

Supplementary Materials: 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 Appendix 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 Appendix 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 Appendix 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 Appendix 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.

Appendix 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).

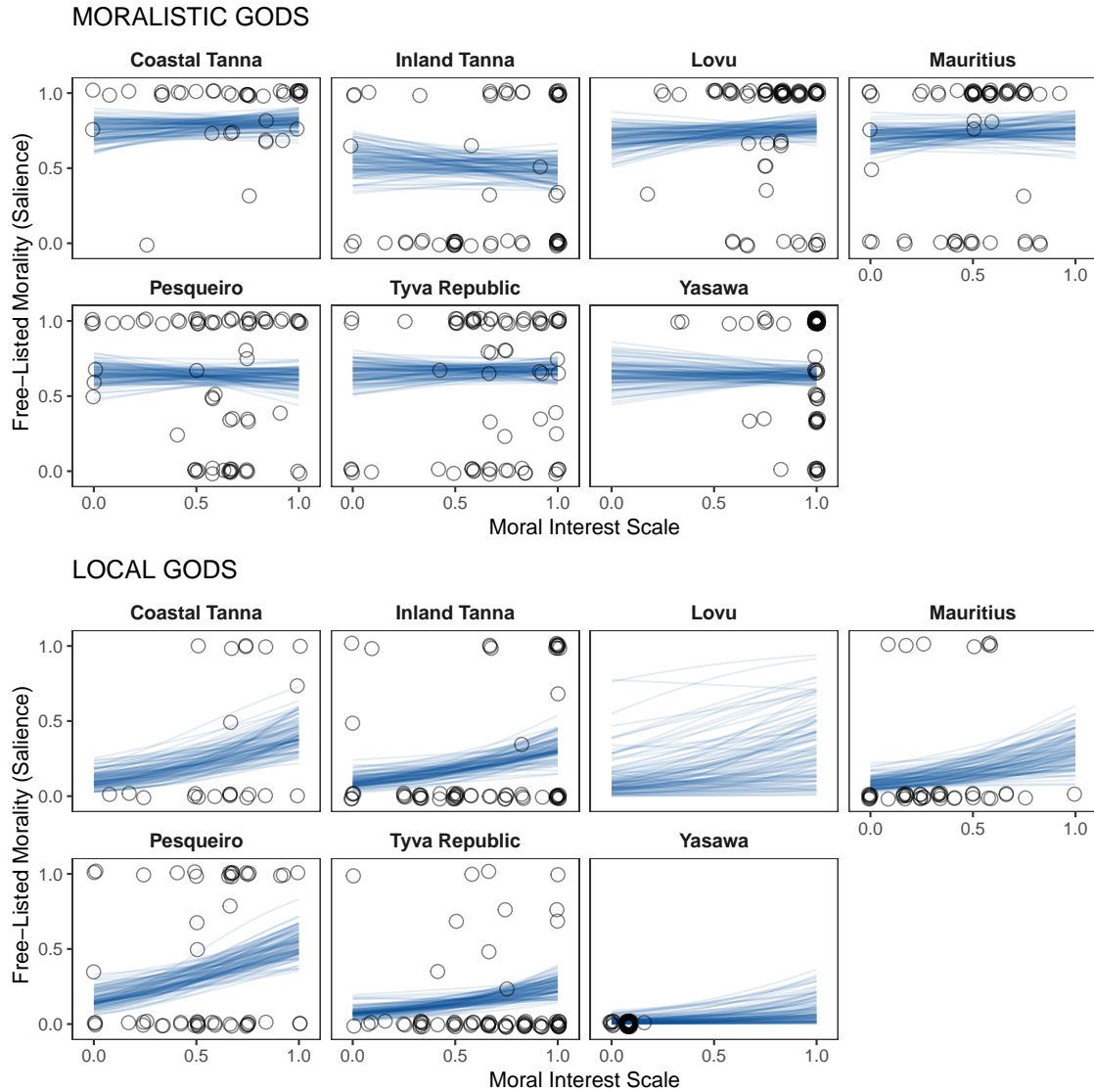


Figure 1. Ordered beta models. Posterior predictions for each field site and each deity from the ordered beta Model 1 and 2 on the relationship between the moral interest scale (x -axis) and saliency of the free-listed general Morality code (y -axis). Lines are draws of expected values from the posterior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

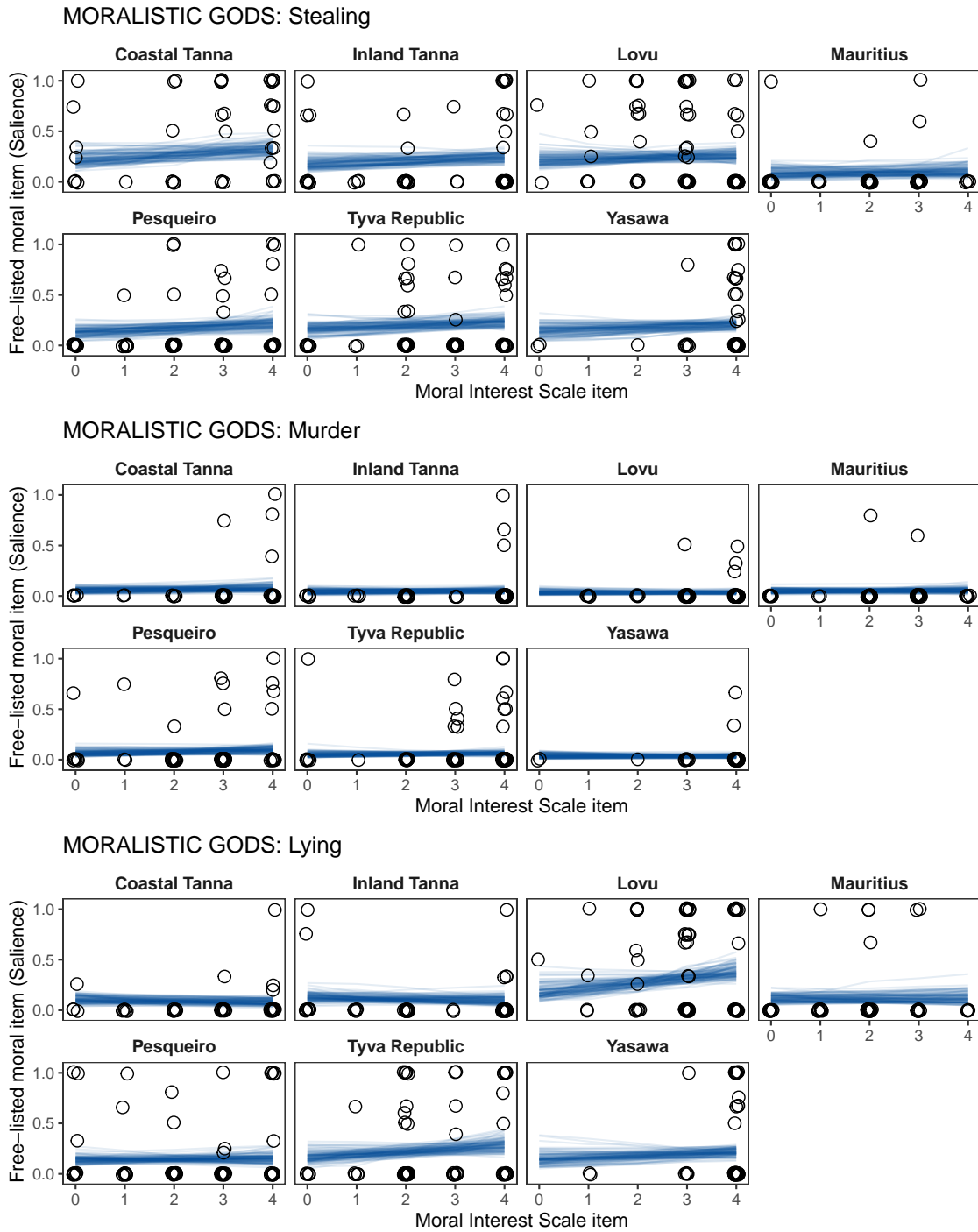


Figure 2. Ordered beta models. Posterior predictions for each field site from the ordered beta Model 3 through 5 on the relationship between the moral interest scale items (x -axis) and salience of the corresponding free-listed code (y -axis). Lines are draws of expected values from the posterior predictive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

