Amazon Reviews

Dr. Jarad Niemi

Iowa State University

March 1, 2016

Amazon Reviews - Upright, bagless, cyclonic vacuum cleaners

	Number of ratings							
product_id	n1	n2	n3	n4	n5	n_total	mean	sd
B000REMVGK	21	17	2	8	7	55	2.33	1.44
B001EFMD8W	40	34	28	77	347	526	4.25	1.26
B001PB51GQ	14	12	13	31	69	139	3.93	1.36
B002DGSJVG	22	8	3	6	10	49	2.47	1.63
B002G9UQZC	8	0	1	1	1	11	1.82	1.47
B002GHBRX4	18	8	9	14	27	76	3.32	1.61
B002HF66BI	9	5	2	2	3	21	2.29	1.49
B003OA77MC	15	7	8	24	42	96	3.74	1.47
B003OAD24Y	7	7	4	9	19	46	3.57	1.53
B003Y3AA3C	20	3	1	2	2	28	1.68	1.28
B0043EW354	40	25	25	60	163	313	3.90	1.44
B00440EO8G	2	1	1	1	7	12	3.83	1.64
B004R9197I	9	1	1	9	26	46	3.91	1.58
B008L5F4H0	3	1	2	12	7	25	3.76	1.27

Model for Amazon Reviews

Let y_{ij} be the jth review for the ith product. Assume

$$y_{ij} \overset{ind}{\sim} N(\theta_i, \sigma^2)$$

and

$$\theta_i \overset{ind}{\sim} N(\mu, \tau^2)$$

and

$$p(\mu, \tau, \sigma) \propto Ca^+(\sigma; 0, 1) Ca^+(\tau; 0, 1)$$

Normal hierarchical model in Stan

```
normal_model = "
data {
  int <lower=1> n;
  int <lower=1> n_products;
  int <lower=1,upper=5> stars[n];
  int <lower=1,upper=n_products> id[n];
}

parameters {
  real mu;                // implied uniform prior
  real<lower=0> sigma;
  real<lower=0> tau;
  real theta[n_products];
}

model {
  // Prior
  sigma ~ cauchy(0,1);
  tau   ~ cauchy(0,1);

  // Hierarchial model
  theta ~ normal(mu,tau);

  // Data model
  for (i in 1:n) stars[i] ~ normal(theta[id[i]], sigma);
}
"
```

Fit model

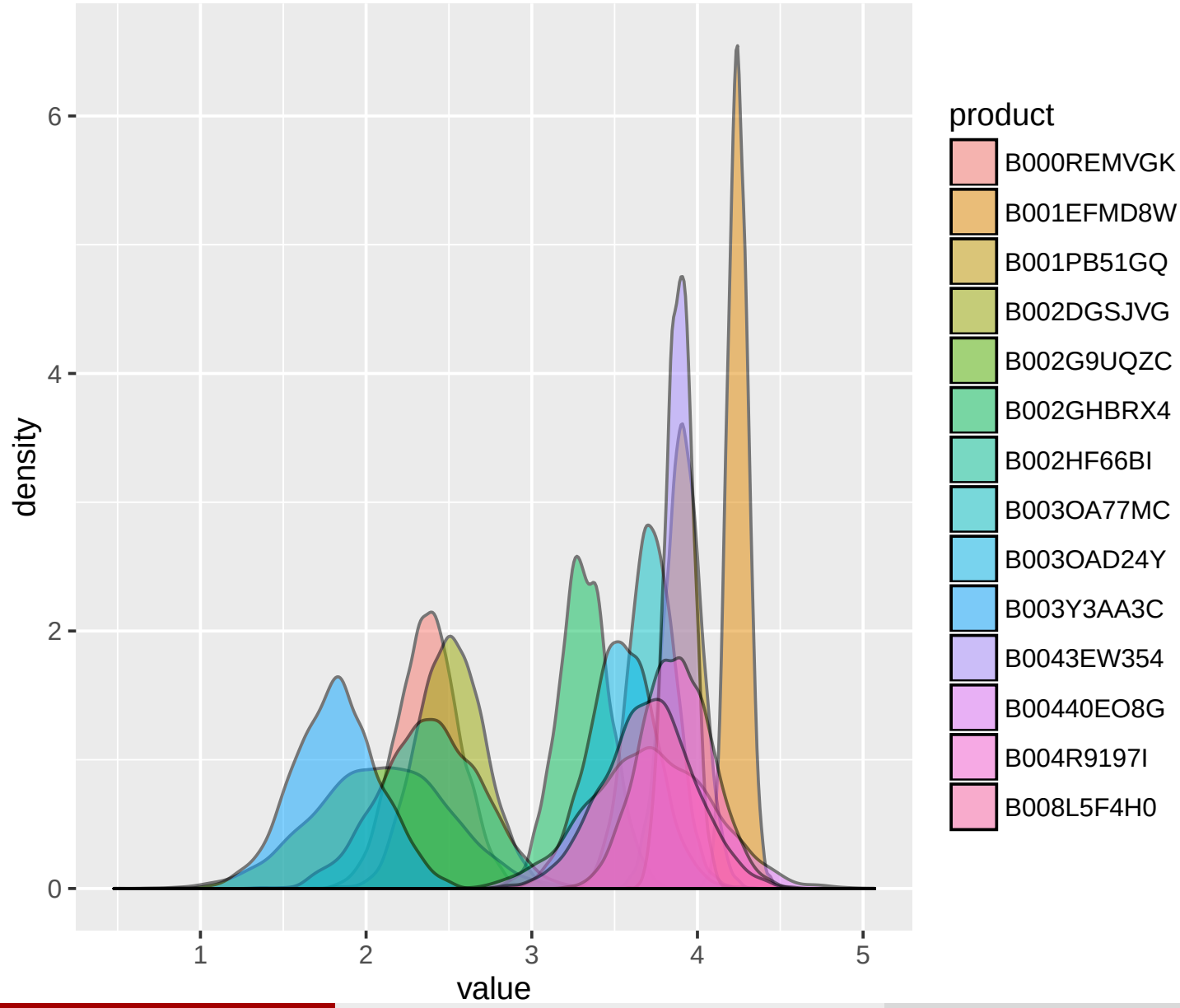
```
m = stan_model(model_code = normal_model)
dat = list(n = nrow(d),
           n_products = nlevels(d$product_id),
           stars = d$stars,
           id = as.numeric(d$product_id))
r = sampling(m, dat)
```

```
SAMPLING FOR MODEL 'ce44497edda7358de70e693408d6c43d' NOW (CHAIN 1).

Chain 1, Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1, Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1, Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1, Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1, Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1, Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1, Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1, Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1, Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1, Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1, Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1, Iteration: 2000 / 2000 [100%]  (Sampling)#
#  Elapsed Time: 1.02 seconds (Warm-up)
#                0.75 seconds (Sampling)
#                1.77 seconds (Total)
#

SAMPLING FOR MODEL 'ce44497edda7358de70e693408d6c43d' NOW (CHAIN 2).

Chain 2, Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2, Iteration:  200 / 2000 [ 10%]  (Warmup)
```

