Generalized linear mixed effects model

Sow culling time

Dr. Jarad Niemi

STAT 544 - Iowa State University

May 2, 2024

Outline

- Mixed effect Poisson regression
  - Modeling
  - Estimation via Stan
  - Posterior
- Decision making
  - Maximize utility

Sow culling time

From Caitlyn Abell:

> *I have attached the data file with 2,868 records from one farm. The contemporary group (cg) is farm, year and season. There are columns for number born alive (nba), number born dead (nbd), and parity. One thing you could look at would be improvement over time or differences of performance between the parities [litters]. I think determining the optimal culling time for a sow given her past history would be interesting.*

Primary question of interest: when should a sow be removed from breeding?

```
d = read.table("Ch16b-farm62.txt", header=T)
d$cg = d$farm = d$yearmo = d$nbd = d$dam = NULL
d = plyr::rename(d, c("sire"="grandsire")) |>
  mutate(sowid = factor(sowid), grandsire = factor(grandsire))
```

```
head(d)
```

```
            sowid nba parity  grandsire
1 985120010234800   3      1  C61LW4846
2 985120011536089  11      1  C60LX1975
3 985120011536089   6      2  C60LX1975
4 985120011537054   8      1 C63LW10719
5 985120011537054  11      2 C63LW10719
6 985120011537120  12      1  C60LX3542
```

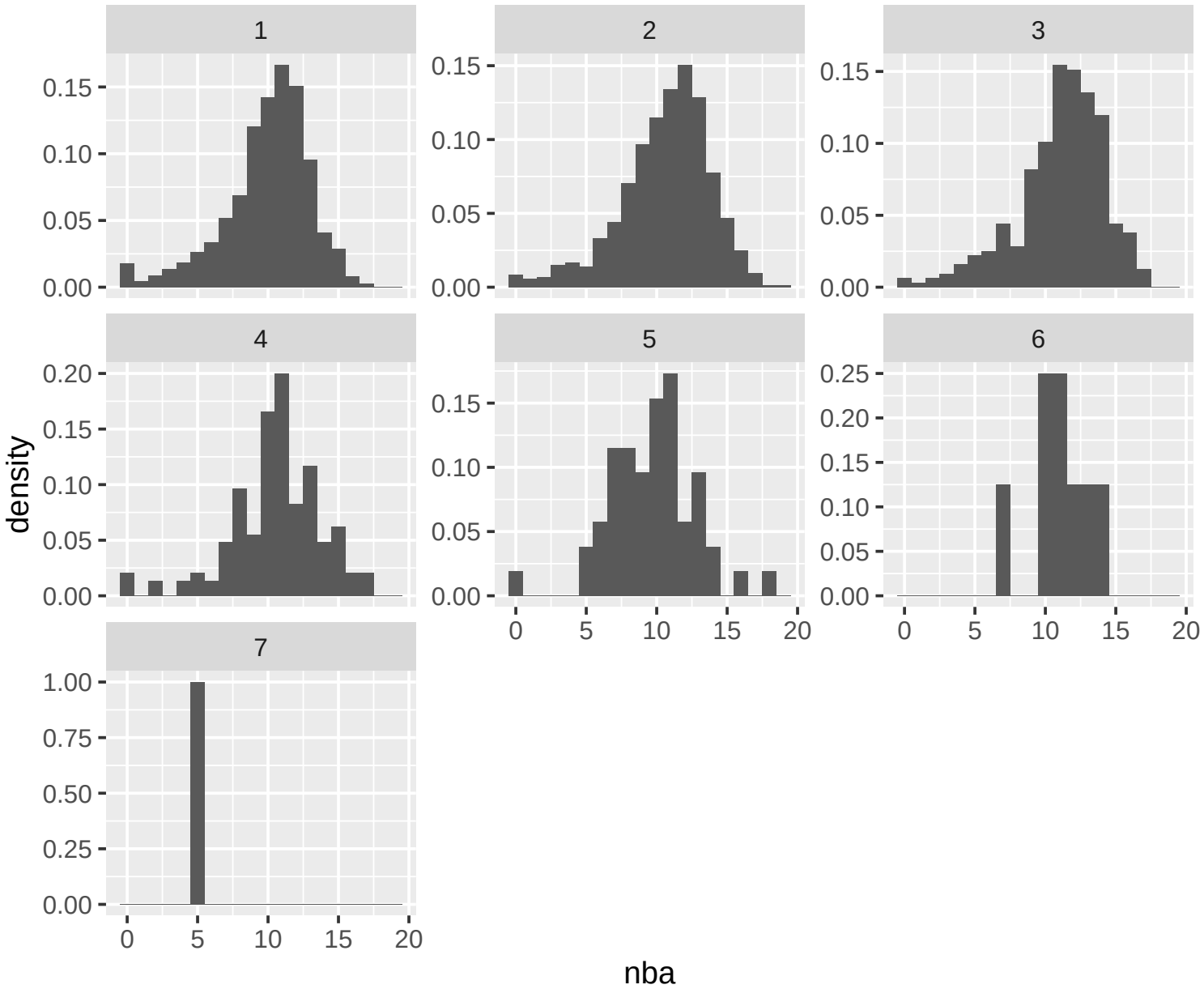
```
summary(d)
```

```
              sowid           nba            parity        grandsire
 985152000271505:   7   Min.   : 0.00   Min.   :1.000   103086 : 238
 985120011545841:   6   1st Qu.: 9.00   1st Qu.:1.000   376475 : 105
 985120025398712:   6   Median :11.00   Median :1.000   514976 :  91
 985152000271655:   6   Mean   :10.28   Mean   :1.714   572703 :  87
 985152002194887:   6   3rd Qu.:12.00   3rd Qu.:2.000   019800 :  84
 985152002429483:   6   Max.   :19.00   Max.   :7.000   376770 :  79
 (Other)        :2831                                   (Other):2184
```

```
dim(d); nlevels(d$sowid); nlevels(d$grandsire)
```

```
[1] 2868    4
[1] 1621
[1] 182
```

```
   1    2    3    4    5    6    7
1621  724  317  145   52    8    1
```



Model

Let y_i be the number born alive for the i^{th} litter. Assume

$$\begin{aligned} y_i &\overset{ind}{\sim} Po(e^{\mu_i}) && i = 1, \ldots, n \\ \mu_i &= \rho_{p[i]} + \alpha_{s[i]} + \beta_{g[i]} \end{aligned}$$

where

- $p[i]$ is the parity for the i^{th} litter,
- $s[i]$ is the sow for the i^{th} litter, and
- $g[i]$ is the grandsire for the i^{th} litter.

