Hierarchical Models and Shrinkage Estimation

Professor Karsten T. Hansen

UC San Diego, Rady School of Management
MGTA 495, Spring 2022

Example: Uber Trip Data

- 37,961 Uber trips for 1,909 users
- Goal: Identify users who on average take high value trips
- This seems like an easy exercise right?

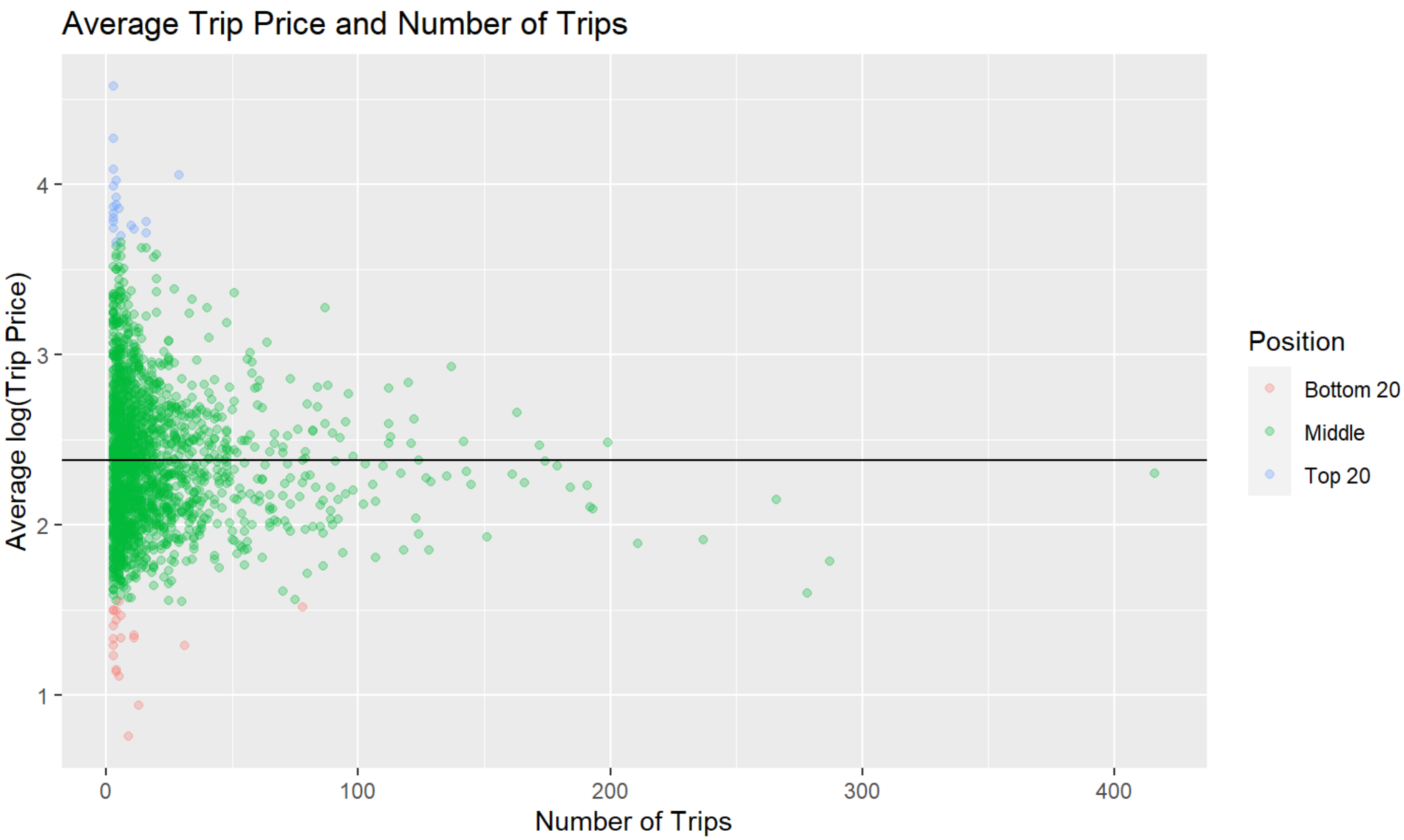
Top 20

```
## # A tibble: 20 x 7
##    userId nTrips avgLogPrice userIdF userIndex empiricalRank Position
##     <dbl>  <int>       <dbl> <fct>       <int>         <dbl> <chr>
##  1  13907      3        4.57 13907        1316             1 Top 20
##  2   9755      3        4.27 9755         1071             2 Top 20
##  3   1849      3        4.09 1849          233             3 Top 20
##  4  13138     29        4.05 13138        1282             4 Top 20
##  5  15994      4        4.02 15994        1414             5 Top 20
##  6   2868      3        3.99 2868          390             6 Top 20
##  7    641      4        3.92 641            48             7 Top 20
##  8   9108      4        3.88 9108         1025             8 Top 20
##  9   3402      3        3.87 3402          465             9 Top 20
## 10  20446      5        3.86 20446        1567            10 Top 20
## 11  15664      3        3.83 15664        1397            11 Top 20
## 12   6885      3        3.80 6885          841            12 Top 20
## 13  18739      3        3.78 18739        1501            13 Top 20
## 14    442     16        3.78 442            36            14 Top 20
## 15  28191     10        3.76 28191        1799            15 Top 20
## 16  15796      3        3.74 15796        1403            16 Top 20
## 17  20822     11        3.74 20822        1579            17 Top 20
## 18   4094     16        3.71 4094          552            18 Top 20
## 19   1837      6        3.70 1837          228            19 Top 20
## 20   9656      4        3.66 9656         1067            20 Top 20
```

Bottom 20

```
## # A tibble: 20 x 7
##    userId nTrips avgLogPrice userIdF userIndex empiricalRank Position
##     <dbl>  <int>       <dbl> <fct>       <int>         <dbl> <chr>
##  1  19288      9       0.759 19288        1526          1909 Bottom 20
##  2   5880     13       0.942 5880          749          1908 Bottom 20
##  3  17527      5       1.11  17527        1459          1907 Bottom 20
##  4   1052      4       1.13  1052          147          1906 Bottom 20
##  5   1106      4       1.15  1106          155          1905 Bottom 20
##  6  12035      3       1.23  12035        1217          1904 Bottom 20
##  7   7851     31       1.29  7851          934          1903 Bottom 20
##  8  26257      3       1.29  26257        1747          1902 Bottom 20
##  9    726      3       1.33  726            68          1901 Bottom 20
## 10  16178     11       1.33  16178        1424          1900 Bottom 20
## 11   2495      6       1.33  2495          328          1899 Bottom 20
## 12   6744     11       1.35  6744          823          1898 Bottom 20
## 13  14161      3       1.41  14161        1330          1897 Bottom 20
## 14   5861      4       1.44  5861          747          1896 Bottom 20
## 15  13819      6       1.47  13819        1313          1895 Bottom 20
## 16   6632      3       1.49  6632          814          1894 Bottom 20
## 17  29331      4       1.49  29331        1838          1893 Bottom 20
## 18   7816      3       1.50  7816          929          1892 Bottom 20
## 19    892     78       1.52  892           110          1891 Bottom 20
## 20  22802      5       1.55  22802        1643          1890 Bottom 20
```

Raw Averages



General Problem!

- Whenever you want to compare averages of something, this problem will show up:

  Averages based on a small number of observations will always be more noisy than averages over a large number of observations

- This means that the extremes will almost always be made up of the averages based on a small number of observations

- Solution: Users are similar but different from each other. We should use this information to our advantage.

- Idea: We should “shrink” the raw means toward the overall mean in a way so that the noisy averages are shrunk more than the non-noisy averages:

$$\hat{\alpha}_i = F_i(\bar{Y}_i)$$

- Standard approach takes $F_i(\bar{Y}_i) = \bar{Y}_i$ but as we have seen that is a bad idea!

- We need $F_i(\bar{Y}_i) < \bar{Y}_i$ for large noisy averages ($F_i(\bar{Y}_i) > \bar{Y}_i$ for small noisy averages) and $F_i(\bar{Y}_i) \approx \bar{Y}_i$ for non-noisy averages

Hiearchical Model

$$
\begin{aligned}
y_{ij}|\alpha_i, \sigma &\sim \mathrm{N}(\alpha_i, \sigma^2), \qquad j = 1, \ldots, N_i; i = 1, \ldots, N, \\
\alpha_i|\mu, \sigma_\alpha &\sim \mathrm{N}(\mu, \sigma_\alpha^2),
\end{aligned}
$$

plus a prior distribution for $\sigma, \mu, \sigma_\alpha$.

- This is also called a Multilevel Model or a model with partial pooling
- We are primarily interested in the posterior distribution of $\{\alpha_i\}_{i=1}^N$ but also in σ_α.
- What happens when $\sigma_\alpha \to 0$ and $\sigma_\alpha \to \infty$?

Example

- Suppose initially that $\sigma, \mu, \sigma_\alpha$ are known. What is the posterior for α_i?
- We can derive this analytically:

$$p(\alpha_i | Y_i, \mu, \sigma, \sigma_\alpha) = \mathrm{N}(\alpha_i | \mu_{\alpha_i}, \tau_{\alpha_i}^{-1}),$$

where

$$\begin{aligned} \tau_{\alpha_i} &\equiv \frac{N_i}{\sigma^2} + \frac{1}{\sigma_\alpha^2}, \\ \mu_{\alpha_i} &\equiv \tau_{\alpha_i}^{-1} \left(\frac{N_i}{\sigma^2} \bar{Y}_i + \frac{1}{\sigma_\alpha^2} \mu \right), \end{aligned}$$

and $\bar{Y}_i \equiv N_i^{-1} \sum_j Y_{ij}$

Special Cases

- We can consider two special cases: $\sigma_\alpha \to 0$ and $\sigma_\alpha \to \infty$
- The posterior mean of α_i in these two special cases is

$$\mathrm{E}[\alpha_i | Y_i, \mu, \sigma_\alpha, \sigma] \to \begin{cases} \bar{Y}_i & \text{for } \sigma_\alpha \to \infty, \\ \mu & \text{for } \sigma_\alpha \to 0. \end{cases}$$

