. (2022). *Rmarkdown: Dynamic documents for r*. Retrieved from <https://github.com/rstudio/rmarkdown>
- Antoine Lucas, D. E. with contributions by, Tuszynski, J., Bengtsson, H., Urbanek, S., Frasca, M., Lewis, B., ... Winston Chang., and. (2021). *Digest: Create compact hash digests of r objects*. Retrieved from <https://CRAN.R-project.org/package=digest>
- Attali, D. (2021a). *Colourpicker: A colour picker tool for shiny and for selecting colours in plots*. Retrieved from <https://CRAN.R-project.org/package=colourpicker>
- Attali, D. (2021b). *Shinyjs: Easily improve the user experience of your shiny apps in seconds*. Retrieved from <https://CRAN.R-project.org/package=shinyjs>
- Attali, D., & Baker, C. (2019). *ggExtra: Add marginal histograms to 'ggplot2', and more 'ggplot2' enhancements*. Retrieved from <https://CRAN.R-project.org/package=ggExtra>
- Auguie, B. (2017). *gridExtra: Miscellaneous functions for "grid" graphics*. Retrieved from <https://CRAN.R-project.org/package=gridExtra>
- Aust, F. (2020). *Rmdfiltr: 'Lua'-filters for r markdown*. Retrieved from <https://CRAN.R-project.org/package=rmdfiltr>
- Aust, F., & Barth, M. (2022). *papaja: Prepare reproducible APA journal articles with R Markdown*. Retrieved from <https://github.com/crsh/papaja>
- Barth, M. (2022). *tinylabls: Lightweight variable labels*. Retrieved from <https://cran.r-project.org/package=tinylabls>
- Bates, D., & Eddelbuettel, D. (2013). Fast and elegant numerical linear algebra using the RcppEigen package. *Journal of Statistical Software*, 52(5), 1–24. Retrieved from <https://www.jstatsoft.org/v52/i05/>
- Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1), 1–48. <https://doi.org/10.18637/jss.v067.i01>
- Bates, D., Mullen, K. M., Nash, J. C., & Varadhan, R. (2014). *Minqa: Derivative-free optimization algorithms by quadratic approximation*. Retrieved from <https://CRAN.R-project.org/package=minqa>
- Beleites, C. (2020). *Arrayhelpers: Convenience functions for arrays*. Retrieved from <https://CRAN.R-project.org/package=arrayhelpers>
- Bendixen, T., Apicella, C., Atkinson, Q., Cohen, E., Henrich, J., McNamara, R. A., ... Purzycki, B. G. (forthcoming). Appealing to the minds of gods: Religious beliefs and appeals correspond to features of local social ecologies. *Religion, Brain & Behavior*.

- Forthcoming. <https://doi.org/10.31234/osf.io/tjn3e>
- Bengtsson, H. (2019). *Listenenv: Environments behaving (almost) as lists*. Retrieved from <https://CRAN.R-project.org/package=listenenv>
- Bengtsson, H. (2020). *Globals: Identify global objects in r expressions*. Retrieved from <https://CRAN.R-project.org/package=globals>
- Bengtsson, H. (2021a). *A unifying framework for parallel and distributed processing in r using futures*. Retrieved from <https://journal.r-project.org/archive/2021/RJ-2021-048/index.html>
- Bengtsson, H. (2021b). *matrixStats: Functions that apply to rows and columns of matrices (and to vectors)*. Retrieved from <https://CRAN.R-project.org/package=matrixStats>
- Bengtsson, H. (2021c). *Parallely: Enhancing the 'parallel' package*. Retrieved from <https://CRAN.R-project.org/package=parallely>
- Bryan, J. (2016). *Cellranger: Translate spreadsheet cell ranges to rows and columns*. Retrieved from <https://CRAN.R-project.org/package=cellranger>
- Bryan, J., Citro, C., & Wickham, H. (2021). *Gargle: Utilities for working with google APIs*. Retrieved from <https://CRAN.R-project.org/package=gargle>
- Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1), 1–28. <https://doi.org/10.18637/jss.v080.i01>
- Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, 10(1), 395–411. <https://doi.org/10.32614/RJ-2018-017>
- Bürkner, P.-C. (2021). Bayesian item response modeling in R with brms and Stan. *Journal of Statistical Software*, 100(5), 1–54. <https://doi.org/10.18637/jss.v100.i05>
- Bürkner, P.-C., Gabry, J., Kay, M., & Vehtari, A. (2022). *Posterior: Tools for working with posterior distributions*. Retrieved from <https://mc-stan.org/posterior/>