The hierarchical structure treats α and β as random effects, i.e.

$$\begin{aligned} \alpha_s &\overset{iid}{\sim} N(0, \sigma_\alpha^2) && s = 1, \ldots, n_{sows} \\ \beta_g &\overset{iid}{\sim} N(0, \sigma_\beta^2) && g = 1, \ldots, n_{grandsires}. \end{aligned}$$

The prior is

$$p(\rho_1, \ldots, \rho_6, \sigma_\alpha, \sigma_\beta) \propto Ca^+(\sigma_\alpha; 0, 1) Ca^+(\sigma_\beta; 0, 1)$$

```
model = "
data {
  int<lower=1> n;
  int<lower=1> np;
  int<lower=1> ns;
  int<lower=1> ng;
  int<lower=0> y[n];
  int<lower=1, upper=np> parity[n];
  int<lower=1, upper=ns> sow[n];
  int<lower=1, upper=ng> grandsire[n];
}

parameters {
  real rho[np];     // implicit prior over whole real line
  real alpha[ns];
  real beta[ng];
  real<lower=0> sigma_alpha;
  real<lower=0> sigma_beta;
}

model {
  for (i in 1:n) {
    y[i] ~ poisson(exp(rho[parity[i]]+alpha[sow[i]]+beta[grandsire[i]]));
  }

  // Random effects
  alpha ~ normal(0, sigma_alpha);
  beta ~ normal(0, sigma_beta);

  sigma_alpha ~ cauchy(0,1);
  sigma_beta  ~ cauchy(0,1);
}"
```

```
d = d[d$parity!=7,]
dat = list(y = d$nba,
           parity = d$parity,
           sow = as.numeric(d$sowid),
           grandsire = as.numeric(d$grandsire))
dat$n  = length(dat$y)
dat$np = max(dat$parity)
dat$ns = max(dat$sow)
dat$ng = max(dat$grandsire)

time = system.time(r <- sampling(object = m,
                                 data = dat,
                                 pars = c("rho","alpha","beta","sigma_alpha","sigma_beta"),
                                 iter = 10000,
                                 thin = 5))
```

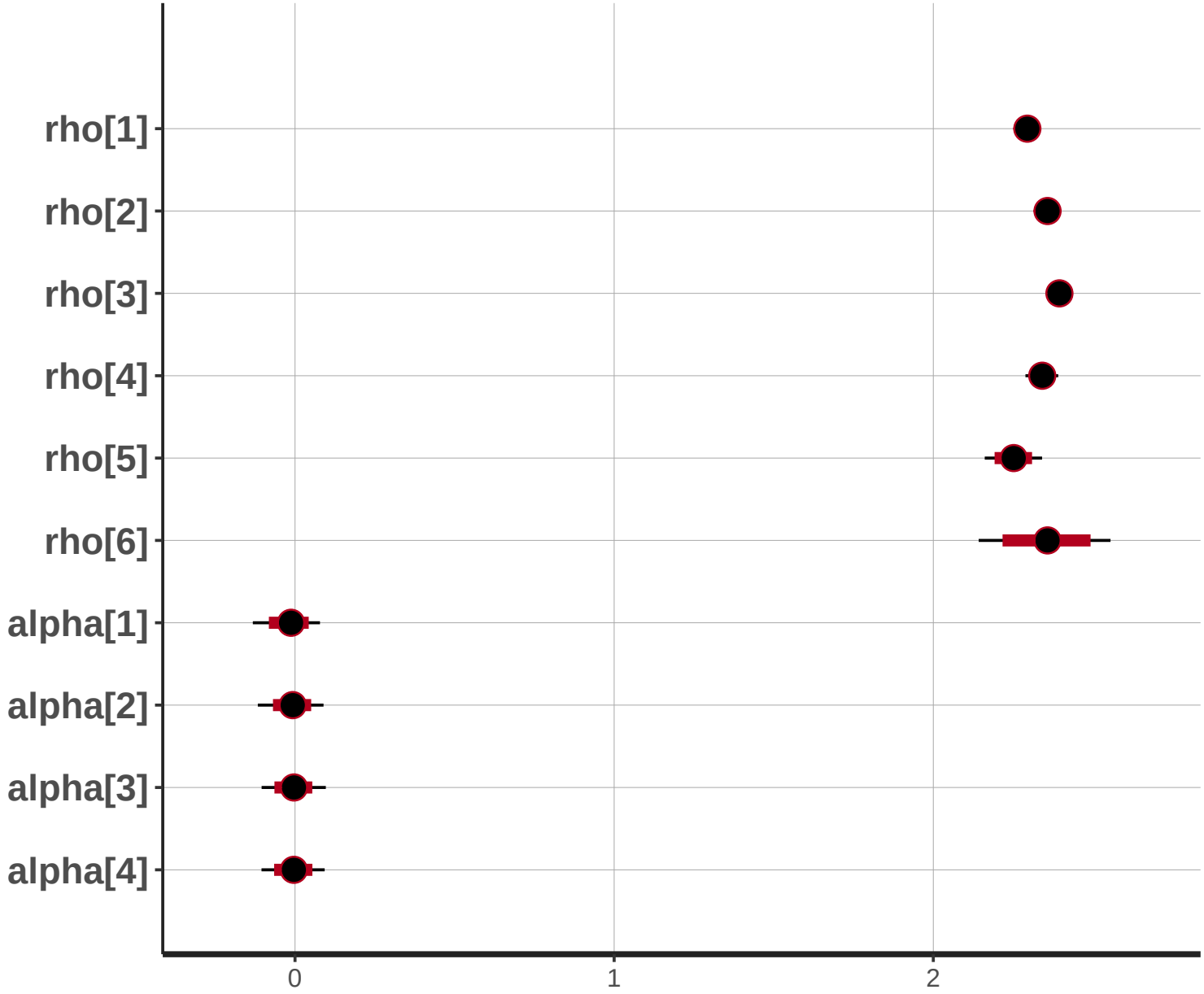
```
SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.000213 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 2.13 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 10000 [  0%]  (Warmup)
Chain 1: Iteration: 1000 / 10000 [ 10%]  (Warmup)
Chain 1: Iteration: 2000 / 10000 [ 20%]  (Warmup)
Chain 1: Iteration: 3000 / 10000 [ 30%]  (Warmup)
Chain 1: Iteration: 4000 / 10000 [ 40%]  (Warmup)
Chain 1: Iteration: 5000 / 10000 [ 50%]  (Warmup)
Chain 1: Iteration: 5001 / 10000 [ 50%]  (Sampling)
Chain 1: Iteration: 6000 / 10000 [ 60%]  (Sampling)
Chain 1: Iteration: 7000 / 10000 [ 70%]  (Sampling)
Chain 1: Iteration: 8000 / 10000 [ 80%]  (Sampling)
Chain 1: Iteration: 9000 / 10000 [ 90%]  (Sampling)
```

