Count Predicted Variable & Contingency Tables

Tim Frasier

Goals and General Idea

Goals
Contingency tables

- When we have count data distributed across a range of different categories
  - Are counts in one category higher than another?
  - Is the count in one category contingent upon the the level in another category (or plural)?

	Eye Colour			
Hair Colour	**Blue**	**Brown**	**Green**	**Hazel**
Black	20	68	5	15
Blond	94	7	16	10
Brunette	84	119	29	54
Red	17	26	14	14

Data from Snee (1974) *The American Statistician* **28**: 9-12, as presented in Kruschke (2015).

Goals
Contingency tables

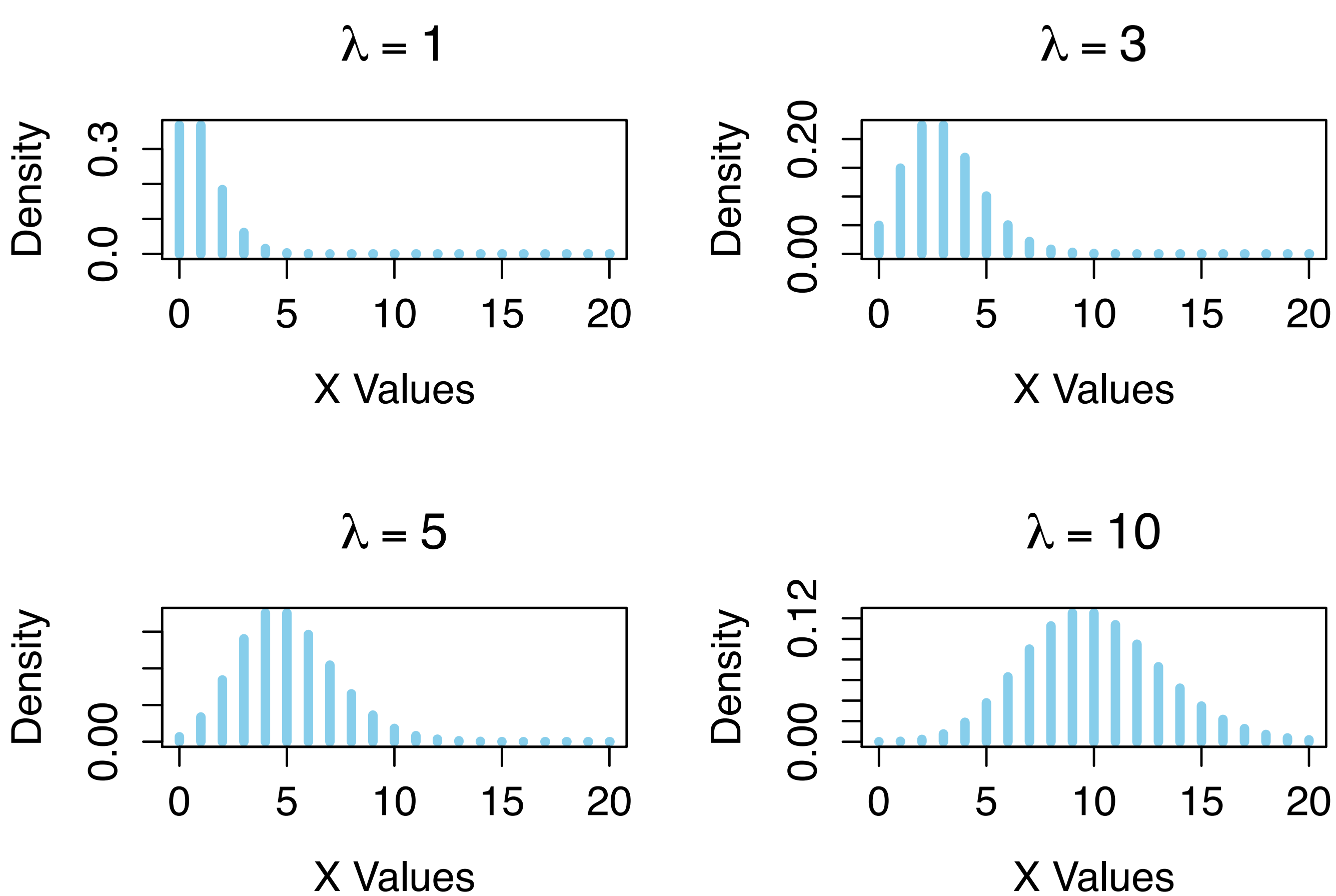
- When we have count data distributed across a range of different categories
  - Are counts in one category higher than another?
  - Is the count in one category contingent upon the the level in another category (or plural)?
- Often addressed with chi-square or Exact test analyses

Goals
Count predicted variable

- Any time our predicted variable is a count
  - Abundance
  - Counts of individuals (or things) with different traits/characteristics
  - etc.

Distribution and Links

- When modelling count data, it is appropriate to use the Poisson distribution
  - Positive integers
  - One parameter - lambda (λ)



Distribution and Links
Contingency table example

	Eye Colour			
Hair Colour	Blue	Brown	Green	Hazel
Black	20	68	5	15
Blond	94	7	16	10
Brunette	84	119	29	54
Red	17	26	14	14

- Cell frequencies are representative of underlying cell probabilities
  - Are nominal variables independent of each other?

