

# Supplementary Material

for the article

Bølstad, Jørgen, “Hierarchical Bayesian Aldrich-McKelvey Scaling”, *Political Analysis*.

## Contents

1	Additional Monte Carlo Results	2
2	Examination of Irregular Response Patterns	4
3	Analysis of ANES 1980	7
4	Code for Generating Data in R	13
5	Model Codes in Stan	14
6	Prior Predictive Simulation	19

The R package accompanying the article is available at:

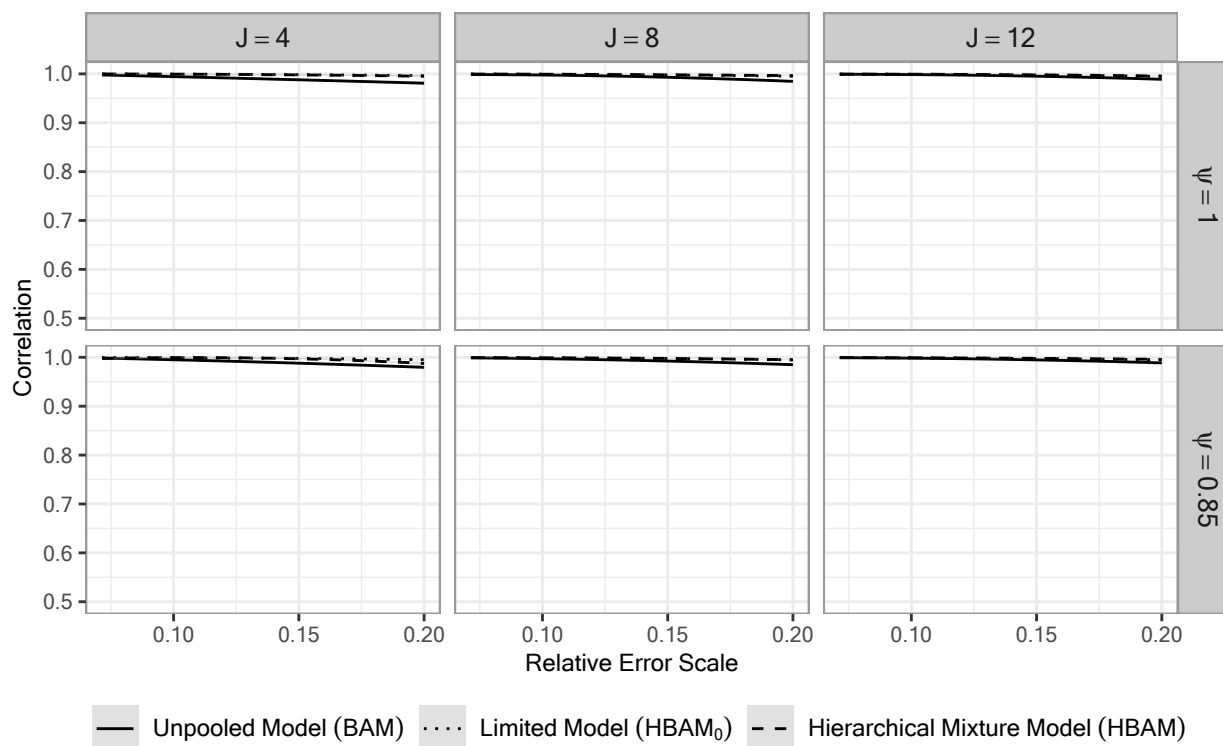
<https://cran.r-project.org/package=hbamr>

The development version of the package is available at:

<https://github.com/jbolstad/hbamr>

# 1 Additional Monte Carlo Results

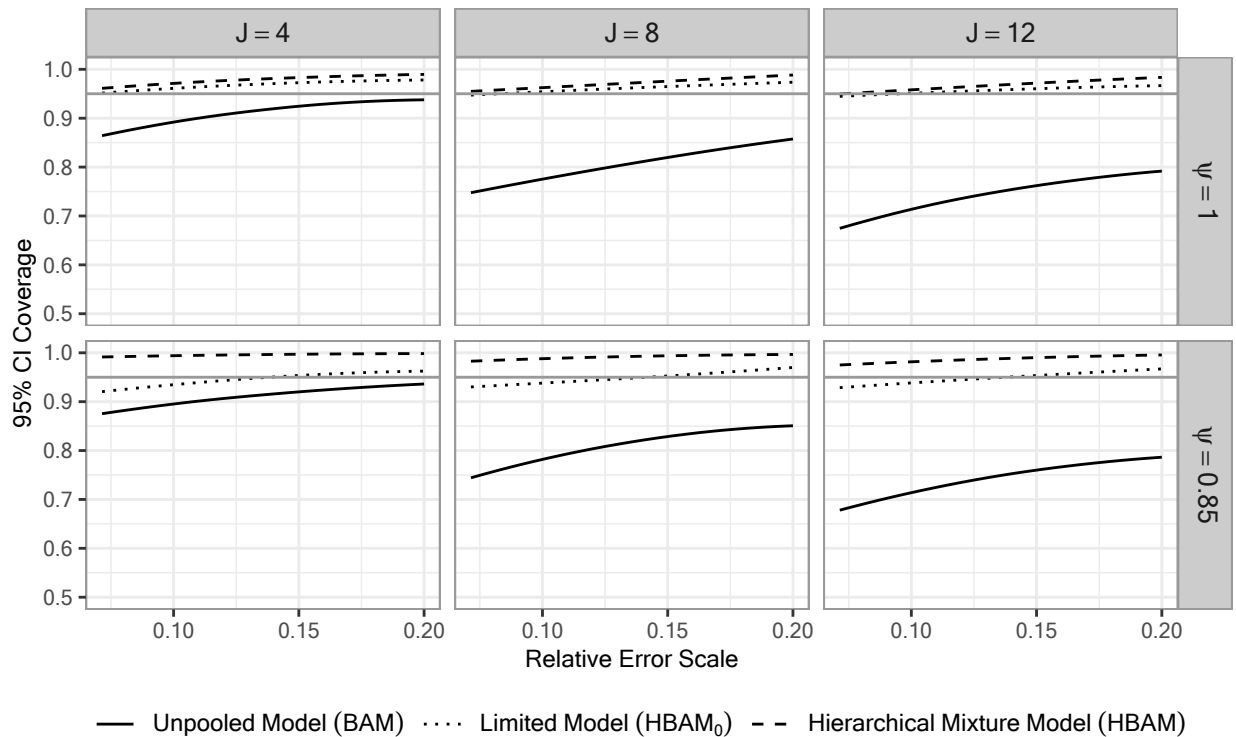
Figure S1 is based on the same Monte Carlo simulations as Figure 7 in the article, but reports results for stimulus positions instead of respondent positions. The correlations between estimated and true stimulus positions are strong for all models across all scenarios. However, the BAM model still performs somewhat worse than the two other models, especially as the scale of the errors increases. The HBAM<sub>0</sub> and HBAM models display very similar performance across all realistic scenarios.



**Figure S1:** Results for stimulus positions in Monte Carlo simulations.

Note: The lines show loess smoothed curves based on 1000 trials per panel.  $J$  is the number of stimuli,  $\psi$  is the probability that respondents do *not* flip the scale, and the relative error scale is  $\tau$  divided by the scale length.

Figure S2 shows the proportion of true respondent positions that are within the 95% credible intervals produced by each model. The BAM model tends to underestimate the uncertainty in its estimates, covering the true parameters in fewer than 95% of the cases. (As discussed in the article, the BAM model also produces some posterior distributions that are far too wide, and the coverage would be lower if these cases were excluded.) The underestimation of uncertainty is likely the result of assuming the self-placements to be made without error. In contrast, the HBAM model covers the true parameters in more than 95% of the cases, especially in scenarios where some respondents flip the scale. This is likely the result of uncertainty regarding whether respondents have flipped the scale or not – making the credible intervals wider.



**Figure S2:** Coverage of true respondent positions in Monte Carlo simulations.

Note: The lines show loess smoothed curves based on 1000 trials per panel.  $J$  is the number of stimuli,  $\psi$  is the probability that respondents do *not* flip the scale, and the relative error scale is  $\tau$  divided by the scale length.

## 2 Examination of Irregular Response Patterns

The BAM and HBAM models assume the answers from each survey respondent follow a meaningful pattern: The respondents may stretch, shift, and flip the generally agreed positions of the stimuli, but their answers still retain some of the original information. An interesting question is what happens when this is not the case – when respondents provide valid survey responses that bear little or no relation to the stimuli positions estimated by the models. More specifically, it is useful to know whether the models differ in how they handle such response patterns.

This section therefore provides data and detailed results for a selection of respondents in the ANES 2012 dataset. In particular, the tables below contain information for the three respondents with the largest estimated error-variances from each model.<sup>1</sup> Simply put, these are the respondents for which the models fit the worst (in the sense that the most variance remains unexplained). The first three rows of each table below contain the three respondents with the largest errors according to the HBAM model, while the last three rows contain the respondents with the largest errors according to the BAM model.