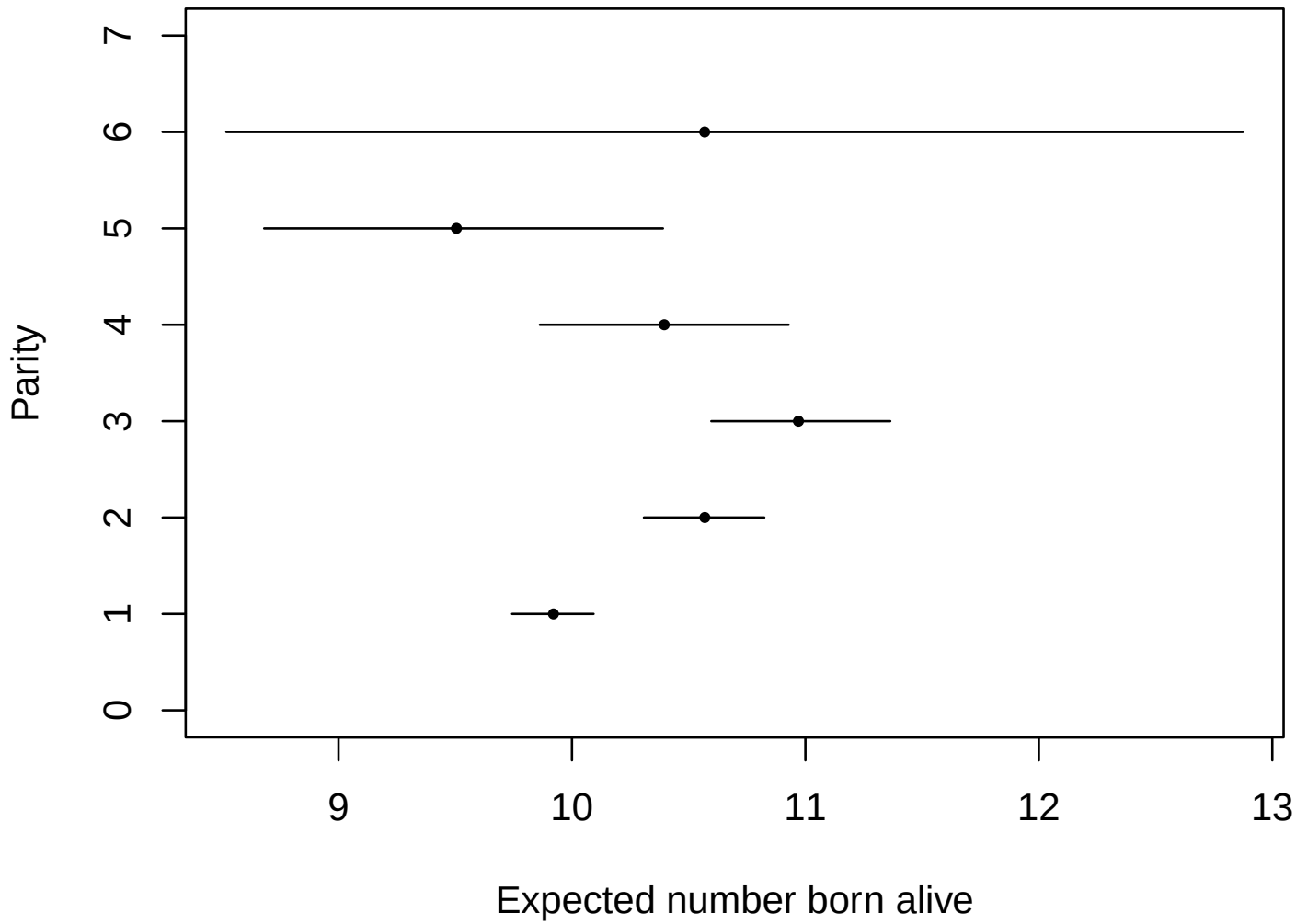
Run time and summary

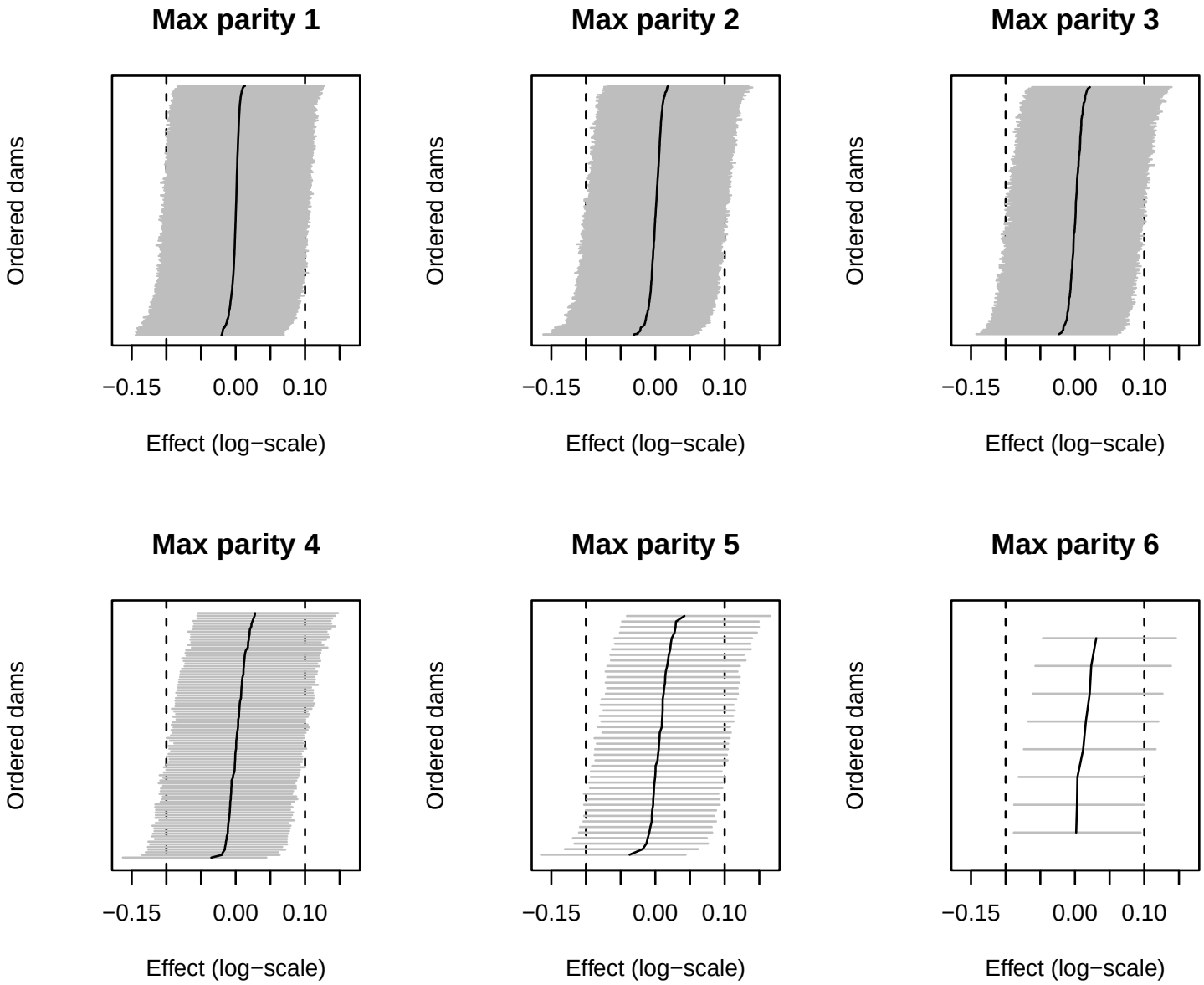
```
time
```

```
   user  system elapsed
425.044   3.078 431.879
```

```
s = summary(r)$summary
```


rho[1]
rho[2]
rho[3]
rho[4]
rho[5]
rho[6]
alpha[1]
alpha[2]
alpha[3]
alpha[4]
0
1
2


Parity
0
1
2
3
4
5
6
7
9
10
11
12
13
Expected number born alive


Max parity 1
Max parity 2
Max parity 3
Max parity 4
Max parity 5
Max parity 6
Ordered dams
−0.15
0.00
0.10
Effect (log−scale)