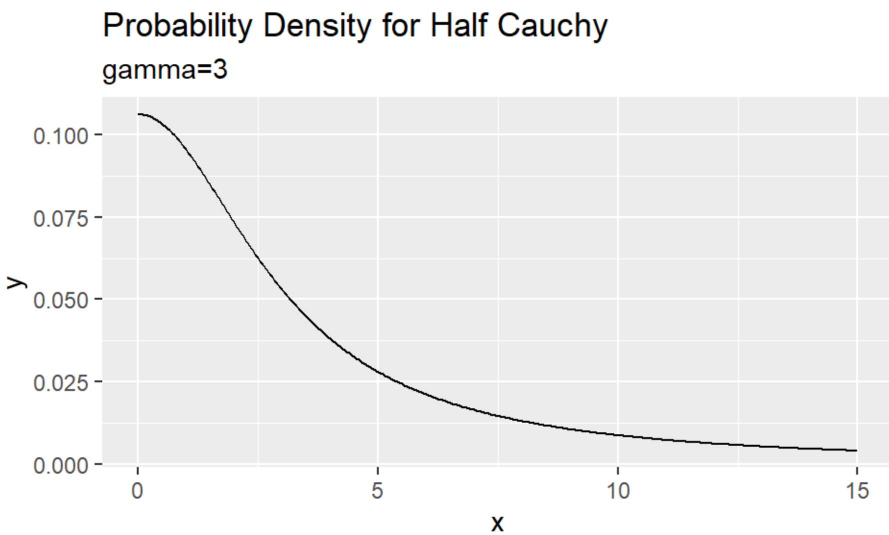
- These cases are also referred to as
  - $\sigma_\alpha \to \infty$ = "No Pooling" (everyone is completely different)
  - $\sigma_\alpha \to 0$ = "Complete Pooling" (everyone is the same)
  - $0 < \sigma_\alpha < \infty$ = "Partial pooling" (everyone is different but similar)
- Preferred approach: Let the data help in determining size of σ_α!
  - If there is evidence in the data that all users are very similar, then we should learn that σ_α is small
  - If there is evidence in the data that users are very different, then we should learn that σ_α is large

Full Model

$$\begin{aligned} y_{ij}|\alpha_i,\sigma &\sim \mathrm{N}(\alpha_i,\sigma^2), \qquad j=1,\dots,N_i; i=1,\dots,N, \\ \alpha_i|\mu,\sigma_\alpha &\sim \mathrm{N}(\mu,\sigma_\alpha^2), \\ \mu &\sim \mathrm{N}(0,5^2), \\ \sigma &\sim \mathrm{Cauchy}_+(0,3), \\ \sigma_\alpha &\sim \mathrm{Cauchy}_+(0,3) \end{aligned}$$

Half Cauchy

$$p(x|\gamma) = \frac{1}{\pi\gamma\left(1 + (\frac{x}{\gamma})^2\right)}$$



Finding the Posterior

- This model - while simple - it already too complex to easily solve analytically
- Note that the joint posterior is a distribution of

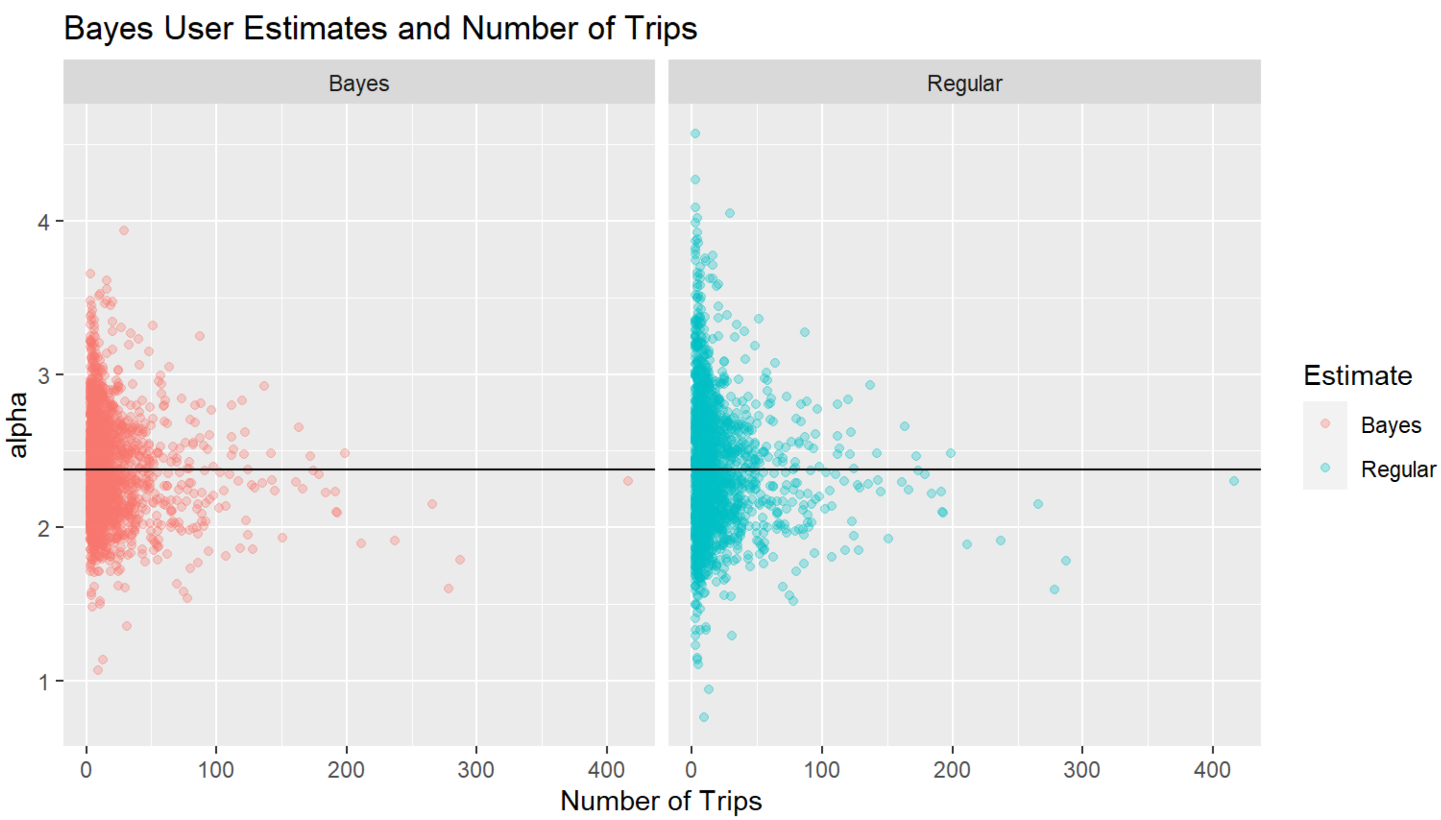
$$(\alpha_1, \ldots, \alpha_N, \mu, \sigma, \sigma_\alpha),$$

  i.e., $N + 3$ parameters where N is the number of users. Therefore this is a probability distribution in close to 2,000 dimensions!
- We will use a Monte Carlo algorithm to numerically simulate this posterior distribution
- This algorithm will simulate the joint posterior by drawing a sequence of S pseudo-random numbers from the posterior distribution: $\{\tilde{\alpha}_s, \tilde{\mu}_s, \tilde{\sigma}_{\alpha,s}, \tilde{\sigma}_s\}_s^S$
- We will map the model into the probabilistic language `Stan`
- We will not go into the details here - more on that next class!

Stan code

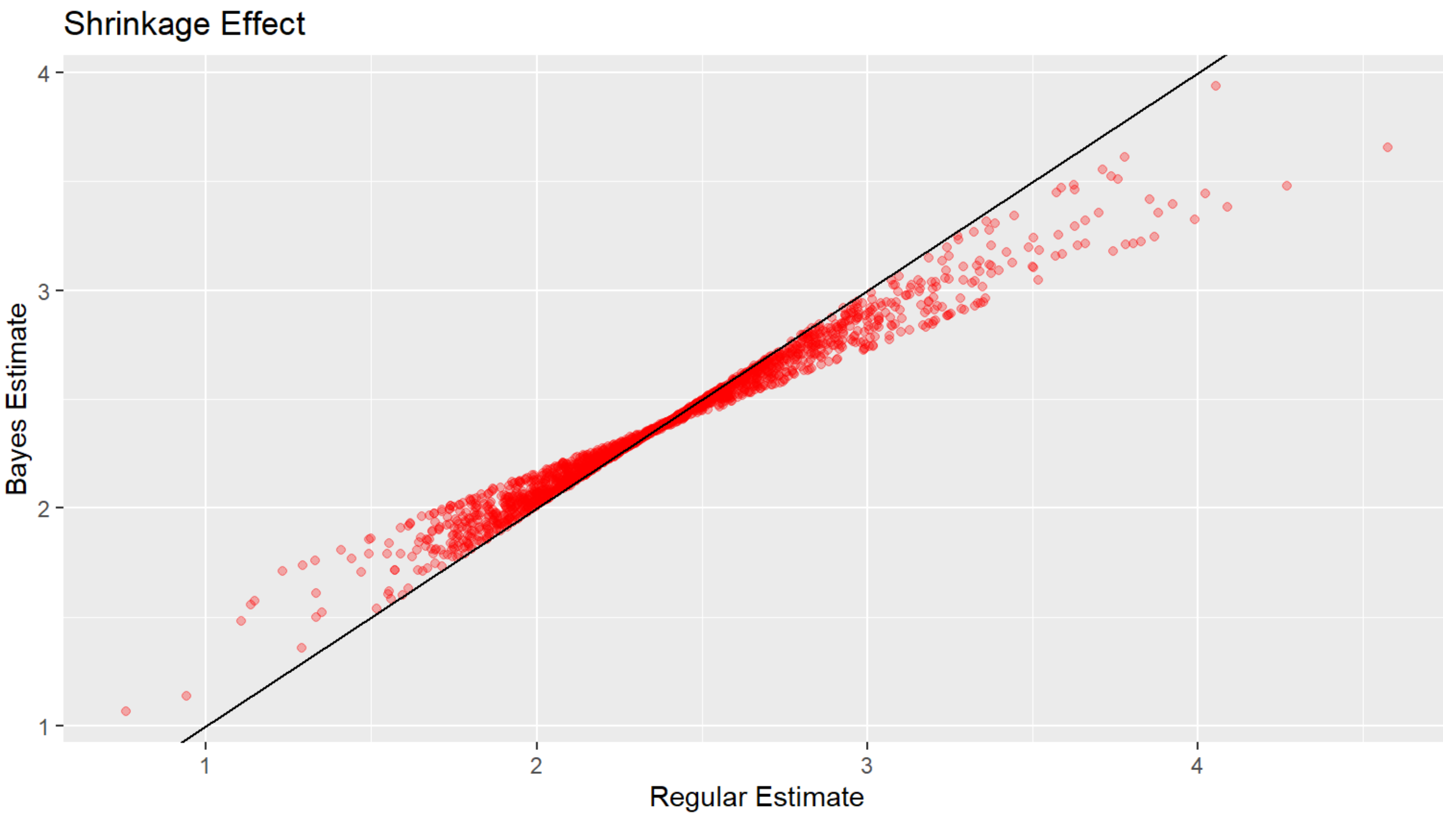
```
data {
  int<lower=0> nObs;                          // number of rows in full data
  int<lower=0> nUsers;                        // number of users
  int<lower=1,upper=nUsers> userID[nObs];     // user index for each row
  vector[nObs] y;                             // log amount
}

parameters {
  real<lower=0> sigma;          // sd alpha
  real mu;                      // mean alpha
  vector[nUsers] alpha;         // user effects
  real<lower=0> sigma_y;        // sd data
}

model {
  sigma ~ cauchy(0, 2.5);
  mu ~ normal(0,5);
  alpha ~ normal(mu, sigma);
  sigma_y ~ cauchy(0, 2.5);

  y ~ normal(alpha[userID], sigma_y);
}
```

Result



Bayes User Estimates and Number of Trips
Bayes
Regular
alpha
Number of Trips
0
100
200
300
400
1
2
3
4
Estimate
Bayes
Regular

Result


Shrinkage Effect
Bayes Estimate
Regular Estimate
1
2
3
4
1
2
3
4

How did we make this?