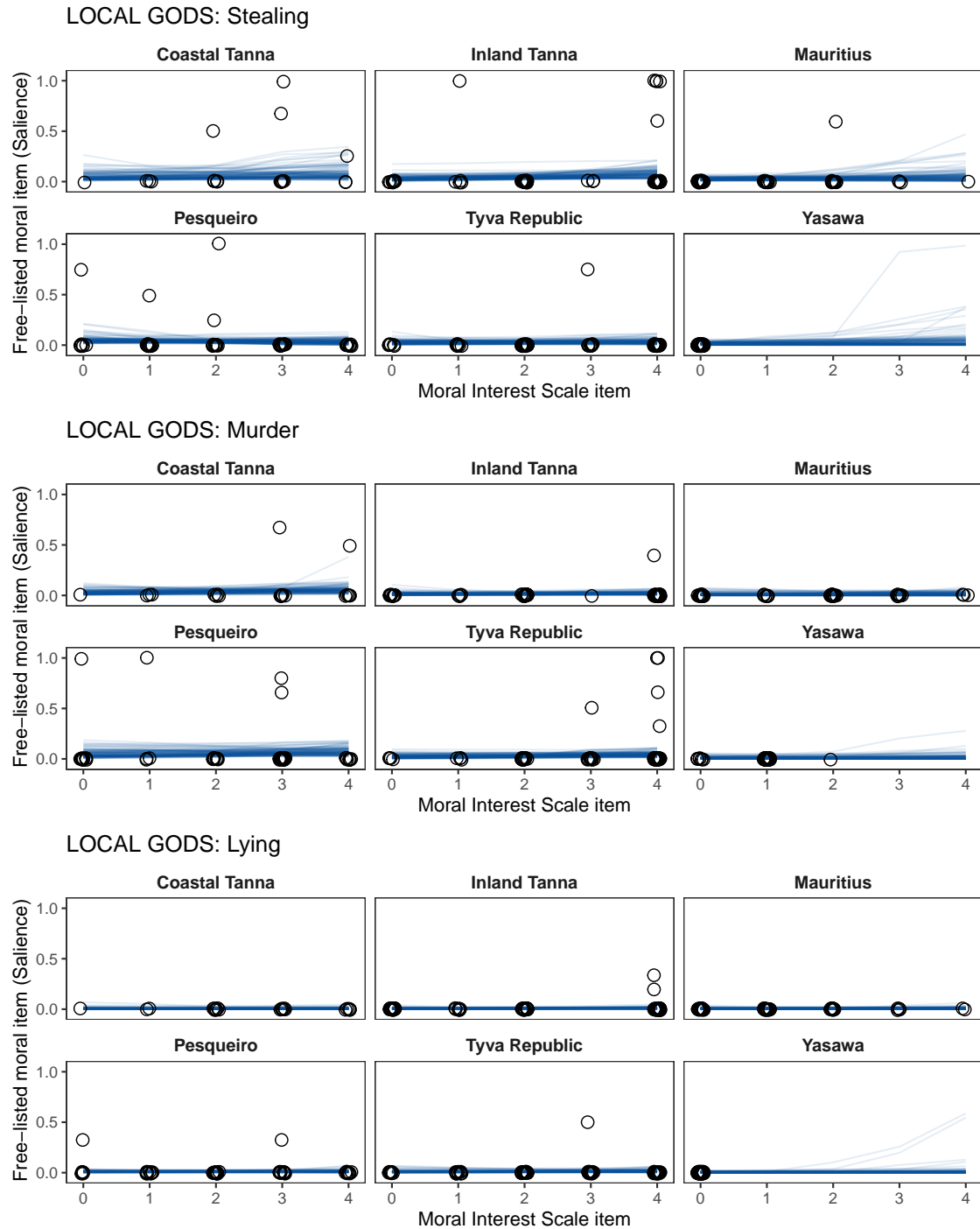


Figure 3. Ordered beta models. Posterior predictions for each field site from ordered beta Model 6 through 8 on the relationship between the moral interest scale items (x -axis) and saliency of the corresponding free-listed code (y -axis). Lines are draws of expected values from the posterior predictive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

Appendix B: Alternative likelihood models for free-list data

Presence/absence

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

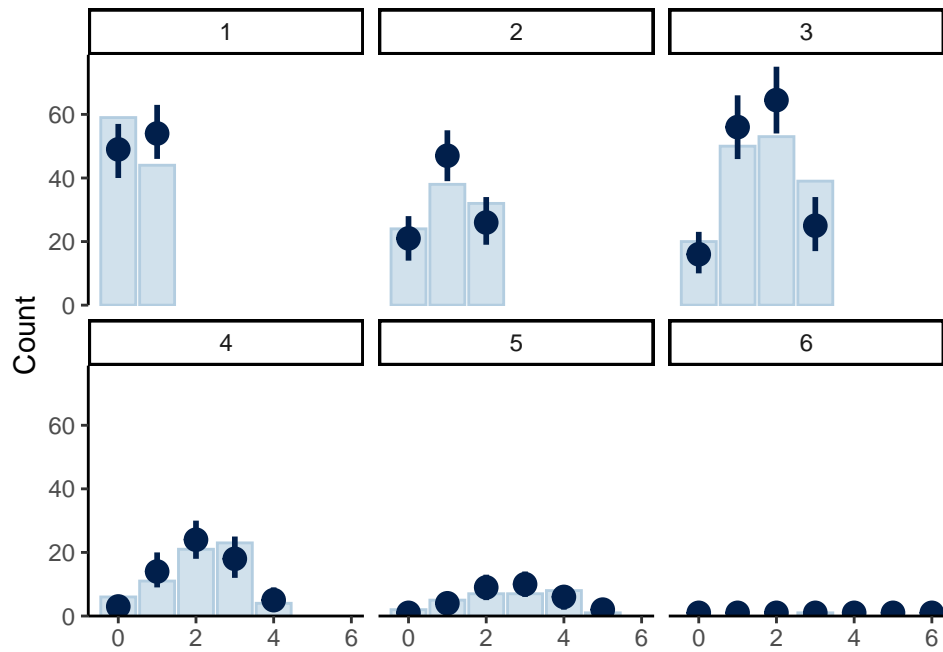


Figure 4. Posterior predictive check for the binomial model. Light blue bars are frequency of each observed outcome while points and intervals are posterior predictions. Panel numbers refer to number of ‘trials’ n , i.e. total number of items listed per participant.