- Chang, W. (2021a). *Cachem: Cache r objects with automatic pruning*. Retrieved from <https://CRAN.R-project.org/package=cachem>
- Chang, W. (2021b). *Fastmap: Fast data structures*. Retrieved from <https://CRAN.R-project.org/package=fastmap>
- Chang, W. (2021c). *R6: Encapsulated classes with reference semantics*. Retrieved from <https://CRAN.R-project.org/package=R6>
- Chang, W. (2021d). *Shinythemes: Themes for shiny*. Retrieved from <https://CRAN.R-project.org/package=shinythemes>
- Chang, W., & Cheng, J. (2021). *Later: Utilities for scheduling functions to execute later with event loops*. Retrieved from <https://CRAN.R-project.org/package=later>
- Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., ... Borges, B. (2021). *Shiny: Web application framework for r*. Retrieved from <https://CRAN.R-project.org/package=shiny>
- Cheng, J. (2018). *miniUI: Shiny UI widgets for small screens*. Retrieved from <https://CRAN.R-project.org/package=miniUI>
- Cheng, J. (2021). *Promises: Abstractions for promise-based asynchronous programming*. Retrieved from <https://CRAN.R-project.org/package=promises>
- Cheng, J., & Chang, W. (2022). *Httpuv: HTTP and WebSocket server library*. Retrieved from <https://CRAN.R-project.org/package=httpuv>
- Cheng, J., Mastny, T., Iannone, R., Schloerke, B., & Sievert, C. (2021). *Sass: Syntactically awesome style sheets ('sass')*. Retrieved from <https://CRAN.R-project.org/package=sass>

- 330 Cheng, J., & Sievert, C. (2021). *Crosstalk: Inter-widget interactivity for HTML widgets*.
 331 Retrieved from <https://CRAN.R-project.org/package=crosstalk>
- 332 Cheng, J., Sievert, C., Schloerke, B., Chang, W., Xie, Y., & Allen, J. (2021). *Htmltools:*
 333 *Tools for HTML*. Retrieved from <https://CRAN.R-project.org/package=htmltools>
- 334 Csardi, G. (2016). *Rematch: Match regular expressions with a nicer 'API'*. Retrieved from
 335 <https://CRAN.R-project.org/package=rematch>
- 336 Csardi, G. (2020). *Prettyunits: Pretty, human readable formatting of quantities*. Retrieved
 337 from <https://CRAN.R-project.org/package=prettyunits>
- 338 Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research.
 339 *InterJournal, Complex Systems*, 1695. Retrieved from <https://igraph.org>
- 340 Csardi, G., & Sorhus, S. (2015). *Praise: Praise users*. Retrieved from <https://CRAN.R-project.org/package=praise>
- 341 Csárdi, G. (2019). *Pkgconfig: Private configuration for 'r' packages*. Retrieved from
 342 <https://CRAN.R-project.org/package=pkgconfig>
- 343 Csárdi, G. (2020). *rematch2: Tidy output from regular expression matching*. Retrieved from
 344 <https://CRAN.R-project.org/package=rematch2>
- 345 Csárdi, G., & Chang, W. (2021a). *Callr: Call r from r*. Retrieved from <https://CRAN.R-project.org/package=callr>
- 346 Csárdi, G., & Chang, W. (2021b). *Processx: Execute and control system processes*. Retrieved
 347 from <https://CRAN.R-project.org/package=processx>
- 348 Csárdi, G., & FitzJohn, R. (2019). *Progress: Terminal progress bars*. Retrieved from
 349 <https://CRAN.R-project.org/package=progress>
- 350 Csárdi, G., Müller, K., & Hester, J. (2021). *Desc: Manipulate DESCRIPTION files*.
 351 Retrieved from <https://CRAN.R-project.org/package=desc>
- 352 Csárdi, G., Podgórski, K., & Geldreich, R. (2021). *Zip: Cross-platform 'zip' compression*.
 353 Retrieved from <https://CRAN.R-project.org/package=zip>
- 354 Dahl, D. B., Scott, D., Roosen, C., Magnusson, A., & Swinton, J. (2019). *Xtable: Export*
 355 *tables to LaTeX or HTML*. Retrieved from <https://CRAN.R-project.org/package=xtable>
- 356 Donnellan, E., Usami, S., & Murayama, K. (2022). *Random item slope regression: Examining*
 357 *both similarities and differences in the association with individual items*. PsyArXiv.
 358 <https://doi.org/10.31234/osf.io/s6erz>
- 359 Dowle, M., & Srinivasan, A. (2021). *Data.table: Extension of 'data.frame'*. Retrieved from
 360 <https://CRAN.R-project.org/package=data.table>
- 361 Eddelbuettel, D. (2013). *Seamless R and C++ integration with Rcpp*. New York: Springer.