Table S1 shows the data for the selected respondents. The stimuli are ordered by their estimated ideological ranking, from leftmost to rightmost, and an accurate set of answers would be expected to approximate a similar order – increasing as we move from left to right. In these worst-fit cases, such order is often hard to find. Respondent 6001, for instance, places both the Democratic party and Mitt Romney on the far right, while placing both the Republican party and Barack Obama on the far left. It is not surprising that the models do not fit well in this case.

Tables S2 and S3 summarize the estimates from each model for these respondents. As we would expect, there is generally more variation and more uncertainty in the estimates from the BAM model. While both models tend to make only modest adjustments to the self-placements, a notable difference between can be seen in their estimates of scale-flipping. The BAM model estimates respondents 6001, 4316 and 6309 to have flipped the scale with a probability of 98-99%, while the HBAM model estimates flipping-parameters of .64, suggesting that these respondents most likely have *not* flipped the scale (while also implying a notable probability that they have done so). For the respondent 6309, who has an off-center position at +2, the BAM model produces an estimated position at the other side of the center, while the HBAM model does not.

---

<sup>1</sup>I am excluding respondents with exactly similar responses to a respondent who is already in the table, as the results tend to be very similar for such respondents.

**Table S1:** Data for six respondents with irregular response patterns

ID	Self	Democrats	Obama	Romney	Republicans
965	0	0	-3	3	0
3502	0	-3	0	0	3
5136	3	0	-3	2	0
6001	0	3	-3	3	-3
4316	0	3	-3	3	-2
6309	2	3	-3	2	-3

Note: The table shows data for the three respondents with the largest errors according to the HBAM model (first three rows) as well as the three with the largest errors according to the BAM model (last three rows). The stimuli have been ordered by their estimated ideological ranking, from leftmost to rightmost.

**Table S2:** BAM-results for six respondents with irregular response patterns

ID	$\alpha$			$\beta$				$\chi$			mean
	2.5%	50%	97.5%	2.5%	50%	97.5%	$> 0$	2.5%	50%	97.5%	
965	-1.06	0.00	1.07	-0.79	0.27	1.65	0.73	-7.12	-0.02	8.83	0.19
3502	-0.98	0.01	0.97	1.86	3.08	4.06	1.00	-0.35	-0.00	0.37	0.01
5136	-0.94	-0.05	0.87	-0.75	0.22	1.35	0.71	-90.93	5.29	106.39	11.12
6001	-1.89	-0.01	2.02	-4.85	-2.80	-0.22	0.02	-1.20	-0.00	1.31	0.05
4316	-1.46	0.43	2.36	-4.23	-2.28	-0.01	0.02	-1.46	0.18	2.24	0.17
6309	-1.84	-0.07	1.74	-4.73	-2.86	-0.56	0.01	-2.60	-0.71	0.02	-0.61

Note: The table summarizes marginal posterior distributions for key individual-level parameters in terms of quantiles and means.  $\alpha$ 's are shift-parameters,  $\beta$ 's are stretch-parameters, and  $\chi$ 's are latent respondent positions. The column " $\beta > 0$ " reports the share of the draws for  $\beta_i$  that are above zero, which indicates the probability that a respondent has *not* flipped the scale.

**Table S3:** HBAM-results for six respondents with irregular response patterns

ID	$\alpha$			abs( $\beta$ )			$\lambda$	$\chi$			mean
	2.5%	50%	97.5%	2.5%	50%	97.5%	50%	2.5%	50%	97.5%	
965	-0.78	0.06	0.84	0.57	0.90	1.44	0.98	-2.03	-0.08	2.07	-0.05
3502	-0.88	-0.09	0.73	0.58	0.94	1.46	0.98	-1.88	0.07	2.04	0.08
5136	-0.89	0.07	0.87	0.58	0.91	1.38	0.98	-2.14	2.08	4.10	1.85
6001	-0.65	0.05	0.75	0.92	1.44	1.89	0.64	-2.41	0.00	2.30	-0.01
4316	-0.59	0.17	0.89	0.86	1.36	1.86	0.64	-2.35	0.05	2.42	0.03
6309	-0.88	-0.12	0.69	0.86	1.32	1.78	0.64	-2.48	0.76	2.79	0.41

Note: The table summarizes marginal posterior distributions for key individual-level parameters in terms of quantiles and means.  $\alpha$ 's are shift-parameters, abs( $\beta$ ) represents absolute values of stretch-parameter draws,  $\chi$  are latent respondent positions, and  $\lambda$  are the estimated probabilities that the respondents have *not* flipped the scale.

The perhaps most notable difference between the models is that the upper and lower bands on some of the respondent positions estimated by the BAM models are extreme and implausible. For 2 out of the 6 respondents in question, the 95% credible intervals produced by the BAM model are considerably wider than the interval covering 95% of the draws for the whole population. One of these respondents also have an extreme posterior median estimate at 5.3 – compared to a median absolute deviation for all respondent position draws of .83. These issues are not present for the HBAM model, which produces highly uncertain, but plausible estimates for all these respondents. Uncertain and plausible estimates may be the best possible outcome here, as we ultimately cannot know what kind of logic (if any) lies behind these response patterns.

### 3 Analysis of ANES 1980

The vignette for the *hbamr* R package uses the same dataset that serves to illustrate the original model by Aldrich and McKelvey (1977) in the R package *basicspace* (Poole et al. 2016). This section presents some key results from the analyses of this dataset.

The data come from the 1980 American National Election Study (ANES) and consist of 7-point Liberal-Conservative scales. There are three reasons for using this dataset: (1) It is a dataset that other authors have considered suitable in the past, (2) it is a relatively small dataset, so it is useful for making illustrations while minimizing the computational cost, and (3) the data are more interesting than typical ANES data, as they cover six stimuli instead of the usual four. In addition to self-placements, and placements of the Democratic and Republican parties, the data cover Democratic presidential candidate Jimmy Carter, Republican candidate Ronald Reagan, independent candidate (and former Republican) John B. Anderson, and Ted Kennedy (who challenged the incumbent Carter, but failed to win the Democratic nomination).<sup>2</sup>

#### 3.1 Cross-Validation and Summary of Results

Table S4 shows summaries of the posterior distributions for the hyperparameters in the HBAM model after fitting the model to this dataset. The table thus parallels Table 2 in the article. Table S5 shows the results of a 10-fold cross-validation using these data. As in the article, the HBAM model clearly outperforms the two other models in terms of out-of-sample prediction accuracy. However, a potential concern with these data is that the two additional stimuli John B. Anderson and Ted Kennedy may have been less well known to voters, and one might wonder whether this could somehow affect the analysis. Table S6 therefore reports a similar cross-validation to the one reported in Table S5, except Anderson and Kennedy have been excluded. The conclusion remains the same in this analysis: The HBAM model outperforms the other two models by a considerable margin. Table S7 similarly replicates Table S4, while excluding Anderson and Kennedy, which makes the results more directly comparable to those reported for ANES 2012.

---

<sup>2</sup>Because the data only contain six stimuli and we want to perform cross validation, I only include those respondents who have complete data in the following analyses. I further require that respondents have used at least two unique positions to place the stimuli. This leaves 643 respondents.

**Table S4:** Posterior summaries for hyper-parameters, HBAM model, ANES 1980

	2.5%	50%	97.5%	$N_{\text{eff}}$	Rhat
$\psi$	0.86	0.88	0.90	626	1.00
$\delta$	2.01	2.10	2.33	2458	1.00
$\sigma_\alpha$	0.29	0.33	0.38	1467	1.00
$\sigma_\beta$	0.27	0.30	0.33	1596	1.00
$\sigma_\chi$	1.13	1.21	1.31	1908	1.00
$\tau$	0.71	0.74	0.77	918	1.00
$\nu$	8.59	11.27	14.97	941	1.00

**Table S5:** Cross-validation, ANES 1980

	BAM	HBAM <sub>0</sub>	HBAM
$\widehat{ELPD}$	-5835.7	-5766.9	-5525.1
$SE_{\widehat{ELPD}}$	58.6	51.2	48.2

Note: Estimated ELPDs based on 10-fold cross-validation. ELPD is the theoretical expected log pointwise predictive density for a new dataset, while  $SE_{\widehat{ELPD}}$  is the standard error of the ELPD estimate.

**Table S6:** Replication of Table S5, excluding Anderson and Kennedy

	BAM	HBAM <sub>0</sub>	HBAM
$\widehat{ELPD}$	-4307.0	-4244.7	-3896.2
$SE_{\widehat{ELPD}}$	55.0	46.8	43.5

Note: Estimated ELPDs based on 10-fold cross-validation. ELPD is the theoretical expected log pointwise predictive density for a new dataset, while  $SE_{\widehat{ELPD}}$  is the standard error of the ELPD estimate.