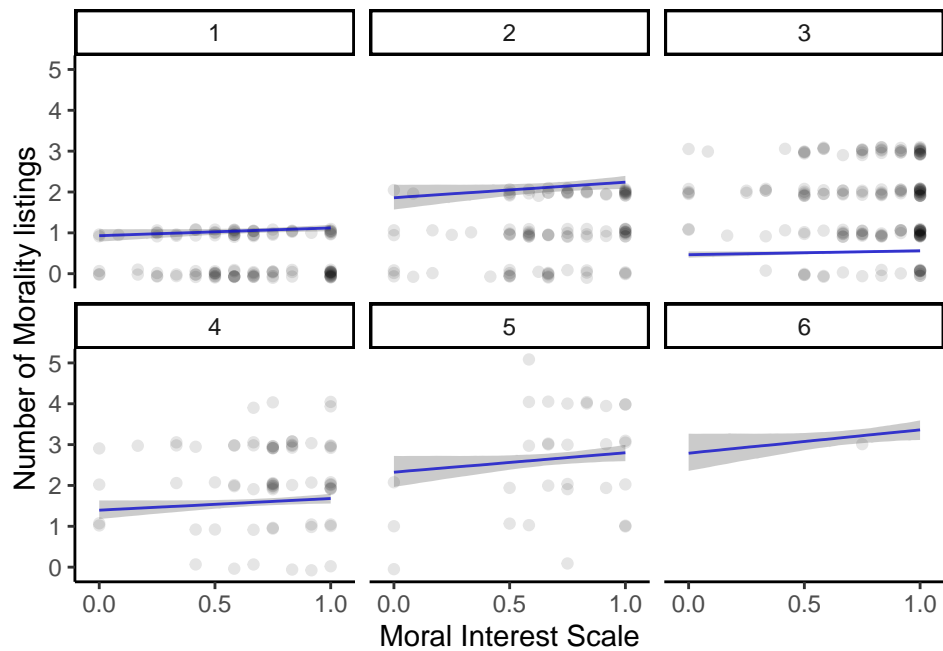


Figure 5. Binomial model. Lines are posterior medians of expected values with 95% interval bands. Raw data points are slightly jittered vertically. Panel numbers refer to number of ‘trials’ n , i.e. total number of items listed per participant.

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 `zi ~ ...` and the negative binomial `y ~`. 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.

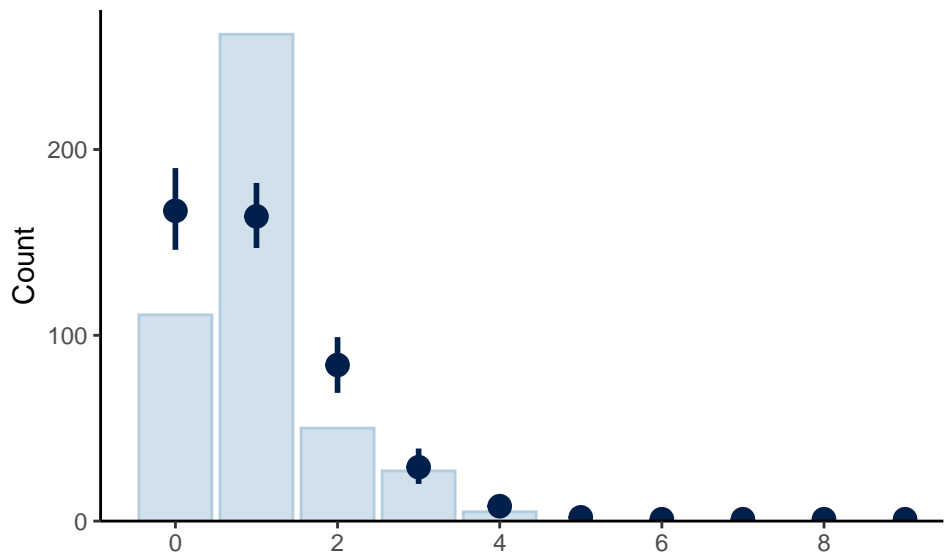


Figure 6. Posterior predictive check for zero-inflated negative binomial. Light blue bars are frequency of each observed outcome while points and intervals are posterior predictions.

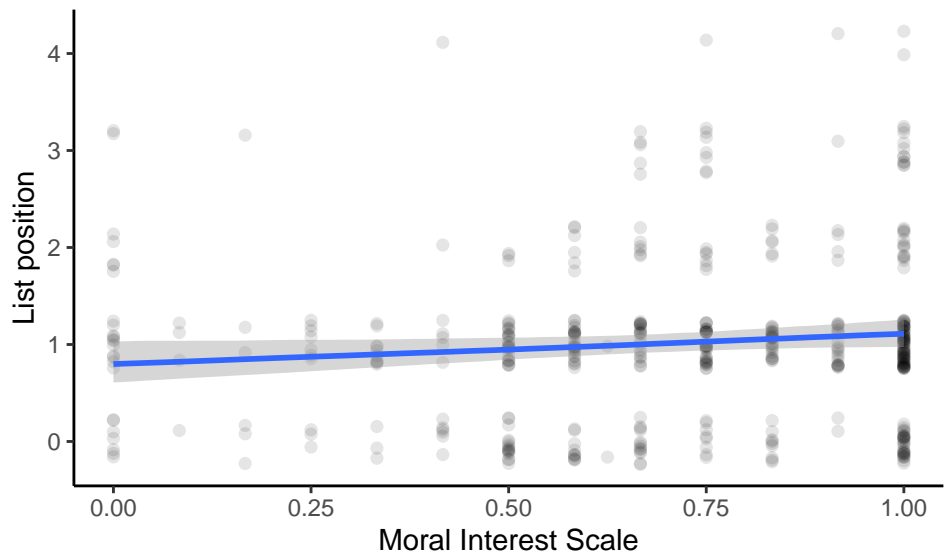


Figure 7. Expected list position according to the zero-inflated negative binomial model. Raw data points are slightly jittered vertically.

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.

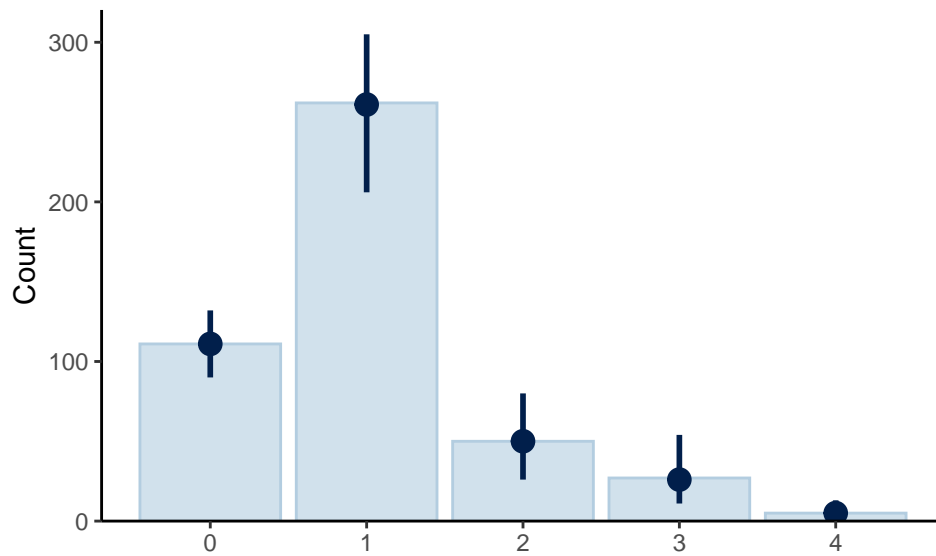


Figure 8. Posterior predictive check for zero-inflated ordinal regression. Light blue bars are frequency of each observed outcome while points and intervals are posterior predictions.