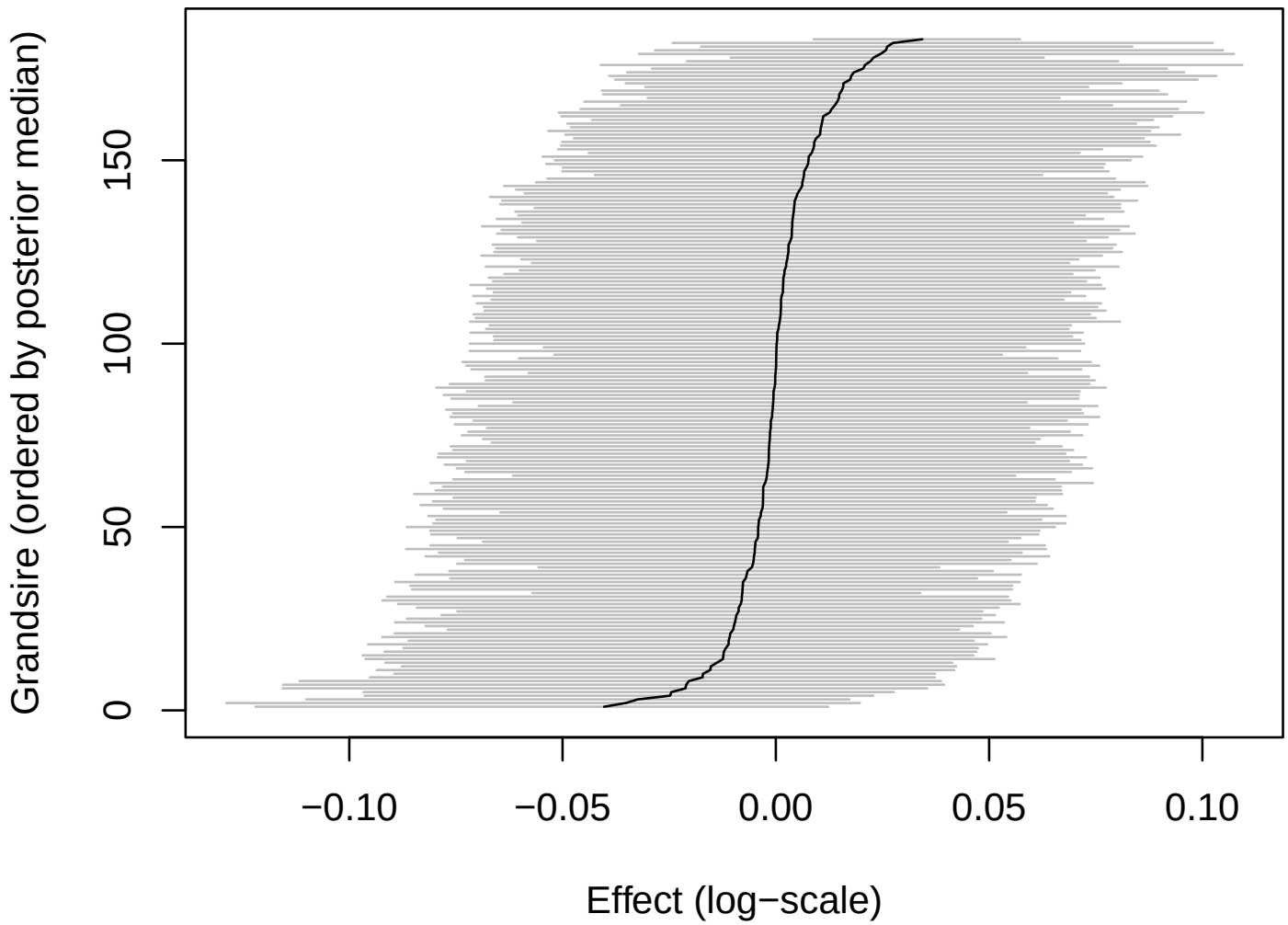
Best and worst performing sows with max parity 4 and 5

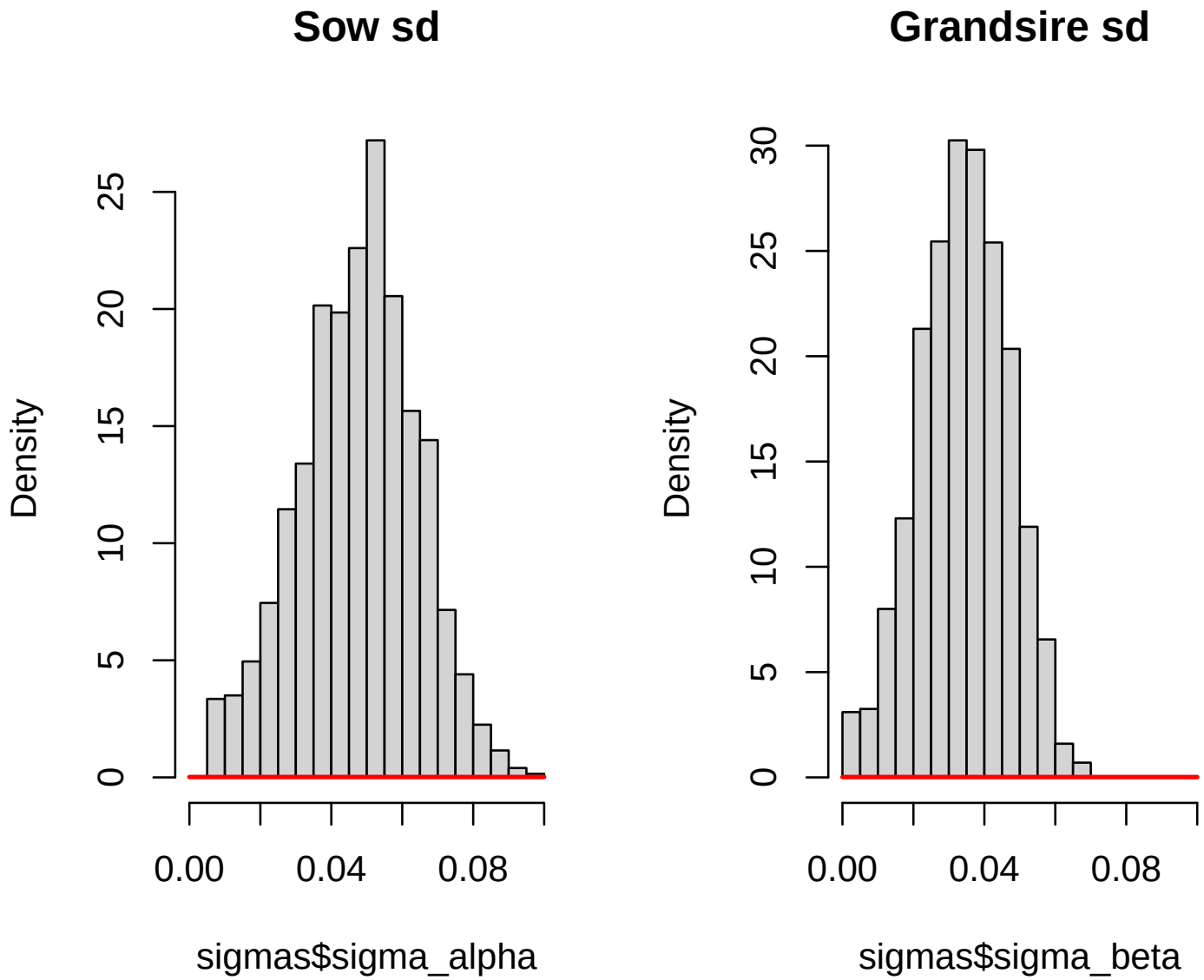
NBA for best and worst performing sows with max parity of 4

```
            sowid  1  2  3  4
1 985152000271823  7  6  5  5
2 985152000856299 15 17 13 13
```