Distribution and Links
Contingency table example

	Eye Colour			
Hair Colour	Blue	Brown	Green	Hazel
Black	20	68	5	15
Blond	94	7	16	10
Brunette	84	119	29	54
Red	17	26	14	14

- Cell frequencies are representative of underlying cell probabilities
    - Are nominal variables independent of each other?
- If independent

$$68 = Pr(\mathrm{BlackHair}) \ \ \mathrm{x} \ \ Pr(\mathrm{BrownEyes})$$

True for all cells

Distribution and Links
Contingency table example

	Eye Colour			
Hair Colour	Blue	Brown	Green	Hazel
Black	20	68	5	15
Blond	94	7	16	10
Brunette	84	119	29	54
Red	17	26	14	14

- Cell frequencies are representative of underlying cell probabilities
  - Are nominal variables independent of each other?
- If independent

$$68 = Pr(\text{BlackHair}) \text{ x } Pr(\text{BrownEyes})$$

True for all cells

- If interaction effects, this will not be the case

Distribution and Links
Contingency table example

	Eye Colour				
Hair Colour	**Blue**	**Brown**	**Green**	**Hazel**	
Black	20	68	5	15	
Blond	94	7	16	10	
Brunette	84	119	29	54	
Red	17	26	14	14	
$f(c)$	**215**	**220**	**64**	**93**	**592** N

marginal frequencies of each eye colour

Distribution and Links
Contingency table example

	Eye Colour			
Hair Colour	**Blue**	**Brown**	**Green**	**Hazel**
Black	20	68	5	15
Blond	94	7	16	10
Brunette	84	119	29	54
Red	17	26	14	14
$f(c)$	**215**	**220**	**64**	**93**	**592**

N

marginal frequencies of each eye colour

$Pr(\text{BlueEyes}) = 215 / 592 = 0.363$

Distribution and Links
Contingency table example

	Eye Colour				
Hair Colour	Blue	Brown	Green	Hazel	
Black	20	68	5	15	**108**
Blond	94	7	16	10	**127**
Brunette	84	119	29	54	**286**
Red	17	26	14	14	**71**
$f(c)$	**215**	**220**	**64**	**93**	**592**
					N

marginal frequencies of each hair colour

marginal frequencies of each eye colour

$Pr(\text{BlueEyes}) = 215 \ / \ 592 = 0.363$

Distribution and Links
Contingency table example

	Eye Colour				
Hair Colour	Blue	Brown	Green	Hazel	
Black	20	68	5	15	**108**
Blond	94	7	16	10	**127**
Brunette	84	119	29	54	**286**
Red	17	26	14	14	**71**
$f(c)$	**215**	**220**	**64**	**93**	**592**

marginal frequencies of each hair colour

N

marginal frequencies of each eye colour

$$Pr(\text{BlueEyes}) = 215 / 592 = 0.363$$

$$Pr(\text{BlackHair}) = 108 / 592 = 0.182$$

Distribution and Links
Contingency table example

	Eye Colour				
Hair Colour	**Blue**	**Brown**	**Green**	**Hazel**	
Black	20	68	5	15	**108**
Blond	94	7	16	10	**127**
Brunette	84	119	29	54	**286**
Red	17	26	14	14	**71**
$f(c)$	**215**	**220**	**64**	**93**	**592**

N

marginal frequencies of each hair colour

marginal frequencies of each eye colour

Pr(BlueEyes) = 215 / 592 = 0.363

Pr(BlackHair) = 108 / 592 = 0.182

Pr(BlueEyes & BlackHair) = (0.363 x 0.182) x 592 = 39

Hmm...must be an interaction effect

Distribution and Links
Contingency table example

$$Pr(\text{BlueEyes \& BlackHair}) = (0.363 \text{ x } 0.182) \text{ x } 592 = 39$$

- Joint probability is the product of the relevant marginal probabilities

- We're used to dealing with additive combinations
  - Can convert to log scale, then they'll be additive

Segue on Logarithms

- Adding logarithms is the same as multiplying original values

$$10 \times 5 = 50$$

$$log(10) + log(5) = log(50)$$

Segue on Logarithms

- Adding logarithms is the same as multiplying original values

$$10 \times 5 = 50$$

$$log(10) + log(5) = log(50)$$

- Can cancel out logarithms (bring back to original scale), by raising them to the exponent

$$exp(log(10) + log(5)) = 50$$

Distribution and Links

$$y_i \sim Poisson(\lambda)$$

$$\lambda = \exp(\beta_0 + \beta_1[x[1]] + \beta_2[x[2]] + \beta_{1,2}[x[1], x[2]])$$

Distribution and Links

$$y_i \sim Poisson(\lambda)$$

$$\lambda = \exp(\beta_0 + \beta_1[x[1]] + \beta_2[x[2]] + \beta_{1,2}[x[1], x[2]])$$

The "black box" into which we can put any of our previous equations, or others

Distribution and Links

$$y_i \sim Poisson(\lambda)$$

$$\lambda = \exp(\beta_0 + \beta_1[x[1]] + \beta_2[x[2]] + \beta_{1,2}[x[1], x[2]])$$

Depends on rest of model, here the average across all categories of all variables

Distribution and Links

$$y_i \sim Poisson(\lambda)$$

$$\lambda = \exp(\beta_0 + \beta_1[x[1]] + \beta_2[x[2]] + \beta_{1,2}[x[1], x[2]])$$

The deflection away from baseline due to being in each category of our first nominal predictor variable.

Distribution and Links

$$y_i \sim Poisson(\lambda)$$

$$\lambda = \exp(\beta_0 + \beta_1[x[1]] + \beta_2[x[2]] + \beta_{1,2}[x[1], x[2]])$$

The deflection away from baseline due to being in each category of our second nominal predictor variable.

Distribution and Links

$$y_i \sim Poisson(\lambda)$$

$$\lambda = \exp(\beta_0 + \beta_1[x[1]] + \beta_2[x[2]] + \boxed{\beta_{1,2}}[x[1], x[2]])$$

Interaction effects.