**Table S7:** Replication of Table S4, excluding Anderson and Kennedy

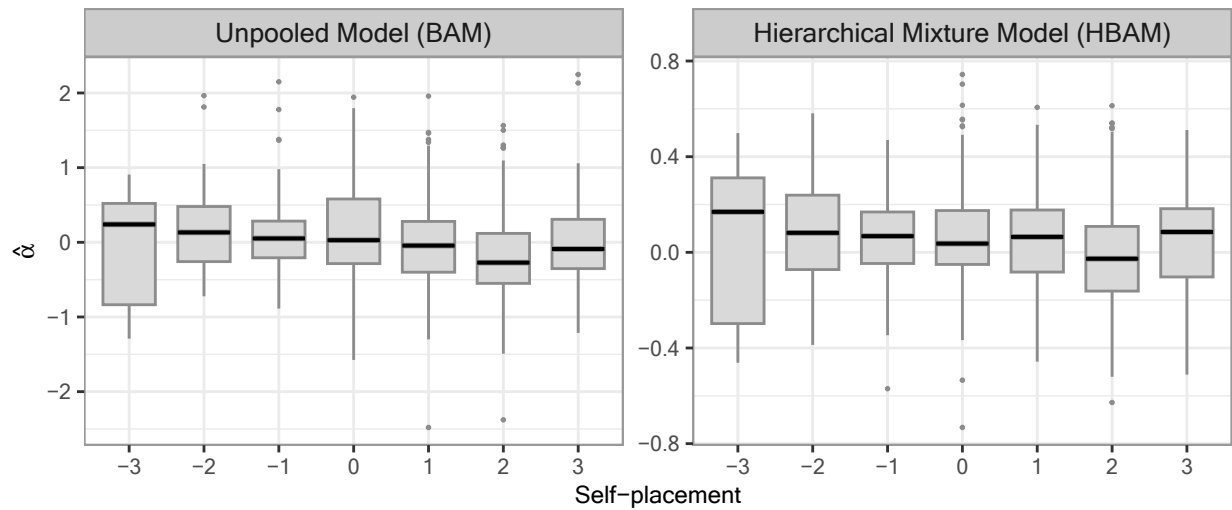
	2.5%	50%	97.5%	$N_{\text{eff}}$	Rhat
$\psi$	0.84	0.86	0.88	887	1.01
$\delta$	2.01	2.06	2.19	3213	1.00
$\sigma_{\alpha}$	0.34	0.38	0.42	1021	1.00
$\sigma_{\beta}$	0.30	0.33	0.35	1474	1.00
$\sigma_{\chi}$	1.11	1.19	1.28	1780	1.00
$\tau$	0.49	0.52	0.55	609	1.01
$\nu$	6.02	8.33	11.56	567	1.01

### 3.2 Estimated Pattern in Shifting

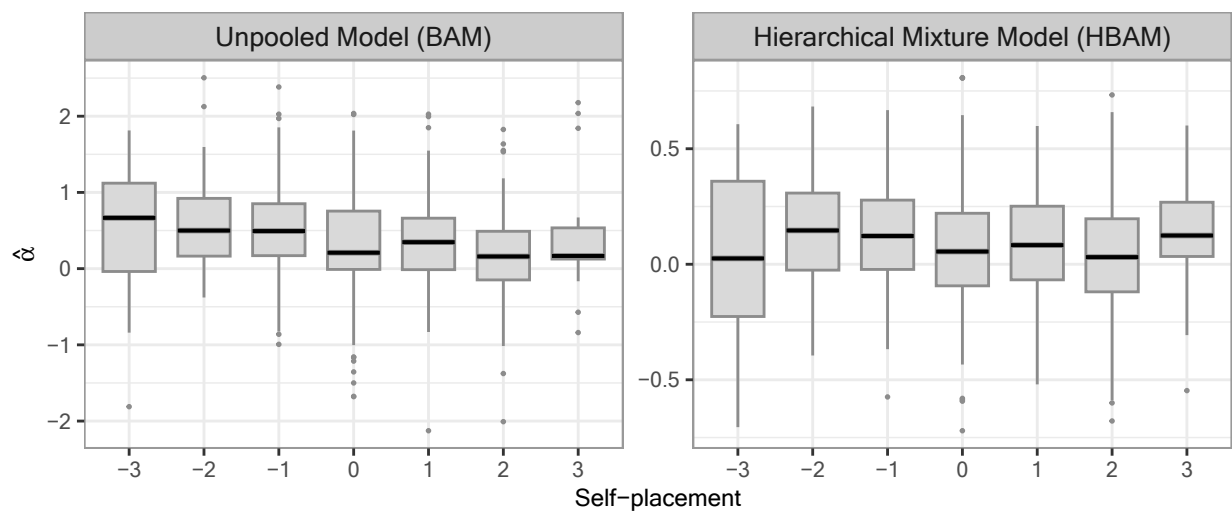
A key finding of Hare et al. (2015) was that respondents on the left side of the scale tend to shift their responses to the right, while those on the right shift their responses to the left. This finding was replicated in Figure 4 in the article. However, this pattern is virtually absent from the results from the ANES 1980 data. Similar to Figure 4 in the article, Figure S3 shows the estimated  $\alpha$  parameters over self-placements. There is a slight downward slope in the median of the estimates as we move from left to right, but the pattern is much weaker than in the data from 2012. As mentioned above, a possible concern might be that the data differ in that the ANES 1980 dataset contains two additional stimuli, and Figure S4 therefore shows the results when John B. Anderson and Ted Kennedy are excluded from the data. However, this does not make the pattern produced by either model much clearer. This raises the question of whether the respondents' behavior has changed over time – perhaps in line with the degree of political polarization. I consider this topic to be outside the scope of this article, but I believe it is an interesting question for future research.

### 3.3 Examination of Irregular Response Patterns, ANES 1980

This section examines irregular response patterns in the ANES 1980 dataset, just like Section 2 did for the ANES 2012 data. Table S8 shows the data for the selected respondents, while Tables S9 and S10 summarize the estimates from each model for these respondents. As in the analysis of the ANES 2012 data, both models tend to make only modest adjustments to the self-placements in these cases. The three respondents that report off-center self-placements are all left at the same side of the center as they were originally, while their relative distances from the center have changed moderately.



**Figure S3:** Median  $\alpha$  parameter estimates over self-placements, ANES 1980



**Figure S4:** Replication of Figure S3, excluding Anderson and Kennedy

Still, there are some substantive differences between the results from the two models. The BAM model estimates that respondent 518 has a large positive shift-parameter of 1.7, which leads the respondent position to be corrected towards the left. The HBAM model estimates only a slightly positive shift-parameter and leaves the respondent close to the original self-placement of zero. Another difference is that the BAM model estimates respondent 1488 to have flipped the scale (with a median  $\beta$  parameter of -0.98 and only 20% of the draws for  $\beta$  being positive), while the HBAM model estimates a flipping-parameter of .66, suggesting this respondent most likely has not flipped the scale (while also implying a notable probability that they have done so). The position estimates for this respondent are still fairly similar across models and close to the original self-placement, which is probably in part due to the original placement being zero.

As in the analysis of the ANES 2012 data, the most notable difference between the models is that the upper and lower bands on the respondent positions estimated by the BAM model are extreme and implausible. For five out of the six respondents in question, the 95% credible intervals produced by the BAM model are considerably wider than the interval covering 95% of the draws for the whole population.

**Table S8:** Data for six respondents with irregular response patterns

ID	Self	Kennedy	Democrats	Carter	Anderson	Republicans	Reagan
380	2	0	-3	-2	0	3	1
1325	-1	0	-2	0	3	2	0
518	0	0	3	-3	-1	3	3
787	0	-2	2	-2	2	-2	3
1488	0	0	-2	3	3	1	-3
1504	2	-2	2	-2	3	-2	2

Note: The table shows data for the three respondents with the largest errors according to the HBAM model (first three rows) as well as the three with the largest errors according to the BAM model (last three rows). The stimuli have been ordered by their estimated ideological ranking, from leftmost to rightmost.

**Table S9:** BAM-results for six respondents with irregular response patterns

ID	$\alpha$			$\beta$				$\chi$			
	2.5%	50%	97.5%	2.5%	50%	97.5%	> 0	2.5%	50%	97.5%	mean
380	-1.71	-0.66	0.42	1.21	2.64	4.23	1.00	0.54	1.01	2.02	1.09
1325	-1.32	-0.27	0.84	-0.33	1.17	2.66	0.94	-5.92	-0.54	3.95	-1.71
518	0.28	1.70	3.18	-0.73	1.40	3.40	0.92	-10.86	-1.04	8.41	-0.16
787	-1.10	0.74	2.44	-1.82	0.75	3.14	0.74	-12.76	-0.32	10.09	-0.28
1488	-2.42	-0.69	0.98	-3.26	-0.98	1.55	0.20	-9.32	-0.33	7.14	-1.58
1504	-1.09	0.53	2.16	-2.10	0.23	2.40	0.58	-19.54	0.79	18.95	-3.42

Note: The table summarizes marginal posterior distributions for key individual-level parameters in terms of quantiles and means.  $\alpha$ 's are shift-parameters,  $\beta$ 's are stretch-parameters, and  $\chi$ 's are latent respondent positions. The column " $\beta > 0$ " reports the share of the draws for  $\beta_i$  that are above zero, which indicates the probability that a respondent has *not* flipped the scale.