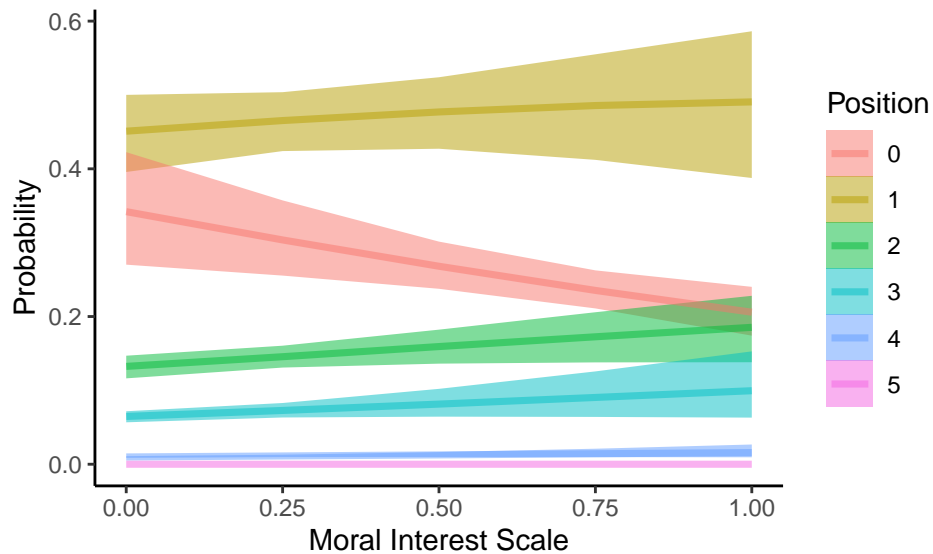


Figure 9. Cumulative probabilities of each list position. 0 = item not listed. Lines are posterior medians and bands are 80% credible intervals.

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.

Appendix 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.

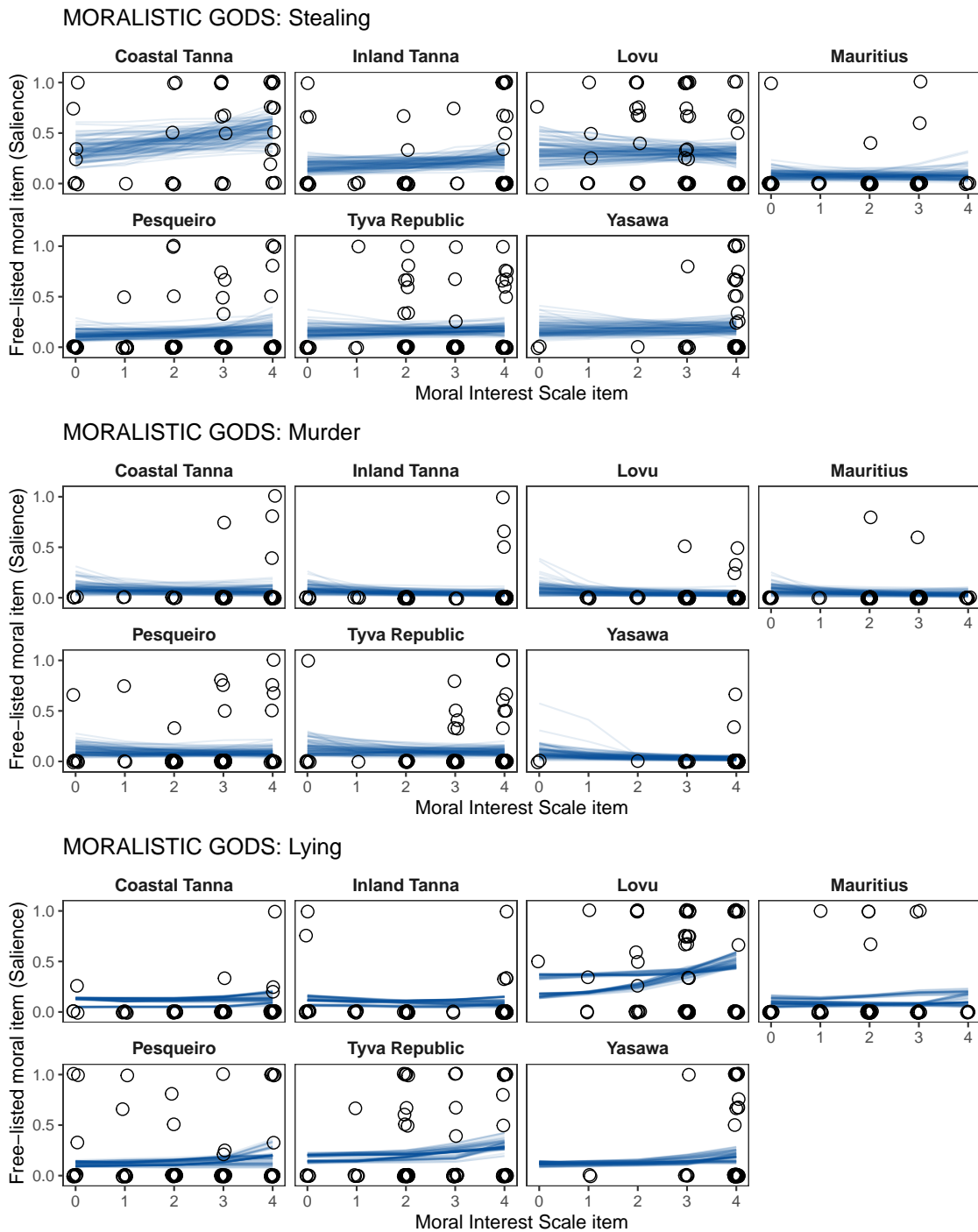


Figure 10. Posterior predictions for each field site from Model 3 through 5 on the relationship between the moral interest scale items (x -axis) and salience of the corresponding free-listed code (y -axis). Lines are draws of expected values from the posterior predictive distributions. Raw data points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