NBA for best and worst performing sows with max parity of 5

```
            sowid  1  2  3  4  5
1 985120012076420 13 14 15 14 18
2 985120026637130  1  7  2 10 10
```


Grandsire (ordered by posterior median)
0
50
100
150
−0.10
−0.05
0.00
0.05
0.10
Effect (log−scale)


Sow sd
Density
0
5
10
15
20
25
0.00
0.04
0.08
sigmas$sigma_alpha
Grandsire sd
Density
0
5
10
15
20
25
30
0.00
0.04
0.08
sigmas$sigma_beta

Culling time

Primary question of interest: when should a sow be removed from breeding?

Who is expected to have more progeny:

- a current sow s
- a new sow

Current sow (for progeny p and grandsire g):

$$\begin{aligned} E[\tilde{y}_s|y] &= E[E[\tilde{y}_s|\rho,\alpha,\beta]|y] \\ &= E\left[e^{\rho_p+\alpha_s+\beta_g}\middle| y\right] \\ &\approx \frac{1}{K}\sum_{k=1}^{K} e^{\rho_p^{(k)}+\alpha_s^{(k)}+\beta_g^{(k)}} \end{aligned}$$

New sow (for progeny 1 and random grandsire):

$$E[\tilde{y}_{new}|y] =\approx \frac{1}{K}\sum_{k=1}^{K} e^{\rho_1^{(k)}+\alpha_{new}^{(k)}+\beta_{new}^{(k)}}$$

where $\alpha_{new}^{(k)} \sim N(0,[\sigma_\alpha^{(k)}]^2)$ and $\beta_{new}^{(k)} \sim N(0,[\sigma_\beta^{(k)}]^2)$.

Simulated answer

For MCMC iterations $k = 1, \ldots, K$,

1. Obtain the k^{th} *joint draw* from the posterior for $\rho_p^{(k)}$, $\rho_1^{(k)}$, $\alpha_s^{(k)}$, $\beta_g^{(k)}$, $\sigma_\alpha^{(k)}$, and $\sigma_\beta^{(k)}$.
2. Calculate $\mu_d^{(k)} = e^{\rho_p^{(k)} + \alpha_s^{(k)} + \beta_g^{(k)}}$.
3. Calculate $\mu_{new}^{(k)} = e^{\rho_p^{(k)} + \alpha_{new}^{(k)} + \beta_{new}^{(k)}}$ where
   a. $\alpha_{new}^{(k)} \sim N\left(0, \left[\sigma_\alpha^{(k)}\right]^2\right)$ and
   b. $\beta_{new}^{(k)} \sim N\left(0, \left[\sigma_\beta^{(k)}\right]^2\right)$.
4. Calculate $\delta^{(k)} = \mu_{new}^{(k)} - \mu_s^{(k)}$.

So $\delta^{(k)}$ is a realization of the expected difference in the number of progeny between a new sow and current sow d.