**Table S10:** HBAM-results for six respondents with irregular response patterns

ID	$\alpha$			abs( $\beta$ )			$\lambda$	$\chi$			mean
	2.5%	50%	97.5%	2.5%	50%	97.5%	50%	2.5%	50%	97.5%	
380	-0.49	0.03	0.55	0.74	1.19	1.69	0.99	-1.01	1.23	2.46	1.15
1325	-0.49	0.01	0.55	0.51	0.83	1.32	0.98	-2.23	-0.70	1.44	-0.64
518	-0.39	0.21	0.78	0.73	1.34	2.06	0.83	-1.79	-0.09	1.70	-0.08
787	-0.40	0.14	0.65	0.85	1.35	1.87	0.66	-1.94	-0.07	1.95	-0.04
1488	-0.56	-0.02	0.55	0.55	0.98	2.03	0.66	-1.82	0.05	1.96	0.04
1504	-0.48	0.06	0.59	0.79	1.22	1.76	0.64	-2.12	0.67	2.41	0.36

Note: The table summarizes marginal posterior distributions for key individual-level parameters in terms of quantiles and means.  $\alpha$ 's are shift-parameters, abs( $\beta$ ) represents absolute values of stretch-parameter draws,  $\chi$ 's are latent respondent positions, and  $\lambda$ 's are estimated probabilities that the respondents have *not* flipped the scale.

## 4 Code for Generating Data in R

```
# Function to draw from the scaled inverse chi-squared distribution:
rinvchisq <- function(n, df, scale = 1 / df) {
  (df * scale) / rchisq(n, df = df)
}

# Function to draw from the Dirichlet distribution:
rdirichlet <- function (n, alpha) {
  l <- length(alpha)
  x <- matrix(rgamma(l * n, alpha), ncol = l, byrow = TRUE)
  sm <- x %*% rep(1, l)
  return(x / as.vector(sm))
}

# Function to generate data:
create_data <- function(N = 500, J = 4, B = 3, nu = 8.4, tau = .19 * B, rho_a = 5, psi = .89,
  sigma_a = .19 * B, sigma_b = .27, sigma_c = .52 * B,
  seed = 1) {
  set.seed(seed)

  # Latent parameters:
  true_alpha <- rnorm(N, 0, sigma_a)
  true_beta0 <- exp(rnorm(N, 0, sigma_b))
  true_kappa <- rbinom(N, 1, psi)
  true_beta <- true_beta0 * true_kappa - true_beta0 * (1 - true_kappa)
  true_chi <- rnorm(N, 0, sigma_c)
  true_theta <- c(runif(1, -B + 1, 0), runif(1, 0, B - 1), runif(J - 2, -B + 1, B - 1))

  # Stimulus positions:
  true_rho <- rdirichlet(1, rep(rho_a, J))
  true_eta <- rinvchisq(N, nu, scale = (tau^2) * (J^2))
  sd_ij <- matrix(rep(sqrt(true_eta), times = J) * rep(true_rho, each = N), nrow = N, ncol = J)
  errors <- matrix(rnorm(N * J, 0, sd_ij), nrow = N, ncol = J)
  ppos <- round(matrix(rep(true_alpha, each = J) + (rep(true_beta, each = J) * true_theta),
    N, J, byrow = TRUE) + errors)
  ppos[ppos < -B] <- -B; ppos[ppos > B] <- B

  # Self-placements:
  self <- round(true_alpha + true_beta * true_chi + rnorm(N, 0, sqrt(true_eta) * min(true_rho)))
  self[self < -B] <- -B; self[self > B] <- B

  dat <- prep_data(self, ppos, B = B)
  dat$true_theta <- true_theta
  dat$true_chi <- true_chi[dat$keep]
  dat$true_alpha <- true_alpha[dat$keep]
  dat$true_beta <- true_beta[dat$keep]
  dat$true_kappa <- true_kappa[dat$keep]

  return(dat)
}
```

## 5 Model Codes in Stan

### 5.1 The BAM Model

```
data {
  int<lower = 1> N;           // n of individuals
  int<lower = 1> J;           // n of items
  int<lower = 1> N_obs;      // n of observations
  int<lower = 1> ii[N_obs];  // index i in matrix
  int<lower = 1> jj[N_obs];  // index j in matrix
  int<lower = 1> B;          // length of scale -1 / 2
  int<lower = 1, upper = J> L; // left pole
  int<lower = 1, upper = J> R; // right pole
  int<lower = -B, upper = B> Y[N_obs]; // reported stimuli positions
  vector<lower = -B, upper = B>[N] V; // reported self-placements
}

transformed data {
  real<lower = 0> tau_prior_rate = (2 - 1) / (B / 5.0);
}

parameters {
  vector[N] alpha;           // shift parameter
  vector[N] beta;           // stretch parameter
  real<lower = -1.1, upper = -.9> thetal; // left pole stimuli
  real<lower = .9, upper = 1.1> thetar; // right pole stimuli
  real thetam[J];           // remaining stimuli
  real<lower = 3, upper = 30> nu; // concentration of etas
  real<lower = 0> tau;        // scale of errors
  vector<lower = 0>[N] eta;   // mean ind. error variance x J^2
  simplex[J] rho;           // stimuli-shares of variance
}

transformed parameters {
  real theta[J];            // latent stimuli position
  vector[N_obs] log_lik;    // pointwise log-likelihood
  real<lower = 0> eta_scale = tau * J;
  theta = thetam;
  theta[L] = thetal;
  theta[R] = thetar;
  for (n in 1:N_obs) {
    log_lik[n] = normal_lpdf(Y[n] | alpha[ii[n]] + beta[ii[n]] * theta[jj[n]],
      sqrt(eta[ii[n]]) * rho[jj[n]]);
  }
}

model {
  alpha ~ uniform(-100, 100);
  beta ~ uniform(-100, 100);
  thetam ~ normal(0, 1);
  thetal ~ normal(0, 1);
  thetar ~ normal(0, 1);
  eta ~ scaled_inv_chi_square(nu, eta_scale);
}
```

```

nu ~ gamma(25, 2.5);
tau ~ gamma(2, 5 / (B * 1.0));
rho ~ dirichlet(rep_vector(5, J));

target += sum(log_lik);
}

generated quantities {
  vector[N] chi = (V - alpha) ./ beta;
}

```

## 5.2 The HBAM<sub>0</sub> Model

```

data {
  int<lower = 1> N;           // n of individuals
  int<lower = 1> J;           // n of items
  int<lower = 1> N_obs;      // n of observations
  int<lower = 1> ii[N_obs];  // index i in matrix
  int<lower = 1> jj[N_obs];  // index j in matrix
  int<lower = 1> B;          // length of scale -1 / 2
  int<lower = -B, upper = B> Y[N_obs]; // reported stimuli positions
  vector<lower = -B, upper = B>[N] V; // reported self-placements
}

transformed data {
  real<lower = 0> sigma_chi_prior_rate = (5 - 1) / (B / 2.0);
  real<lower = 0> sigma_alpha_prior_rate = (2 - 1) / (B / 5.0);
  real<lower = 0> tau_prior_rate = (2 - 1) / (B / 5.0);
}

parameters {
  vector[N] alpha_raw;      // shift parameter, raw
  vector[N] beta_raw;       // stretch parameter, raw
  real theta[J];           // latent stimuli position
  real<lower = 0> sigma_alpha; // sd of alpha
  real<lower = 0, upper = 2> sigma_beta; // sd of log(beta)
  real<lower = 0> sigma_chi; // sd of chi0
  real<lower = 3, upper = 30> nu; // concentration of etas
  real<lower = 0> tau;       // scale of errors
  vector<lower = 0>[N] eta;  // mean ind. error variance x J^2
  simplex[J] rho;          // stimuli-shares of variance
  vector[N] chi;           // latent respondent positions
}

transformed parameters {
  vector[N] alpha;         // shift parameter
  vector[N] beta;          // stretch parameter
  vector[N_obs] log_lik;   // pointwise log-likelihood for Y
  vector[N] log_lik_V;     // pointwise log-likelihood for V
  real<lower = 0> eta_scale = tau * J;
  real<lower=0> min_rho = min(rho);
  alpha = alpha_raw * sigma_alpha; // non-centered specifications
  beta = exp(beta_raw * sigma_beta);
}

```

```

for (n in 1:N_obs) {
  log_lik[n] = normal_lpdf(Y[n] | alpha[ii[n]] + beta[ii[n]] * theta[jj[n]],
    sqrt(eta[ii[n]]) * rho[jj[n]]);
}
for (i in 1:N) {
  log_lik_V[i] = normal_lpdf(V[i] | alpha[i] + beta[i] * chi[i], sqrt(eta[i]) * min_rho);
}
}

model {
  theta ~ normal(0, B * 2);
  alpha_raw ~ normal(0, 1);
  sigma_alpha ~ gamma(2, sigma_alpha_prior_rate);
  beta_raw ~ normal(0, 1);
  sigma_beta ~ gamma(3, 10);
  chi ~ normal(0, sigma_chi);
  sigma_chi ~ gamma(5, sigma_chi_prior_rate);
  eta ~ scaled_inv_chi_square(nu, eta_scale);
  nu ~ gamma(25, 2.5);
  tau ~ gamma(2, tau_prior_rate);
  rho ~ dirichlet(rep_vector(5, J));

  target += sum(log_lik_V);
  target += sum(log_lik);
}