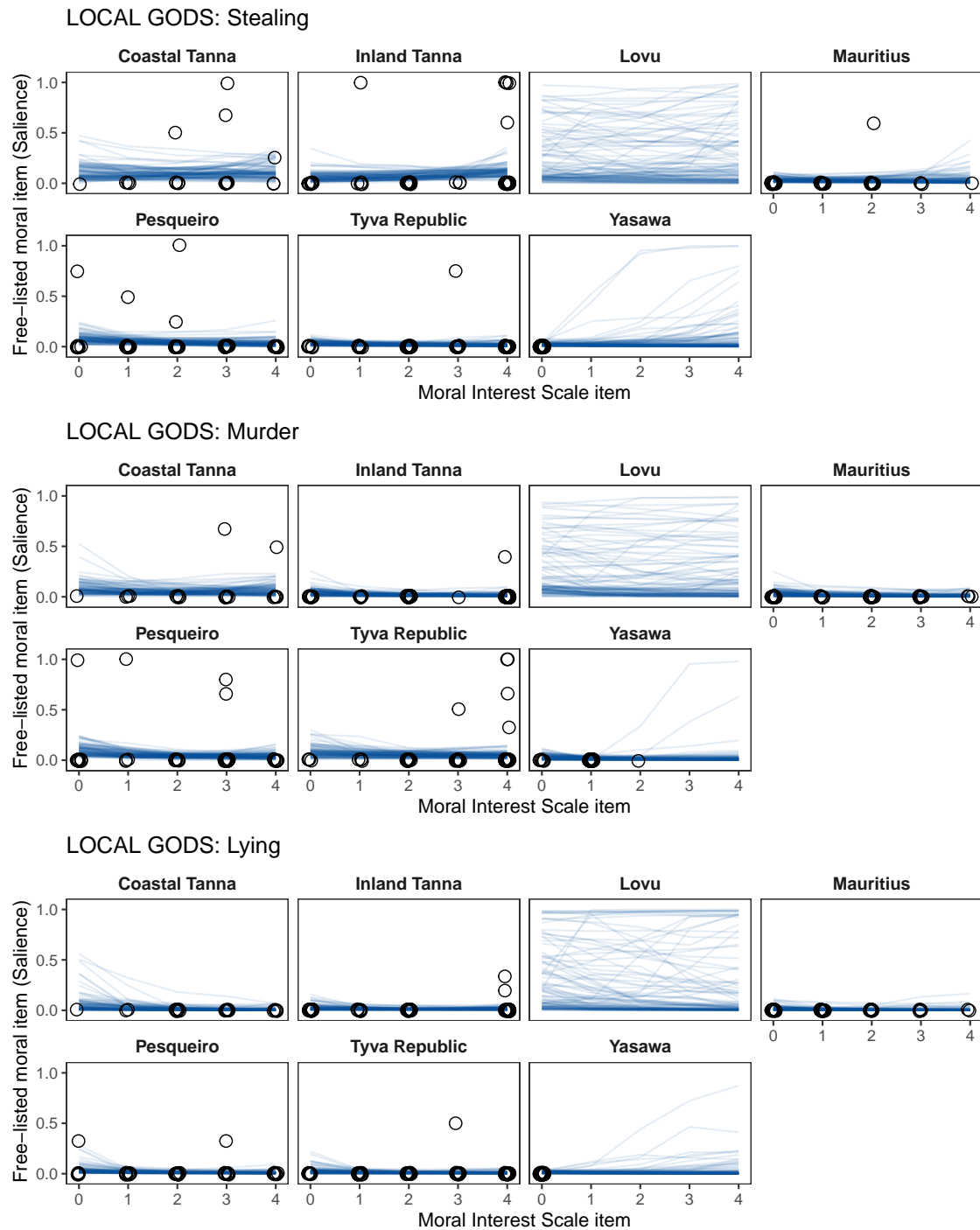


Figure 11. Posterior predictions for each field site from Model 6 through 8 on the relationship between the moral interest scale items (x -axis) and saliency of the corresponding free-listed code (y -axis). Lines are draws of expected values from the posterior predictive distributions. Raw data points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

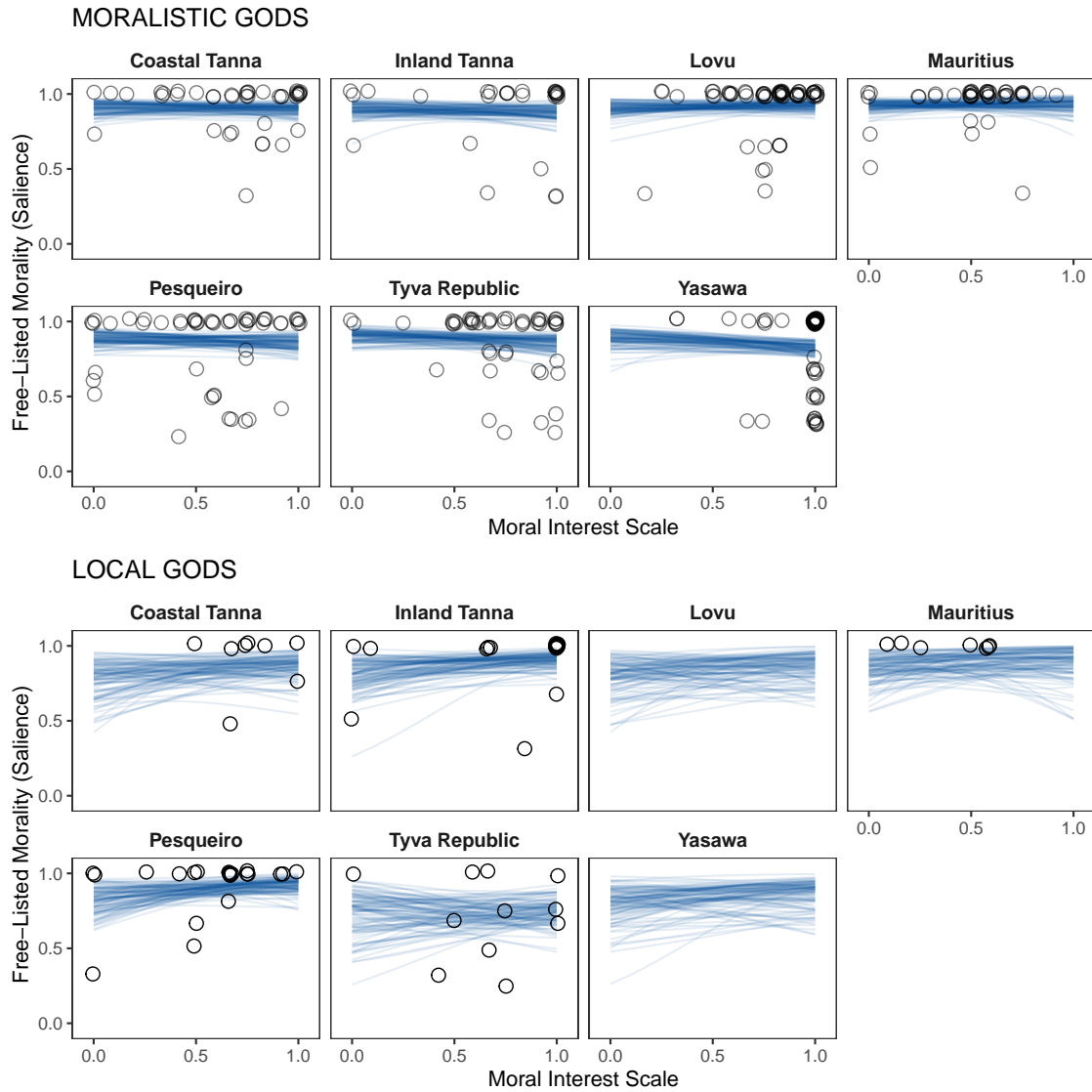


Figure 12. Reduced models. Posterior predictions for each field site from Model 1 and 2 on the relationship between the moral interest scale (x -axis) and saliency of free-listed Morality (y -axis) with only non-zero free-list responses. Lines are draws of expected values from the posterior predictive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