- The algorithm generates a stream of pseudo random numbers from the distribution we are interested in
- Why is this useful? Note that if $\{\tilde{x}_s\}_{s=1}^S$ are random draws from a distribution π then

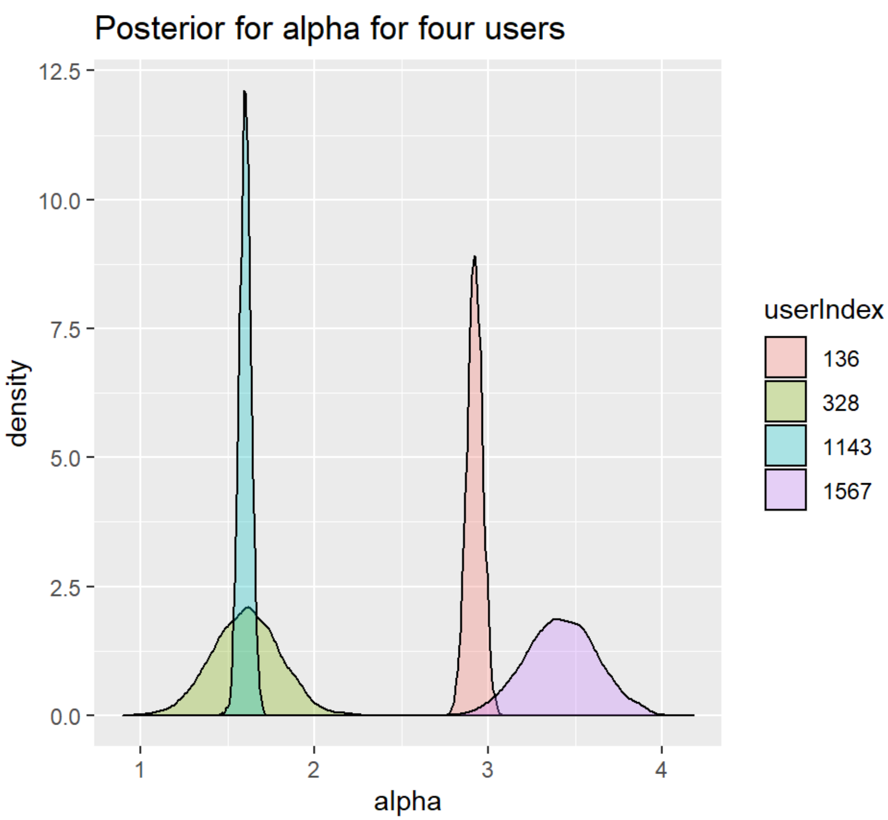
$$\mathrm{E}_\pi[f(x)] \approx \frac{1}{S}\sum_{s=1}^{S} f(\tilde{x}_s),$$

  for a function f, where we can control the approximation error by choosing large enough enough M.
- This is called Monte Carlo simulation
- Almost all Bayesian models are trained this way
- Next week we will look at the details of how these algorithms are designed

Using Draws

- Posterior average: $\bar{\tilde{x}} = \frac{1}{S}\sum_{s=1}^{S} \tilde{x}_s$
- Posterior standard deviation: $\sqrt{\frac{1}{S}\sum_{s=1}^{S} (\tilde{x}_s - \bar{\tilde{x}})^2}$
- Posterior quantiles: Empirical quantiles of $\{\tilde{x}_s\}_{s=1}^{S}$
- Posterior distribution: histogram or density of $\{\tilde{x}_s\}_{s=1}^{S}$

Using Draws




```
## # A tibble: 4 x 6
##   userIndex value .lower .upper avgLogPrice nTrips
##       <int> <dbl>  <dbl>  <dbl>       <dbl>  <int>
## 1       136  2.92   2.84   3.01        2.93    137
## 2       328  1.61   1.23   1.97        1.33      6
## 3      1143  1.60   1.54   1.67        1.60    278
## 4      1567  3.42   3.02   3.82        3.86      5
```

Using Draws: Posterior Ranks

- Suppose want to identify the top 15 users in terms of value (measured as average spend per trip)
- This is a question about rank. For example the top ranked user is

$$user_1(\alpha_1, \ldots, \alpha_N) \equiv \{i : \alpha_i \geq \alpha_j,\ \forall j \in \{1, \ldots, N\}\}$$

- Note that the rank depends on the unknown parameters $\alpha_1, \ldots, \alpha_N$.
- We can use the posterior draws of the α vector to simulate posterior ranks:
  - For each draw $\tilde{\alpha}$ find the rank of each user: $\tilde{r}_1, \ldots, \tilde{r}_N$
  - This creates S draws of each user's ranking
  - Then we can summarize these S draws and find the mean rank, min rank, max rank etc for each user

Posterior Ranks

```
## # A tibble: 15 x 7
##    userIndex meanPostRank minPostRank maxPostRank nTrips avgLogPrice
##        <int>        <dbl>       <dbl>       <dbl>  <int>       <dbl>
##  1      1282         1.35           1           7     29        4.05
##  2        36         7.43           1          51     16        3.78
##  3      1316         9.89           1         154      3        4.57
##  4       552        10.1            1          78     16        3.71
##  5      1579        12.6            1          85     11        3.74
##  6      1799        13.8            1         103     10        3.76
##  7       877        14.4            1          79     16        3.63
##  8      1129        14.7            1          66     20        3.59
##  9       879        16.5            2         107     14        3.63
## 10      1743        16.6            1          75     19        3.57
## 11      1071        21.5            1         361      3        4.27
## 12      1414        22.4            1         218      4        4.02
## 13      1567        24.7            1         327      5        3.86
## 14       271        27.1            2         161     20        3.44
## 15      1375        28.2            6          68     51        3.36
## # ... with 1 more variable: empiricalRank <dbl>
```

Multilevel Regression Model

- The basic idea of shrinkage estimation can be applied to any model
- Let's consider a regression model with a multilevel/hierarchical structure:

$$
\begin{aligned}
y_{ij}|\alpha_i, \beta_i, \sigma &\sim \mathrm{N}(\alpha_i + \beta_i x_i, \sigma^2), \qquad j = 1, \ldots, N_i; i = 1, \ldots, N, \\
\alpha_i, \beta_i|\mu, \Sigma &\sim \mathrm{N}(\mu, \Sigma),
\end{aligned}
$$

- Note that without the second stage, this would be like training an independent regression model for each i

- The model is closed by specifying a prior for σ, μ, Σ.
- Note that

$$
\Sigma \to \begin{cases} 0 & \text{One pooled regression,} \\ \infty & N \text{ independent regressions} \end{cases}
$$

Case Study: Multilevel Demand Model

- Weekly sales and prices of Frito Lay Pretzels for 76 Stores
- 3 Years of data
- We want to allow stores to have different baseline sales and different demand price effects

$$\begin{aligned} \log y_{sw} &= \alpha_s + \beta_s \log p_{sw} + \varepsilon_{sw}, \\ \varepsilon_{sw} | \sigma &\sim \mathrm{N}(0, \sigma^2), \\ \alpha_s, \beta_s | \mu, \Sigma &\sim \mathrm{N}(\mu, \Sigma), \end{aligned}$$

where y_{sw} is sales volume for store s in week w, and p_{sw} is the brand price for store s in week w

- Let's try two different priors:
  - A shrinkage prior where we allow the model to learn the degree to which stores are similar
  - An independence prior with zero pooling, i.e., we treat the 76 stores as independent

Priors

- To specify a prior on the covariance matrix Σ, we use the decomposition

$$\Sigma \equiv \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix} = \begin{bmatrix} \tau_1 & 0 \\ 0 & \tau_2 \end{bmatrix} \times \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \times \begin{bmatrix} \tau_1 & 0 \\ 0 & \tau_2 \end{bmatrix},$$