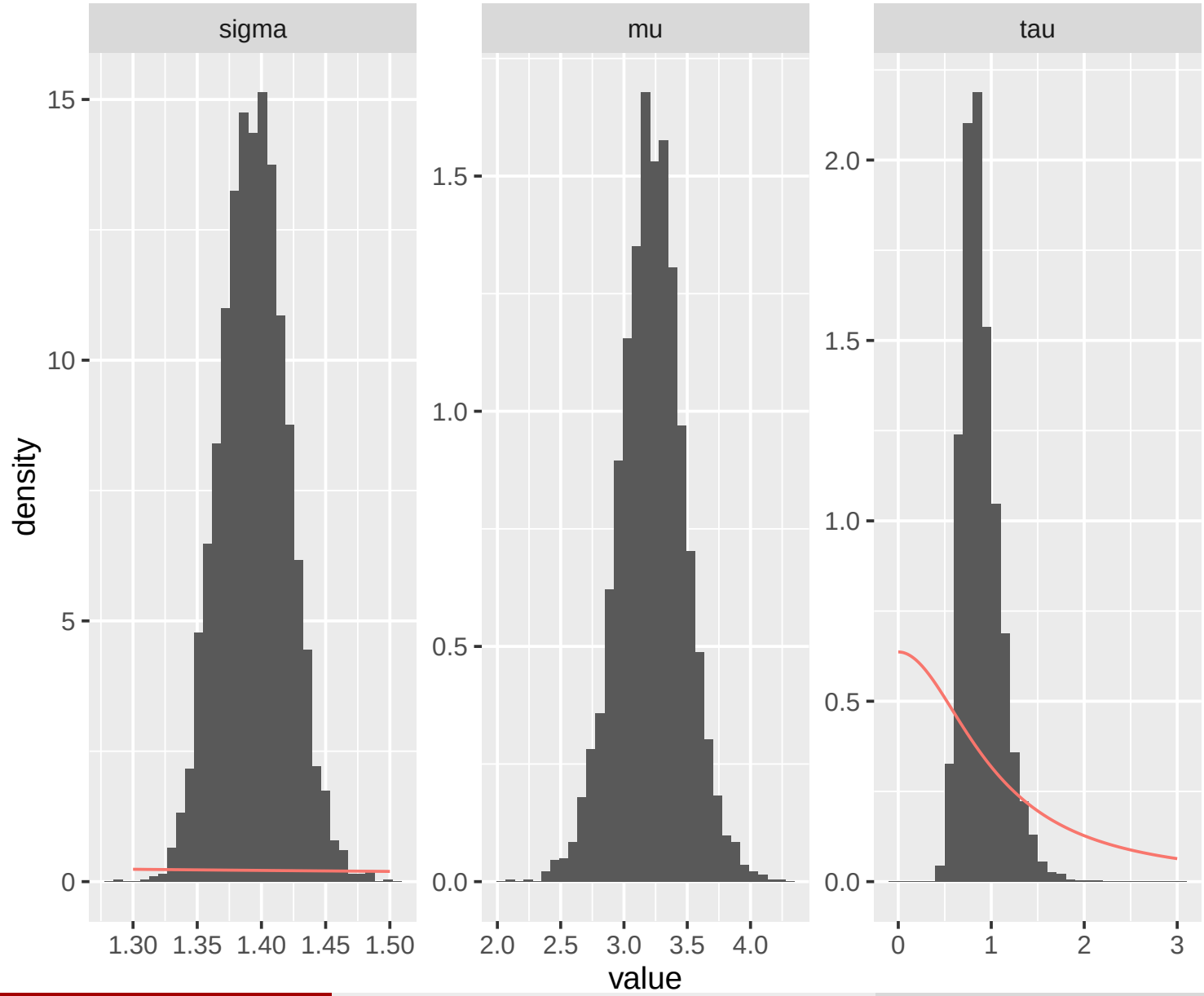
Tabular summary

```
Inference for Stan model: ce44497edda7358de70e693408d6c43d.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

             mean se_mean   sd     2.5%      25%      50%      75%    97.5% n_eff Rhat
mu           3.22    0.00 0.26     2.70     3.05     3.22     3.38     3.73  4000    1
sigma        1.39    0.00 0.03     1.34     1.38     1.39     1.41     1.45  4000    1
tau          0.89    0.00 0.21     0.58     0.74     0.85     1.01     1.40  4000    1
theta[1]     2.37    0.00 0.19     2.01     2.25     2.37     2.50     2.74  4000    1
theta[2]     4.24    0.00 0.06     4.12     4.20     4.24     4.29     4.37  4000    1
theta[3]     3.91    0.00 0.11     3.69     3.84     3.91     3.99     4.14  4000    1
theta[4]     2.51    0.00 0.20     2.14     2.37     2.51     2.65     2.90  4000    1
theta[5]     2.09    0.01 0.40     1.30     1.82     2.09     2.37     2.85  4000    1
theta[6]     3.32    0.00 0.16     3.01     3.21     3.31     3.42     3.63  4000    1
theta[7]     2.39    0.00 0.30     1.80     2.19     2.39     2.60     2.97  4000    1
theta[8]     3.73    0.00 0.14     3.45     3.63     3.72     3.82     4.00  4000    1
theta[9]     3.55    0.00 0.20     3.14     3.41     3.55     3.68     3.95  4000    1
theta[10]    1.82    0.00 0.26     1.31     1.65     1.82     1.99     2.32  4000    1
theta[11]    3.89    0.00 0.08     3.74     3.84     3.89     3.95     4.04  4000    1
theta[12]    3.72    0.01 0.36     3.00     3.47     3.71     3.96     4.42  4000    1
theta[13]    3.87    0.00 0.21     3.47     3.73     3.87     4.02     4.28  4000    1
theta[14]    3.70    0.00 0.27     3.15     3.53     3.71     3.88     4.24  4000    1
lp__      -1207.53    0.08 2.91 -1214.01 -1209.23 -1207.20 -1205.37 -1202.81  1453    1

Samples were drawn using NUTS(diag_e) at Tue Mar  1 09:51:25 2016.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

Movie mean posteriors (θ_i)



Other parameter posteriors



A quick rating

Suppose a new vacuum cleaner comes on the market and there are two Amazon reviews both with 5 stars. What do you think the averaging star rating will be (in the future) for this new product?

Let n^* be the number of new ratings and $\overline{y}^*$ be the average of those ratings, then

$$\begin{aligned} E[\theta^*|\overline{y}^*, n^*, \sigma, \mu, \tau] &= \frac{\frac{n^*}{\sigma^2}}{\frac{n^*}{\sigma^2}+\frac{1}{\tau^2}}\overline{y}^* + \frac{\frac{1}{\tau^2}}{\frac{n^*}{\sigma^2}+\frac{1}{\tau^2}}\mu \\ &= \frac{n^*}{n^*+\frac{\sigma^2}{\tau^2}}\overline{y}^* + \frac{\frac{\sigma^2}{\tau^2}}{n^*+\frac{\sigma^2}{\tau^2}}\mu \\ &= \frac{n^*}{n^*+m}\overline{y}^* + \frac{m}{n^*+m}\mu \end{aligned}$$

where $m = \sigma^2/\tau^2$ is a measure of how many *prior* samples there are.

IMDB rating

From `http://www.imdb.com/chart/top.html`:

```
weighted rating (WR) = (v / (v+m))  R + (m / (v+m))  C

Where:

R = average for the movie (mean) = (Rating)
v = number of votes for the movie = (votes)
m = minimum votes required to be listed in the Top 250
    (currently 25000)
C = the mean vote across the whole report (currently 7.1)
```

Thus IMDB uses a Bayesian estimate for the rating for each movie where $m = \sigma^2/\tau^2 = 25,000$. IMDB has enough data that the uncertainty in $\mu(C)$, σ^2, and τ^2 is pretty minimal.

Clearly incorrect model

We assumed

$$y_{ij} \overset{ind}{\sim} N(\theta_i, \sigma^2)$$

for the jth star rating of product i. Clearly this model is incorrect since $y_{ij} \in \{1, 2, 3, 4, 5\}$.

An alternative model is

$$z_{ij} \overset{ind}{\sim} Bin(4, \theta_i)$$

where $z_{ij} = y_{ij} - 1$ is the jth star rating minus 1 of product i and

$$\theta_i \sim Be(\alpha, \beta) \qquad \text{and} \qquad p(\alpha, \beta) \propto (\alpha + \beta)^{-5/2}.$$

The idea behind this model would be that product i has an independent, but common probability (θ_i) of earning each star.

Binomial hierarchical model in Stan

```
binomial_model = "
data {
  int <lower=1> n;
  int <lower=1> n_products;
  int <lower=0,upper=4> z[n];
  int <lower=1,upper=n_products> id[n];
}

parameters {
  real<lower=0> alpha;
  real<lower=0> beta;
  real<lower=0,upper=1> theta[n_products];
}

model {
  // Prior
  increment_log_prob(-5*log(alpha+beta)/2); // improper prior

  // Hierarchical model
  theta ~ beta(alpha,beta);

  // Data model
  for (i in 1:n) z[i] ~ binomial(4, theta[id[i]]);
}
"
```

Fit model

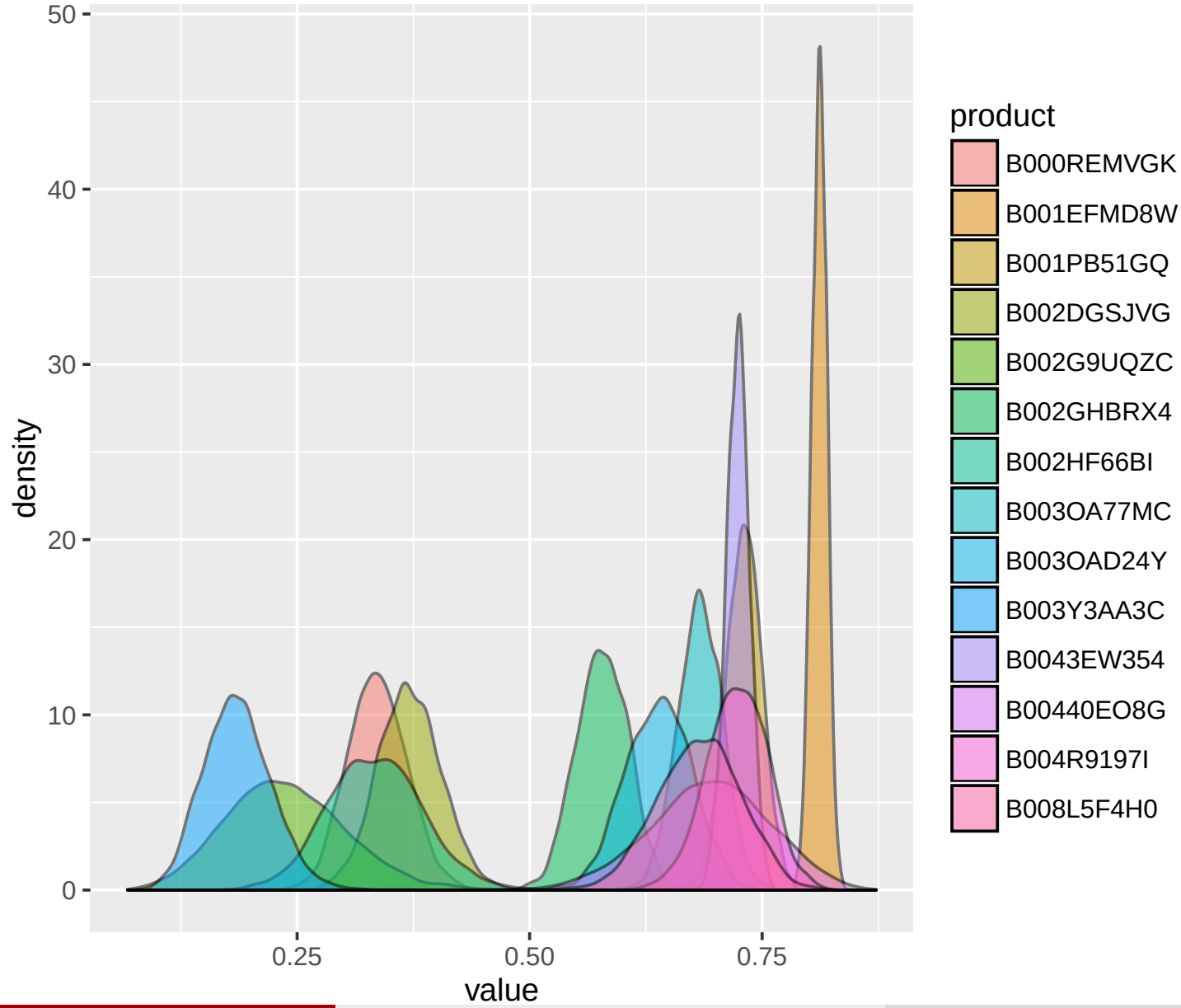
```
m = stan_model(model_code = binomial_model)
dat = list(n = nrow(d),
           n_products = nlevels(d$product_id),
           z = d$stars-1,
           id = as.numeric(d$product_id))
r = sampling(m, dat)
```

