NRT Lectures - Statistical Modeling

Bayesian Hierarchical Model

Dr. Bo Li
Department of Statistics
University of Illinois at Urbana-Champaign

Spring 2020

Slides credit to Dr. Trevor Park

Motivation: Rat Tumor Data

Rat Tumor Example

Rats have been used in many experiments to evaluate the effect of drugs on development of tumors.

Tumors may develop even if no drug is administered (zero dose).

What is the baseline probability of a tumor?

A survey of 71 different experiments is conducted, recording

- Number of rats in control (zero-dose) group
- Number of control group rats developing a tumor

Let

$$
\begin{aligned}
n_j &= \text{total number of rats in control group of experiment } j \\
y_j &= \text{number in control group of experiment } j \text{ that develop a tumor} \\
j &= 1, \ldots, 71
\end{aligned}
$$

Data pairs (y_j, n_j):

$$(0,20), \quad (0,20), \quad (0,20), \quad \ldots, \quad (9,24), \quad (4,14)$$

Assume n_j fixed, but y_j random.

If control group rats in experiment j develop tumors independently, and with the same probability,

$$y_j \quad \sim \quad \mathrm{Bin}(n_j, \text{ ? })$$

If

$$\theta_j \quad = \quad \text{control-group tumor probability in experiment } j$$

then

$$y_j \mid \theta_j \quad \sim \quad \mathrm{Bin}(n_j, \theta_j)$$

If there was only one experiment, could use methods described at GLM

What can we do with data from *several* experiments?
What assumptions should we make?

Three options:

1. Same probability for all experiments ($\theta_j = \theta$)
2. Completely unrelated probabilities for different experiments
3. Compromise: Distinct, but related, probabilities

Option 1: Common Probability θ

Just pool data into one binomial sample:

$$y_1 + \cdots + y_{71} \mid \theta \quad \sim \quad \mathrm{Bin}(n_1 + \cdots + n_{71},\, \theta)$$

Give a prior to θ (e.g., beta distribution), and compute posterior.

Drawbacks:

- Contradicted by data (results not shown here)
- No way to assess variability among experiments

Option 2: Unrelated Probabilities θ_j

Analyze each experiment separately (with its own prior).

Drawbacks:

- Limited by precision of each individual experiment
- No Bayesian way to answer questions about *overall* probability
- No way to predict result of a *new* experiment

Option 3: Compromise

Allow separate tumor probabilities

$$\theta_1, \ \ldots, \ \theta_{71}$$

but assume they have something in common.

Idea: Regard the experiments as if sampled from a population of possible experiments.

Then the θ_js are independently random from a common (population) distribution.

Advantages:

- Can define "overall" probability of tumor
- Can assess variability among experiments
- Can possibly improve estimation of individual θ_js through a more informative prior (by pooling information across experiments – later)

But distribution of θ_js is unknown.

Need a higher-level model to make inference based on the data.

Leads to idea of *hierarchical models* ...

Hierarchical Model for Rat Tumors

Recall

$$
\begin{aligned}
n_j &= \text{total number of rats in control group of experiment } j \\
y_j &= \text{number in control group of experiment } j \text{ that develop a tumor} \\
\theta_j &= \text{control-group tumor probability in experiment } j \\
j &= 1, \ldots, 71
\end{aligned}
$$

$$y_j \mid \theta_j \sim \mathrm{Bin}(n_j, \theta_j)$$

Seek to model θ_js as if independent from same distribution.

Natural parametric choice for a distribution of probabilities:

$$\mathrm{Beta}(\alpha, \beta) \qquad\qquad \alpha > 0, \quad \beta > 0$$

Continuous, and gives probability 1 to interval $(0, 1)$

Recall density:

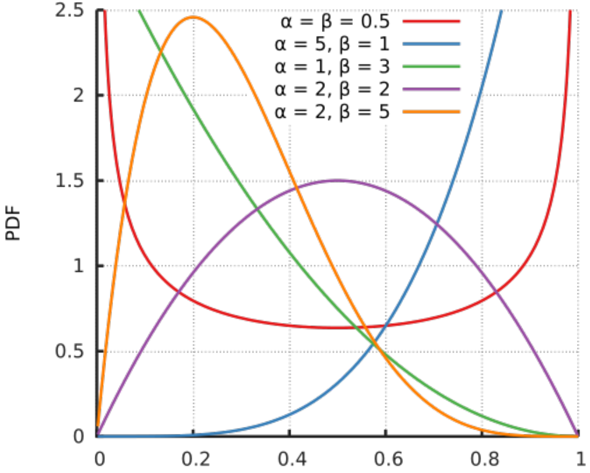
$$p(\theta) \;\; = \;\; \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\, \theta^{\alpha-1}\,(1-\theta)^{\beta-1} \qquad\quad 0 < \theta < 1$$

($\mathrm{U}(0, 1)$ is a special case: $\alpha = \beta = 1$)

So try

$$\theta_j \;\; \sim \;\; \mathrm{Beta}(\alpha, \beta)$$

Some $\mathrm{Beta}(\alpha, \beta)$ densities:



From: Beta distribution. (2017, May 29). In *Wikipedia, The Free Encyclopedia*. Retrieved June 2, 2017 from `https://en.wikipedia.org/w/index.php?title=Beta_distribution&oldid=782786187`

Model now has two levels:

- Lower level:

$$y_j \mid \theta_j \quad \sim \quad \mathrm{Bin}(n_j, \theta_j)$$

- Higher level:

$$\theta_j \quad \sim \quad \mathrm{Beta}(\alpha, \beta)$$

Called a **hierarchical** (or **multilevel**) model

Call α and β **hyperparameters**.

How to choose them?

- Guess (subjective)
- Use prior information (if available)
- Estimate from data ("empirical Bayes")
- Give them a prior distribution ("hierarchical Bayes")

A prior on hyperparameters is a **hyperprior**.

How to choose a hyperprior for $\alpha > 0$ and $\beta > 0$?

No natural (conjugate) choice

Want something that is

- Convenient to specify and use
- Not too informative

For example:

$$\alpha,\, \beta \quad \sim \quad \text{iid Expon}(\lambda) \qquad\qquad \text{some } \lambda > 0$$

so that

$$p(\alpha, \beta) \quad = \quad p(\alpha)\, p(\beta) \quad = \quad \lambda e^{-\lambda\alpha} \cdot \lambda e^{-\lambda\beta} \qquad \alpha > 0,\ \beta > 0$$

This can be made less informative (flatter) by choosing λ closer to zero.

Alternative suggestion:

$$p\left(\frac{\alpha}{\alpha+\beta}, (\alpha+\beta)^{-1/2}\right) \;\propto\; 1 \qquad 0 < \frac{\alpha}{\alpha+\beta} < 1, \quad (\alpha+\beta)^{-1/2} > 0$$

Motivation:

$$\frac{\alpha}{\alpha+\beta} \;=\; \mathrm{E}(\theta_j \mid \alpha, \beta) \;=\; \mu$$

$$(\alpha+\beta)^{-1/2} \;\approx\; \sqrt{\frac{\mathrm{var}(\theta_j \mid \alpha, \beta)}{\mu(1-\mu)}}$$

Note: Improper, but can be shown to give proper posterior

Warning: Using improper hyperpriors can be dangerous (example in BDA3).

Must be able to verify posterior is proper

Independence Assumptions

Natural to assume different experiments are independent (different times, different places, different researchers, ...)

So pairs

$$(y_1, \theta_1), \quad (y_2, \theta_2), \quad \ldots, \quad (y_{71}, \theta_{71})$$

are conditionally independent of each other, given the population of experiments (i.e., given α and β).

Thus θ_js are conditionally independent, given α and β.

We also assume y_j is conditionally independent of α and β, given θ_j.

Can express these relationships through the joint density:

$$
\begin{aligned}
p\big(\{y_j\}, \{\theta_j\}, \alpha, \beta\big) &= p(\alpha, \beta)\, p\big(\{y_j\}, \{\theta_j\} \mid \alpha, \beta\big) \\
&= p(\alpha, \beta) \prod_{j=1}^{71} p(y_j, \theta_j \mid \alpha, \beta) \\
&= p(\alpha, \beta) \prod_{j=1}^{71} p(\theta_j \mid \alpha, \beta)\, p(y_j \mid \theta_j, \alpha, \beta) \\
&= p(\alpha, \beta) \prod_{j=1}^{71} p(\theta_j \mid \alpha, \beta)\, p(y_j \mid \theta_j)
\end{aligned}
$$

Technically more precise to write

$$\{y_j\} \mid \{\theta_j\}, \alpha, \beta \quad \sim \quad \text{indep. Bin}\big(\{n_j\}, \{\theta_j\}\big)$$

For simplicity just write

$$y_j \mid \theta_j \ \sim \ \text{Bin}(n_j, \theta_j)$$

Conditional independence assumptions are implicit.