Distribution and Links

$$y_i \sim Poisson(\lambda)$$

$$\lambda = \exp(\beta_0 + \beta_1[x[1]] + \beta_2[x[2]] + \beta_{1,2}[x[1], x[2]])$$

Note that coefficient estimates will now be on the log scale
(even though we haven't explicitly specified them as such)

Distribution and Links

- The link between the predictor and predicted variables is based on logarithms
  - Previous models (other than logistic) have been based on identity
  - These types of models are called log linear models

Bayesian Approach

Load Libraries & Functions

```
library(rstan)
source("plotPost.R")
```

Organize the Data

```
# Y-Data
y = as.integer(haireye$Freq)
N = length(y)
yLogMean = log(mean(y))
yLogSD = log(sd(c(rep(0, N - 1), sum(y))))
```

Organize the Data

```
# Y-Data
y = as.integer(haireye$Freq)
N = length(y)
yLogMean = log(mean(y))
yLogSD = log(sd(c(rep(0, N - 1), sum(y))))
```

Will see why we need these in a minute...

Organize the Data

```
# Y-Data
y = as.integer(haireye$Freq)
N = length(y)
yLogMean = log(mean(y))
yLogSD = log(sd(c(rep(0, N - 1), sum(y))))
```

Largest possible sd would occur if the sum of all values was in one cell, and all others were zero

Organize the Data

```
# Eye data
eye = as.numeric(haireye$Eye)
eyeColours = levels(haireye$Eye)
nEyeColours = length(unique(eye))

# Hair data
hair = as.numeric(haireye$Hair)
hairColours = levels(haireye$Hair)
nHairColours = length(unique(hair))
```

Make Data List For Stan

```
dataList = list(
  y = y,
  N = N,
  yLogMean = yLogMean,
  yLogSD = yLogSD,
  eye = eye,
  hair = hair,
  nEyeColours = nEyeColours,
  nHairColours = nHairColours
)
```

Define the Model

- The **data** block

```
modelstring = "
  data {
    int N;                 // Sample size
    int nEyeColours;       // Number of different eye colours in data set
    int nHairColours;      // Number of different hair colours in data set
    real yLogMean;         // The log mean of the observed y data
    real yLogSD;           // The log SD of the observed y data
    int<lower=0> y[N];     // The y data, remember they are integers, and must
                              be defined as such
    int eye[N];            // The eye data, contains indicators of eye colour
    int hair[N];           // The hair data, contains indicators of hair colour
  }
```

Define the Model

- The **parameters** block

```
parameters {
    real b0;
    real b1[nEyeColours];                  // Effect of eye colour
    real b2[nHairColours];                 // Effect of hair colour
    real b3[nEyeColours, nHairColours];    // Interaction effect between hair and
                                              eye colour
  }
```

Define the Model

- The **model** block

```
model {
    // Definitions
    vector[N] lambda;

    // Likelihood
    for (i in 1:N) {
      lambda[i] = exp(b0 + b1[eye[i]] + b2[hair[i]] + b3[eye[i], hair[i]]);
      y[i] ~ poisson(lambda[i]);
    }

    // Priors
    b0 ~ normal(yLogMean, yLogSD);

    for (j in 1:nEyeColours) {
      b1[j] ~ normal(0, 1);
    }

    for (j in 1:nHairColours) {
      b2[j] ~ normal(0, 1);
    }

    for (j in 1:nEyeColours) {
      for (k in 1:nHairColours) {
        b3[j, k] ~ normal(0, 1);
      }
    }
  }
```

Define the Model

- The **generated quantities** block

```
generated quantities {
    // Definitions
    vector[N] lambda_pred;
    vector[N] y_pred;

    for (i in 1:N) {
      lambda_pred[i] = exp(b0 + b1[eye[i]] + b2[hair[i]] + b3[eye[i], hair[i]]);
      y_pred[i] = poisson_rng(lambda_pred[i]);
    }
  }
"
writeLines(modelstring, con = "model.stan")
```

Run the Model

```
stanFit <- stan(file = "model.stan",
                data = dataList,
                pars = c("b0", "b1", "b2", "b3", "y_pred"),
                warmup = 2000,
                iter = 10000,
                chains = 3)
```

