# ADVANCED BAYESIAN MODELING

# BINOMIAL HIERARCHICAL MODEL IN R/JAGS: **RAT TUMOR RESULTS**

Recall:

- Rat tumor hierarchical model with (diffuse) independent exponential hyperpriors
- Run JAGS model in R using `rjags` package
- `x` has posterior samples of $\alpha$ and $\beta$.

Now to analyze results ...

```
> summary(x)

Iterations = 3501:13500
Thinning interval = 1
Number of chains = 1
Sample size per chain = 10000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

        Mean    SD Naive SE Time-series SE
alpha  3.427 1.360  0.01360         0.1453
beta  20.496 8.053  0.08053         0.8360

2. Quantiles for each variable:

       2.5%    25%    50%   75%  97.5%
alpha 1.595  2.447  3.156  4.09  6.871
beta  9.722 14.599 18.946 24.40 41.474
```

Information from summary:

$$\mathrm{E}(\alpha \mid y) \approx 3.4 \qquad \sqrt{\mathrm{var}(\alpha \mid y)} \approx 1.4$$

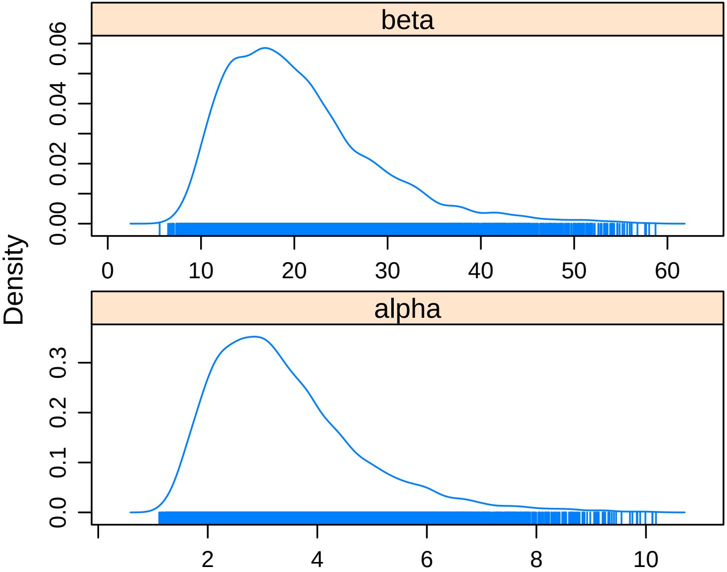
$$\mathrm{E}(\beta \mid y) \approx 20 \qquad \sqrt{\mathrm{var}(\beta \mid y)} \approx 8$$

Approx. 95% central posterior intervals:

$$\alpha : \quad (1.6,\ 6.9) \qquad \beta : \quad (9.7,\ 41)$$

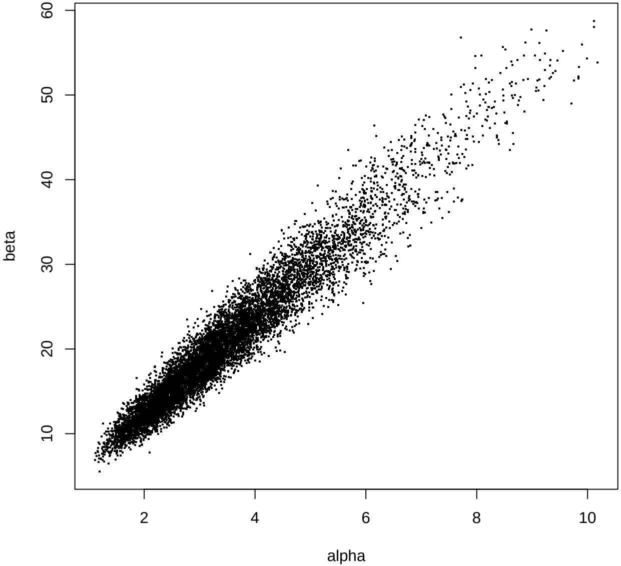
Get estimated posterior densities of $\alpha$ and $\beta$:

```
> require(lattice)
Loading required package: lattice

> densityplot(x)
```


beta
alpha
Density
0.00
0.02
0.04
0.06
0
10
20
30
40
50
60
0.0
0.1
0.2
0.3
2
4
6
8
10

Examine joint posterior distribution of $\alpha$ and $\beta$:

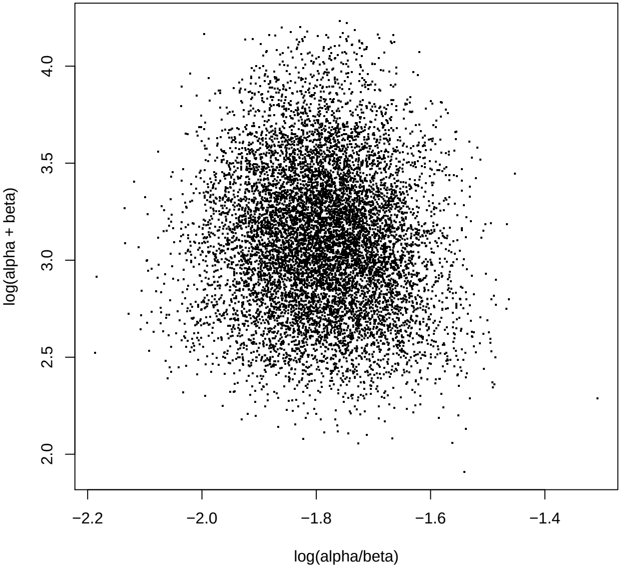
```
> alpha <- as.matrix(x)[,"alpha"]

> beta <- as.matrix(x)[,"beta"]

> plot(alpha, beta, pch=".", cex=2)
```


60
50
40
30
20
10
beta
2
4
6
8
10
alpha

May be more meaningful to graph $\alpha/\beta$ and $\alpha + \beta$, and to use log scales:

```
> plot(log(alpha/beta), log(alpha+beta), pch=".", cex=2)
```

(See BDA3, Sec. 5.3.)


log(alpha + beta)
2.0
2.5
3.0
3.5
4.0
−2.2
−2.0
−1.8
−1.6
−1.4
log(alpha/beta)

Were exponential hyperpriors diffuse enough?

Try different (less informative?) prior ...

# Alternative Model

Recall flat hyperprior proposed in BDA3, Sec. 5.3:

$$p(\phi_1, \phi_2) \;\propto\; 1 \qquad \phi_1 \;=\; \frac{\alpha}{\alpha+\beta} \;\in\; (0,1) \qquad \phi_2 \;=\; (\alpha+\beta)^{-1/2} \;\in\; (0,\infty)$$

Solving,

$$\alpha \;=\; \phi_1/\phi_2^2 \qquad\qquad \beta \;=\; (1-\phi_1)/\phi_2^2$$

Approximate that improper hyperprior by a wide but proper one:

$$\phi_1 \;\sim\; \mathrm{U}(0,1) \qquad \phi_2 \;\sim\; \mathrm{U}(0,1000) \qquad \text{independent}$$

File `rattumor2.bug`:

```
model {

  for (j in 1:length(y)) {
    y[j] ~ dbin(theta[j], N[j])
    theta[j] ~ dbeta(alpha, beta)
  }

  alpha <- phi1 / phi2^2
  beta <- (1-phi1) / phi2^2

  phi1 ~ dunif(0,1)
  phi2 ~ dunif(0,1000)

}
```

Try to run JAGS (in R using `rjags`) as before:

```
> m <- jags.model("rattumor2.bug", d)
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
   Observed stochastic nodes: 71
   Unobserved stochastic nodes: 73
   Total graph size: 223

Initializing model

  |+++++++++++++                                     |  26%
Error: Error in node phi1
Slicer stuck at value with infinite density
```

Origin of error is obscure, relating to misbehavior of a built-in “sampler” in JAGS.

Three options:

- Make hyperprior on $\phi_2$ more informative:

```
phi2 ~ dunif(0,10)
```

- Truncate the beta distribution away from its problematic endpoints:

```
    theta[j] ~ dbeta(alpha, beta) T(0.0001,0.9999)
```

- Turn off the sampler causing the problem.

We choose the third option. Turning off a sampler generally causes JAGS to fall back on another sampler that may not have the same problem.

In R:

```
> set.factory("bugs::BinomSlice","sampler",FALSE)
```

Now try again:

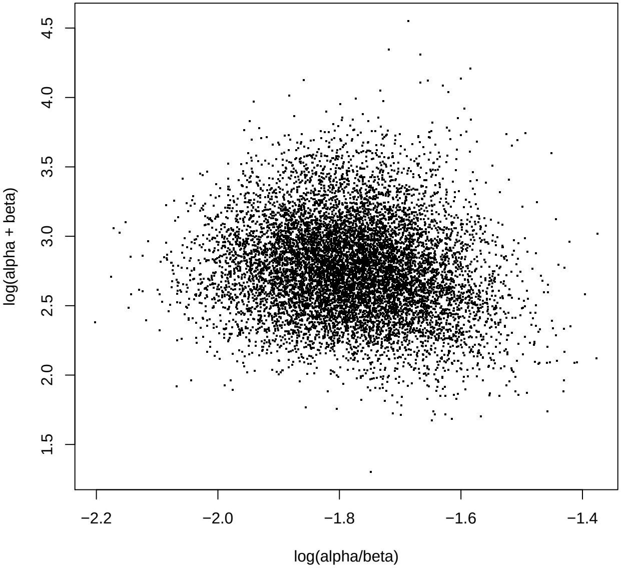
```
> m <- jags.model("rattumor2.bug", d)

...


> update(m, 2500)
  |**************************************************| 100%

> x <- coda.samples(m, c("alpha","beta"), n.iter=10000)
  |**************************************************| 100%
```

Plot posterior jointly distributed samples of $\alpha/\beta$ and $\alpha + \beta$, on log scales:

```
> alpha <- as.matrix(x)[,"alpha"]

> beta <- as.matrix(x)[,"beta"]

> plot(log(alpha/beta), log(alpha+beta), pch=".", cex=2)
```

Compare BDA3, Fig. 5.3.