Similarly, more precise to write

$$\{\theta_j\} \mid \alpha, \beta \quad \sim \quad \text{iid Beta}(\alpha, \beta)$$

For simplicity just write

$$\theta_j \mid \alpha, \beta \;\; \sim \;\; \text{Beta}(\alpha, \beta)$$

Conditional independence assumption is implicit.

Putting it all together:

$$\begin{aligned} y_j \mid \theta_j &\sim \mathrm{Bin}(n_j, \theta_j) \\ \theta_j \mid \alpha, \beta &\sim \mathrm{Beta}(\alpha, \beta) \\ \alpha, \beta &\sim \text{some joint distribution} \end{aligned}$$

Is there a representation that makes hierarchical structure more obvious?

Graphical Models

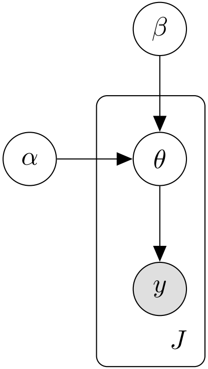
The figure on the right represents a **graphical model**.

Variables are **nodes** and connections are **edges**.

It is a **directed acyclic graph (DAG)**: All edges are arrows in one direction, and there are no cycles.

Each variable in a circled node is random. If the circle is shaded, the variable is observed (data).

The rounded rectangle is a **plate**: All nodes on the plate represent variables that are vectors of length J.

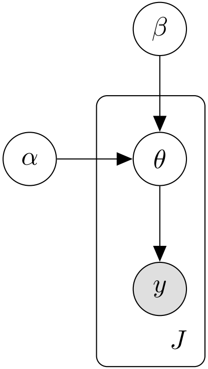


Each edge is from a **parent** to a **child**.

E.g., θ is a child of parents α and β.

In a hierarchical representation, a child's distribution is specified conditionally on its parents only.

Top-level variables (α and β) have distributions specified unconditionally (marginally). They are (marginally) independent if they occupy different nodes.

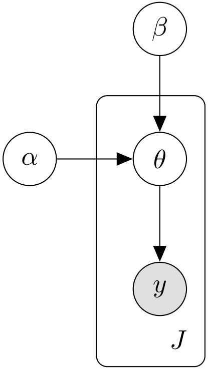


On a plate, different elements of the vector variables are assumed conditionally independent (given the parents).

E.g., the θ_js are conditionally independent given α and β.

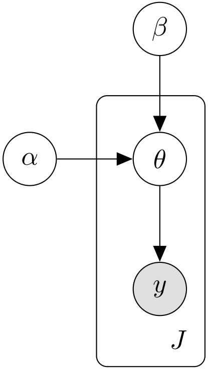
A parent-child dependence on a plate usually indicates that each child *element* conditionally depends only on the corresponding parent *element*.

E.g., y_j (conditionally) depends only on θ_j.



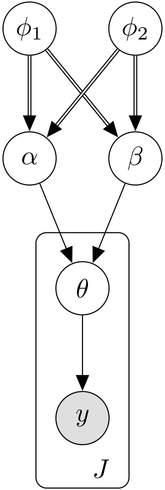
Compare:

$$\begin{aligned} y_j \mid \theta_j &\sim \text{Bin}(n_j, \theta_j) \qquad j = 1, \ldots, J \\ \theta_j \mid \alpha, \beta &\sim \text{Beta}(\alpha, \beta) \qquad j = 1, \ldots, J \\ \alpha, \beta &\sim \text{indep. Expon}(\lambda) \end{aligned}$$



Alternative model:

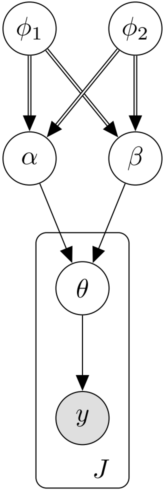
$$
\begin{aligned}
y_j \mid \theta_j &\sim \mathrm{Bin}(n_j, \theta_j) \qquad j = 1, \ldots, J \\
\theta_j \mid \alpha, \beta &\sim \mathrm{Beta}(\alpha, \beta) \qquad j = 1, \ldots, J \\
\phi_1 &= \frac{\alpha}{\alpha + \beta} \sim \mathrm{U}(0, 1) \\
\phi_2 &= (\alpha + \beta)^{-1/2} \sim \text{flat on } (0, \infty)
\end{aligned}
$$



Double arrows represent deterministic relationships.

Nodes α and β are **deterministic**: Defined as an exact function of their parents.

All other nodes are **stochastic**: Defined in terms of a distribution. Their parents (if any) define parameters of the distribution.



If a proper prior is used, then DAG models are well-defined – joint distribution exists and is unique.

Advice: Always make sure your model is a DAG model (unless you really know what you are doing).

Bayesian Simulation Software

BUGS: Bayesian inference Using Gibbs Sampling – software project

- WinBUGS
- OpenBUGS – Windows, Linux, Mac (under Wine)
- JAGS: Just Another Gibbs Sampler

All attempt to automate posterior simulation, requiring only a DAG model to be specified.

BUGS Modeling Languages

BUGS project developed specialized language for model specification, based on DAG models.

JAGS uses a variant of this language.

We will use the JAGS variant, described in manual here:
`https://sourceforge.net/projects/mcmc-jags/files/`

Compare:

$$
\begin{aligned}
y_j \mid \theta_j &\sim \mathrm{Bin}(n_j, \theta_j) \quad j = 1, \ldots, J \\
\theta_j \mid \alpha, \beta &\sim \mathrm{Beta}(\alpha, \beta) \quad j = 1, \ldots, J \\
\alpha, \beta &\sim \text{indep. Expon}(0.001)
\end{aligned}
$$

```
model {

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

  alpha ~ dexp(0.001)
  beta ~ dexp(0.001)

}
```

In JAGS, “~” defines a **stochastic relation**: Variable on the left-hand side is a stochastic node.

Note parameterization of `dbin`. Always check JAGS manual!

Consider this alternative model:

$$\phi_1 \;=\; \frac{\alpha}{\alpha+\beta} \;\sim\; \mathrm{U}(0,1)$$

$$\phi_2 \;=\; (\alpha+\beta)^{-1/2} \;\sim\; \mathrm{U}(0,1000)$$

$$\alpha \;=\; \phi_1/\phi_2^2$$

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

```
model {

  for (j in 1:J) {
    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)

}
```

In JAGS, "<-" defines a **deterministic relation**: Variable on left-hand side is a deterministic (or **logical**) node.

Notes about JAGS:

- Statements within a block may be listed in any order.
- Improper priors are not allowed.
- Data values are not allowed for deterministic nodes.
- See full manual for definitions of distributions.

Running JAGS in R

Rat Tumor Example

$$\begin{aligned}
n_j &= \text{total number of rats in control group of experiment } j \\
y_j &= \text{number in control group of experiment } j \text{ that develop a tumor} \\
\theta_j &= \text{control group tumor probability in experiment } j \\
j &= 1, \ldots, 71
\end{aligned}$$

Data in file `rattumor.txt`:

```
# Rat tumor data from Tarone (1982).  Data from Table 5.1 of Bayesian
# Data Analysis.
# From:  http://www.stat.columbia.edu/~gelman/book/data/rats.asc

y N
0 20
0 20
0 20
0 20
0 20

...

9 24
4 14
```