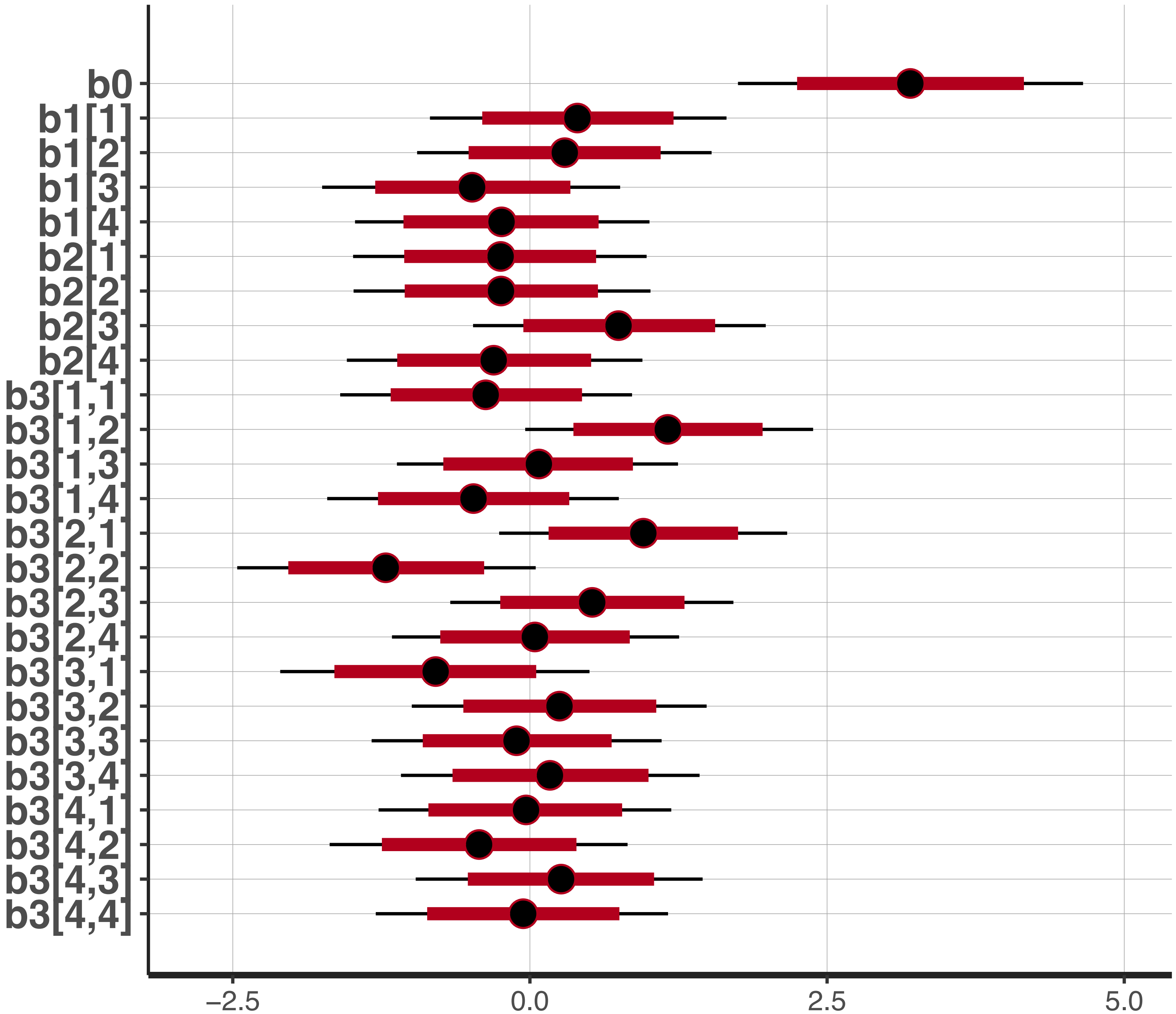
Check MCMC Performance

```
print(stanFit)

Inference for Stan model: model.
3 chains, each with iter=10000; warmup=2000; thin=1;
post-warmup draws per chain=8000, total post-warmup draws=24000.

             mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
b0           3.20    0.01 0.74   1.75   2.70   3.20   3.70   4.65 19019    1
b1[1]        0.40    0.00 0.63  -0.84  -0.02   0.40   0.83   1.65 23941    1
b1[2]        0.30    0.00 0.63  -0.95  -0.13   0.29   0.72   1.53 23930    1
b1[3]       -0.48    0.00 0.64  -1.75  -0.91  -0.49  -0.05   0.76 23508    1
b1[4]       -0.24    0.00 0.64  -1.47  -0.67  -0.24   0.20   1.01 22565    1
b2[1]       -0.25    0.00 0.63  -1.49  -0.68  -0.25   0.18   0.98 24810    1
b2[2]       -0.24    0.00 0.64  -1.48  -0.67  -0.24   0.18   1.01 24454    1
b2[3]        0.75    0.00 0.63  -0.48   0.33   0.75   1.17   1.98 23263    1
b2[4]       -0.30    0.00 0.63  -1.54  -0.73  -0.30   0.13   0.95 25520    1
b3[1,1]     -0.37    0.00 0.63  -1.60  -0.79  -0.37   0.06   0.86 25801    1
b3[1,2]      1.16    0.00 0.62  -0.04   0.74   1.16   1.58   2.38 25822    1
...
```

View Posteriors

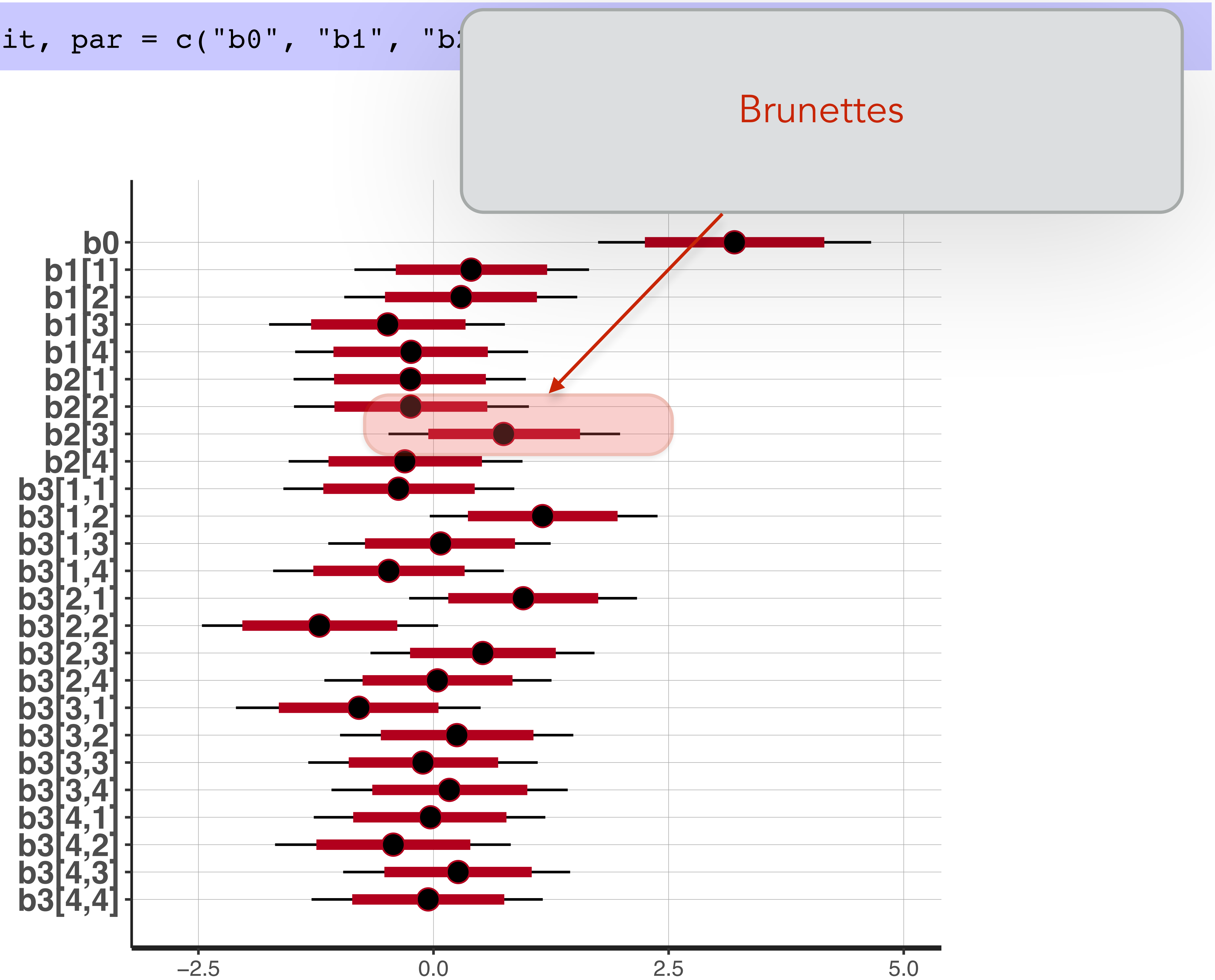
Plotting Posterior Distributions

```
stan_plot(stanFit, par = c("b0", "b1", "b2", "b3"))
```