 362 <https://doi.org/10.1007/978-1-4614-6868-4>
- 363 Eddelbuettel, D., & Balamuta, J. J. (2018). Extending extitR with extitC++: A Brief
 364 Introduction to extitRcpp. *The American Statistician*, 72(1), 28–36. <https://doi.org/10.1080/00031305.2017.1375990>
- 365 Eddelbuettel, D., Emerson, J. W., & Kane, M. J. (2021). *BH: Boost c++ header files*.
 366 Retrieved from <https://CRAN.R-project.org/package=BH>
- 367 Eddelbuettel, D., & François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal*
 368 *of Statistical Software*, 40(8), 1–18. <https://doi.org/10.18637/jss.v040.i08>
- 369 FitzJohn, R. (2017). *Ids: Generate random identifiers*. Retrieved from <https://CRAN.R-project.org/package=ids>
- 370 Gabry, J. (2018). *Shinystan: Interactive visual and numerical diagnostics and posterior*

- 375 *analysis for bayesian models*. Retrieved from <https://CRAN.R-project.org/package=shiny>
 376 *inystan*
- 377 Gabry, J., Goodrich, B., & Lysy, M. (2020). *Rstantools: Tools for developing r packages*
 378 *interfacing with 'stan'*. Retrieved from <https://CRAN.R-project.org/package=rstantools>
- 379 Gabry, J., & Mahr, T. (2021). *Bayesplot: Plotting for bayesian models*. Retrieved from
 380 <https://mc-stan.org/bayesplot/>
- 381 Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., & Gelman, A. (2019). Visualization in
 382 bayesian workflow. *J. R. Stat. Soc. A*, 182, 389–402. <https://doi.org/10.1111/rssa.12378>
- 383 Gagolewski, M. (2021a). Stringi: Fast and portable character string processing in r. *Journal*
 384 *of Statistical Software*.
- 385 Gagolewski, M. (2021b). *Stringi: Fast and portable character string processing in r*. Retrieved
 386 from <https://stringi.gagolewski.com/>
- 387 Garbett, S. P., Stephens, J., Simonov, K., Xie, Y., Dong, Z., Wickham, H., ... Kamvar,
 388 Z. N. (2022). *Yaml: Methods to convert r data to YAML and back*. Retrieved from
 389 <https://CRAN.R-project.org/package=yaml>
- 390 Garnier, Simon, Ross, Noam, Rudis, Robert, ... Cédric. (2021). *viridis - colorblind-friendly*
 391 *color maps for r*. <https://doi.org/10.5281/zenodo.4679424>
- 392 Gaslam, B. (2021). *Diffobj: Diffs for r objects*. Retrieved from <https://CRAN.R-project.org/package=diffobj>
- 393
- 394 Gaslam, B. (2022). *Fansi: ANSI control sequence aware string functions*. Retrieved from
 395 <https://CRAN.R-project.org/package=fansi>
- 396 Genz, A., & Bretz, F. (2009). *Computation of multivariate normal and t probabilities*.
 397 Heidelberg: Springer-Verlag.
- 398 Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., & Hothorn, T. (2021).
 399 *mvtnorm: Multivariate normal and t distributions*. Retrieved from <https://CRAN.R-project.org/package=mvtnorm>
- 400
- 401 Gilbert, P., & Varadhan, R. (2019). *numDeriv: Accurate numerical derivatives*. Retrieved
 402 from <https://CRAN.R-project.org/package=numDeriv>
- 403 Gronau, Q. F., Singmann, H., & Wagenmakers, E.-J. (2020). bridgesampling: An R package
 404 for estimating normalizing constants. *Journal of Statistical Software*, 92(10), 1–29.
 405 <https://doi.org/10.18637/jss.v092.i10>
- 406 Grosjean, P. (2021). *SciViews-r*. MONS, Belgium: UMONS. Retrieved from <https://www.sciviews.org/SciViews-R/>
- 407
- 408 Hankin, R. K. S. (2007). Very large numbers in r: Introducing package brobdingnag. *R*
 409 *News*, 7.
- 410 Harrison, E., Drake, T., & Ots, R. (2021). *Finalfit: Quickly create elegant regression results*
 411 *tables and plots when modelling*. Retrieved from <https://CRAN.R-project.org/package=finalfit>
- 412
- 413 Hasselman, B. (2018). *Nleqslv: Solve systems of nonlinear equations*. Retrieved from
 414 <https://CRAN.R-project.org/package=nleqslv>
- 415 Henry, L., & Wickham, H. (2021a). *Lifecycle: Manage the life cycle of your package functions*.