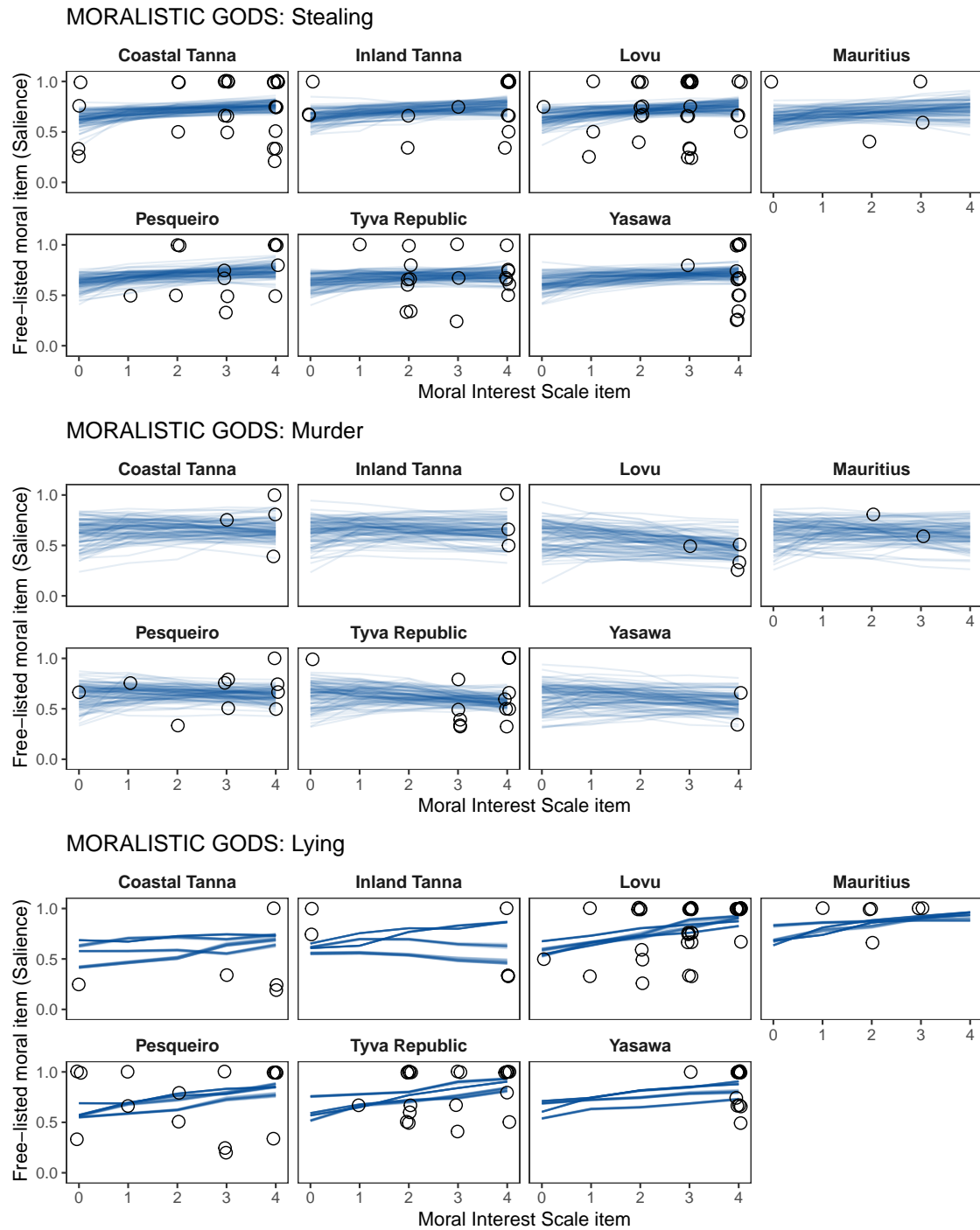


Figure 13. Reduced models. Posterior predictions for each field site from Model 3 through 5 on the relationship between the moral interest scale (x -axis) and saliency of the corresponding free-listed code (y -axis) with only non-zero free-list responses. Lines are draws of expected values from the posterior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

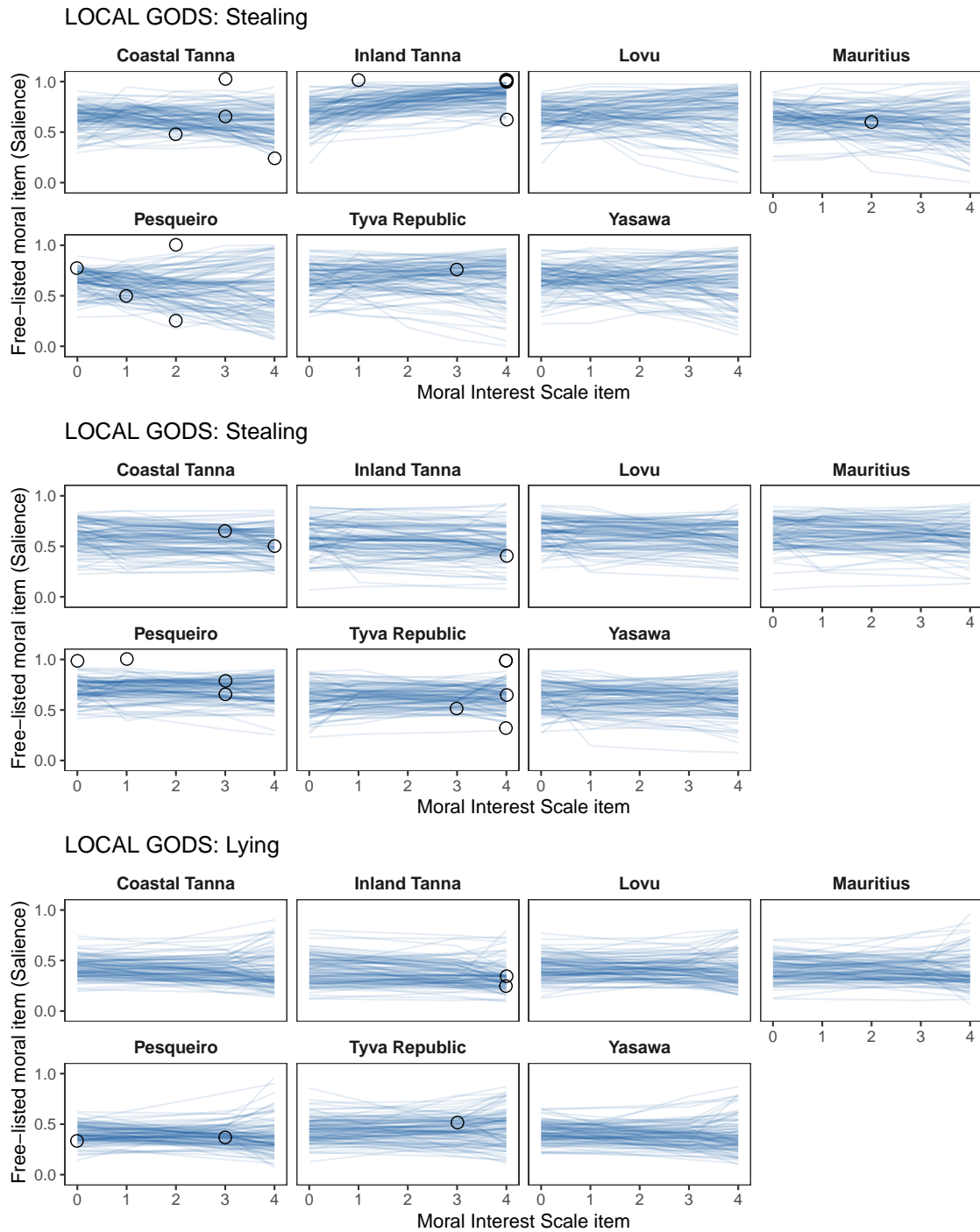


Figure 14. Reduced models. Posterior predictions for each field site from Model 6 through 8 on the relationship between the moral interest scale (x -axis) and saliency of the corresponding free-listed code (y -axis) with only non-zero free-list responses. Lines are draws of expected values from the posterior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