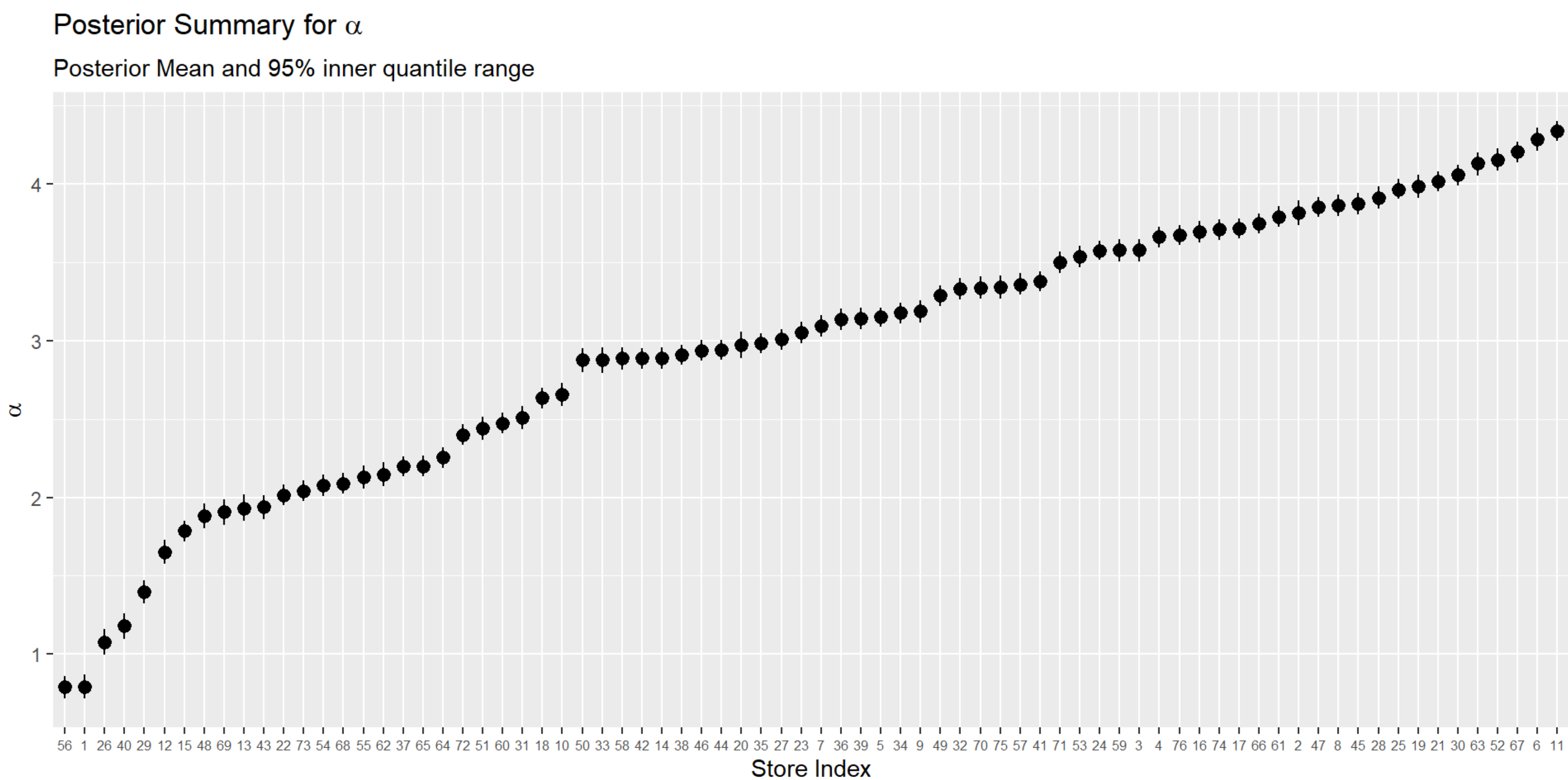
  where ρ is the correlation coefficient between α_i and β_i.

- Priors are then assigned to τ_1, τ_2 and ρ:

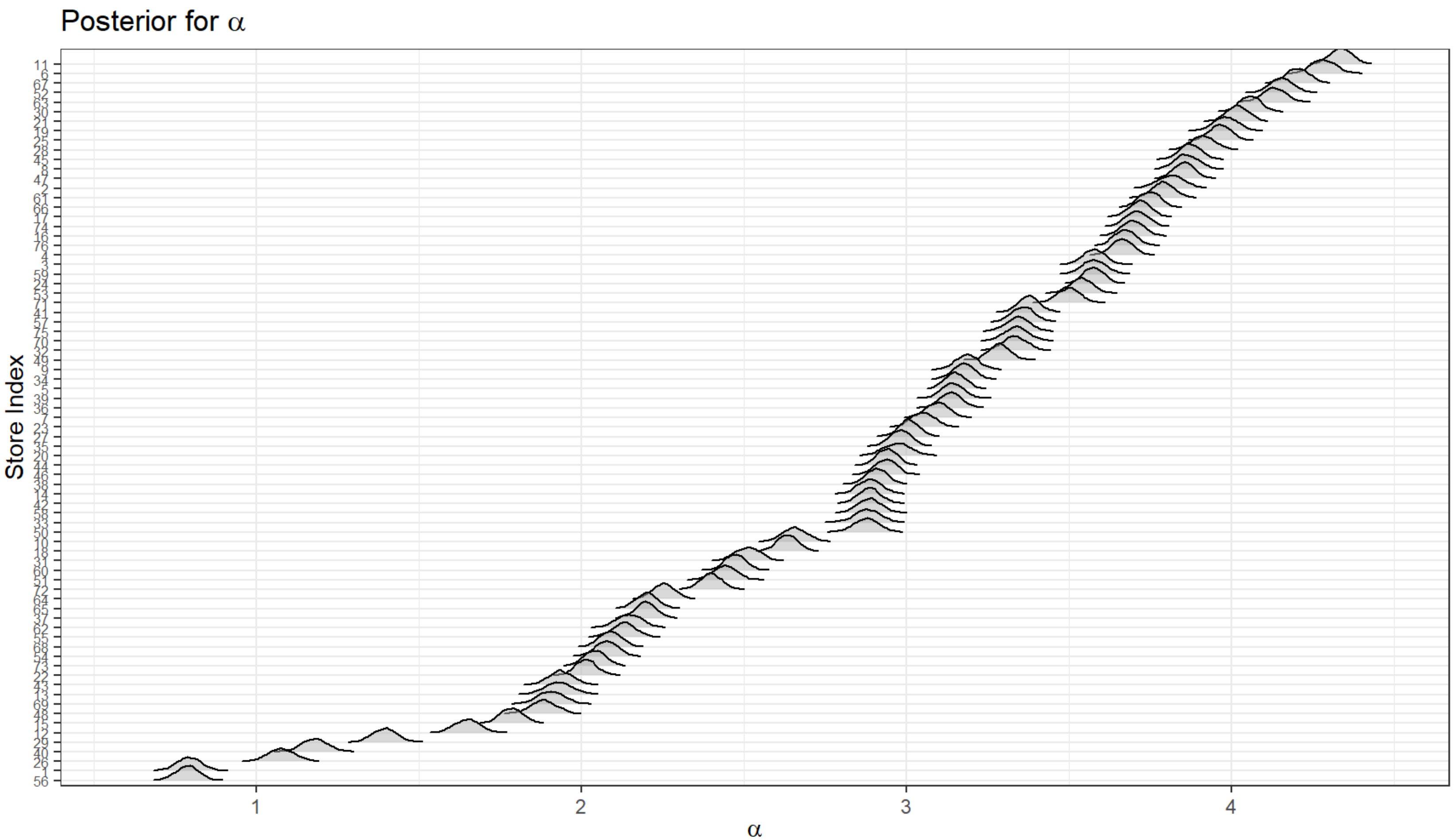
$$\begin{aligned} \tau_1 &\sim \mathrm{Cauchy}_+(0, 2.5), \quad \tau_2 \sim \mathrm{Cauchy}_+(0, 2.5) \\ \rho &\sim \mathrm{U}(-1, 1). \end{aligned}$$

- For the independence prior we just assign independent diffuse (meaning large variance) normal distributions for α_s and β_s, e.g., $\alpha_s \sim \mathrm{N}(0, 10),\ \beta_s \sim \mathrm{N}(0, 10)$

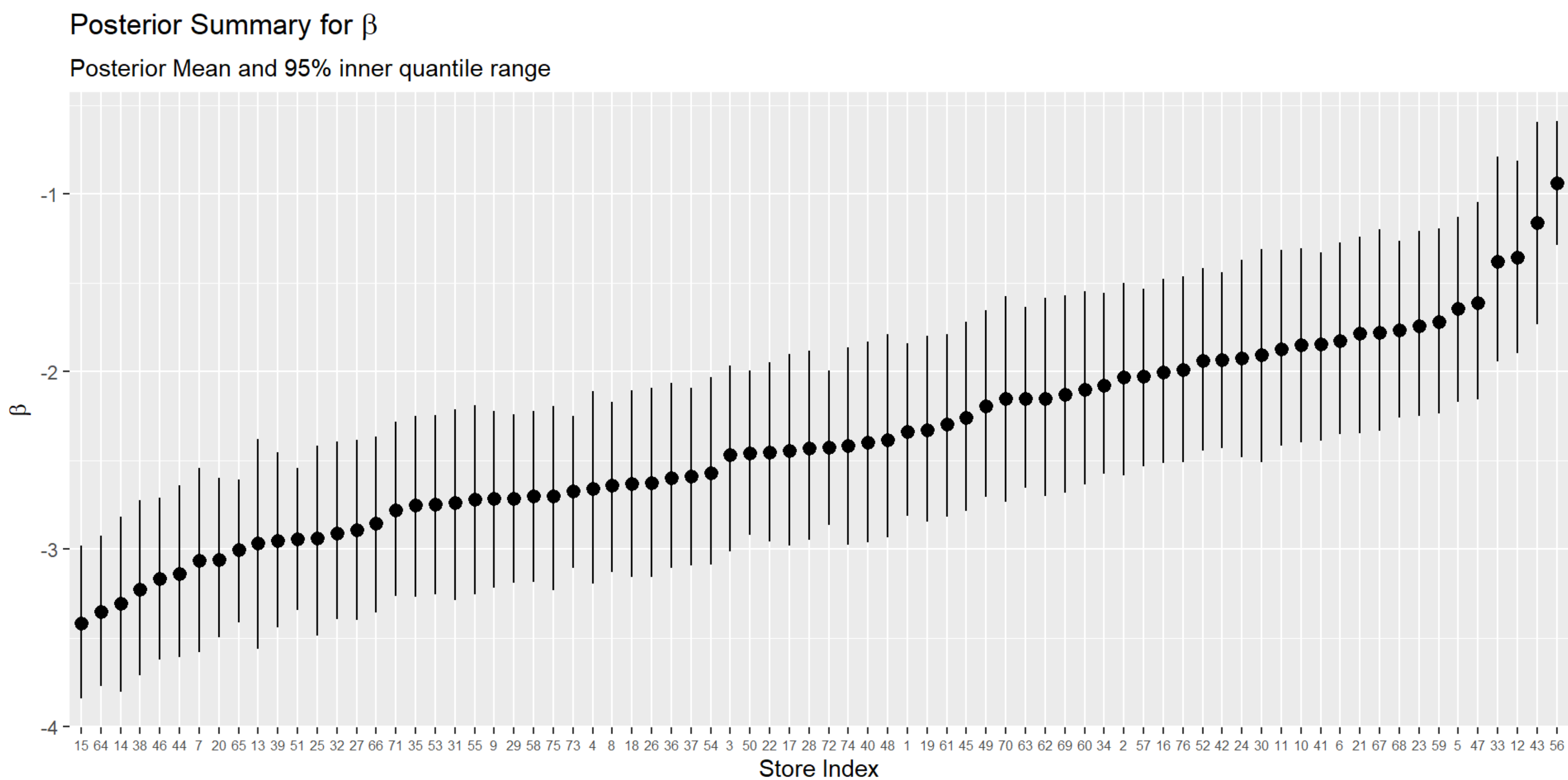
Results (With Shrinkage)