 416 Retrieved from <https://CRAN.R-project.org/package=lifecycle>
- 417 Henry, L., & Wickham, H. (2021b). *Tidysselect: Select from a set of strings*. Retrieved from
 418 <https://CRAN.R-project.org/package=tidysselect>
- 419 Hester, J., & Bryan, J. (2022). *Glue: Interpreted string literals*. Retrieved from <https://CRAN.R-project.org/package=glue>

- 420 [//CRAN.R-project.org/package=glue](https://CRAN.R-project.org/package=glue)
- 421 Hester, J., & Csárdi, G. (2021). *Brio: Basic r input output*. Retrieved from [https://CRAN.R-](https://CRAN.R-project.org/package=brio)
- 422 [project.org/package=brio](https://CRAN.R-project.org/package=brio)
- 423 Hester, J., & François, R. (2021). *cpp11: A c++11 interface for r's c interface*. Retrieved
- 424 from <https://CRAN.R-project.org/package=cpp11>
- 425 Hester, J., Henry, L., Müller, K., Ushey, K., Wickham, H., & Chang, W. (2021). *Withr:*
- 426 *Run code 'with' temporarily modified global state*. Retrieved from [https://CRAN.R-](https://CRAN.R-project.org/package=withr)
- 427 [project.org/package=withr](https://CRAN.R-project.org/package=withr)
- 428 Hester, J., Wickham, H., & Bryan, J. (2021). *Vroom: Read and write rectangular text data*
- 429 *quickly*. Retrieved from <https://CRAN.R-project.org/package=vroom>
- 430 Hester, J., Wickham, H., & Csárdi, G. (2021). *Fs: Cross-platform file system operations*
- 431 *based on 'libuv'*. Retrieved from <https://CRAN.R-project.org/package=fs>
- 432 Iannone, R. (2021). *Fontawesome: Easily work with 'font awesome' icons*. Retrieved from
- 433 <https://CRAN.R-project.org/package=fontawesome>
- 434 Johnson, S. G. (?). The NLOpt nonlinear-optimization package. ?, ?(?), ?
- 435 Justin Talbot. (2020). *Labeling: Axis labeling*. Retrieved from [https://CRAN.R-project.or](https://CRAN.R-project.org/package=labeling)
- 436 [g/package=labeling](https://CRAN.R-project.org/package=labeling)
- 437 Kay, M. (2021). *ggdist: Visualizations of distributions and uncertainty*. [https://doi.org/10.5](https://doi.org/10.5281/zenodo.3879620)
- 438 [281/zenodo.3879620](https://doi.org/10.5281/zenodo.3879620)
- 439 Kay, M. (2022). *tidybayes: Tidy data and geoms for Bayesian models*. [https://doi.org/10.5](https://doi.org/10.5281/zenodo.1308151)
- 440 [281/zenodo.1308151](https://doi.org/10.5281/zenodo.1308151)
- 441 Kubinec, R. (2022). Ordered beta regression: A parsimonious, well-fitting model for
- 442 continuous data with lower and upper bounds. *Political Analysis*. [https://doi.org/10.101](https://doi.org/10.1017/pan.2022.20)
- 443 [7/pan.2022.20](https://doi.org/10.1017/pan.2022.20)
- 444 Lane, A. J., & Purzycki, B. G. (2016). *AnthroTools: Some custom tools for anthropology*.
- 445 Lang, M. (2017). checkmate: Fast argument checks for defensive r programming. *The R*
- 446 *Journal*, 9(1), 437–445. Retrieved from [https://journal.r-project.org/archive/2017/RJ-](https://journal.r-project.org/archive/2017/RJ-2017-028/index.html)
- 447 [2017-028/index.html](https://journal.r-project.org/archive/2017/RJ-2017-028/index.html)
- 448 Lang, M., & R Core Team. (2021). *Backports: Reimplementations of functions introduced*
- 449 *since r-3.0.0*. Retrieved from <https://CRAN.R-project.org/package=backports>
- 450 Lewis, B. W. (2020). *Threejs: Interactive 3D scatter plots, networks and globes*. Retrieved
- 451 from <https://CRAN.R-project.org/package=threejs>
- 452 Lincoln, M. (2020). *Clipr: Read and write from the system clipboard*. Retrieved from
- 453 <https://CRAN.R-project.org/package=clipr>
- 454 Loden, J., Daeschler, D., Rodola, G., & Csárdi, G. (2021). *Ps: List, query, manipulate*
- 455 *system processes*. Retrieved from <https://CRAN.R-project.org/package=ps>
- 456 Meredith, M., & Kruschke, J. (2020). *HDInterval: Highest (posterior) density intervals*.