```

### 5.3 The HBAM Model

```

data {
  int<lower = 1> N; // n of individuals
  int<lower = 1> J; // n of items
  int<lower = 1> N_obs; // n of observations
  int<lower = 1> ii[N_obs]; // index i in matrix
  int<lower = 1> jj[N_obs]; // index j in matrix
  int<lower = 1> B; // length of scale -1 / 2
  int<lower = 1, upper = J> L; // left pole
  int<lower = 1, upper = J> R; // right pole
  int<lower = -B, upper = B> Y[N_obs]; // reported stimuli positions
  vector<lower = -B, upper = B>[N] V; // reported self-placements
}

transformed data {
  real<lower = 0> sigma_chi_prior_rate = (5 - 1) / (B / 2.0);
  real<lower = 0> sigma_alpha_prior_rate = (2 - 1) / (B / 5.0);
  real<lower = 0> tau_prior_rate = (2 - 1) / (B / 5.0);
}

parameters {
  matrix[N, 2] alpha_raw; // shift parameter, split, raw
  matrix[N, 2] beta_raw; // stretch parameter, split, raw
  ordered[2] theta_lr; // left and right pole
  real theta_raw[J]; // remaining stimuli
}

```



```

real<lower = 0> sigma_alpha;           // sd of alpha
real<lower = 0, upper = 2> sigma_beta; // sd of log(beta)
real<lower = 0> sigma_chi;            // sd of chi0
real<lower = 3, upper = 30> nu;       // concentration of etas
real<lower = 0> tau;                   // scale of errors
vector<lower = 0>[N] eta;              // mean ind. error variance x J^2
simplex[J] rho;                         // stimuli-shares of variance
vector<lower = 0, upper = 1>[N] lambda; // mixing proportion, flipping
real<lower = .5, upper = 1> psi;       // mean of prior on lambda
real<lower = 2, upper = 100> delta;    // concentration of prior on lambda
matrix[N, 2] chi0;                     // latent respondent positions, split
}

transformed parameters {
  real<lower=0> alpha_lambda = delta * psi; // reparameterization
  real<lower=0> beta_lambda = delta * (1 - psi);
  real theta[J];                          // latent stimuli position
  matrix[N, 2] alpha0;                    // shift parameter, split
  matrix[N, 2] beta0;                     // stretch parameter, split
  vector[N_obs] log_lik;                   // pointwise log-likelihood for Y
  vector[N] log_lik_V;                     // pointwise log-likelihood for V
  real<lower = 0> eta_scale = tau * J;
  real<lower=0> min_rho = min(rho);
  theta = theta_raw;
  theta[L] = theta_lr[1];                  // safeguard to ensure identification
  theta[R] = theta_lr[2];
  alpha0[, 1] = alpha_raw[, 1] * sigma_alpha; // non-centered specifications
  alpha0[, 2] = alpha_raw[, 2] * sigma_alpha;
  beta0[, 1] = exp(beta_raw[, 1] * sigma_beta);
  beta0[, 2] = -exp(beta_raw[, 2] * sigma_beta);

  for (n in 1:N_obs) {
    log_lik[n] = log_mix( lambda[ii[n]],
      normal_lpdf(Y[n] | alpha0[ii[n], 1] + beta0[ii[n], 1] * theta[jj[n]],
        sqrt(eta[ii[n]]) * rho[jj[n]]),
      normal_lpdf(Y[n] | alpha0[ii[n], 2] + beta0[ii[n], 2] * theta[jj[n]],
        sqrt(eta[ii[n]]) * rho[jj[n]]));
  }
  for (i in 1:N) {
    log_lik_V[i] = log_mix( lambda[i],
      normal_lpdf(V[i] | alpha0[i, 1] + beta0[i, 1] * chi0[i, 1],
        sqrt(eta[i]) * min_rho),
      normal_lpdf(V[i] | alpha0[i, 2] + beta0[i, 2] * chi0[i, 2],
        sqrt(eta[i]) * min_rho));
  }
}

model {
  theta_raw ~ normal(0, B);
  theta_lr ~ normal(0, B);
  alpha_raw[, 1] ~ normal(0, 1);
  alpha_raw[, 2] ~ normal(0, 1);
  sigma_alpha ~ gamma(2, sigma_alpha_prior_rate);
  beta_raw[, 1] ~ normal(0, 1);

```

```

beta_raw[, 2] ~ normal(0, 1);
sigma_beta ~ gamma(3, 10);
chi0[, 1] ~ normal(0, sigma_chi);
chi0[, 2] ~ normal(0, sigma_chi);
sigma_chi ~ gamma(5, sigma_chi_prior_rate);
eta ~ scaled_inv_chi_square(nu, eta_scale);
nu ~ gamma(25, 2.5);
tau ~ gamma(2, tau_prior_rate);
rho ~ dirichlet(rep_vector(5, J));
lambda ~ beta(alpha_lambda, beta_lambda);
psi ~ beta(8.5, 1.5);
delta - 2 ~ gamma(2, .1);

target += sum(log_lik_V);
target += sum(log_lik);
}

generated quantities {
  vector[N] kappa = to_vector(bernoulli_rng(lambda));
  vector[N] chi = (kappa .* chi0[, 1]) + ((1 - kappa) .* chi0[, 2]);
  vector[N] alpha = (kappa .* alpha0[, 1]) + ((1 - kappa) .* alpha0[, 2]);
  vector[N] beta = (kappa .* beta0[, 1]) + ((1 - kappa) .* beta0[, 2]);
}

```

# Prior Predictive Simulation

The following section illustrates the prior specifications for the HBAM model by simulating draws from the various prior distributions. The prior specifications reflect two principles: (1) Including enough information to identify the model and regularize the estimates while letting the data drive the results, and (2) employing hierarchical priors wherever possible, letting the degree of shrinkage depend on the data. R code for running the simulations is included throughout this section.

## Setup

```
### Simulation and plotting parameters
J <- 8           # Number of stimuli
N <- 500        # Number of respondents
B <- 5          # Scale bound
d <- 1e6        # Number of draws per simulation
nbin <- 150     # Number of bins for histograms
set.seed(1)

### Packages
library(ggplot2)

### Necessary functions
# Function to draw from the scaled inverse chi-squared distribution:
rinvsq <- function(n, df, scale = 1 / df) {
  (df * scale) / rchisq(n, df = df)
}

# Function to draw from the Dirichlet distribution:
rdirichlet <- function(n, alpha) {
  l <- length(alpha)
  x <- matrix(rgamma(l * n, alpha), ncol = l, byrow = TRUE)
  sm <- x %*% rep(1, l)
  return(x / as.vector(sm))
}
```

## Scale flipping

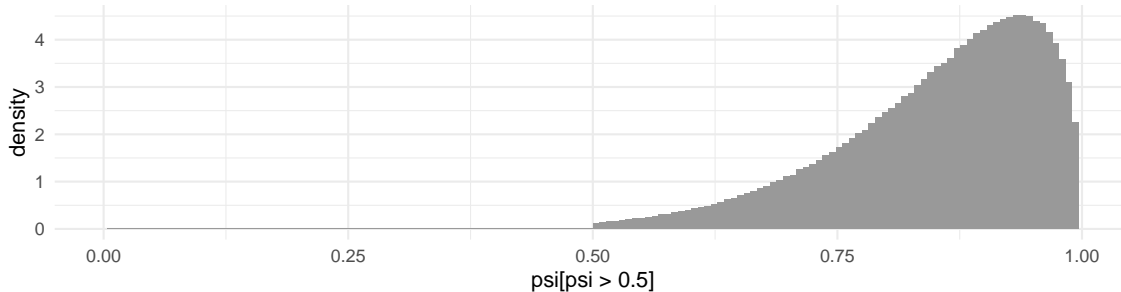
The individual flipping parameters,  $\kappa$ , are marginalized out of the HBAM likelihood, while their expectations,  $\lambda$ , are estimated in the model.  $\lambda_i$  is given a hierarchical beta prior, specified in terms of its expectation,  $\psi$ , and concentration,  $\delta$ :  $\lambda_i \sim \text{Beta}(\psi\delta, (1 - \psi)\delta)$ .