```
SAMPLING FOR MODEL 'c80c4bc1ba93fb522fa00d9419c9dda2' NOW (CHAIN 1).

Chain 1, Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1, Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1, Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1, Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1, Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1, Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1, Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1, Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1, Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1, Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1, Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1, Iteration: 2000 / 2000 [100%]  (Sampling)#
#  Elapsed Time: 1.91 seconds (Warm-up)
#                1.97 seconds (Sampling)
#                3.88 seconds (Total)
#

SAMPLING FOR MODEL 'c80c4bc1ba93fb522fa00d9419c9dda2' NOW (CHAIN 2).

Chain 2, Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2, Iteration:  200 / 2000 [ 10%]  (Warmup)
```

Tabular summary

```
Inference for Stan model: c80c4bc1ba93fb522fa00d9419c9dda2.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

             mean se_mean   sd     2.5%      25%      50%      75%    97.5% n_eff Rhat
alpha        2.71    0.03 1.11     1.09     1.92     2.52     3.31     5.27  1946    1
beta         2.28    0.02 0.89     0.95     1.63     2.16     2.78     4.40  1969    1
theta[1]     0.34    0.00 0.03     0.28     0.31     0.34     0.36     0.40  4000    1
theta[2]     0.81    0.00 0.01     0.80     0.81     0.81     0.82     0.83  4000    1
theta[3]     0.73    0.00 0.02     0.69     0.72     0.73     0.74     0.77  3912    1
theta[4]     0.37    0.00 0.03     0.31     0.35     0.37     0.39     0.44  3900    1
theta[5]     0.24    0.00 0.06     0.13     0.19     0.23     0.28     0.37  3693    1
theta[6]     0.58    0.00 0.03     0.52     0.56     0.58     0.60     0.63  3241    1
theta[7]     0.33    0.00 0.05     0.24     0.30     0.33     0.37     0.44  3805    1
theta[8]     0.68    0.00 0.02     0.64     0.67     0.68     0.70     0.73  3961    1
theta[9]     0.64    0.00 0.04     0.57     0.61     0.64     0.66     0.70  3254    1
theta[10]    0.19    0.00 0.04     0.12     0.16     0.18     0.21     0.26  4000    1
theta[11]    0.72    0.00 0.01     0.70     0.72     0.72     0.73     0.75  4000    1
theta[12]    0.69    0.00 0.06     0.56     0.65     0.70     0.74     0.81  3945    1
theta[13]    0.72    0.00 0.03     0.66     0.70     0.72     0.75     0.79  3733    1
theta[14]    0.68    0.00 0.04     0.59     0.65     0.68     0.71     0.77  3637    1
lp__      -3265.20    0.07 2.82 -3271.56 -3266.87 -3264.92 -3263.22 -3260.60  1584    1

Samples were drawn using NUTS(diag_e) at Tue Mar  1 09:52:08 2016.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

Review mean posteriors (θ_i)