Plotting Posterior Distributions

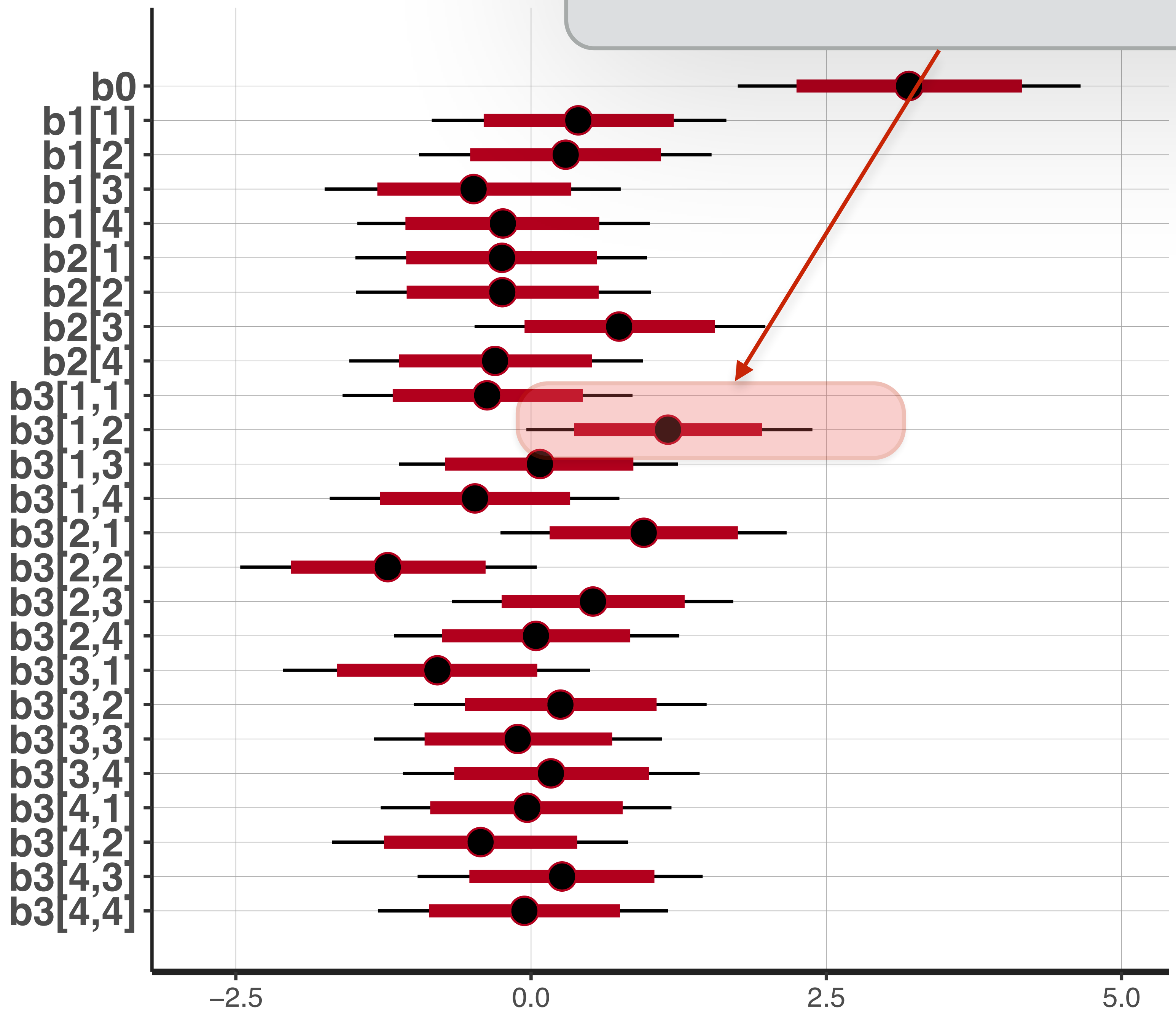
```
stan_plot(stanFit, par = c("b0", "b1", "b2
```