Appendix 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. 0.24 (Xie, 2016, 2021a), 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. 0.14 (H. Wickham & Xie, 2019), 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. 1.37 (Xie, 2014, 2015, 2021b), 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, 2021c), 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 et al., 2022), rematch v. 1.0.1
 221 (Csardi, 2016), rematch2 v. 2.1.2 (Csárdi, 2020), renv v. 0.15.2 (Ushey, 2022), reshape v.
 222 0.8.8 (H. Wickham, 2007a), reshape2 v. 1.4.4 (H. Wickham, 2007b), rmarkdown v. 2.11
 223 (Allaire et al., 2021; Xie, Allaire, & Golemund, 2018; Xie, Dervieux, & Riederer, 2020),
 224 rmdfiltr 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), tinylabels v.
 234 0.2.3 (Barth, 2022), tinytex v. 0.36 (Xie, 2019, 2021d), 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, 2021e), xtable v. 1.8.4 (Dahl, Scott, Roosen, Magnusson, &
 239 Swinton, 2019), xts v. 0.12.1 (Ryan & Ulrich, 2020), yaml v. 2.2.2 (Stephens et al., 2022),
 240 zip v. 2.2.0 (Csárdi, Podgórski, & Geldreich, 2021), zoo v. 1.8.9 (Zeileis & Grothendieck,
 241 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. (2021). *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). *Listenr: Environments behaving (almost) as lists*. Retrieved from <https://CRAN.R-project.org/package=listenr>
- 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). *Parallelly: Enhancing the 'parallel' package*. Retrieved from <https://CRAN.R-project.org/package=parallelly>
- 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>

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

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

- 421 Hester, J., & François, R. (2021). *cpp11: A c++11 interface for r's c interface*. Retrieved
422 from <https://CRAN.R-project.org/package=cpp11>
- 423 Hester, J., Henry, L., Müller, K., Ushey, K., Wickham, H., & Chang, W. (2021). *Withr:*
424 *Run code 'with' temporarily modified global state*. Retrieved from [https://CRAN.R-](https://CRAN.R-project.org/package=withr)
425 [project.org/package=withr](https://CRAN.R-project.org/package=withr)
- 426 Hester, J., Wickham, H., & Bryan, J. (2021). *Vroom: Read and write rectangular text data*
427 *quickly*. Retrieved from <https://CRAN.R-project.org/package=vroom>
- 428 Hester, J., Wickham, H., & Csárdi, G. (2021). *Fs: Cross-platform file system operations*
429 *based on 'libuv'*. Retrieved from <https://CRAN.R-project.org/package=fs>
- 430 Iannone, R. (2021). *Fontawesome: Easily work with 'font awesome' icons*. Retrieved from
431 <https://CRAN.R-project.org/package=fontawesome>
- 432 Johnson, S. G. (?). The NLOpt nonlinear-optimization package. ?, ?(?), ?
- 433 Justin Talbot. (2020). *Labeling: Axis labeling*. Retrieved from [https://CRAN.R-project.or](https://CRAN.R-project.org/package=labeling)
434 [g/package=labeling](https://CRAN.R-project.org/package=labeling)
- 435 Kay, M. (2021). *ggdist: Visualizations of distributions and uncertainty*. [https://doi.org/10.5](https://doi.org/10.5281/zenodo.3879620)
436 [281/zenodo.3879620](https://doi.org/10.5281/zenodo.3879620)
- 437 Kay, M. (2022). *tidybayes: Tidy data and geoms for Bayesian models*. [https://doi.org/10.5](https://doi.org/10.5281/zenodo.1308151)
438 [281/zenodo.1308151](https://doi.org/10.5281/zenodo.1308151)
- 439 Kubinec, R. (2022). Ordered beta regression: A parsimonious, well-fitting model for
440 continuous data with lower and upper bounds. *Political Analysis*. [https://doi.org/10.101](https://doi.org/10.1017/pan.2022.20)
441 [7/pan.2022.20](https://doi.org/10.1017/pan.2022.20)
- 442 Lane, A. J., & Purzycki, B. G. (2016). *AnthroTools: Some custom tools for anthropology*.
- 443 Lang, M. (2017). checkmate: Fast argument checks for defensive r programming. *The R*
444 *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)
445 [2017-028/index.html](https://journal.r-project.org/archive/2017/RJ-2017-028/index.html)
- 446 Lang, M., & R Core Team. (2021). *Backports: Reimplementations of functions introduced*
447 *since r-3.0.0*. Retrieved from <https://CRAN.R-project.org/package=backports>
- 448 Lewis, B. W. (2020). *Threejs: Interactive 3D scatter plots, networks and globes*. Retrieved
449 from <https://CRAN.R-project.org/package=threejs>
- 450 Lincoln, M. (2020). *Clipr: Read and write from the system clipboard*. Retrieved from
451 <https://CRAN.R-project.org/package=clipr>
- 452 Loden, J., Daeschler, D., Rodola', G., & Csárdi, G. (2021). *Ps: List, query, manipulate*
453 *system processes*. Retrieved from <https://CRAN.R-project.org/package=ps>
- 454 Meredith, M., & Kruschke, J. (2020). *HDInterval: Highest (posterior) density intervals*.
455 Retrieved from <https://CRAN.R-project.org/package=HDInterval>
- 456 Müller, K. (2020). *Rprojroot: Finding files in project subdirectories*. Retrieved from
457 <https://CRAN.R-project.org/package=rprojroot>
- 458 Neuwirth, E. (2014). *RColorBrewer: ColorBrewer palettes*. Retrieved from [https://CRAN.R-](https://CRAN.R-project.org/package=RColorBrewer)
459 [project.org/package=RColorBrewer](https://CRAN.R-project.org/package=RColorBrewer)
- 460 O'Hara-Wild, M., Kay, M., & Hayes, A. (2022). *Distributional: Vectorised probability*
461 *distributions*. Retrieved from <https://CRAN.R-project.org/package=distributional>
- 462 Oehlschlägel, J., & Ripley, B. (2020). *Bit: Classes and methods for fast memory-efficient*
463 *boolean selections*. Retrieved from <https://CRAN.R-project.org/package=bit>
- 464 Oehlschlägel, J., & Silvestri, L. (2020). *bit64: A S3 class for vectors of 64bit integers*.
465 Retrieved from <https://CRAN.R-project.org/package=bit64>