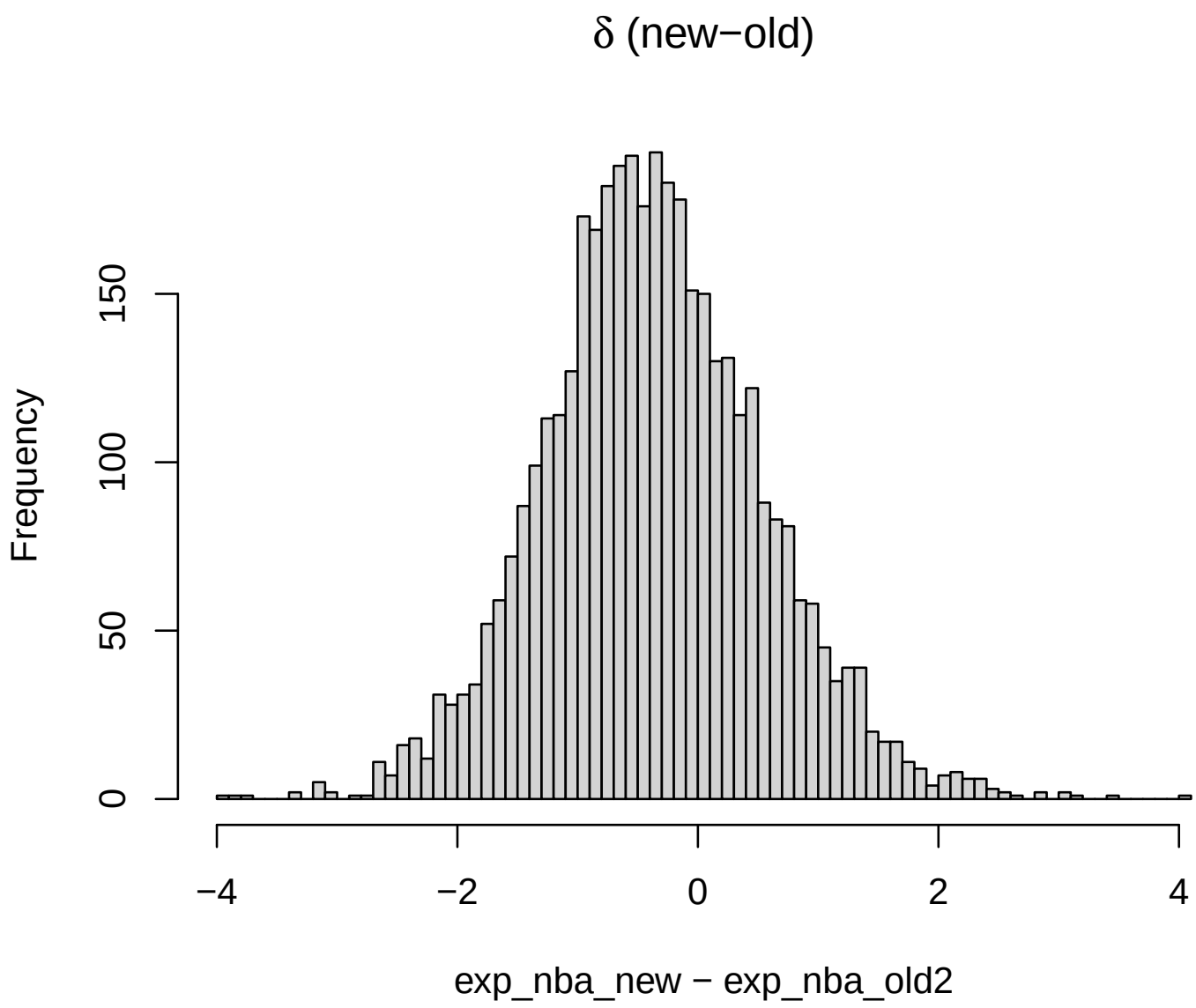
```
sow_summary = ddply(d, .(sowid), summarize,
                    max_parity = max(parity),
                    sum_nba    = sum(nba),
                    mean_nba   = sum_nba/max_parity)

# Find a sow with 0 born alive in parity 1 (and no other data)
sow = which(sow_summary$max_parity==1 & sow_summary$sum_nba==0)[1]
grandsire = as.numeric(d$sire[d$sowid==sow_summary$sowid[sow]]); grandsire = 2 # no idea why

# Expected nba in parity 2 for sow with 0 born alive in first parity
alphas <- extract(r, "alpha")$alpha
betas  <- extract(r, "beta")$beta
rhos   <- extract(r, "rho")$rho
exp_nba_old2 <- exp(rhos[, 2] + alphas[, sow] + betas[, grandsire])

# Expected nba in parity 1 for random sow
alpha_new   <- rnorm(nrow(rhos), 0, sigmas$sigma_alpha)