Plotting Posterior Distributions

```
stan_plot(stanFit, par = c("b0", "b1", "b2
```

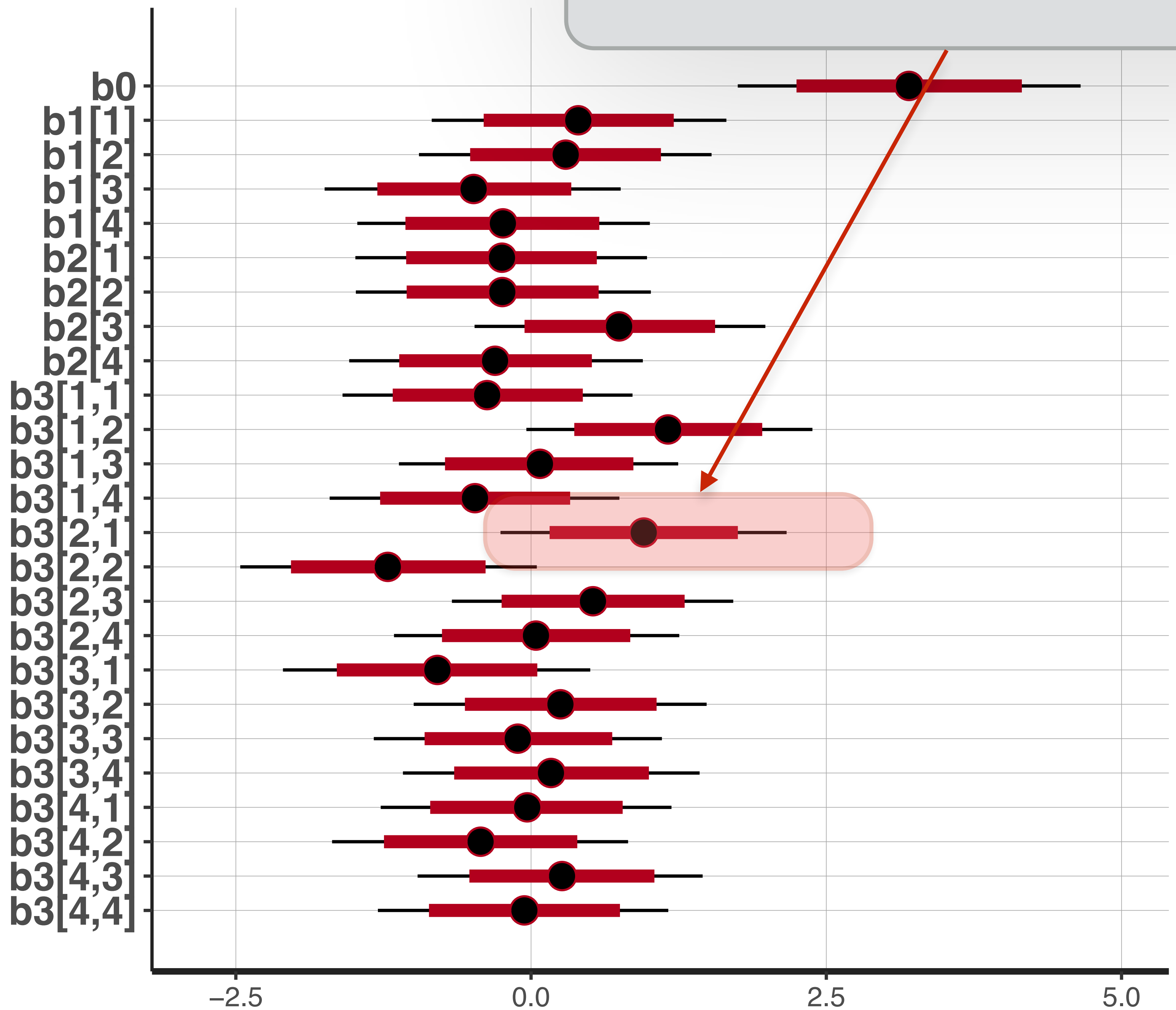
Blue eyes & blonde hair



Plotting Posterior Distributions

```
stan_plot(stanFit, par = c("b0", "b1", "b
```