- 457 Retrieved from <https://CRAN.R-project.org/package=HDInterval>
- 458 Müller, K. (2020). *Rprojroot: Finding files in project subdirectories*. Retrieved from
- 459 <https://CRAN.R-project.org/package=rprojroot>
- 460 Neuwirth, E. (2014). *RColorBrewer: ColorBrewer palettes*. Retrieved from [https://CRAN.R-](https://CRAN.R-project.org/package=RColorBrewer)
- 461 [project.org/package=RColorBrewer](https://CRAN.R-project.org/package=RColorBrewer)
- 462 O'Hara-Wild, M., Kay, M., & Hayes, A. (2022). *Distributional: Vectorised probability*
- 463 *distributions*. Retrieved from <https://CRAN.R-project.org/package=distributional>
- 464 Oehlschlägel, J., & Ripley, B. (2020). *Bit: Classes and methods for fast memory-efficient*

- 465 *boolean selections*. Retrieved from <https://CRAN.R-project.org/package=bit>
- 466 Oehlschlägel, J., & Silvestri, L. (2020). *bit64: A S3 class for vectors of 64bit integers*.
 467 Retrieved from <https://CRAN.R-project.org/package=bit64>
- 468 Ooms, J. (2018). *Commonmark: High performance CommonMark and github markdown*
 469 *rendering in r*. Retrieved from <https://CRAN.R-project.org/package=commonmark>
- 470 Ooms, J. (2019). *Askpass: Safe password entry for r, git, and SSH*. Retrieved from <https://CRAN.R-project.org/package=askpass>
- 471
 472 Ooms, J. (2020). *Sys: Powerful and reliable tools for running system commands in r*.
 473 Retrieved from <https://CRAN.R-project.org/package=sys>
- 474 Ooms, J. (2021a). *Curl: A modern and flexible web client for r*. Retrieved from <https://CRAN.R-project.org/package=curl>
- 475
 476 Ooms, J. (2021b). *Openssl: Toolkit for encryption, signatures and certificates based on*
 477 *OpenSSL*. Retrieved from <https://CRAN.R-project.org/package=openssl>
- 478 Pedersen, T. L. (2020). *Patchwork: The composer of plots*. Retrieved from <https://CRAN.R-project.org/package=patchwork>
- 479
 480 Pedersen, T. L., Nicolae, B., & François, R. (2021). *Farver: High performance colour space*
 481 *manipulation*. Retrieved from <https://CRAN.R-project.org/package=farver>
- 482 Perry, P. O. (2021). *utf8: Unicode text processing*. Retrieved from <https://CRAN.R-project.org/package=utf8>
- 483
 484 Plate, T., & Heiberger, R. (2016). *Abind: Combine multidimensional arrays*. Retrieved from
 485 <https://CRAN.R-project.org/package=abind>
- 486 Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). CODA: Convergence diagnosis
 487 and output analysis for MCMC. *R News*, 6(1), 7–11. Retrieved from <https://journal.r-project.org/archive/>
- 488
 489 Potter, S. (2012). *Introducing the selectr package*. Auckland, New Zealand: The University
 490 of Auckland. Retrieved from The University of Auckland website: <http://stattech.wordpress.fos.auckland.ac.nz/2012-10-introducing-the-selectr-package/>
- 491
 492 Purzycki, B. G., & Bendixen, T. (2020). Examining Values, Virtues, and Tradition in the
 493 Tyva Republic with Free-List and Demographic Data. *The New Research in Tuva*, 4,
 494 6–18.
- 495 Purzycki, B. G., Stagnaro, M. N., & Sasaki, J. (2020). Breaches of Trust Change the
 496 Content and Structure of Religious Appeals. *Journal for the Study of Religion, Nature*
 497 *and Culture*, 14(1), 71–94. <https://doi.org/10.1558/jsrnc.38786>
- 498 R Core Team. (2021). *R: A language and environment for statistical computing*. Vienna,
 499 Austria: R Foundation for Statistical Computing. Retrieved from <https://www.R-project.org/>
- 500
 501 R Special Interest Group on Databases (R-SIG-DB), Wickham, H., & Müller, K. (2021).
 502 *DBI: R database interface*. Retrieved from <https://CRAN.R-project.org/package=DBI>
- 503 Ratnakumar, S., Mick, T., & Davis, T. (2021). *Rappdirs: Application directories: Determine*
 504 *where to save data, caches, and logs*. Retrieved from <https://CRAN.R-project.org/package=rappdirs>
- 505
 506 Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., & Müller, M.
 507 (2011). pROC: An open-source package for r and s+ to analyze and compare ROC curves.
 508 *BMC Bioinformatics*, 12, 77.