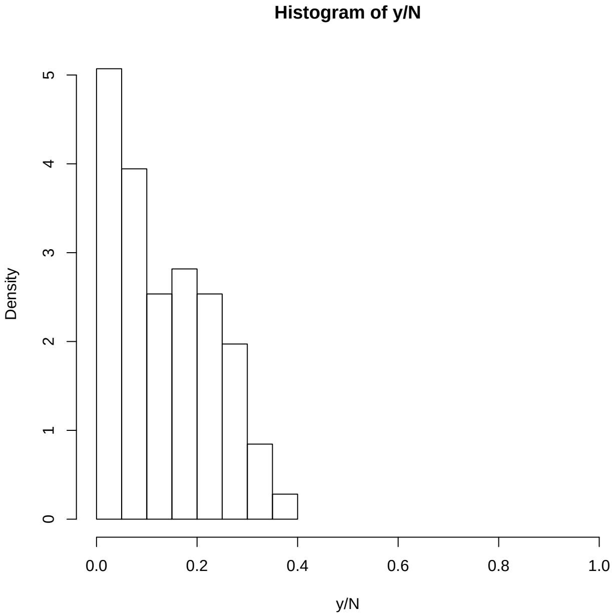
```
> d <- read.table("rattumor.txt", header=TRUE)

> head(d)
  y  N
1 0 20
2 0 20
3 0 20
4 0 20
5 0 20
6 0 20

> summary(d)
       y                N        
 Min.   : 0.000   Min.   :10.00  
 1st Qu.: 1.000   1st Qu.:19.00  
 Median : 3.000   Median :20.00  
 Mean   : 3.761   Mean   :24.49  
 3rd Qu.: 5.000   3rd Qu.:22.50  
 Max.   :16.000   Max.   :52.00  
```

A naive estimate of θ_j:

$$\hat{\theta}_j \quad = \quad y_j/n_j$$

Histogram of $\hat{\theta}_j$s:

```
> with(d, hist(y/N, freq=FALSE, xlim=c(0,1)))
```


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

Model

First try this hierarchical model:

$$\begin{aligned} y_j \mid \theta_j &\sim \mathrm{Bin}(n_j, \theta_j) \\ \theta_j \mid \alpha, \beta &\sim \mathrm{Beta}(\alpha, \beta) \\ \alpha, \beta &\sim \text{indep. Expon}(\lambda) \end{aligned}$$

Choosing λ near zero makes hyperprior flatter (more diffuse).

Later: Choosing λ too small can lead to problems.

JAGS model specification in file `rattumor1.bug`:

```
model {

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

  alpha ~ dexp(0.001)
  beta ~ dexp(0.001)

}
```

- Using $\lambda = 0.001$
- JAGS allows use of `length` function.

Using rjags

First install JAGS (Windows, Mac OS X, Linux):

http://mcmc-jags.sourceforge.net

We will use R package rjags to access JAGS within R:

```
> install.packages("rjags")
```

Now make it available to use:

```
> library(rjags)  # automatically loads coda package
Loading required package: coda
Linked to JAGS 4.3.0
Loaded modules: basemod,bugs
```

Now create a JAGS model object from model in file `rattumor1.bug` and data in data frame `d`:

```
> m <- jags.model("rattumor1.bug", d)
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
   Observed stochastic nodes: 71
   Unobserved stochastic nodes: 73
   Total graph size: 216

Initializing model

  |++++++++++++++++++++++++++++++++++++++++++++++++++| 100%
```

Object `m` defines an iterative random sampling scheme (more later).

First run it for many iterations until it becomes reliable:

```
> update(m, 2500)  # burn-in
  |**************************************************| 100%
```

Then run it for many more iterations, storing samples from selected nodes:

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

Now `x` is a special object containing posterior samples of α and β.

```
> head(x)
[[1]]
Markov Chain Monte Carlo (MCMC) output:
Start = 3501
End = 3507
Thinning interval = 1
        alpha     beta
[1,] 5.204525 33.83398
[2,] 5.092650 32.32364
[3,] 4.945052 32.34294
[4,] 5.384586 29.77359
[5,] 5.831973 29.91007
[6,] 5.660368 29.43142
[7,] 5.809329 31.59240

attr(,"class")
[1] "mcmc.list"
```

Converting x to a matrix makes the variates easier to access:

```
> head(as.matrix(x))
        alpha     beta
[1,] 5.204525 33.83398
[2,] 5.092650 32.32364
[3,] 4.945052 32.34294
[4,] 5.384586 29.77359
[5,] 5.831973 29.91007
[6,] 5.660368 29.43142
```

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 α and β.