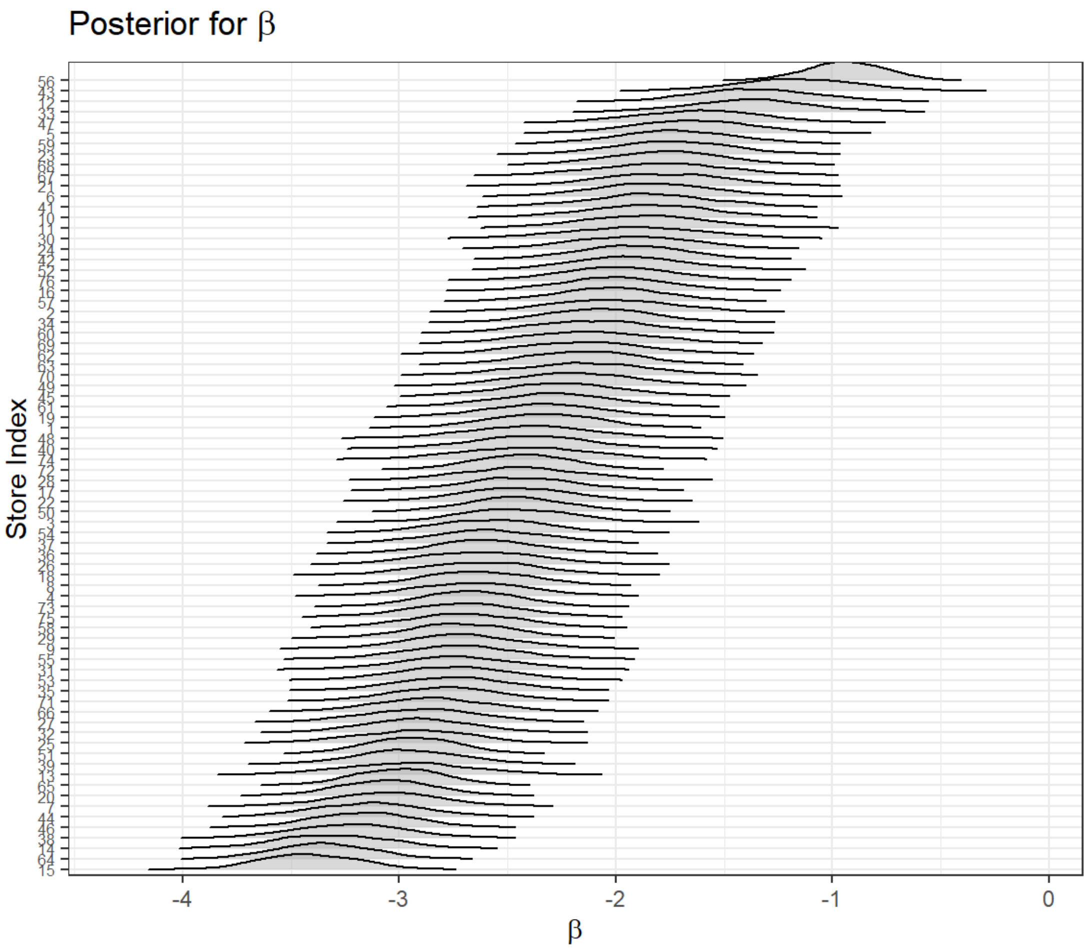
Results (With Shrinkage)



Results (With Shrinkage)



Results (With Shrinkage)



Covariation

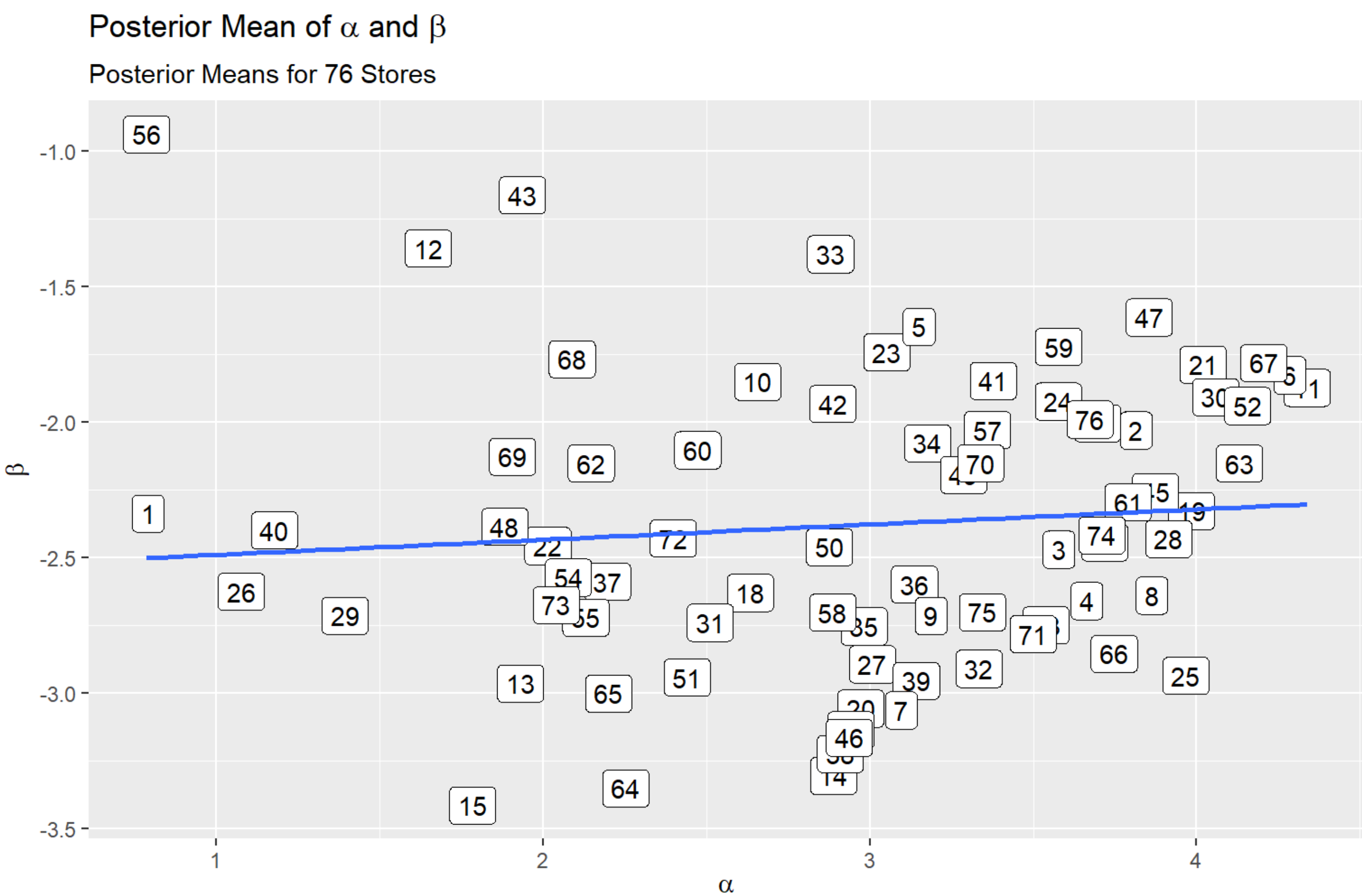
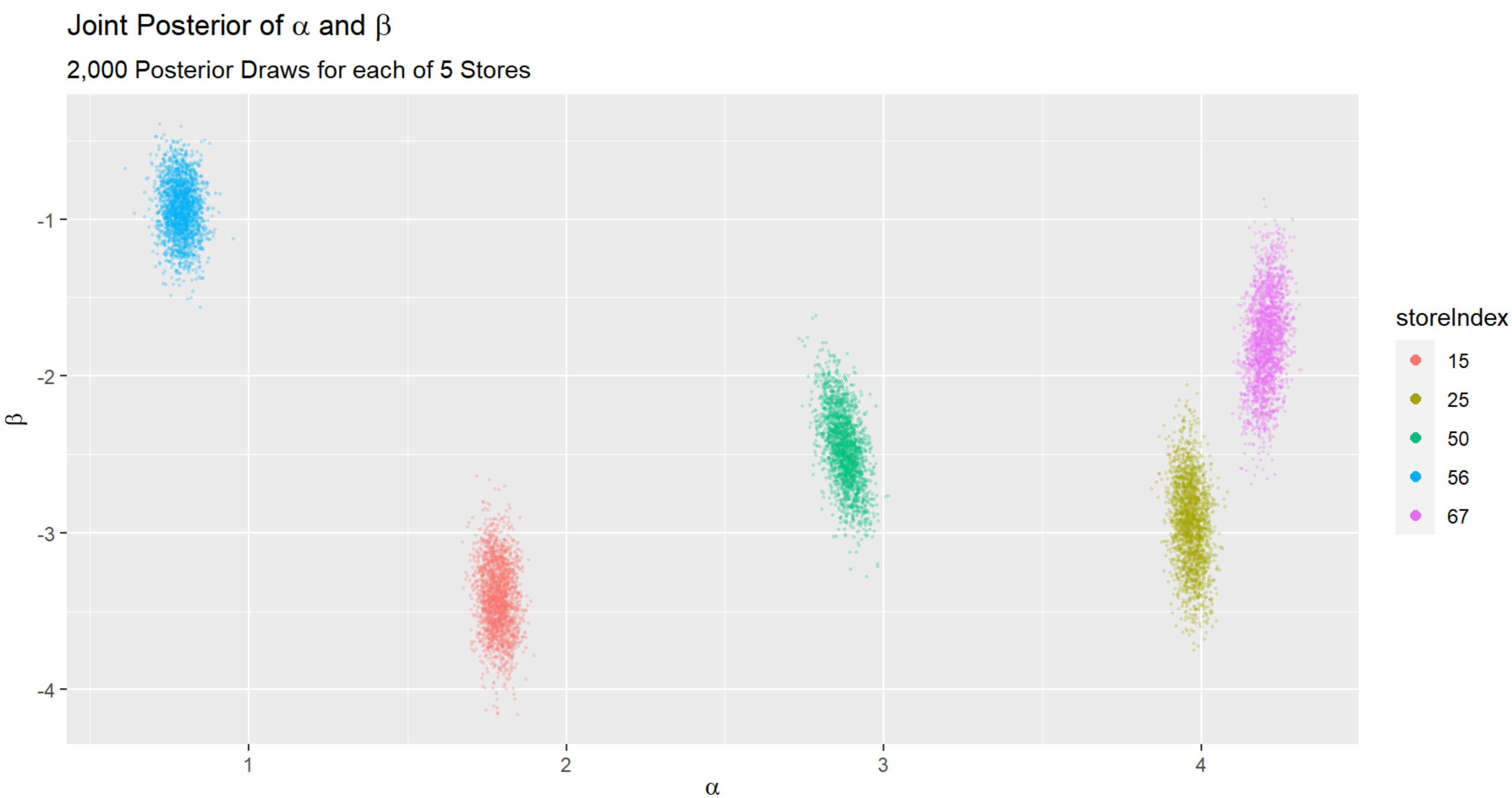
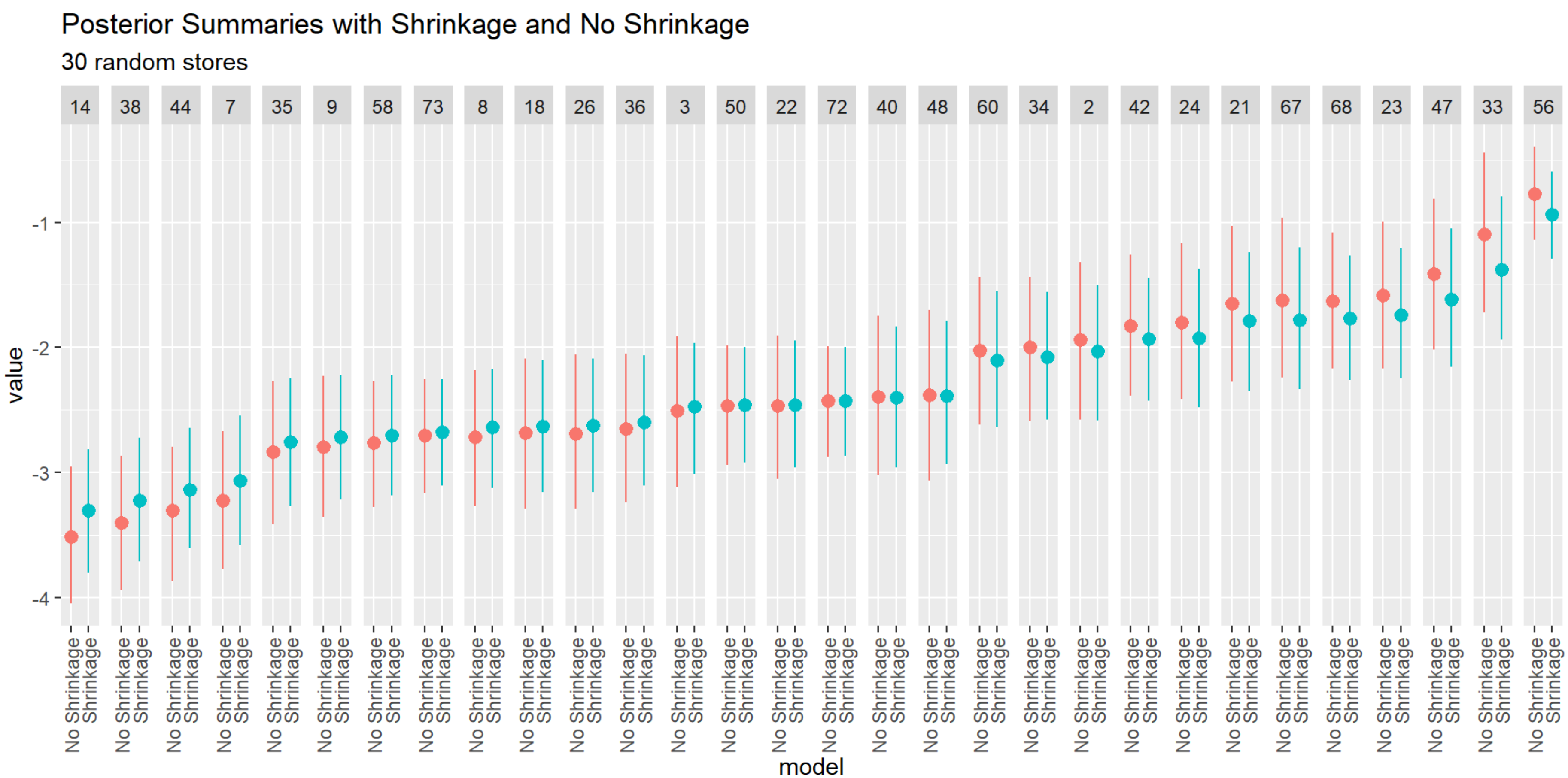