Other parameter posteriors

Recall that

- α is the prior success
- β is the prior failures

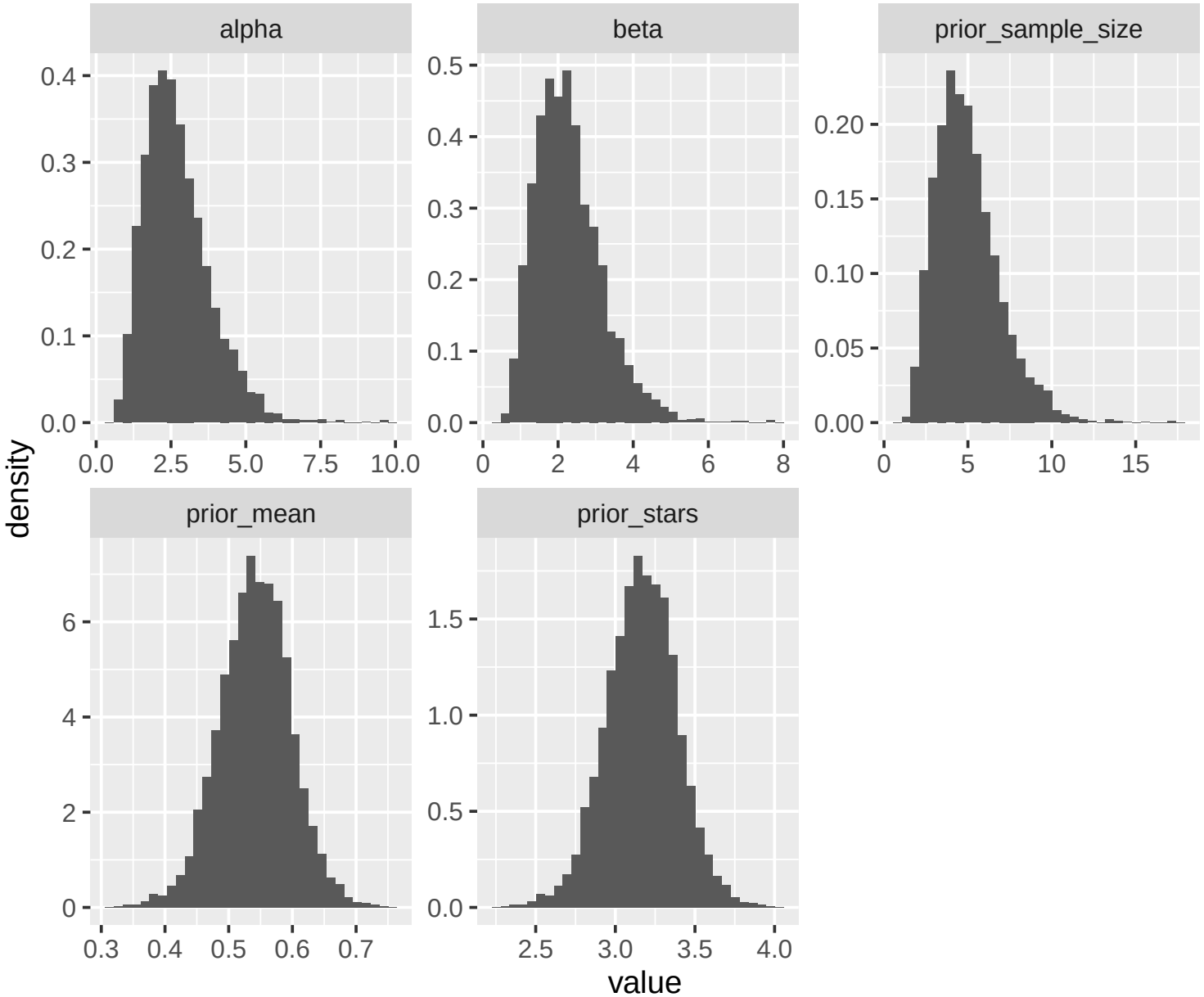
So

- $\alpha + \beta$ is the prior sample size
- $E[\theta_i|\alpha, \beta] = \frac{\alpha}{\alpha+\beta}$ is the prior expectation for the probability

But we might want to show results on the original scale (stars), so the expected number of stars for a new product is

$$\begin{aligned} E[\text{stars}_{*j}|\alpha, \beta] &= E[\text{y}_{*j} + 1|\alpha, \beta] = E[\text{y}_{*j}|\alpha, \beta] + 1 \\ &= E[E[y_{*j}|\theta^*]|\alpha, \beta] + 1 = E[4\theta^*|\alpha, \beta] + 1 \\ &= 4\frac{\alpha}{\alpha+\beta} + 1 \end{aligned}$$

Other parameter posteriors



Uniform use of star ratings

This binomial model has the proper support $\{0, 1, 2, 3, 4\}$ for stars minus 1, but does it have the correct proportion of observations in each star category?

As an example, $\hat{\theta}_2 = 0.81$ (and a 95% CI is (0.79, 0.83)). Thus, we would expect

stars	theoretical	observed
1	0.001	0.076
2	0.022	0.065
3	0.142	0.053
4	0.404	0.146
5	0.430	0.660

But this ignores the uncertainty in θ_2, so perhaps this difference is due to this uncertainty.

Posterior predictive pvalue

To assess this model fit, we will simulate posterior predictive star ratings for product 2 and compare to the observed ratings:

product_id	n1	n2	n3	n4	n5	n_total
B001EFMD8W	40	34	28	77	347	526

Let $\tilde{y}_2$ be all the predictive data for product 2, i.e. $\tilde{y}_2 = (\tilde{y}_{21}, \ldots, \tilde{y}_{2J})$ with $J = 526$ where $\tilde{y}_{2j}$ is the jth predictive star rating minus 1 for review j of product 2. Then

$$p(\tilde{y}_2|y) = \int \left[\prod_{j=1}^{J} p(\tilde{y}_{2j}|\theta_2) \right] p(\theta_2|y) d\theta_2$$

Thus the following procedure will simulation from the joint distribution for the predictive ratings:

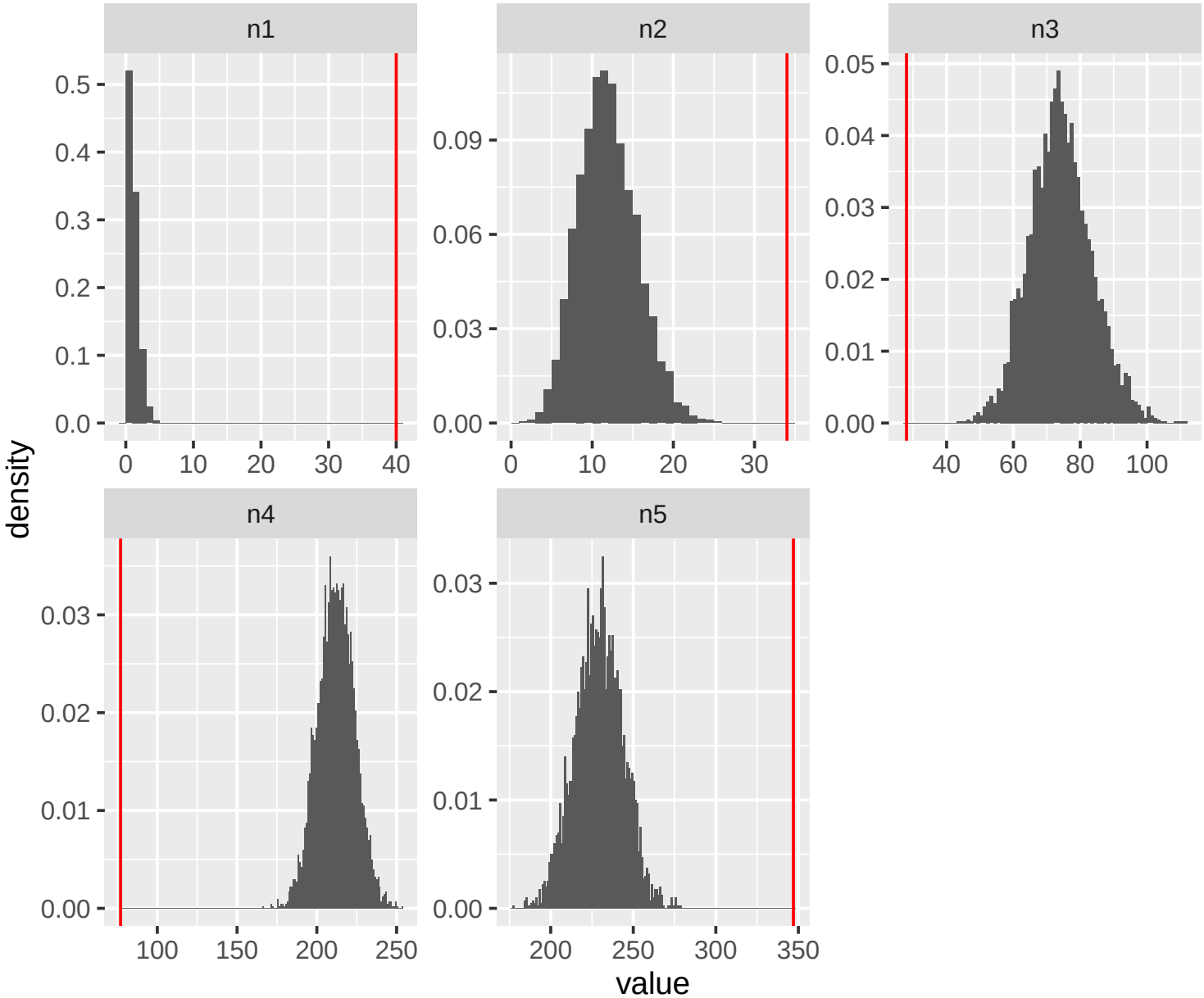
1. $\theta_2 \sim p(\theta_2|y)$,
2. For $j = 1, \ldots, 526$, $y_{2j} \overset{ind}{\sim} Bin(4, \theta_2)$, and
3. $\text{star}_{2j} = y_{2j} + 1$.