Based on existing studies, we would expect the probability of flipping to be well below 50%, and (as  $\kappa_i$  is 1 when a respondent does *not* flip the scale) this implies a  $\psi$  well above .5. The model places a lower limit on  $\psi$  at .5 to ensure identification (i.e. an upper limit of 50% flipping, which would be an extreme case).  $\psi$  is further given a Beta(8.5, 1.5) prior, which implies a 15% probability that respondents flip the scale. This prior allows all plausible values, while placing the most probability mass around the values found in previous studies. The prior is also uni-modal and curved at the upper limit, which can be beneficial for the HMC algorithm.

```
psi <- rbeta(d, 8.5, 1.5)
mean(psi[psi > .5])

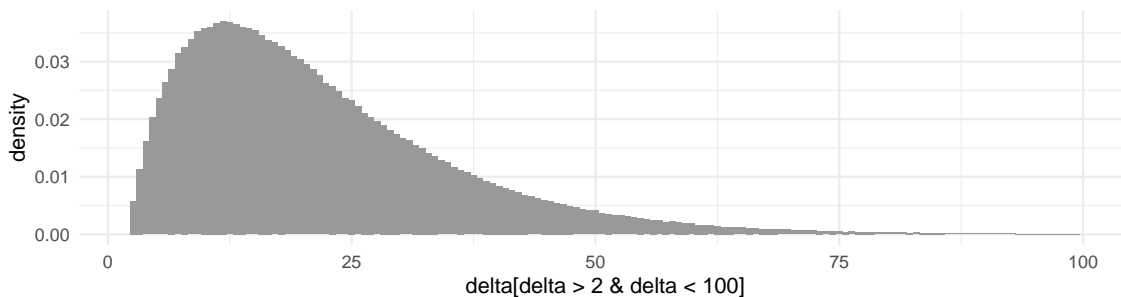
## [1] 0.852709

qplot(psi[psi > .5], bins = nbin, xlim = c(0, 1)) + aes(y = after_stat(density))
```



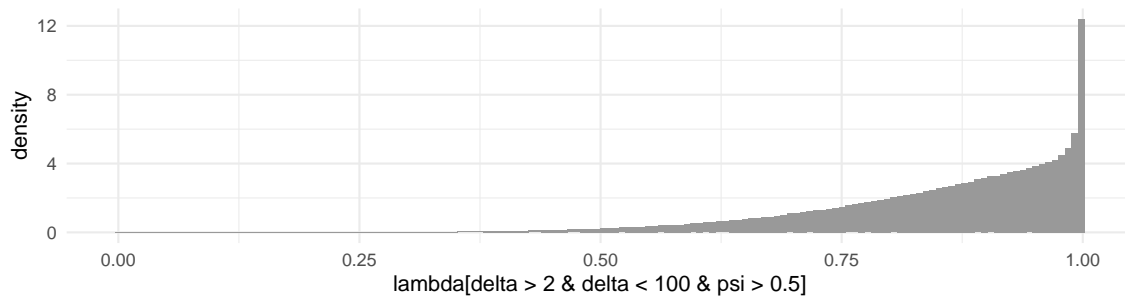
The concentration parameter,  $\delta$ , is likely to be low when the model is applied to real data that contain at least some instances of scale flipping. However, in artificially created data with zero flipping, this parameter could potentially reach very high values, and letting the parameter take on higher values helps the model produce optimal results in such settings. At the same time, the parameter may be poorly identified in these artificial scenarios, so the prior should be also sufficiently short-tailed to enable efficient sampling. Another consideration is that the beta distribution turns bi-modal when delta is below 2, and such a distribution is very challenging for an HMC-type sampling algorithm. The parameter is therefore given a lower limit of 2, which still allows for a flat distribution if the data call for it. The parameter is also given an upper limit of 100 to reduce the risk that the sampler gets stuck at extreme values when analyzing artificial data. The prior for  $\delta$  is specified as 2 plus a gamma distribution with a shape parameter of 2 and a rate parameter of .1, which gives a mode at 12:

```
delta <- 2 + rgamma(d, 2, rate = .1)
qplot(delta[delta > 2 & delta < 100], bins = nbin, xlim = c(2, 100)) +
  aes(y = after_stat(density))
```



To see the implications of the priors above for the expectations of the flipping parameters,  $\lambda$ , we can plug the simulated values into the prior for  $\lambda_i$ :

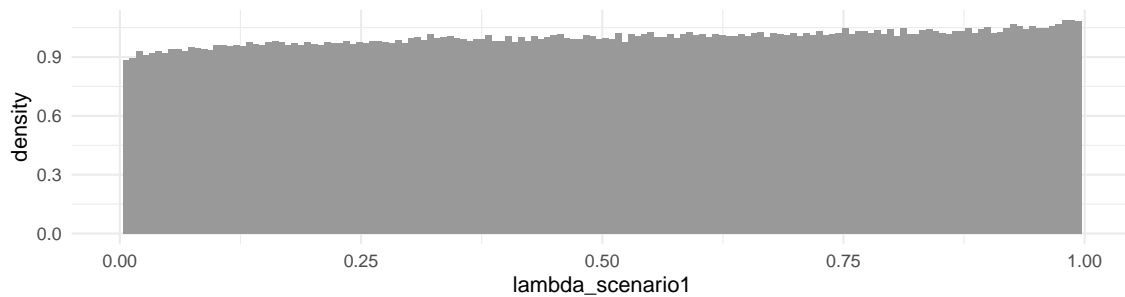
```
alpha_lambda <- delta * psi
beta_lambda <- delta * (1 - psi)
lambda <- rbeta(d, alpha_lambda, beta_lambda)
qplot(lambda[delta > 2 & delta < 100 & psi > .5], bins = nbin) +
  aes(y = after_stat(density))
```



This distribution implies that unambiguous non-flipping is the most likely, but it also preserves the expectation of  $\psi$  – implying a 15% probability of flipping the scale.

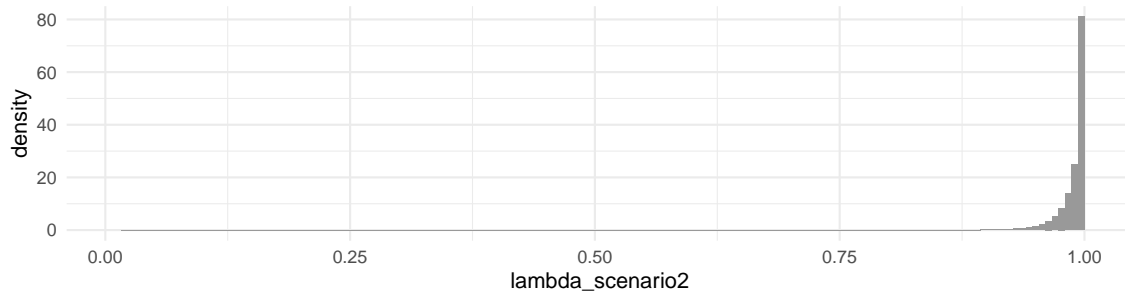
It should also be noted that these priors are flexible, due to the hierarchical setup. If the data call for it, the priors allow for a scenario where  $\delta = 2.01$  and  $\psi = .51$ , which gives an almost flat prior for  $\lambda_i$ :

```
delta_scenario1 <- 2.01
psi_scenario1 <- .51
alpha_lambda <- delta_scenario1 * psi_scenario1
beta_lambda <- delta_scenario1 * (1 - psi_scenario1)
lambda_scenario1 <- rbeta(d, alpha_lambda, beta_lambda)
qplot(lambda_scenario1, bins = nbin, xlim = c(0, 1)) + aes(y = after_stat(density))
```



At the other extreme, the priors allow for a scenario where flipping is essentially non-existent:

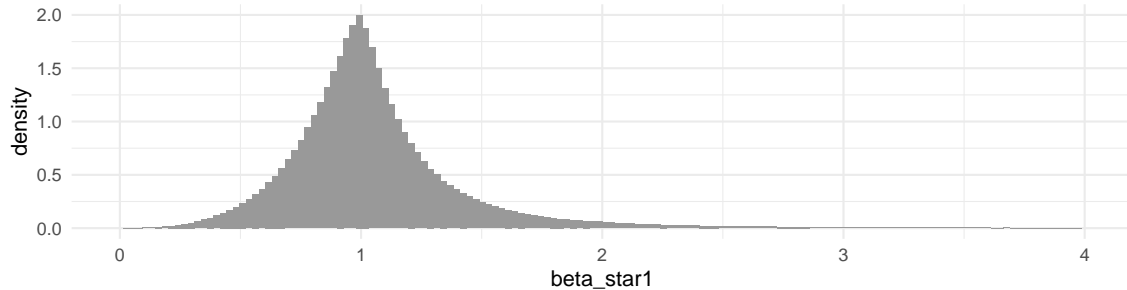
```
delta_scenario2 <- 50
psi_scenario2 <- .99
alpha_lambda <- delta_scenario2 * psi_scenario2
beta_lambda <- delta_scenario2 * (1 - psi_scenario2)
lambda_scenario2 <- rbeta(d, alpha_lambda, beta_lambda)
qplot(lambda_scenario2, bins = nbin, xlim = c(.01, 1)) + aes(y = after_stat(density))
```