Illustration of Posterior Uncertainty




Posterior Summaries with Shrinkage and No Shrinkage
30 random stores
14
38
44
7
35
9
58
73
8
18
26
36
3
50
22
72
40
48
60
34
2
42
24
21
67
68
23
47
33
56
value
-1
-2
-3
-4
No Shrinkage
Shrinkage
model

Expected Demand

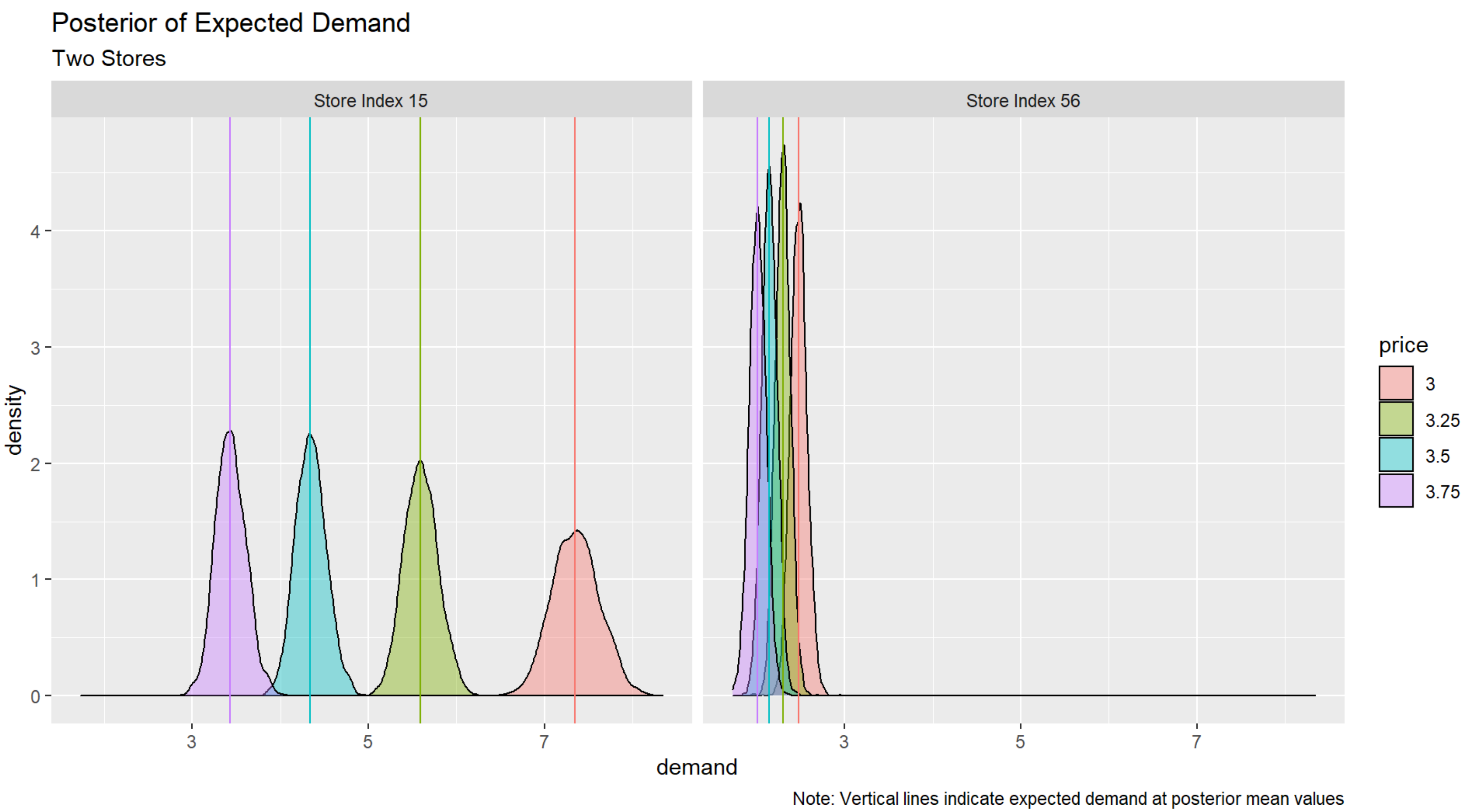
- Let's try to use the model to predict expected future demand Y_s^* for a store s
- Note that the generative model of sales is a log-normal distribution. Therefore,

$$\mathrm{E}[Y_s^*|\theta_s, \sigma] = \exp\left\{\alpha_s + \beta_s \log p + \frac{\sigma^2}{2}\right\} \equiv g(\theta_s, \sigma; p)$$

- Notice that this is a simple function of $\theta_s \equiv (\alpha_s, \beta_s)$ and p.
- Since our algorithm already provides us draws from the posterior for θ_s, σ, we can easily generated draws of g simply as

$$\left\{g(\tilde{\theta}_{sd}, \tilde{\sigma}_d; p)\right\}_{d=1}^{D}$$

- By varying p we can then easily trace out the effect of price changes on expected demand



Posterior of Expected Demand
Two Stores
Store Index 15
Store Index 56
density
demand
0
1
2
3
4
3
5
7
price
3
3.25
3.5
3.75
Note: Vertical lines indicate expected demand at posterior mean values

Price Setting?