Posterior predictive distribution in R

```
theta2 = as.numeric(draws$theta[,2])
ytilde2 = adply(theta2, 1, function(x) {
  ytilde = rbinom(526, 4, x) + 1
  data.frame(n1 = sum(ytilde==1),
             n2 = sum(ytilde==2),
             n3 = sum(ytilde==3),
             n4 = sum(ytilde==4),
             n5 = sum(ytilde==5))
})
head(ytilde2)
```

```
  X1 n1 n2 n3  n4  n5
1  1  0 13 76 211 226
2  2  2 13 88 206 217
3  3  0 10 85 190 241
4  4  0 13 63 205 245
5  5  1 15 79 199 232
6  6  0  9 76 195 246
```

Posterior predictive distribution in R



Ordinal data model

Let $y_i = (y_{i1}, \ldots, y_{i5})$ be the vector of 1-star to 5-star ratings, assume

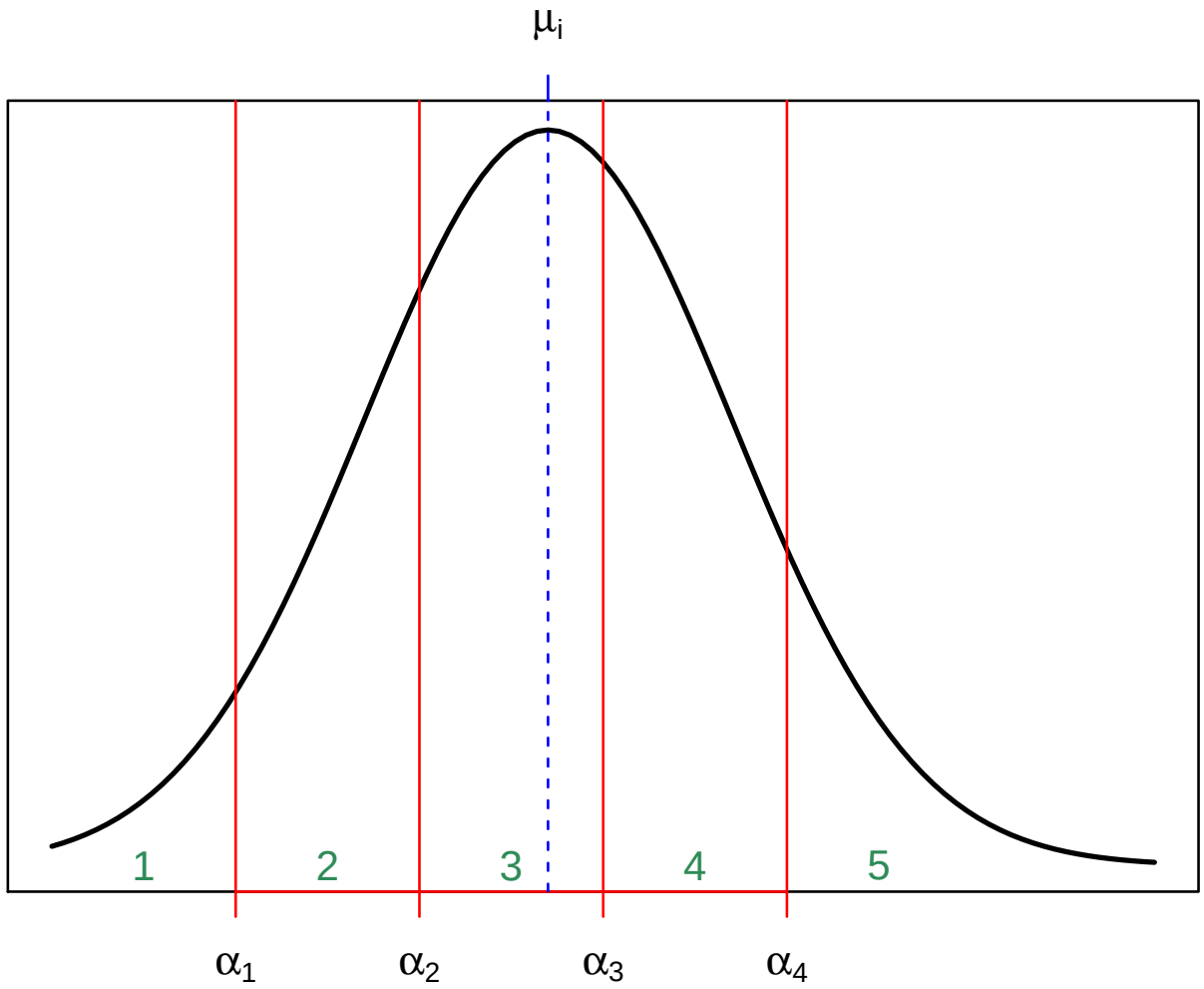
$$Y_i \overset{ind}{\sim} Mult(n_i, \theta_i)$$

where θ_i is a probability vector

$$\theta_{ik} = \int_{\alpha_{k-1}}^{\alpha_k} N(x|\mu_i, 1)dx = \Phi(\alpha_k - \mu_i) - \Phi(\alpha_{k-1} - \mu_i)$$

where $\alpha_0 = -\infty$, $\alpha_1 = 0$, and $\alpha_5 = \infty$, and Φ is the standard normal cumulative distribution function (cdf).