beta_new    <- rnorm(nrow(rhos), 0, sigmas$sigma_beta)
exp_nba_new <- exp(rhos[, 1] + alpha_new + beta_new)
```

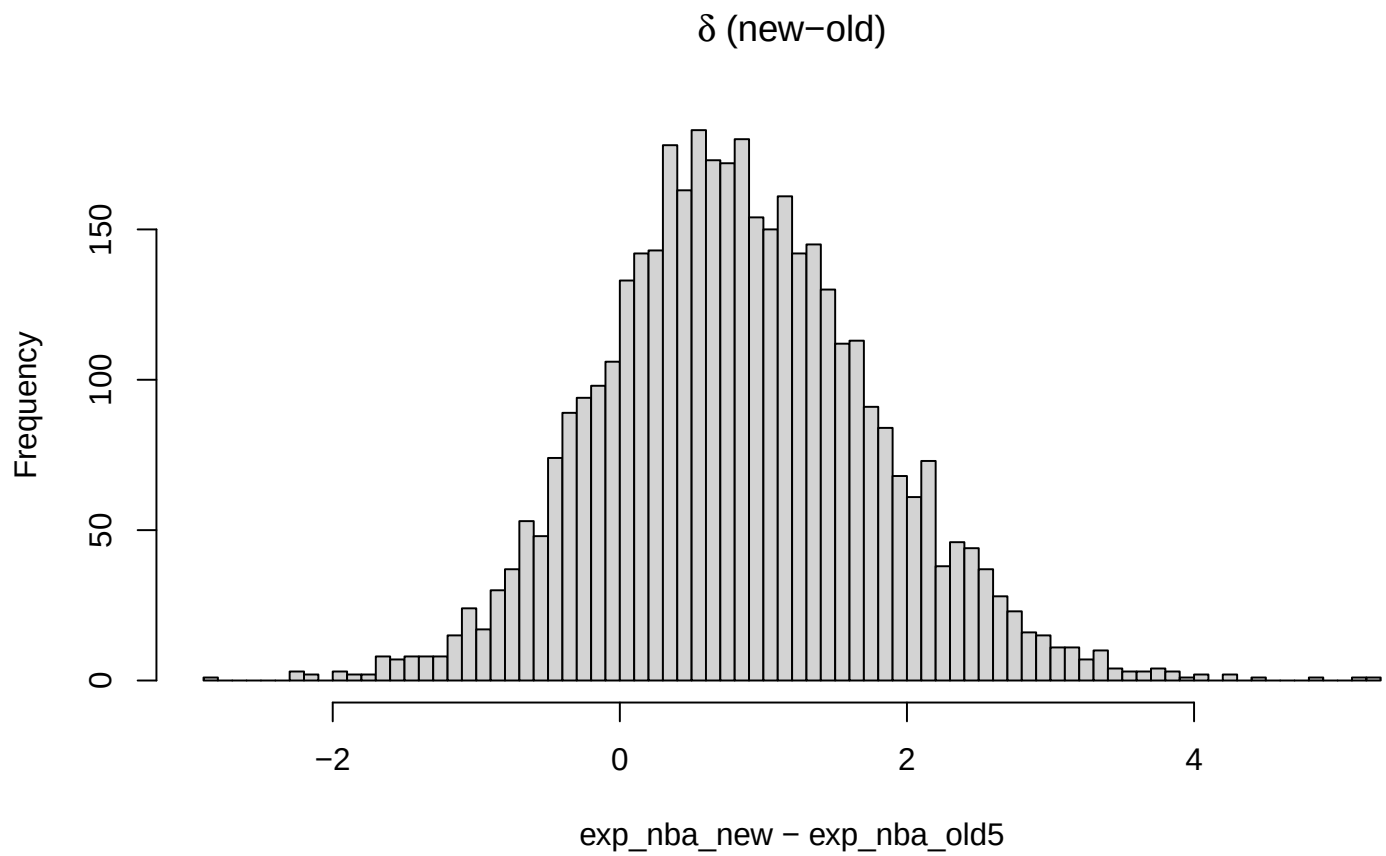
```
hist(exp_nba_new - exp_nba_old2, 100, main=expression(paste(delta," (new-old)")))
```



```r
# Replace sow with 23 born alive after 4 parities?
sow = which(sow_summary$max_parity==4 & sow_summary$sum_nba==23)[1]
grandsire = as.numeric(d$sire[d$sowid==sow_summary$sowid[sow]]); grandsire = 2 # no idea why

# Expected nba in parity 3 for sow with 0 born alive in first parity
exp_nba_old5 = exp(rhos[,5]+alphas[,sow]+betas[,grandsire])

# Difference in expected nba
hist(exp_nba_new-exp_nba_old5, 100, main=expression(paste(delta," (new-old)")))
```