## Beta

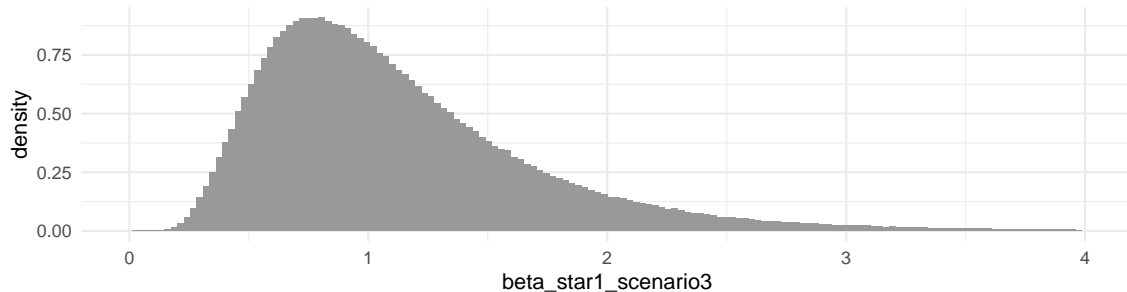
In the HBAM model, the  $\beta$  parameters are modeled separately for the flipped and non-flipped states:  $\beta_{1i}^*$  and  $\beta_{2i}^*$ . These sets of parameters are given the same log-normal prior, except the prior the parameters in flipped state has a negative sign. The parameters are modeled hierarchically, with the hyperparameter  $\sigma_\beta$  representing the standard deviation of the logged parameters.  $\sigma_\beta$  is given a Gamma(3, 10) prior, which permits all values that this parameter could plausibly take, while concentrating the probability mass around the more plausible values. The resulting prior distribution for  $\beta_{1i}^*$  is as follows:

```
sigma_beta <- rgamma(d, shape = 3, rate = 10)
beta_star1 <- exp(rnorm(d, 0, sigma_beta))
qplot(beta_star1, bins = nbin, xlim = c(0, 4)) + aes(y = after_stat(density))
```



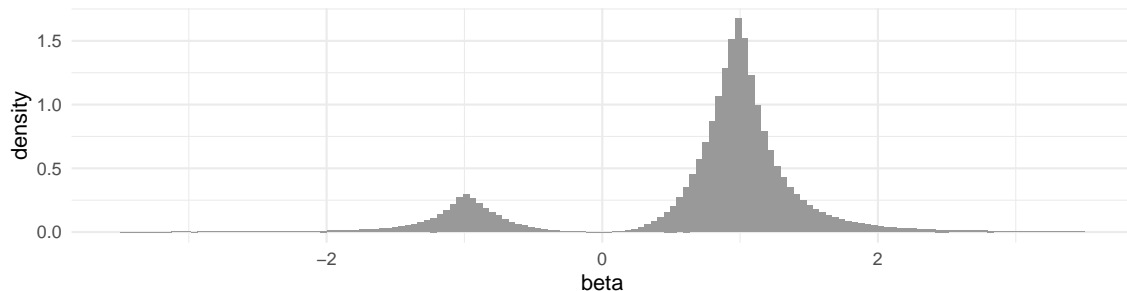
Again, it should be noted that the hierarchical setup entails a large degree of flexibility and would allow for a scenario like the following if the data were to call for it:

```
sigma_beta_scenario3 <- .5
beta_star1_scenario3 <- exp(rnorm(d, 0, sigma_beta_scenario3))
qplot(beta_star1_scenario3, bins = nbin, xlim = c(0, 4)) + aes(y = after_stat(density))
```



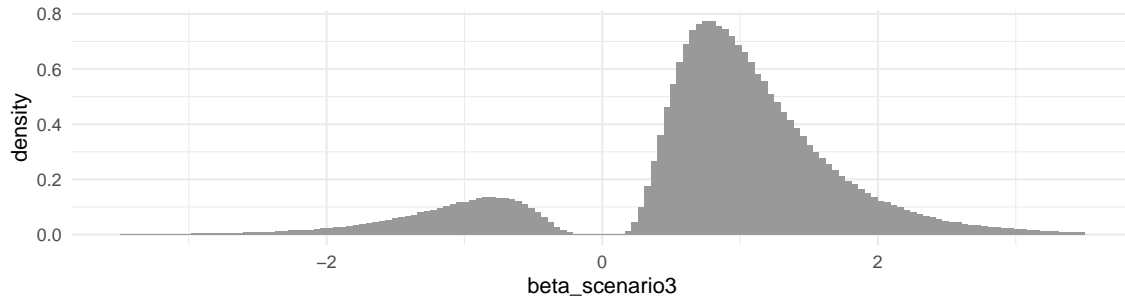
Finally, we can combine the draws for  $\beta_{1i}^*$  and  $\beta_{2i}^*$  with the draws for  $\lambda$  to see the full implications for the  $\beta$  parameters:

```
beta_star2 <- -exp(rnorm(d, 0, sigma_beta))
kappa <- rbinom(d, 1, lambda)
beta <- kappa * beta_star1 + (1 - kappa) * beta_star2
qplot(beta, bins = nbin, xlim = c(-3.5, 3.5)) + aes(y = after_stat(density))
```



In the extreme scenario where  $\sigma_\beta$  is .5, we get the following distribution:

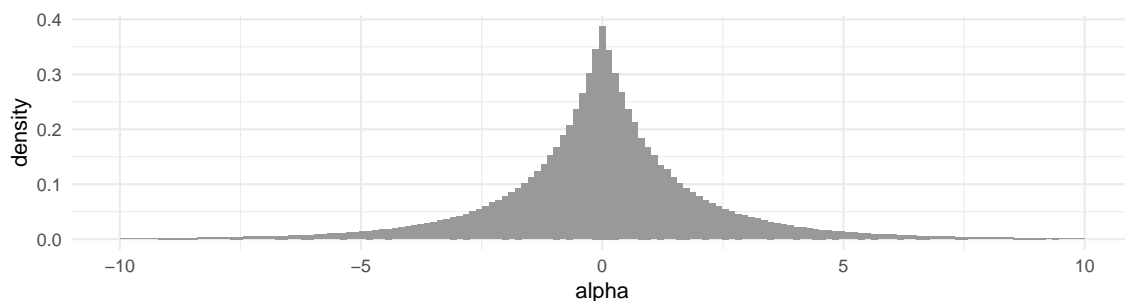
```
beta_star2_scenario3 <- -exp(rnorm(d, 0, sigma_beta_scenario3))
kappa <- rbinom(d, 1, lambda)
beta_scenario3 <- kappa * beta_star1_scenario3 + (1 - kappa) * beta_star2_scenario3
qplot(beta_scenario3, bins = nbin, xlim = c(-3.5, 3.5)) + aes(y = after_stat(density))
```



## Alpha

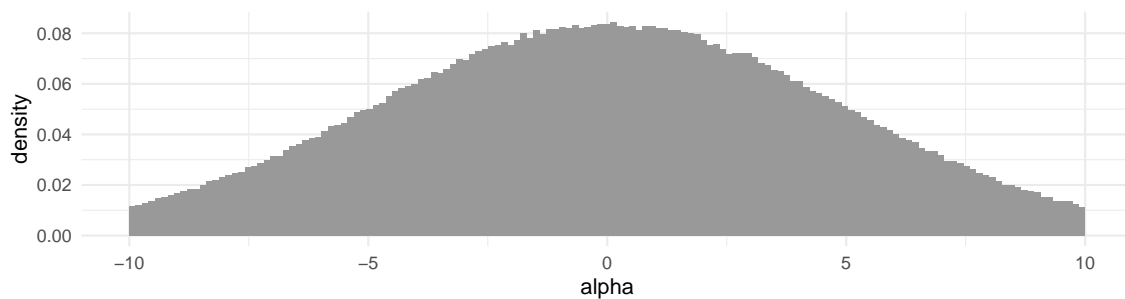
The shift parameters,  $\alpha$ , are also modeled hierarchically. In the HBAM model,  $\alpha_1^*$  and  $\alpha_2^*$  are given the same prior, and this simulation is therefore simplified to a single set of  $\alpha$  parameters (like in the HBAM<sup>0</sup> model). The hyperparameter  $\sigma_\alpha$  is given a prior of Gamma(2, 5/B), which implies a mode at B/5. This results in the following prior distribution for  $\alpha_i$ :

```
sigma_alpha <- rgamma(d, shape = 2, rate = (2 - 1) / (B / 5))
alpha <- rnorm(d, 0, sigma_alpha)
qplot(alpha, bins = nbin, xlim = c(-10, 10)) + aes(y = after_stat(density))
```



Again, the hierarchical setup entails a large degree of flexibility, and would allow for an extreme scenario like this one if the data were to call for it:

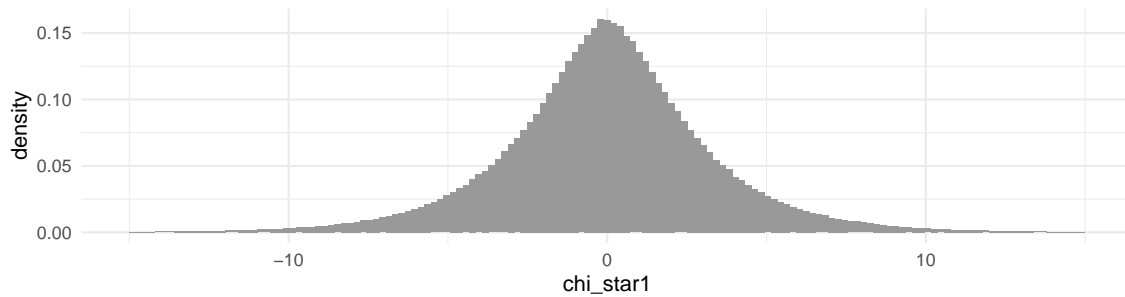
```
alpha <- rnorm(d, 0, B)
qplot(alpha, bins = nbin, xlim = c(-10, 10)) + aes(y = after_stat(density))
```



## Respondent positions

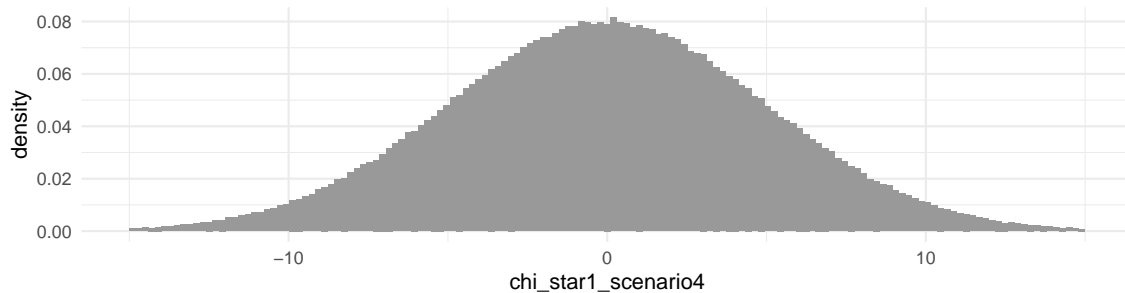
In the HBAM<sub>0</sub> and HBAM models, the latent respondent positions are treated as unknown parameters and given a hierarchical prior structure. The standard deviation of the latent positions,  $\sigma_\chi$ , is given a prior of Gamma(5, 8/B). This implies that values around half  $B$  are the most plausible, while allowing for a wide range of possible values. The resulting prior for the latent respondent positions,  $\chi$ , is as follows:

```
sigma_chi <- rgamma(d, 5, rate = (5 - 1) / (B / 2))
chi_star1 <- rnorm(d, 0, sigma_chi)
qplot(chi_star1, bins = nbin, xlim = c(-15, 15)) + aes(y = after_stat(density))
```



Again, it should be noted that the hierarchical setup entails a large degree of flexibility, and would allow an extreme scenario like this one if the data were to call for it:

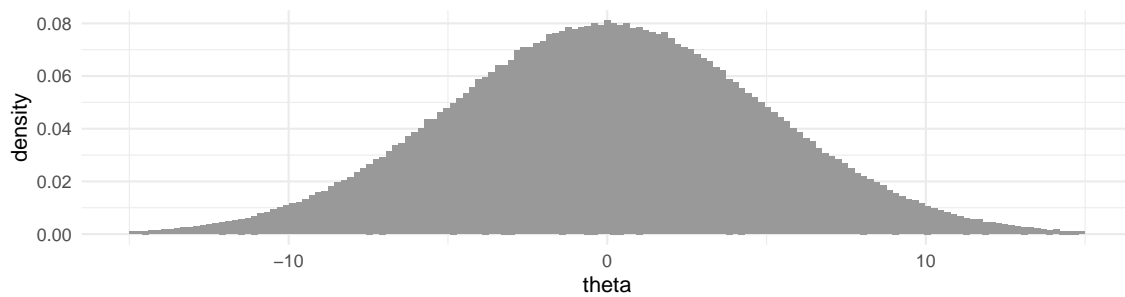
```
sigma_chi_scenario4 <- B
chi_star1_scenario4 <- rnorm(d, 0, sigma_chi_scenario4)
qplot(chi_star1_scenario4, bins = nbin, xlim = c(-15, 15)) + aes(y = after_stat(density))
```



## Stimulus positions

The scale of the latent stimulus positions will be determined by the other parts of the model, and these positions are therefore given a wide normal prior, with mean a mean of zero and a standard deviation of  $B$ :

```
theta <- rnorm(d, 0, B)
qplot(theta, bins = nbin, xlim = c(-15, 15)) + aes(y = after_stat(density))
```

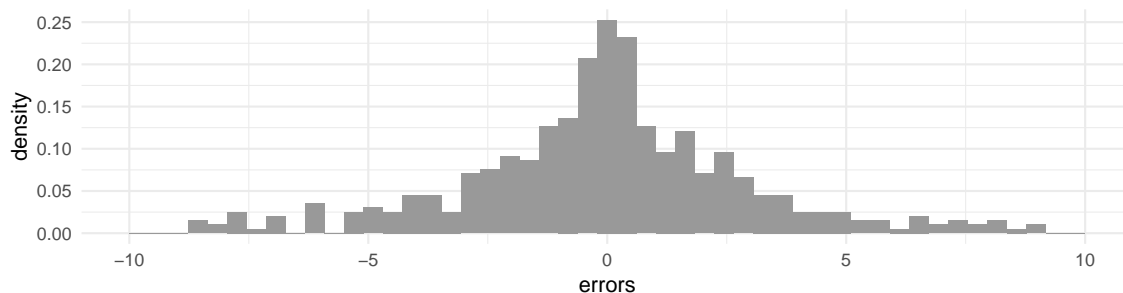




## Errors

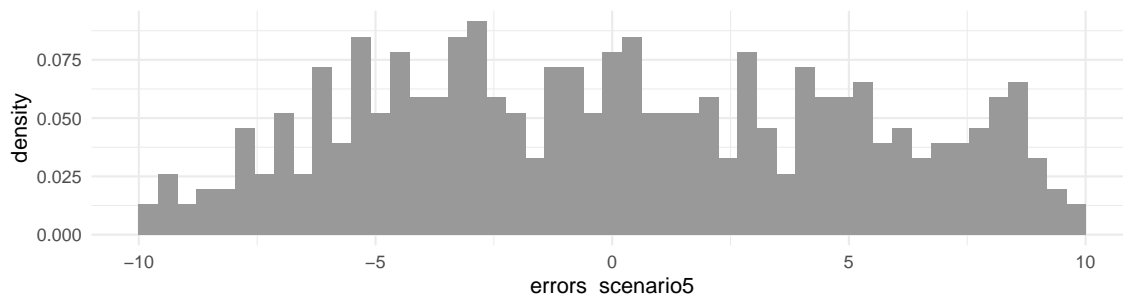
All three models that are discussed in the article, and most models in the *hbamr* package, treat the errors in a way that allows for heteroskedasticity. The respondent-specific error-variance parameters,  $\eta$ , are given a hierarchical prior structure. Combining the priors for the different parameters involved in the error specification, we get the following prior distribution for the errors across all respondents and parties:

```
d <- 2e5      # Reduce number of draws due to heavier calculations
nbin <- 50    # Reduce number of bins due to fewer draws
rho <- rdirichlet(d, rep(5, J))
nu <- rgamma(d, shape = 25, rate = 2.5)
tau <- rgamma(d, shape = 2, rate = (2 - 1) / (B / 5))
eta <- rinvcchisq(d, nu, (tau * J)^2)
sd_ij <- matrix(rep(sqrt(eta), times = J) * rep(rho, each = N), nrow = N, ncol = J)
errors <- matrix(rnorm(N * J, 0, sd_ij), nrow = N, ncol = J)
qplot(errors, bins = nbin, xlim = c(-10, 10)) + aes(y = after_stat(density))
```



Again, it should be noted that the hierarchical setup entails a large degree of flexibility, and would allow for an extreme scenario like this one if the data were to call for it:

```
tau_scenario5 <- B
eta_scenario5 <- rinvcchisq(d, nu, (tau_scenario5 * J)^2);
sd_ij <- matrix(rep(sqrt(eta_scenario5), times = J) * rep(rho, each = N),
               nrow = N, ncol = J)
errors_scenario5 <- matrix(rnorm(N * J, 0, sd_ij), nrow = N, ncol = J)
qplot(errors_scenario5, bins = nbin, xlim = c(-10, 10)) + aes(y = after_stat(density))
```



## References

- Aldrich, John H and Richard D McKelvey. 1977. “A Method of Scaling with Applications to the 1968 and 1972 Presidential Elections.” *American Political Science Review* 71(1):111–130.
- Hare, Christopher, David A. Armstrong, Ryan Bakker, Royce Carroll and Keith T. Poole. 2015. “Using Bayesian Aldrich-McKelvey Scaling to Study Citizens’ Ideological Preferences and Perceptions.” *American Journal of Political Science* 59(3):759–774.
- Poole, Keith, Jeffrey Lewis, Howard Rosenthal, James Lo and Royce Carroll. 2016. “Recovering a Basic Space from Issue Scales in R.” *Journal of Statistical Software* 69(7):1–21.