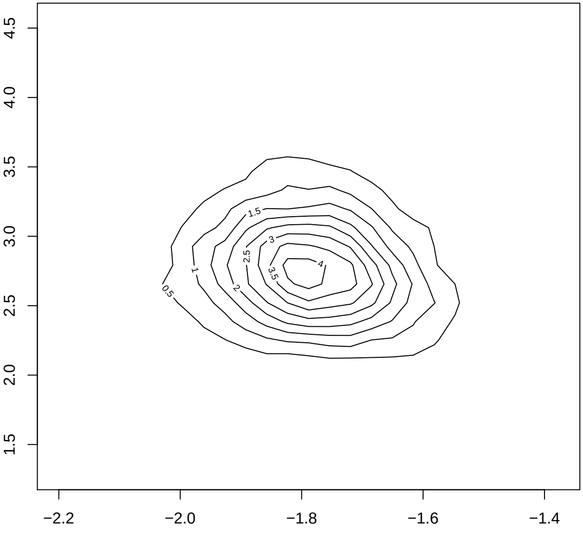
(Also compare with previous model results – try yourself.)


4.5
4.0
3.5
3.0
2.5
2.0
1.5
log(alpha + beta)
−2.2
−2.0
−1.8
−1.6
−1.4
log(alpha/beta)

Try also a contour plot of estimated joint posterior density:

```
> library(MASS)

> contour(kde2d(log(alpha/beta), log(alpha+beta)))
```

Compare BDA3, Fig. 5.3.


4.5
4.0
3.5
3.0
2.5
2.0
1.5
−2.2
−2.0
−1.8
−1.6
−1.4
0.5
1
1.5
2
2.5
3
3.5
4

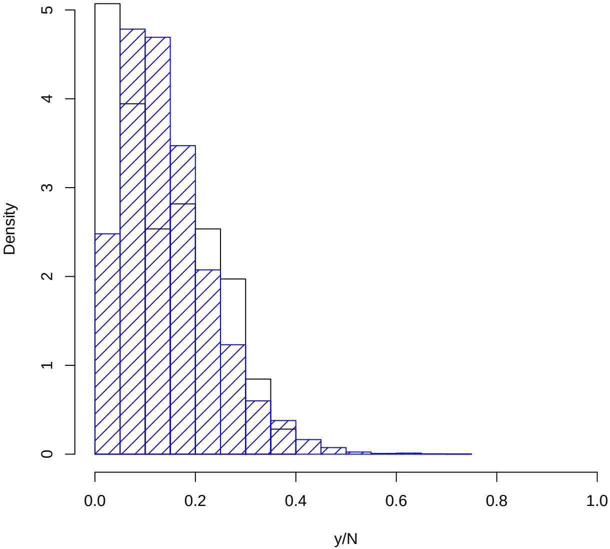
Now consider the posterior predictive distribution of $\tilde{\theta}$, the tumor probability for a "new" experiment, exchangeable with the others.

Can directly simulate $\tilde{\theta}$ using posterior $\alpha$ and $\beta$ samples:

```
> thetatilde <- rbeta(10000, alpha, beta)
```

Then plot as histogram, on same plot with naive empirical histogram:

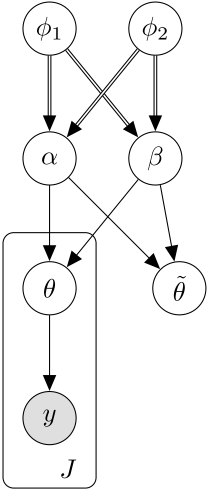
```
> with(d, hist(y/N, freq=FALSE, main="", xlim=c(0,1)))
> hist(thetatilde, freq=FALSE, density = 10, col="blue", border="blue",
+       add=TRUE)
```


Density
0
1
2
3
4
5
0.0
0.2
0.4
0.6
0.8
1.0
y/N

We can add $\tilde{\theta}$ to the DAG.

Note that it is a single (scalar) node with no observed descendants. It is conditionally independent of $\theta$, given its parents ($\alpha$ and $\beta$).

Since $\tilde{\theta}$ is just another node, we can alternatively simulate it using JAGS ...



```
model {

  for (j in 1:length(y)) {
    y[j] ~ dbin(theta[j], N[j])
    theta[j] ~ dbeta(alpha, beta)
  }

  thetatilde ~ dbeta(alpha, beta)

  alpha <- phi1 / phi2^2
  beta <- (1-phi1) / phi2^2

  phi1 ~ dunif(0,1)
  phi2 ~ dunif(0,1000)

}
```

Remark:

Since JAGS uses simulation, seeded differently each time, you should expect slightly different results in each run.

Setting an R seed does *not* set the JAGS seed.

To set the JAGS seed, see the JAGS manual.