Visualizing the model



Hierarchical model

So each product has its own mean μ_i. The larger μ_i is the more 5-star ratings the product will receive and the fewer 1-star ratings the product will review.

In order to borrow information across different products, we might assume a hierarchical model for the μ_i, e.g.

$$\mu_i \overset{ind}{\sim} N(\eta, \tau^2)$$

with a prior

$$p(\eta, \tau) \propto Ca(\tau; 0, 1).$$

```
ordinal_model = "
data {
  int <lower=1> n_products;
  int <lower=0> y[n_products,5]; // summarized count by product
}

parameters {
  real<lower=0> alpha_diff[3];
  real mu[n_products];
  real eta;
  real<lower=0> tau;
}

transformed parameters {
  ordered[4] alpha;              // cut points
  simplex[5] theta[n_products]; // each theta vector sums to 1

  alpha[1] <- 0; for (i in 1:3) alpha[i+1] <- alpha[i] + alpha_diff[i];

  for (p in 1:n_products) {
    theta[p,1] <- Phi(-mu[p]);
    for (j in 2:4)
      theta[p,j] <- Phi(alpha[j]-mu[p]) - Phi(alpha[j-1]-mu[p]);
    theta[p,5] <- 1-Phi(alpha[4]-mu[p]);
  }
}

model {
  tau ~ cauchy(0,1);
  mu ~ normal(eta, tau);
  for (p in 1:n_products) y[p] ~ multinomial(theta[p]); // n_reviews[p] is implicit
}
"
```

Fit model

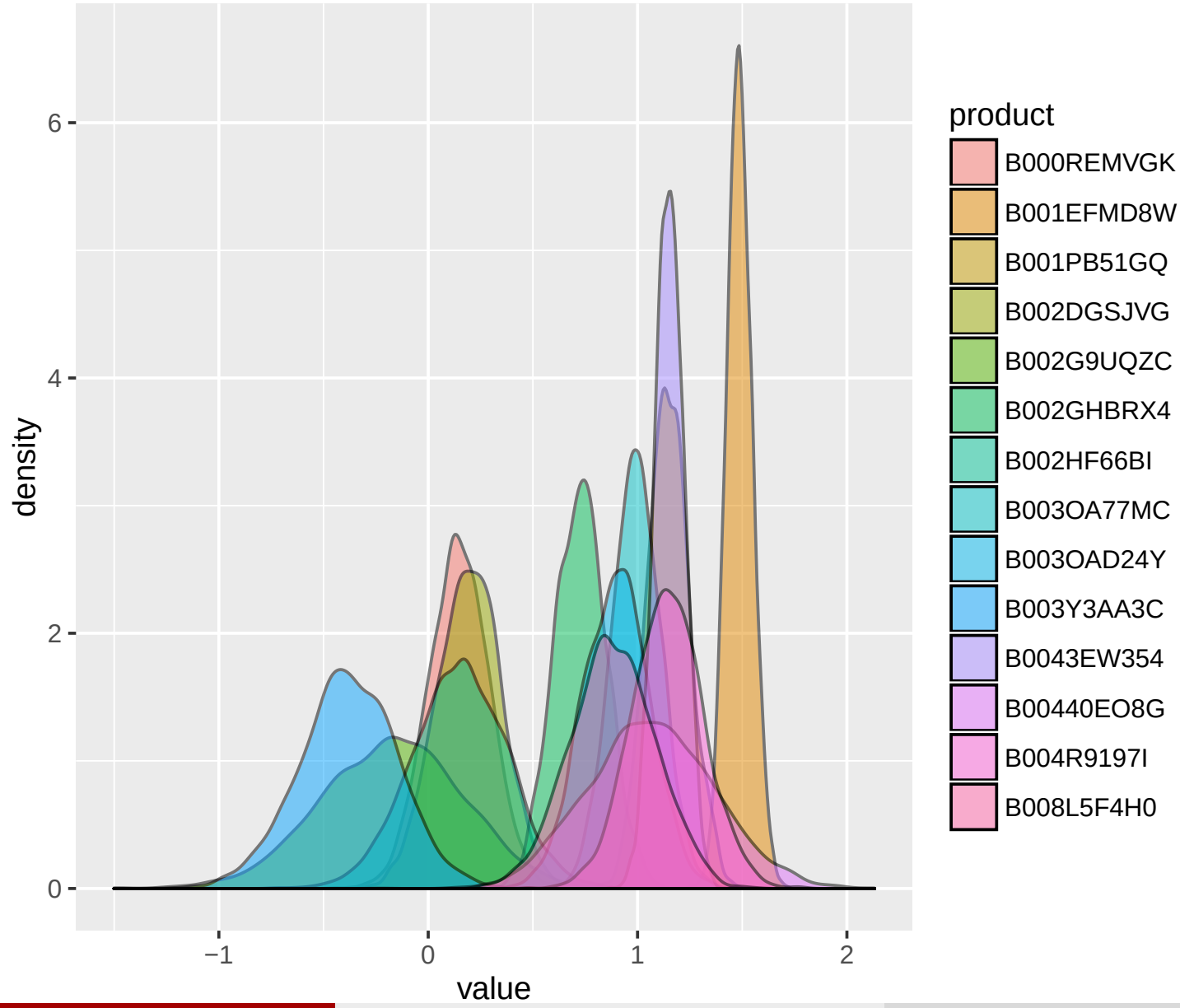
```
m = stan_model(model_code = ordinal_model)
dat = list(n_products = nrow(for_table),
           y = as.matrix(for_table[,2:6]))
r = sampling(m, dat, pars = c("alpha","eta","tau","mu"))
```

```
SAMPLING FOR MODEL '57287ba9467231c84b741a8223f095b4' NOW (CHAIN 1).

Chain 1, Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1, Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1, Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1, Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1, Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1, Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1, Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1, Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1, Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1, Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1, Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1, Iteration: 2000 / 2000 [100%]  (Sampling)#
#  Elapsed Time: 0.25 seconds (Warm-up)
#                0.21 seconds (Sampling)
#                0.46 seconds (Total)
#

SAMPLING FOR MODEL '57287ba9467231c84b741a8223f095b4' NOW (CHAIN 2).

Chain 2, Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2, Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 2, Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 2, Iteration:  600 / 2000 [ 30%]  (Warmup)
```