- Ooms, J. (2018). *Commonmark: High performance CommonMark and github markdown rendering in r*. Retrieved from <https://CRAN.R-project.org/package=commonmark>
- Ooms, J. (2019). *Askpass: Safe password entry for r, git, and SSH*. Retrieved from <https://CRAN.R-project.org/package=askpass>
- Ooms, J. (2020). *Sys: Powerful and reliable tools for running system commands in r*. Retrieved from <https://CRAN.R-project.org/package=sys>
- Ooms, J. (2021a). *Curl: A modern and flexible web client for r*. Retrieved from <https://CRAN.R-project.org/package=curl>
- Ooms, J. (2021b). *Openssl: Toolkit for encryption, signatures and certificates based on OpenSSL*. Retrieved from <https://CRAN.R-project.org/package=openssl>
- Pedersen, T. L. (2020). *Patchwork: The composer of plots*. Retrieved from <https://CRAN.R-project.org/package=patchwork>
- Pedersen, T. L., Nicolae, B., & François, R. (2021). *Farver: High performance colour space manipulation*. Retrieved from <https://CRAN.R-project.org/package=farver>
- Perry, P. O. (2021). *utf8: Unicode text processing*. Retrieved from <https://CRAN.R-project.org/package=utf8>
- Plate, T., & Heiberger, R. (2016). *Abind: Combine multidimensional arrays*. Retrieved from <https://CRAN.R-project.org/package=abind>
- Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1), 7–11. Retrieved from <https://journal.r-project.org/archive/>
- Potter, S. (2012). *Introducing the selectr package*. Auckland, New Zealand: The University of Auckland. Retrieved from The University of Auckland website: <http://stattech.wordpress.fos.auckland.ac.nz/2012-10-introducing-the-selectr-package/>
- Purzycki, B. G., & Bendixen, T. (2020). Examining Values, Virtues, and Tradition in the Tyva Republic with Free-List and Demographic Data. *The New Research in Tuva*, 4, 6–18.
- Purzycki, B. G., Stagnaro, M. N., & Sasaki, J. (2020). Breaches of Trust Change the Content and Structure of Religious Appeals. *Journal for the Study of Religion, Nature and Culture*, 14(1), 71–94. <https://doi.org/10.1558/jsrnc.38786>
- R Core Team. (2021). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from <https://www.R-project.org/>
- R Special Interest Group on Databases (R-SIG-DB), Wickham, H., & Müller, K. (2021). *DBI: R database interface*. Retrieved from <https://CRAN.R-project.org/package=DBI>
- Ratnakumar, S., Mick, T., & Davis, T. (2021). *Rappdirs: Application directories: Determine where to save data, caches, and logs*. Retrieved from <https://CRAN.R-project.org/package=rappdirs>
- Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., & Müller, M. (2011). pROC: An open-source package for r and s+ to analyze and compare ROC curves. *BMC Bioinformatics*, 12, 77.
- Rodríguez-Sánchez, F., Jackson, C. P., & Hutchins, S. D. (2022). *Grateful: Facilitate citation of r packages*. Retrieved from <https://github.com/Pakillo/grateful>
- Ryan, J. A., & Ulrich, J. M. (2020). *Xts: eXtensible time series*. Retrieved from <https://CRAN.R-project.org/package=xts>

- Schloerke, B., Cook, D., Larmarange, J., Briatte, F., Marbach, M., Thoen, E., ... Crowley, J. (2021). *GGally: Extension to 'ggplot2'*. Retrieved from <https://CRAN.R-project.org/package=GGally>
- Sievert, C., & Cheng, J. (2021a). *Bslib: Custom 'bootstrap' 'sass' themes for 'shiny' and 'rmarkdown'*. Retrieved from <https://CRAN.R-project.org/package=bslib>
- Sievert, C., & Cheng, J. (2021b). *Jquerylib: Obtain 'jQuery' as an HTML dependency object*. Retrieved from <https://CRAN.R-project.org/package=jquerylib>
- Sklyar, O., Murdoch, D., Smith, M., Eddelbuettel, D., Francois, R., Soetaert, K., & Ranke, J. (2021). *Inline: Functions to inline c, c++, fortran function calls from r*. Retrieved from <https://CRAN.R-project.org/package=inline>
- Stan Development Team. (2020). *StanHeaders: Headers for the R interface to Stan*. Retrieved from <https://mc-stan.org/>
- Stan Development Team. (2021). *RStan: The R interface to Stan*. Retrieved from <https://mc-stan.org/>
- Stauffer, R., Mayr, G. J., Dabernig, M., & Zeileis, A. (2009). Somewhere over the rainbow: How to make effective use of colors in meteorological visualizations. *Bulletin of the American Meteorological Society*, 96(2), 203–216. <https://doi.org/10.1175/BAMS-D-13-00155.1>
- Stephens, J., Simonov, K., Xie, Y., Dong, Z., Wickham, H., Horner, J., ... Kamvar, Z. N. (2022). *Yaml: Methods to convert r data to YAML and back*. Retrieved from <https://CRAN.R-project.org/package=yaml>
- Therneau, T. (2020). *Bdsmatrix: Routines for block diagonal symmetric matrices*. Retrieved from <https://CRAN.R-project.org/package=bdsmatrix>
- Urbanek, S. (2015). *base64enc: Tools for base64 encoding*. Retrieved from <https://CRAN.R-project.org/package=base64enc>
- Urbanek, S., & Ts'o, T. (2021). *Uuid: Tools for generating and handling of UUIDs*. Retrieved from <https://CRAN.R-project.org/package=uuid>
- Ushey, K. (2018). *Sourcetools: Tools for reading, tokenizing and parsing r code*. Retrieved from <https://CRAN.R-project.org/package=sourcetools>
- Ushey, K. (2022). *Renv: Project environments*. Retrieved from <https://CRAN.R-project.org/package=renv>
- Vaidyanathan, R., Xie, Y., Allaire, J., Cheng, J., Sievert, C., & Russell, K. (2021). *Htmlwidgets: HTML widgets for r*. Retrieved from <https://CRAN.R-project.org/package=htmlwidgets>
- van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3), 1–67. <https://doi.org/10.18637/jss.v045.i03>
- van den Boogaart, K. G. (2020). *tensorA: Advanced tensor arithmetic with named indices*. Retrieved from <https://CRAN.R-project.org/package=tensorA>
- Vanderkam, D., Allaire, J., Owen, J., Gromer, D., & Thieurmel, B. (2018). *Dygraphs: Interface to 'dygraphs' interactive time series charting library*. Retrieved from <https://CRAN.R-project.org/package=dygraphs>
- Vaughan, D. (2021). *Tzdb: Time zone database information*. Retrieved from <https://CRAN.R-project.org/package=tzdb>
- Vehtari, A., Gabry, J., Magnusson, M., Yao, Y., Bürkner, P.-C., Paananen, T., & Gelman,

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

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

- 646 Zeileis, A., Fisher, J. C., Hornik, K., Ihaka, R., McWhite, C. D., Murrell, P., ... Wilke,
647 C. O. (2020). colorspace: A toolbox for manipulating and assessing colors and palettes.
648 *Journal of Statistical Software*, 96(1), 1–49. <https://doi.org/10.18637/jss.v096.i01>
- 649 Zeileis, A., & Grothendieck, G. (2005). Zoo: S3 infrastructure for regular and irregular time
650 series. *Journal of Statistical Software*, 14(6), 1–27. <https://doi.org/10.18637/jss.v014.i06>
- 651 Zeileis, A., Hornik, K., & Murrell, P. (2009). Escaping RGBland: Selecting colors for
652 statistical graphics. *Computational Statistics & Data Analysis*, 53(9), 3259–3270. <https://doi.org/10.1016/j.csda.2008.11.033>
- 653