Full Uncertainty

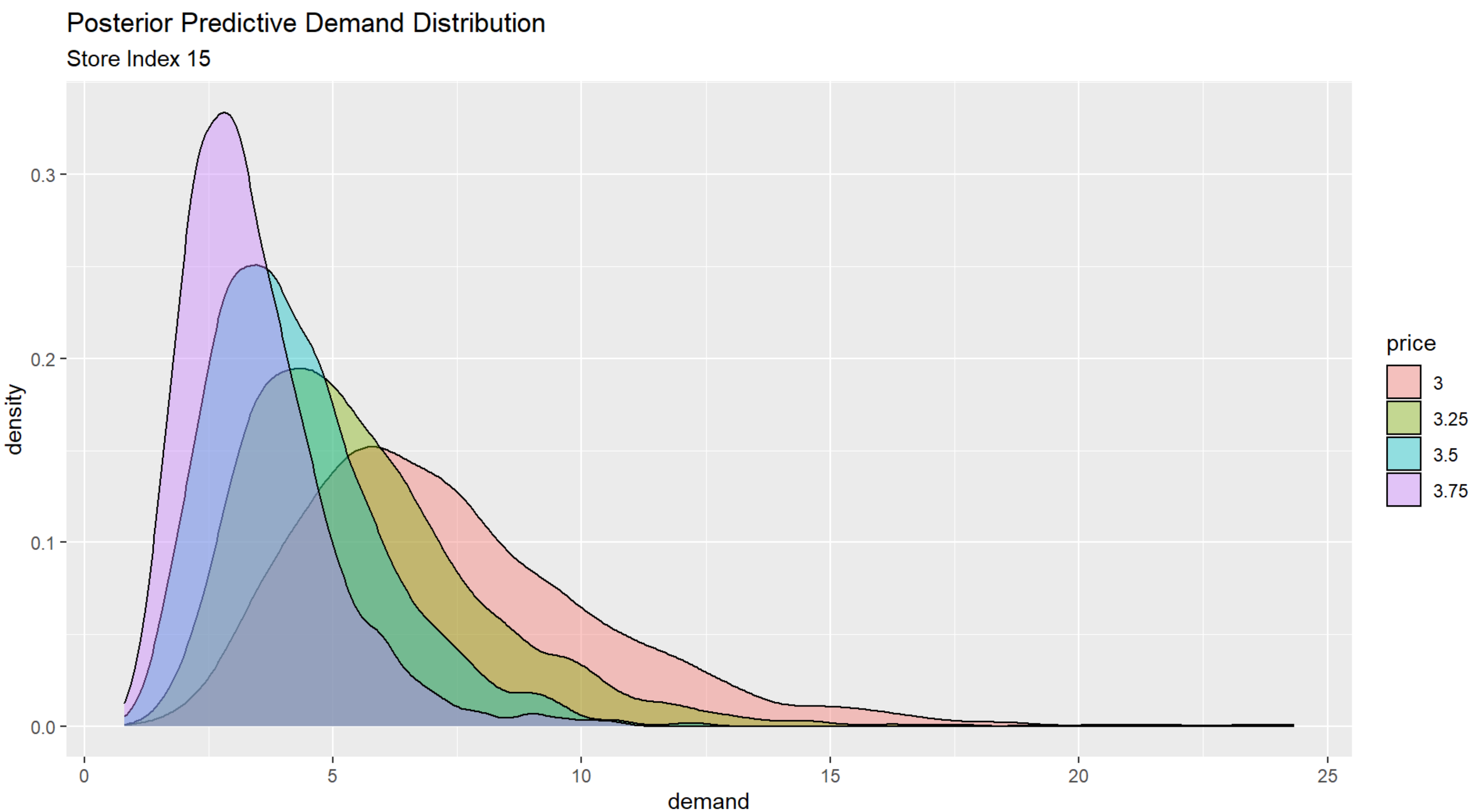
- What is the full uncertainty facing the store about next week's demand?
- This involves two sources: model uncertainty and the specific draw of demand that will materialize conditional on a specific model
- The answer is the posterior predictive distribution:

$$p(Y_s{}^*|p, \text{data}) = \int p(Y_s{}^*|p, \theta_s, \sigma)p(\theta_s, \sigma|\text{data})d\theta_s d\sigma$$

- We can simulate this quite easily:

1. For each simulated draw $\tilde{\theta}_s, \tilde{\sigma}$,
2. Sample $\tilde{Y}_s^*$ as

$$\tilde{Y}^* \sim \text{LogNormal}(\tilde{\alpha}_s + \tilde{\beta}_s \log p, \tilde{\sigma})$$


Posterior Predictive Demand Distribution
Store Index 15
density
0.0
0.1
0.2
0.3
0
5
10
15
20
25
demand
price
3
3.25
3.5
3.75

Model 2: Explain variation

- Our model above naturally incorporates variation in parameters across stores
- Can we explain this variation? Why are some stores price sensitive and others not? Why do some stores have low baseline sales?
- We could do some simple correlations/regressions of parameter estimates on store characteristics….BUT…a much better approach is to incorporate store characteristics explicitly in the model and then see if we can learn any dependencies
- All we have to do is modify the prior distribution of $\theta_s = (\alpha_s, \beta_s)$

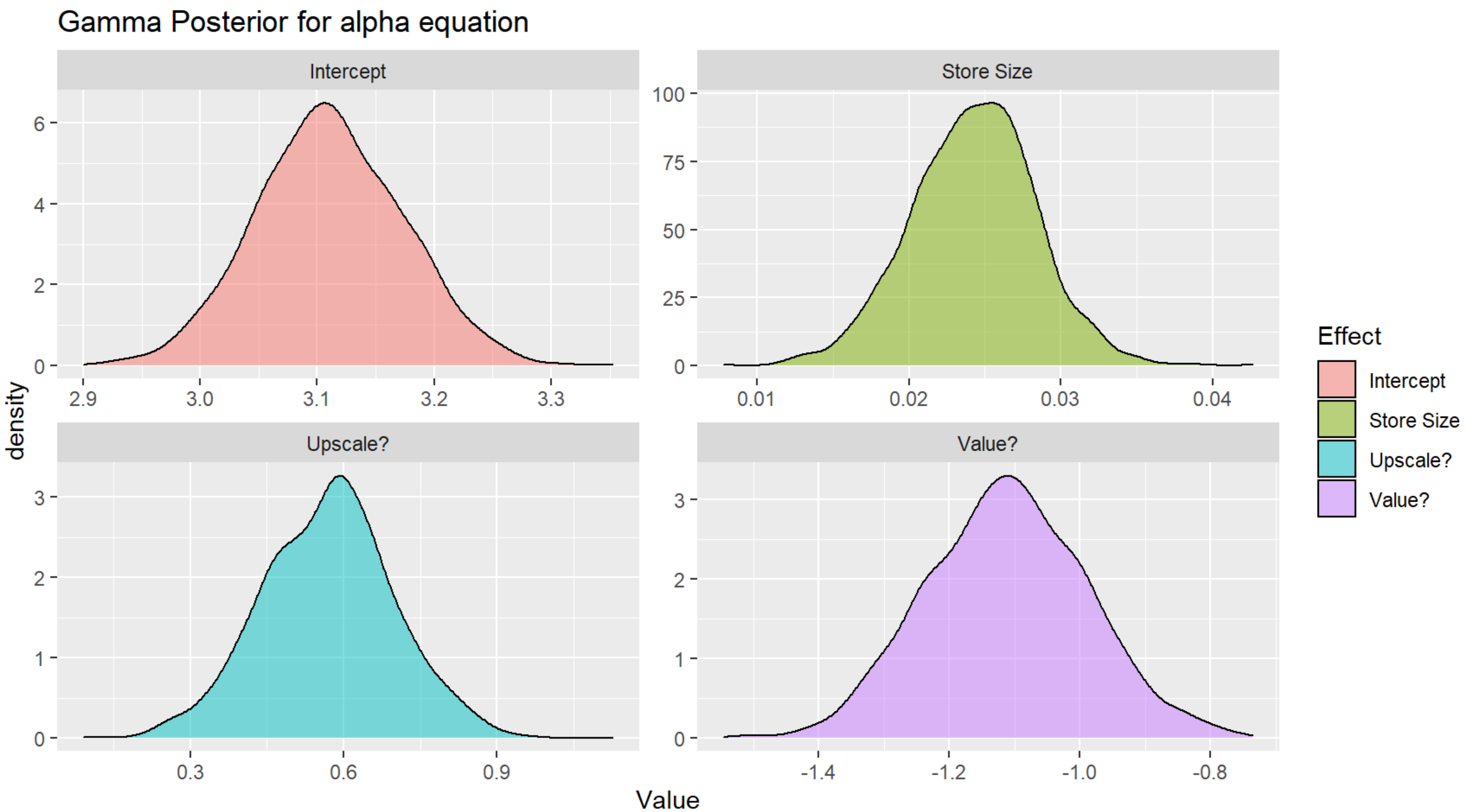
Model 2

$$
\begin{aligned}
\log y_{sw} &= \alpha_s + \beta_s \log p_{sw} + \varepsilon_{sw}, \\
\varepsilon_{sw}|\sigma &\sim \mathrm{N}(0, \sigma^2), \\
\alpha_s &= \gamma_\alpha' Z_s + \psi_{\alpha,s}, \\
\beta_s &= \gamma_\beta' Z_s + \psi_{\beta,s}, \\
\psi_s \equiv (\psi_{\alpha,s}, \psi_{s,\beta})|\Sigma &\sim \mathrm{N}(0, \Sigma),
\end{aligned}
$$

- Here Z_s is a vector store characteristics for store s
- The previous model is a special case of this with $Z_s = 1$
- We can use the same priors as the previous mmodel plus a prior on the γ parameters, e.g.,