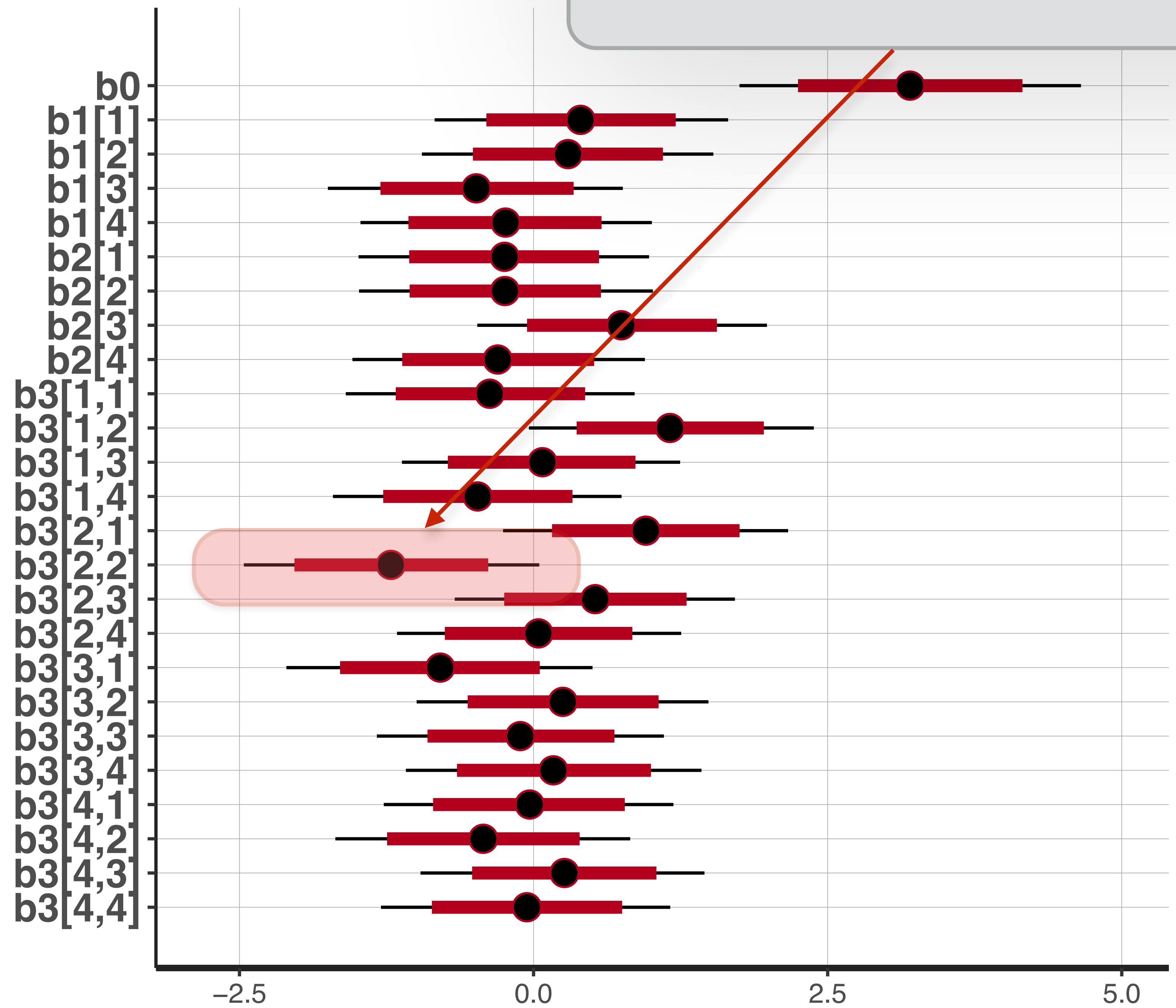
Brown eyes and black hair



Plotting Posterior Distributions

```
stan_plot(stanFit, par = c("b0", "b1", "b2
```

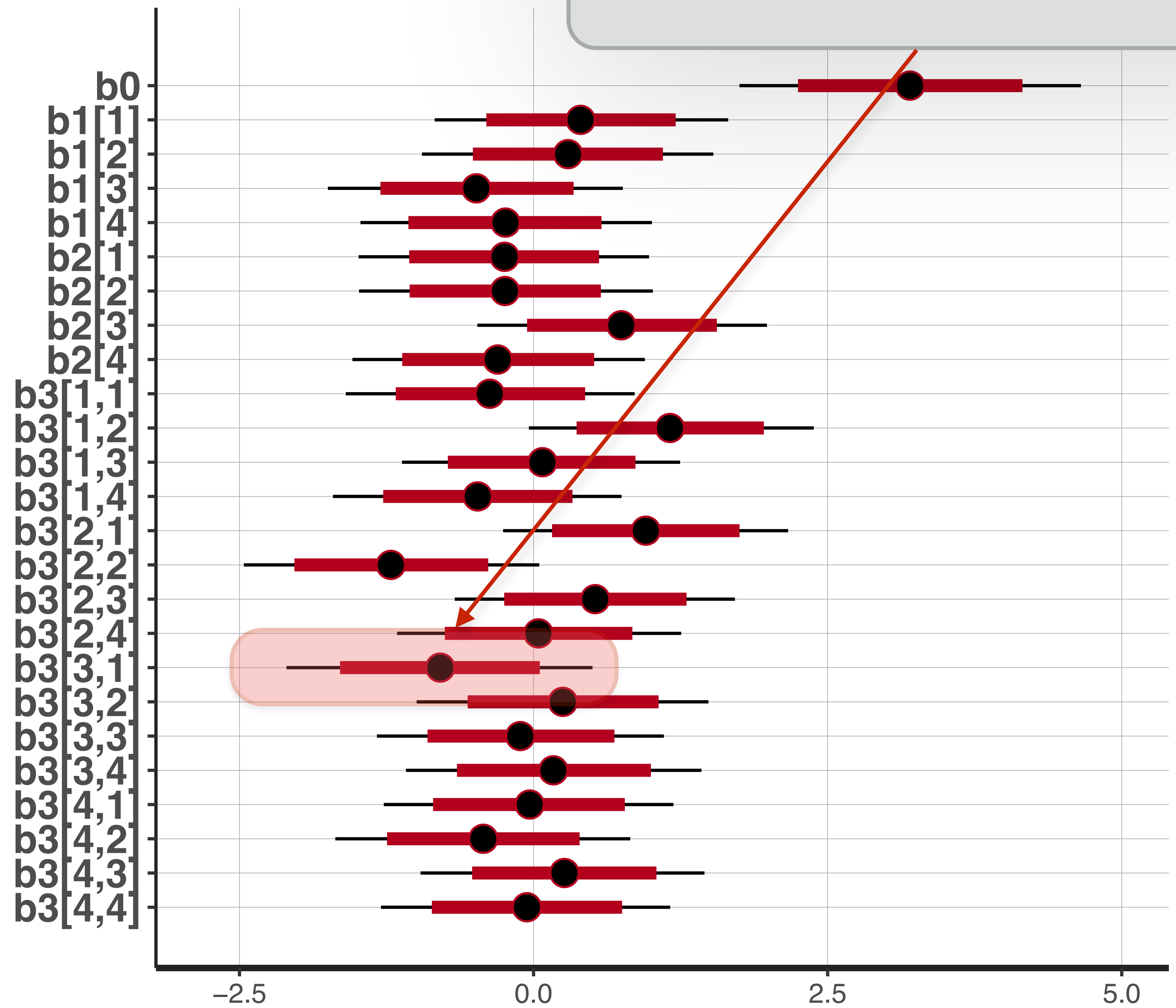
Brown eyes and blonde hair



Plotting Posterior Distributions

```
stan_plot(stanFit, par = c("b0", "b1", "b2
```