- 509 Rodríguez-Sánchez, F., Jackson, C. P., & Hutchins, S. D. (2022). *Grateful: Facilitate citation*

- 510 *of r packages*. Retrieved from <https://github.com/Pakillo/grateful>
- 511 Ryan, J. A., & Ulrich, J. M. (2020). *Xts: eXtensible time series*. Retrieved from <https://CRAN.R-project.org/package=xts>
- 512
- 513 Schloerke, B., Cook, D., Larmarange, J., Briatte, F., Marbach, M., Thoen, E., ... Crowley,
514 J. (2021). *GGally: Extension to 'ggplot2'*. Retrieved from [https://CRAN.R-project.org/](https://CRAN.R-project.org/package=GGally)
515 [package=GGally](https://CRAN.R-project.org/package=GGally)
- 516 Sievert, C., & Cheng, J. (2021a). *Bslib: Custom 'bootstrap' 'sass' themes for 'shiny' and*
517 *'rmarkdown'*. Retrieved from <https://CRAN.R-project.org/package=bslib>
- 518 Sievert, C., & Cheng, J. (2021b). *Jquerylib: Obtain 'jQuery' as an HTML dependency object*.
519 Retrieved from <https://CRAN.R-project.org/package=jquerylib>
- 520 Sklyar, O., Murdoch, D., Smith, M., Eddelbuettel, D., Francois, R., Soetaert, K., & Ranke,
521 J. (2021). *Inline: Functions to inline c, c++, fortran function calls from r*. Retrieved
522 from <https://CRAN.R-project.org/package=inline>
- 523 Stan Development Team. (2020). *StanHeaders: Headers for the R interface to Stan*.
524 Retrieved from <https://mc-stan.org/>
- 525 Stan Development Team. (2021). *RStan: The R interface to Stan*. Retrieved from
526 <https://mc-stan.org/>
- 527 Stauffer, R., Mayr, G. J., Dabernig, M., & Zeileis, A. (2009). Somewhere over the rainbow:
528 How to make effective use of colors in meteorological visualizations. *Bulletin of the*
529 *American Meteorological Society*, 96(2), 203–216. [https://doi.org/10.1175/BAMS-D-13-](https://doi.org/10.1175/BAMS-D-13-00155.1)
530 [00155.1](https://doi.org/10.1175/BAMS-D-13-00155.1)
- 531 Therneau, T. (2020). *Bdsmatrix: Routines for block diagonal symmetric matrices*. Retrieved
532 from <https://CRAN.R-project.org/package=bdsmatrix>
- 533 Urbanek, S. (2015). *base64enc: Tools for base64 encoding*. Retrieved from [https://CRAN.R-](https://CRAN.R-project.org/package=base64enc)
534 [project.org/package=base64enc](https://CRAN.R-project.org/package=base64enc)
- 535 Urbanek, S., & Ts'o, T. (2021). *Uuid: Tools for generating and handling of UUIDs*. Retrieved
536 from <https://CRAN.R-project.org/package=uuid>
- 537 Ushey, K. (2018). *Sourcetools: Tools for reading, tokenizing and parsing r code*. Retrieved
538 from <https://CRAN.R-project.org/package=sourcetools>
- 539 Ushey, K. (2022). *Renv: Project environments*. Retrieved from [https://CRAN.R-project.or](https://CRAN.R-project.org/package=renv)
540 [g/package=renv](https://CRAN.R-project.org/package=renv)
- 541 Vaidyanathan, R., Xie, Y., Allaire, J., Cheng, J., Sievert, C., & Russell, K. (2021). *Htmlwid-*
542 *gets: HTML widgets for r*. Retrieved from <https://CRAN.R-project.org/package=html>
543 [widgets](https://CRAN.R-project.org/package=html)
- 544 van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by
545 chained equations in r. *Journal of Statistical Software*, 45(3), 1–67. [https://doi.org/10.1](https://doi.org/10.18637/jss.v045.i03)
546 [8637/jss.v045.i03](https://doi.org/10.18637/jss.v045.i03)
- 547 van den Boogaart, K. G. (2020). *tensorA: Advanced tensor arithmetic with named indices*.
548 Retrieved from <https://CRAN.R-project.org/package=tensorA>
- 549 Vanderkam, D., Allaire, J., Owen, J., Gromer, D., & Thieurmél, B. (2018). *Dygraphs:*
550 *Interface to 'dygraphs' interactive time series charting library*. Retrieved from <https://CRAN.R-project.org/package=dygraphs>
551 <https://CRAN.R-project.org/package=dygraphs>
- 552 Vaughan, D. (2021). *Tzdb: Time zone database information*. Retrieved from [https://CRAN](https://CRAN.R-project.org/package=tzdb)
553 [.R-project.org/package=tzdb](https://CRAN.R-project.org/package=tzdb)
- 554 Vehtari, A., Gabry, J., Magnusson, M., Yao, Y., Bürkner, P.-C., Paananen, T., & Gelman,

- 555 A. (2020). *Loo: Efficient leave-one-out cross-validation and WAIC for bayesian models*.
 556 Retrieved from <https://mc-stan.org/loo/>
- 557 Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical bayesian model evaluation using
 558 leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27, 1413–1432.