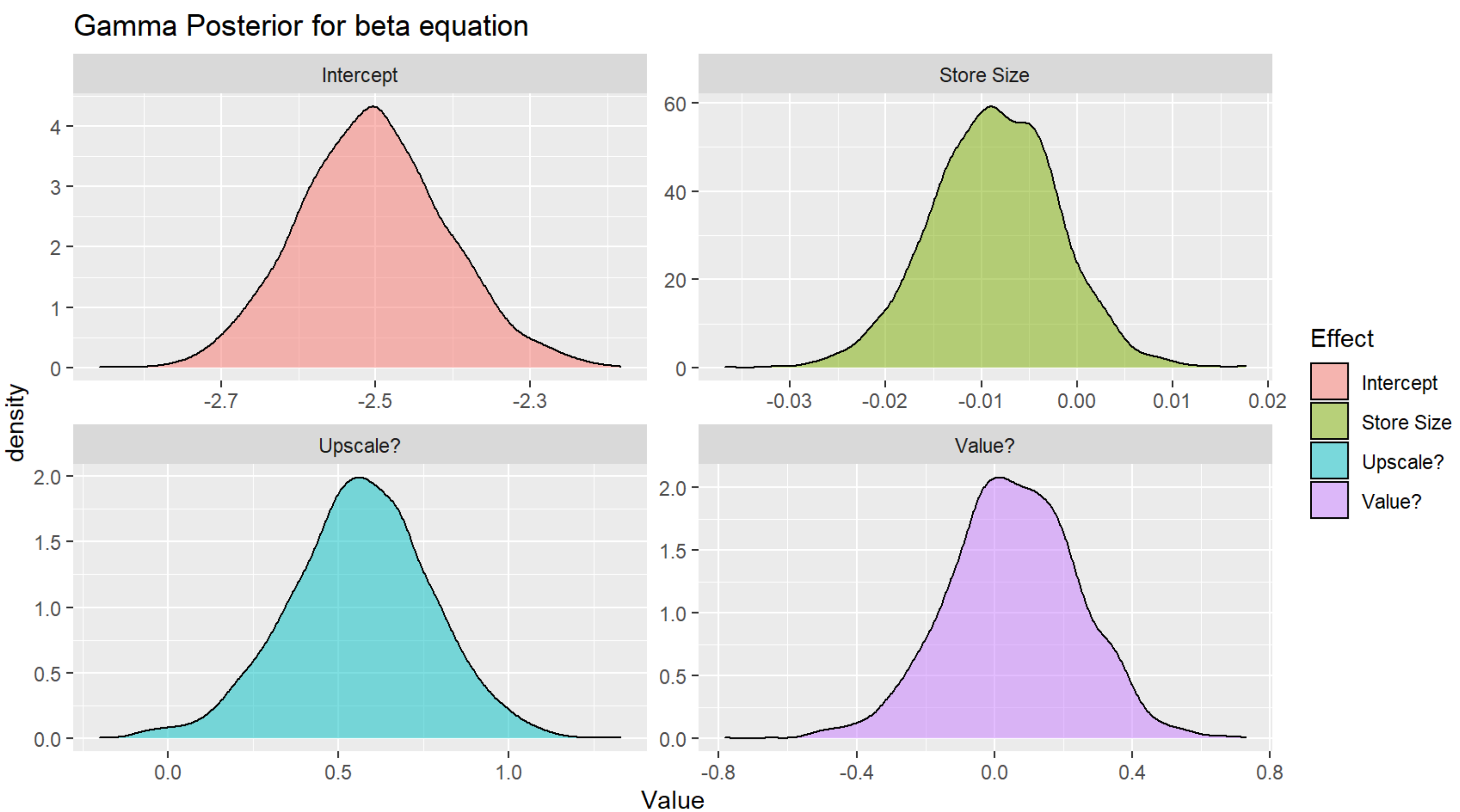
$$
\gamma_\alpha \sim \mathrm{N}(0, 5^2 I_K), \;\; \gamma_\beta \sim \mathrm{N}(0, 5^2 I_K)
$$

Results



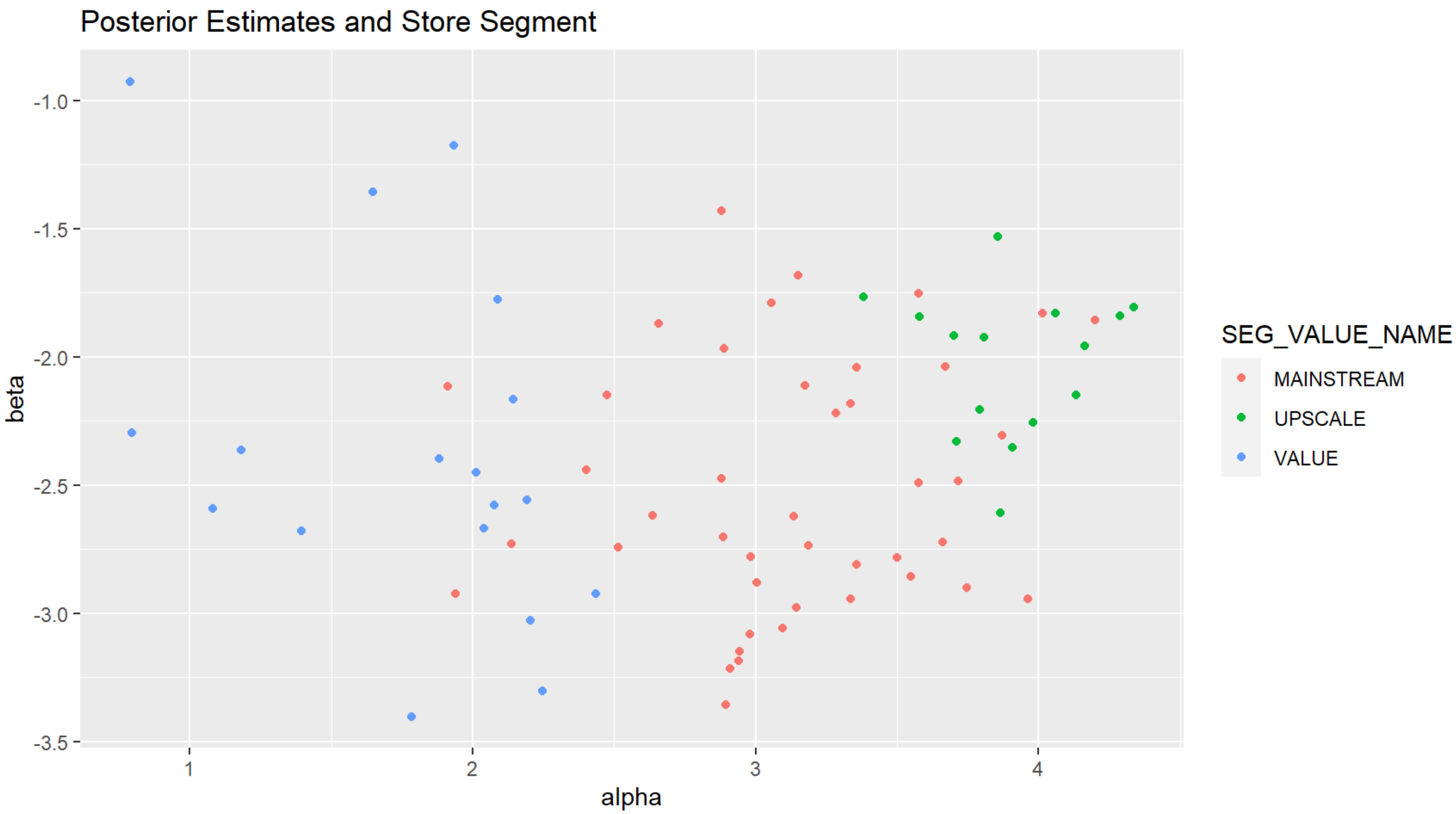
Gamma Posterior for alpha equation
Intercept
Store Size
Upscale?
Value?
density
Value
Effect
Intercept
Store Size
Upscale?
Value?

Results



Gamma Posterior for beta equation
Intercept
Store Size
Upscale?
Value?
density
Value
Effect
Intercept
Store Size
Upscale?
Value?
-2.7
-2.5
-2.3
-0.03
-0.02
-0.01
0.00
0.01
0.02
0.0
0.5
1.0
-0.8
-0.4
0.0
0.4
0.8

Results


Posterior Estimates and Store Segment
-1.0
-1.5
-2.0
-2.5
-3.0
-3.5
beta
1
2
3
4
alpha
SEG_VALUE_NAME
MAINSTREAM
UPSCALE
VALUE

Appendix

Deriving Posterior for Normal Model

The model is

$$\begin{aligned} y_{ij}|\alpha_i, \sigma &\sim \mathrm{N}(\alpha_i, \sigma^2), \qquad j = 1, \ldots, N_i; i = 1, \ldots, N, \\ \alpha_i|\mu, \sigma_\alpha &\sim \mathrm{N}(\mu, \sigma_\alpha^2), \end{aligned}$$

- To get the posterior for α_i conditional on the remaining parameters, we need to calculate

$$p(\alpha_i|y_i) = \frac{p(y_i|\alpha_i)p(\alpha_i)}{\int p(y_i|\alpha_i)p(\alpha_i)d\alpha_i},$$

where the likelihood function is

$$\begin{aligned} p(y_i|\alpha_i) &= \prod_{j=1}^{N_i} \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(-\frac{1}{2\sigma^2}(y_{ij} - \alpha_i)^2 \Big) \\ &= \Big(\frac{1}{\sqrt{2\pi}\sigma} \Big)^{N_i} \exp\Big(-\frac{1}{2\sigma^2} \sum_{j=1}^{N_i} (y_{ij} - \alpha_i)^2 \Big) \end{aligned}$$

- Note that we can rewrite the sum as

$$
\begin{aligned}
\sum_{j=1}^{N_i}(y_{ij}-\alpha_i)^2 &= \sum_j (Y_{ij}^2+\alpha_i^2-2Y_{ij}\alpha_i) \\
&= N_i(\alpha_i^2-2\alpha_i\bar{y}_i)+\sum_j y_{ij}^2 \\
&= N_i(\alpha_i-\bar{y}_i)^2+\sum_j y_{ij}^2 - N_i\bar{y}_i^2
\end{aligned}
$$

- The second term doesn't depend on α_i and will cancel out in the fraction definining the posterior. Therefore, we can write the numerator as

$$
\exp\Big(-\frac{N_i}{2\sigma^2}(\alpha_i-\bar{y}_i)^2\Big)\times\exp\Big(-\frac{1}{2\sigma_\alpha^2}(\alpha_i-\mu)^2\Big)=
$$

$$
\exp\Big(-\frac{1}{2}\Big[\frac{N_i}{\sigma^2}(\alpha_i-\bar{y}_i)^2+\frac{1}{\sigma_\alpha^2}(\alpha_i-\mu)^2\Big]\Big)
$$

- Using the “completing the square” result from week 1, slide 30, we can write the term in square brackets as

$$\frac{N_i}{\sigma^2}(\alpha_i - \bar{y}_i)^2 + \frac{1}{\sigma_\alpha^2}(\alpha_i - \mu)^2 =$$
$$\left(\frac{N_i}{\sigma^2} + \frac{1}{\sigma_\alpha^2}\right)\left[\alpha_i - \mu_{\alpha_i}\right]^2 + C,$$

  where

$$\mu_{\alpha_i} \equiv \frac{\frac{N_i}{\sigma^2}\bar{y}_i + \frac{1}{\sigma_\alpha^2}\mu}{\frac{N_i}{\sigma^2} + \frac{1}{\sigma_\alpha^2}},$$

and C is a constant that doesn’t depend on α_i.

- Collecting terms we then have the posterior for α_i:

$$p(\alpha_i|y_i) = \frac{\exp\left(-\frac{\tau_{\alpha_i}}{2}(\alpha - \mu_{\alpha_i})^2\right)}{\int \exp\left(-\frac{\tau_{\alpha_i}}{2}(\alpha - \mu_{\alpha_i})^2\right) d\alpha_i},$$

  where $\tau_{\alpha_i} = \frac{N_i}{\sigma^2} + \frac{1}{\sigma_\alpha^2}$. We can either solve the integral in the denominator or simply realize that the numerator is proportional to the density for normal distribution. Either way we have

$$p(\alpha_i|y_i) = \mathrm{N}(\mu_{\alpha_i}, \tau_{\alpha_i}^{-1})$$