Estimated number born alive

Estimates of the number born alive for each sow with Monte Carlo uncertainty:

```
  variable       est         se
1      new  9.937792 0.01009904
2     old2 10.306574 0.01002058
3     old5  9.107997 0.01103353
```

Estimates of the expected difference with Monte Carlo uncertainty:

```
delta = rbind(as.data.frame(mcse(exp_nba_old2-exp_nba_new)),
              as.data.frame(mcse(exp_nba_old5-exp_nba_new)),
              as.data.frame(mcse(exp_nba_old5-exp_nba_old2)))
rownames(delta) = c("parity2-new","parity5-new","parity5-parity2")
delta
```

```
                      est         se
parity2-new      0.3687822 0.01440155
parity5-new     -0.8297950 0.01506756
parity5-parity2 -1.1985771 0.01274723
```

Utility functions

Suppose the only cost difference is in the number of progeny, then $U(\tilde{y}_i) = u_1 \tilde{y}_i$ and want

$$\max_{i \in \{d, new\}} E[U(\tilde{y}_i)|y]$$

or, equivalently, pick new if

$$E[U(\tilde{y}_{new})|y] - E[U(\tilde{y}_s)|y] > 0$$

But, since expectation is a linear operator,

$$\begin{aligned} E[U(\tilde{y}_{new})|y] - E[U(\tilde{y}_s)|y] &= E[U(\tilde{y}_{new}) - U(\tilde{y}_s)|y] \\ &= E[u_1 \tilde{y}_{new} - u_1 \tilde{y}_s|y] \\ &= u_1 E[\tilde{y}_{new}|y] - u_1 E[\tilde{y}_s|y] \\ &= u_1 E[\delta|y] \end{aligned}$$

So u_1 just scales our posterior expectation for the difference.

Utility functions

Suppose the utility function also involves moving a new sow in, then $U(\tilde{y}_i) = u_1\tilde{y}_i - u_2\mathrm{I}(i = new)$ and you should pick new if

$$E[U(\tilde{y}_{new})|y] - E[U(\tilde{y}_s)|y] = u_1 E[\delta|y] - u_2 > 0.$$

Now suppose an older sow (or this particular sow) needs more medications, then $U(\tilde{y}_i) = u_1\tilde{y}_i - u_2\mathrm{I}(i = new) - u_3\mathrm{I}(i = d)$ and you should pick new if

$$E[U(\tilde{y}_{new})|y] - E[U(\tilde{y}_s)|y] = u_1 E[\delta|y] - u_2 + u_3 > 0.$$

Now, the decision will depend on the individual utilities u_1, u_2, and u_3.

Cost functions

So far, all cost functions have been linear in $\tilde{y}$, but suppose $U(\tilde{y}_i)$ is a complicated function of $\tilde{y}_i$. Then to pick new, we want

$$E[U(\tilde{y}_{new})|y] - E[U(\tilde{y}_s)|y] > 0$$

This may be analytically intractable, but we can easily simulate from it. Suppose

$$U(\tilde{y}_i) = \tilde{y}_i + 0.1(\tilde{y}_i - 10)\mathrm{I}(\tilde{y}_i > 10) - 0.2\mathrm{I}(i = new).$$

New data realizations

To estimate these utility functions, we will need predictive simulations $\tilde{y}$ for the old sow and the new sow.

We can obtain these simulations via

- Old sow (progeny p and average grand-sire):

$$\tilde{y}_s^{(k)} \sim Po\left(e^{\rho_p^{(k)}+\alpha_s^{(k)}}\right)$$

- New sow (average grand-sire):

$$\tilde{y}_{new}^{(k)} \sim Po\left(e^{\rho_1^{(k)}+\alpha_{new}^{(k)}}\right)$$

where $e^{\rho_p^{(k)}+\alpha_s^{(k)}}$ and $e^{\rho_1^{(k)}+\alpha_{new}^{(k)}}$ are simulations previously drawn.

Summary

Lecture demonstrated

- mixed effect Poisson regression model,
- implementation in Stan,
- posterior summaries, and
- using the analysis to make a decision regarding sow culling time.