Green eyes and black hair



Posterior
Predictive Check

Posterior Predictive Check

- Extract the predicted values

```
yPred = matrix(0, nrow = chainLength, ncol = N)

for (i in 1:N) {
  yPred[, i] = mcmcChains[, paste("y_pred[", i, "]", sep = "")]
}
```

Posterior Predictive Check

- Calculate the mean and HDI predicted values

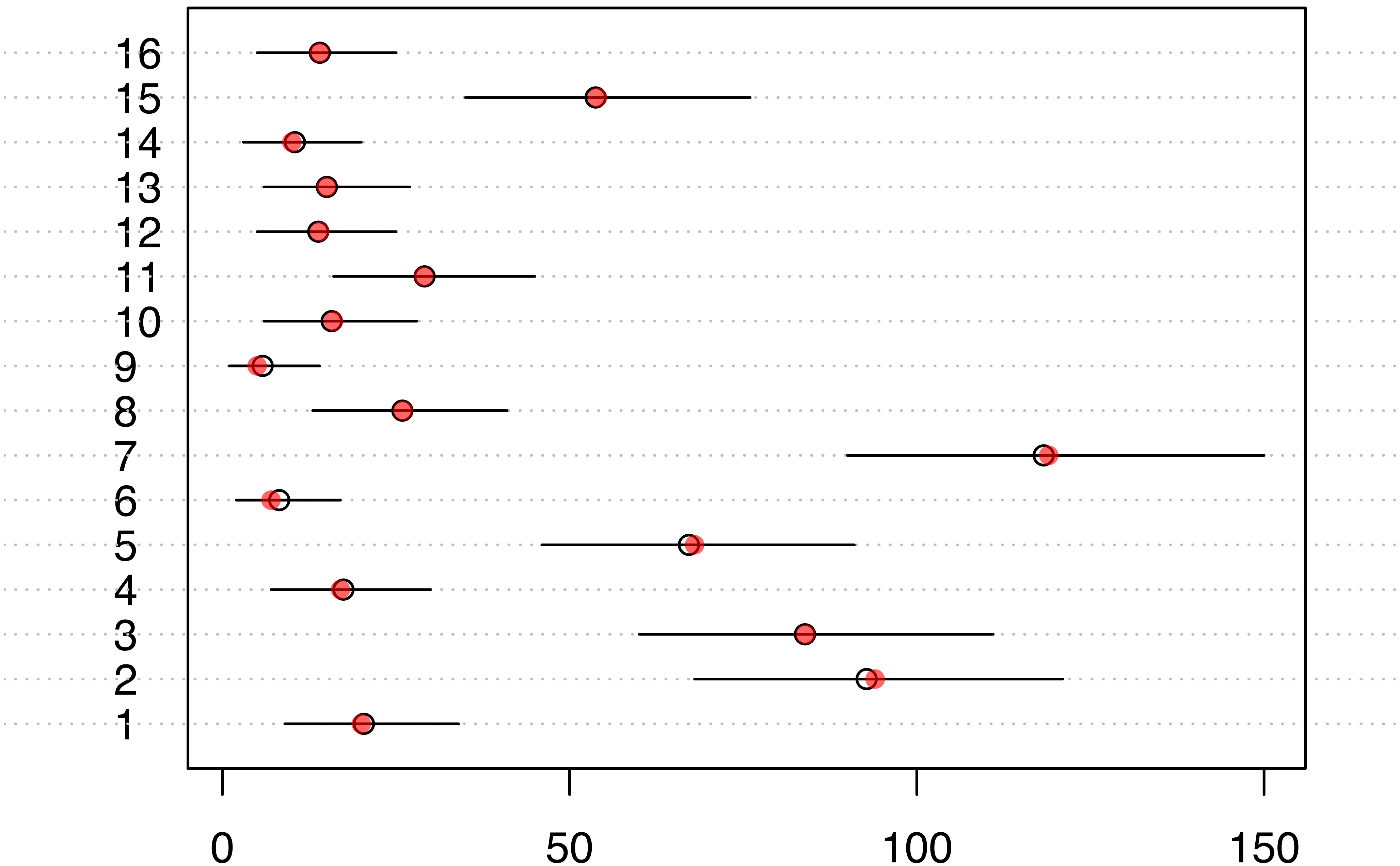
```
yPredMean = apply(yPred, 2, mean)
yPredLow = apply(yPred, 2, quantile, probs = 0.025)
yPredHigh = apply(yPred, 2, quantile, probs = 0.975)
```

Posterior Predictive Check

- Plot the data

```
#--- Plot predicted values ---#
par(mfrow = c(1, 1))
combination = 1:N
dotchart(x = yPredMean, labels = combination, xlim = c(min(yPredLow),
max(yPredHigh)))

#--- Add HDI lines ---#
segments(x0 = yPredLow, y0 = combination, x1 = yPredHigh, y1 =
combination)

#--- Add observed values ---#
points(x = y, y = combination, pch = 16, col = rgb(1, 0, 0, 0.6))
```

Posterior Predictive Check



Questions?