 559 <https://doi.org/10.1007/s11222-016-9696-4>
- 560 Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Bürkner, P.-C. (2021). Rank-
 561 normalization, folding, and localization: An improved rhat for assessing convergence of
 562 MCMC (with discussion). *Bayesian Analysis*.
- 563 Warnes, G. R., Bolker, B., & Lumley, T. (2021). *Gtools: Various r programming tools*.
 564 Retrieved from <https://CRAN.R-project.org/package=gtools>
- 565 Wickham, C. (2018). *Munsell: Utilities for using munsell colours*. Retrieved from <https://CRAN.R-project.org/package=munsell>
- 566 Wickham, H. (2007a). Reshaping data with the reshape package. *Journal of Statistical*
 567 *Software*, 21(12). Retrieved from <http://www.jstatsoft.org/v21/i12/paper>
- 568 Wickham, H. (2007b). Reshaping data with the reshape package. *Journal of Statistical*
 569 *Software*, 21(12), 1–20. Retrieved from <http://www.jstatsoft.org/v21/i12/>
- 570 Wickham, H. (2011a). Testthat: Get started with testing. *The R Journal*, 3, 5–10. Retrieved
 571 from https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf
- 572 Wickham, H. (2011b). The split-apply-combine strategy for data analysis. *Journal of*
 573 *Statistical Software*, 40(1), 1–29. Retrieved from <http://www.jstatsoft.org/v40/i01/>
- 574 Wickham, H. (2019a). *Assertthat: Easy pre and post assertions*. Retrieved from <https://CRAN.R-project.org/package=assertthat>
- 575 Wickham, H. (2019b). *Lazyeval: Lazy (non-standard) evaluation*. Retrieved from <https://CRAN.R-project.org/package=lazyeval>
- 576 Wickham, H. (2021a). *Blob: A simple S3 class for representing vectors of binary data*
 577 *(‘BLOBS’)*. Retrieved from <https://CRAN.R-project.org/package=blob>
- 578 Wickham, H. (2021b). *Ellipsis: Tools for working with ...*. Retrieved from <https://CRAN.R-project.org/package=ellipsis>
- 579 Wickham, H. (2021c). *Waldo: Find differences between r objects*. Retrieved from <https://CRAN.R-project.org/package=waldo>
- 580 Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., ... Yutani,
 581 H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1686.
 582 <https://doi.org/10.21105/joss.01686>
- 583 Wickham, H., Chang, W., Hester, J., & Henry, L. (2021). *Pkgload: Simulate package*
 584 *installation and attach*. Retrieved from <https://CRAN.R-project.org/package=pkgload>
- 585 Wickham, H., Henry, L., & Vaughan, D. (2021). *Vctrs: Vector helpers*. Retrieved from
 586 <https://CRAN.R-project.org/package=vctrs>
- 587 Wickham, H., Hester, J., Chang, W., Müller, K., & Cook, D. (2021). *Memoise: ‘Memoisation’*
 588 *of functions*. Retrieved from <https://CRAN.R-project.org/package=memoise>
- 589 Wickham, H., Hester, J., & Csárdi, G. (2021). *Pkgbuild: Find tools needed to build r packages*.
 590 Retrieved from <https://CRAN.R-project.org/package=pkgbuild>
- 591 Wickham, H., Kuhn, M., & Vaughan, D. (2022). *Generics: Common S3 generics not*
 592 *provided by base r methods related to model fitting*. Retrieved from <https://CRAN.R-project.org/package=generics>
- 593 Wickham, H., & Pedersen, T. L. (2019). *Gtable: Arrange ‘grobs’ in tables*. Retrieved from

- 600 <https://CRAN.R-project.org/package=gtable>
- 601 Wickham, H., & Seidel, D. (2020). *Scales: Scale functions for visualization*. Retrieved from
- 602 <https://CRAN.R-project.org/package=scales>
- 603 Wickham, H., & Xie, Y. (2022). *Evaluate: Parsing and evaluation tools that provide more*
- 604 *details than the default*. Retrieved from <https://CRAN.R-project.org/package=evaluate>
- 605 Wilke, C. O. (2021). *Ggribes: Ridgeline plots in 'ggplot2'*. Retrieved from [https://CRAN.R-](https://CRAN.R-project.org/package=ggribes)
- 606 [project.org/package=ggribes](https://CRAN.R-project.org/package=ggribes)
- 607 Wilke, C. O., & Pedersen, T. L. (2021). *Isoband: Generate isolines and isobands from*
- 608 *regularly spaced elevation grids*. Retrieved from [https://CRAN.R-project.org/package=](https://CRAN.R-project.org/package=isoband)
- 609 [isoband](https://CRAN.R-project.org/package=isoband)
- 610 Wolodzko, T. (2020). *extraDistr: Additional univariate and multivariate distributions*.