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:

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

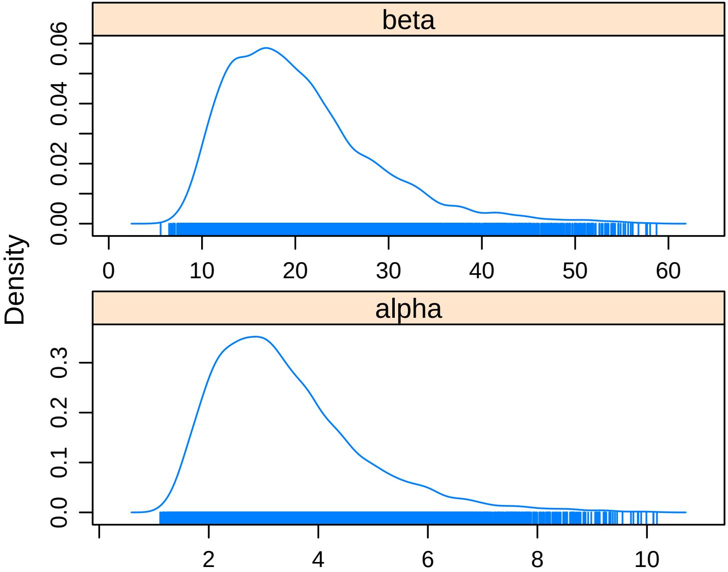
$$E(\beta \mid y) \approx 20 \qquad \sqrt{\mathrm{var}(\beta \mid y)} \approx 8$$

Approx. 95% central posterior intervals:

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

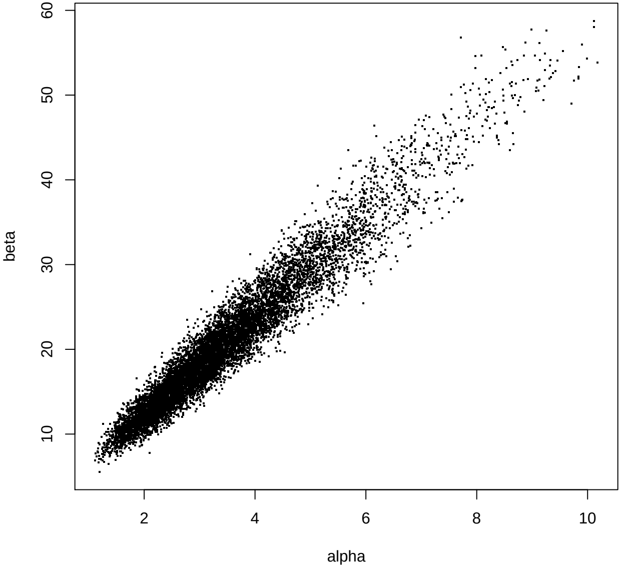
Get estimated posterior densities of α and β:

```
> 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 α and β:

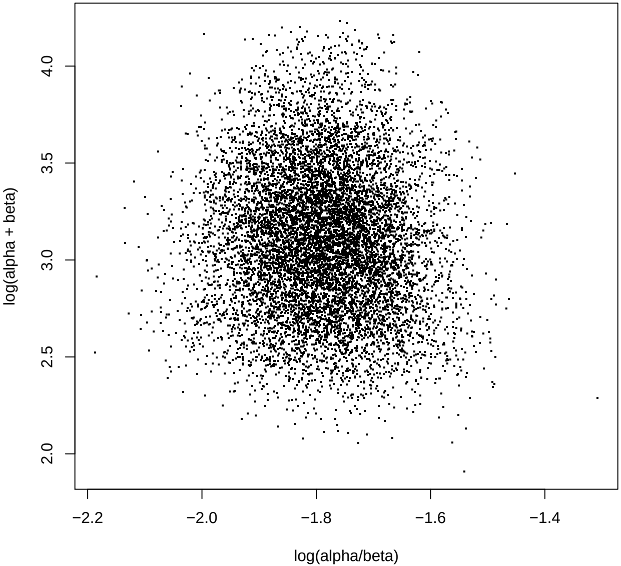
```
> alpha <- as.matrix(x)[,"alpha"]

> 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 α/β 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 ϕ_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:

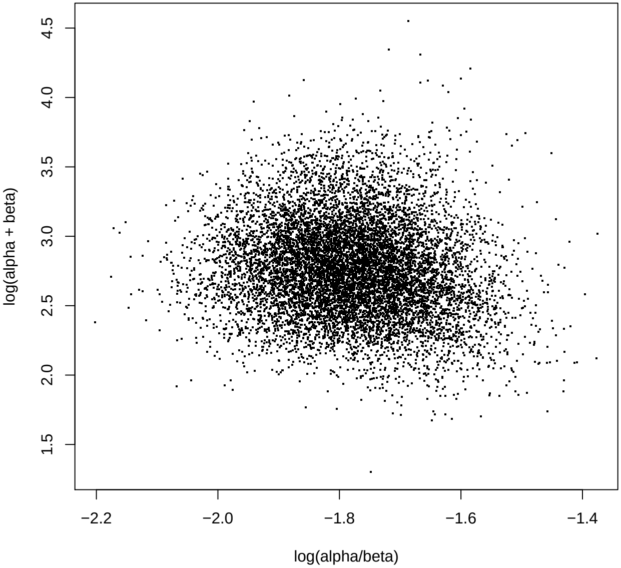
```
> m <- jags.model("rattumor2.bug", d)

...


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

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

Plot posterior jointly distributed samples of α/β 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.

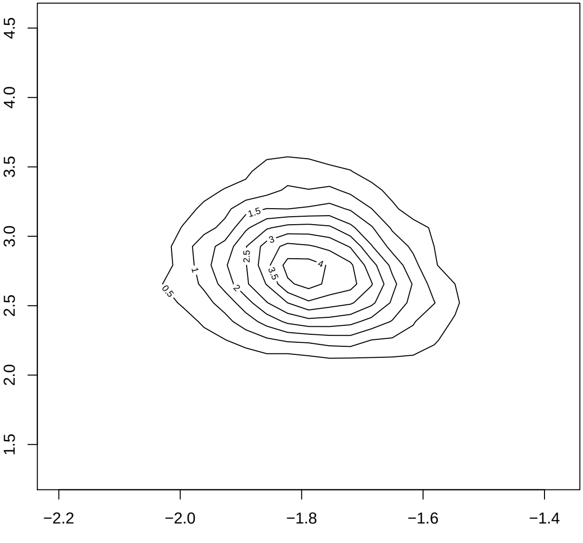
(Also compare with previous model results – try yourself.)


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

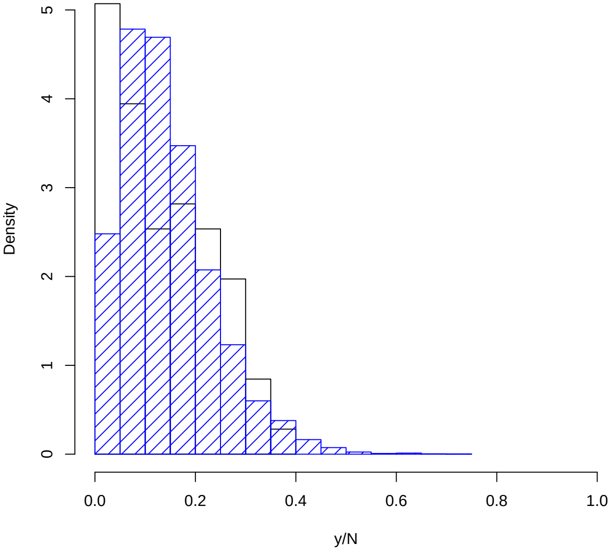
Now consider the posterior predictive distribution of $\tilde{\theta}$, the tumor probability for a "new" experiment, exchangeable with the others.

Can directly simulate $\tilde{\theta}$ using posterior α and β samples:

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

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

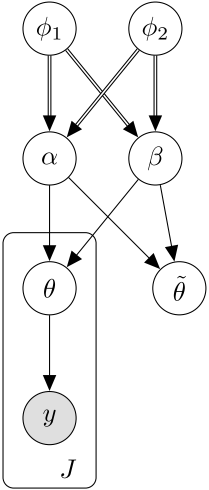
```
> with(d, hist(y/N, freq=FALSE, main="", xlim=c(0,1)))
> hist(thetatilde, freq=FALSE, density = 10, col="blue", border="blue",
+       add=TRUE)
```


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 θ, given its parents (α and β).

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.