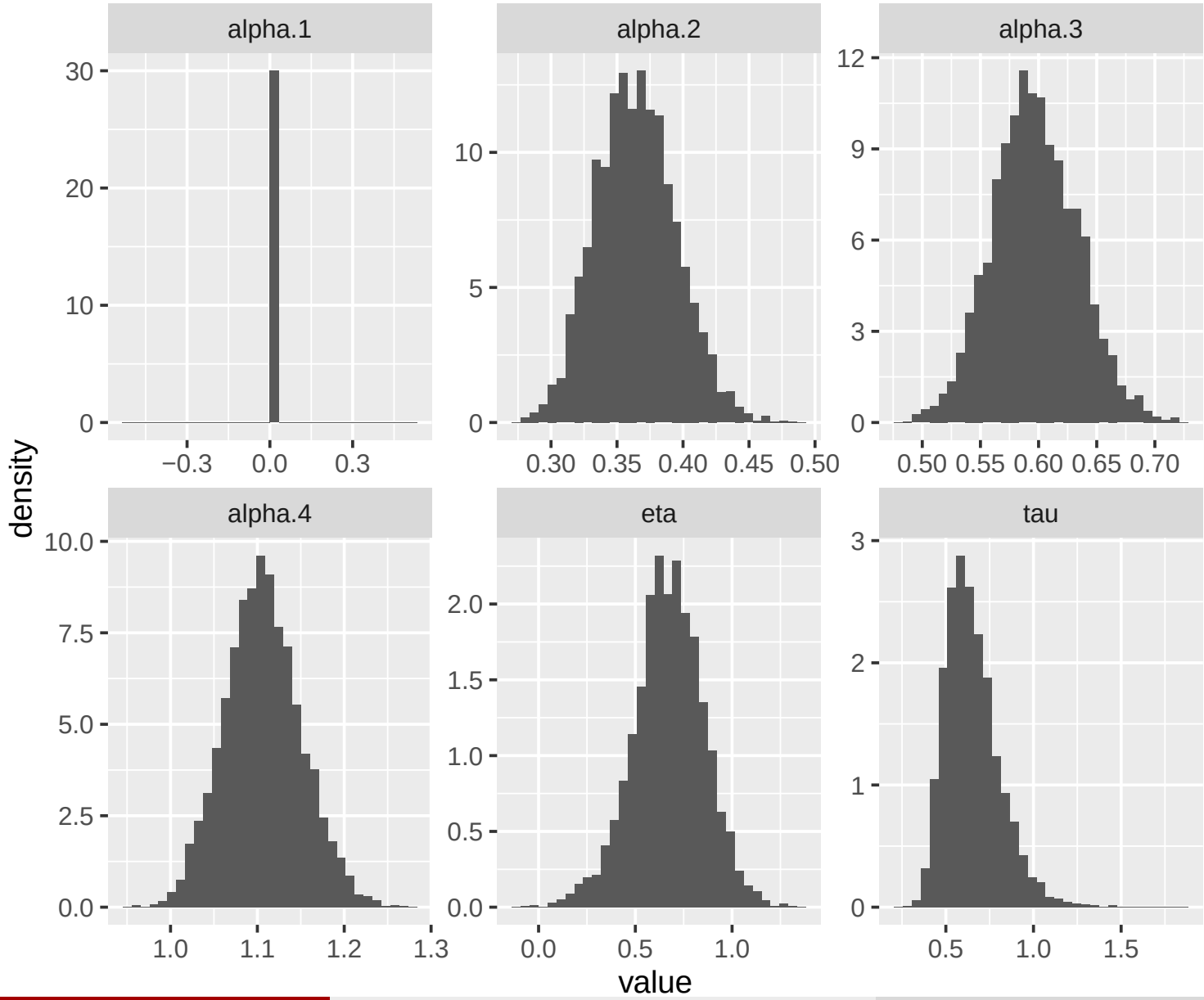
Fit model

```
r


Inference for Stan model: 57287ba9467231c84b741a8223f095b4.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

           mean se_mean   sd     2.5%      25%      50%      75%    97.5% n_eff Rhat
alpha[1]   0.00    0.00 0.00     0.00     0.00     0.00     0.00     0.00  4000  NaN
alpha[2]   0.36    0.00 0.03     0.31     0.34     0.36     0.38     0.43  3708    1
alpha[3]   0.60    0.00 0.04     0.53     0.57     0.60     0.62     0.67  3471    1
alpha[4]   1.11    0.00 0.04     1.02     1.08     1.11     1.14     1.20  3118    1
eta        0.67    0.00 0.18     0.28     0.56     0.67     0.79     1.02  4000    1
tau        0.65    0.00 0.16     0.41     0.54     0.63     0.74     1.02  3182    1
mu[1]      0.15    0.00 0.15    -0.14     0.05     0.15     0.24     0.43  4000    1
mu[2]      1.49    0.00 0.06     1.37     1.44     1.48     1.53     1.61  4000    1
mu[3]      1.15    0.00 0.10     0.96     1.08     1.14     1.21     1.35  4000    1
mu[4]      0.20    0.00 0.15    -0.10     0.09     0.20     0.30     0.49  4000    1
mu[5]     -0.17    0.01 0.34    -0.84    -0.40    -0.17     0.05     0.47  4000    1
mu[6]      0.73    0.00 0.12     0.49     0.64     0.73     0.81     0.97  4000    1
mu[7]      0.15    0.00 0.23    -0.30     0.00     0.15     0.30     0.59  4000    1
mu[8]      0.99    0.00 0.12     0.76     0.91     0.99     1.07     1.23  4000    1
mu[9]      0.90    0.00 0.16     0.59     0.79     0.90     1.00     1.20  4000    1
mu[10]    -0.38    0.00 0.23    -0.85    -0.53    -0.38    -0.22     0.07  4000    1
mu[11]     1.15    0.00 0.07     1.01     1.10     1.15     1.20     1.29  4000    1
mu[12]     1.06    0.00 0.30     0.50     0.87     1.06     1.27     1.67  4000    1
mu[13]     1.14    0.00 0.17     0.81     1.03     1.14     1.26     1.48  4000    1
mu[14]     0.88    0.00 0.20     0.48     0.75     0.88     1.02     1.27  4000    1
lp__   -1835.74    0.08 3.07 -1842.64 -1837.52 -1835.39 -1833.56 -1830.79  1479    1

Samples were drawn using NUTS(diag_e) at Tue Mar  1 10:17:31 2016.
For each parameter, n_eff is a crude measure of effective sample size,
```

Review mean posteriors (θ_i)



Other parameter posteriors



Visualizing the model