- 611 Retrieved from <https://CRAN.R-project.org/package=extraDistr>
- 612 Xie, Y. (2014). Knitr: A comprehensive tool for reproducible research in R. In V. Stodden,
- 613 F. Leisch, & R. D. Peng (Eds.), *Implementing reproducible computational research*.
- 614 Chapman; Hall/CRC. Retrieved from [http://www.crcpress.com/product/isbn/97814665](http://www.crcpress.com/product/isbn/9781466561595)
- 615 [61595](http://www.crcpress.com/product/isbn/9781466561595)
- 616 Xie, Y. (2015). *Dynamic documents with R and knitr* (2nd ed.). Boca Raton, Florida:
- 617 Chapman; Hall/CRC. Retrieved from <https://yihui.org/knitr/>
- 618 Xie, Y. (2016). *Bookdown: Authoring books and technical documents with R markdown*.
- 619 Boca Raton, Florida: Chapman; Hall/CRC. Retrieved from [https://bookdown.org/yihui](https://bookdown.org/yihui/bookdown)
- 620 [/bookdown](https://bookdown.org/yihui/bookdown)
- 621 Xie, Y. (2019). TinyTeX: A lightweight, cross-platform, and easy-to-maintain LaTeX
- 622 distribution based on TeX live. *TUGboat*, (1), 30–32. Retrieved from [https://tug.org/](https://tug.org/TUGboat/Contents/contents40-1.html)
- 623 [TUGboat/Contents/contents40-1.html](https://tug.org/TUGboat/Contents/contents40-1.html)
- 624 Xie, Y. (2021a). *Mime: Map filenames to MIME types*. Retrieved from [https://CRAN.R-](https://CRAN.R-project.org/package=mime)
- 625 [project.org/package=mime](https://CRAN.R-project.org/package=mime)
- 626 Xie, Y. (2021b). *Tinytex: Helper functions to install and maintain TeX live, and compile*
- 627 *LaTeX documents*. Retrieved from <https://github.com/yihui/tinytex>
- 628 Xie, Y. (2021c). *Xfun: Supporting functions for packages maintained by 'yihui xie'*. Retrieved
- 629 from <https://CRAN.R-project.org/package=xfun>
- 630 Xie, Y. (2022a). *Bookdown: Authoring books and technical documents with r markdown*.
- 631 Retrieved from <https://github.com/rstudio/bookdown>
- 632 Xie, Y. (2022b). *Knitr: A general-purpose package for dynamic report generation in r*.
- 633 Retrieved from <https://yihui.org/knitr/>
- 634 Xie, Y., Allaire, J. J., & Golemund, G. (2018). *R markdown: The definitive guide*. Boca
- 635 Raton, Florida: Chapman; Hall/CRC. Retrieved from [https://bookdown.org/yihui/rma](https://bookdown.org/yihui/rmarkdown)
- 636 [rkdown](https://bookdown.org/yihui/rmarkdown)
- 637 Xie, Y., Cheng, J., & Tan, X. (2021). *DT: A wrapper of the JavaScript library 'DataTables'*.
- 638 Retrieved from <https://CRAN.R-project.org/package=DT>
- 639 Xie, Y., Dervieux, C., & Riederer, E. (2020). *R markdown cookbook*. Boca Raton, Florida:
- 640 Chapman; Hall/CRC. Retrieved from <https://bookdown.org/yihui/rmarkdown-cookbook>
- 641 Xie, Y., & Qiu, Y. (2021). *Highr: Syntax highlighting for r source code*. Retrieved from
- 642 <https://CRAN.R-project.org/package=highr>
- 643 Yao, Y., Vehtari, A., Simpson, D., & Gelman, A. (2017). Using stacking to average bayesian
- 644 predictive distributions. *Bayesian Analysis*. <https://doi.org/10.1214/17-BA1091>

- 645 Zeileis, A., Fisher, J. C., Hornik, K., Ihaka, R., McWhite, C. D., Murrell, P., ... Wilke,
646 C. O. (2020). colorspace: A toolbox for manipulating and assessing colors and palettes.
647 *Journal of Statistical Software*, 96(1), 1–49. <https://doi.org/10.18637/jss.v096.i01>
- 648 Zeileis, A., & Grothendieck, G. (2005). Zoo: S3 infrastructure for regular and irregular time
649 series. *Journal of Statistical Software*, 14(6), 1–27. <https://doi.org/10.18637/jss.v014.i06>
- 650 Zeileis, A., Hornik, K., & Murrell, P. (2009). Escaping RGBland: Selecting colors for
651 statistical graphics. *Computational Statistics & Data Analysis*, 53(9), 3259–3270. <https://doi.org/10.1016/j.csda.2008.11.033>
- 652