JAGS: hierarchical model

Radon is a naturally occurring radioactive gas. Fun fact that I learned: there is no known safe level of exposure to radon. You should always aim to have the lowest level possible. That being said, the EPA recommends homes be fixed if the radon level is 4 pCi/L (picocuries per liter) or higher. It's also been known that most indoor radon comes from the naturally occurring radon in the soil. Thus, basements probably have higher levels of radon compared to floors higher up in a house.

We have observed radon levels from households in Minnesota. For each house, we take a measurement of radon from either the first floor or the basement. We also record the county in which the house is located. Note that houses within a county are physically/geographically close to each other.

From an environmental protection and resources perspective, we are interested in learning if certain counties have higher levels of radon than others, and if this differs by basement levels versus floor at or above ground-level.

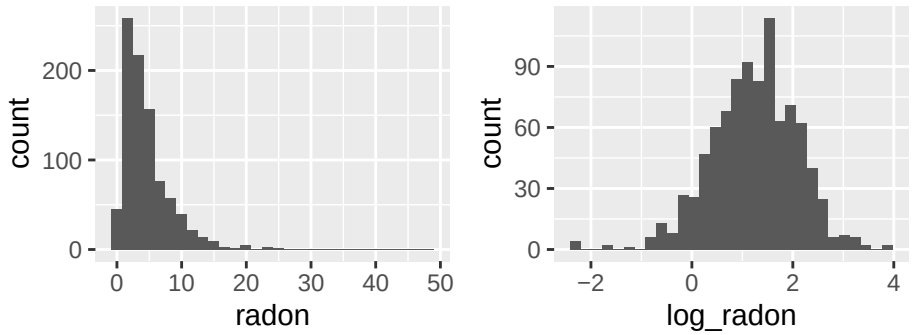
EDA

```
library(tidyverse)
library(rjags)
library(MCMCvis)
mn_radon <- readRDS("mn_radon.Rda")
```

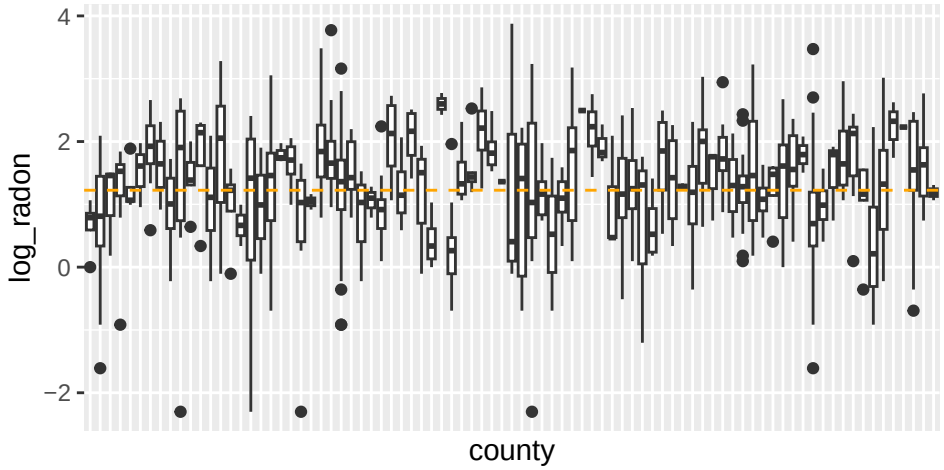
idnum	state	zip	county	floor	radon	radon2	log_radon
5081	MN	55735	AITKIN	first	2.2	2.2	0.7884574
5082	MN	55748	AITKIN	basement	2.2	2.2	0.7884574
5083	MN	55748	AITKIN	basement	2.9	2.9	1.0647107
5084	MN	56469	AITKIN	basement	1.0	1.0	0.0000000
5085	MN	55011	ANOKA	basement	3.1	3.1	1.1314021
5086	MN	55014	ANOKA	basement	2.5	2.5	0.9162907

As you see, the observed radon values are right-skewed with some very large values. We will work the log-transformed versions (a common technique for skewed data) to obtain a roughly

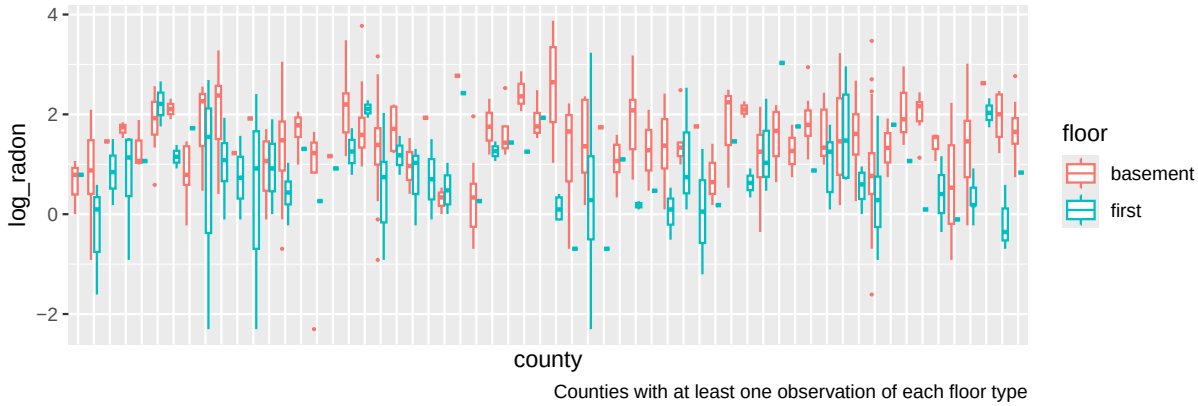
normal-looking distribution on the log scale. However, there were some observed radon values of 0, so we first convert observed radon levels of 0 to 0.1 prior to logging. (This data manipulation "technique" of adding a positive jitter to the 0s is a bit controversial in the modeling world.)



Let's examine the observed distribution of log radon levels by county:



Might this vary by the `floor` type?



Simple linear regression

Let Y_{ij} denote the log-radon level in houes i in county j. We might consider the following simple linear regression model:

$$Y_{ij}|\beta_{0,j}, \beta_{1,j}, \sigma^2, x_{ij} \overset{ind}{\sim} N(\beta_{0,j} + \beta_{1,j}x_{ij}, \sigma^2) \qquad i = 1, \dots, n_j, j = 1, \dots, J$$

Depending on the priors we choose, we could make this complete pooling model, a no-pooling model*, or a hierarchical (partial-pooling) model.

- *We'll consider pooling the variances, but not the regression coefficients.*

These are the first few n_j:

```
AITKIN     ANOKA    BECKER  BELTRAMI    BENTON  BIGSTONE BLUEEARTH     BROWN
     4        52         3         7         4         3        14         4
```

Check: why might I not want a complete pooling model?

No-pool model

To achieve a no-pooling* model for the coefficients, we'll give county-specific independent priors for the coefficients:

$$\beta_{0,j} \overset{iid}{\sim} N(\mu_0, \sigma_0^2) \qquad j = 1, \dots, J$$
$$\beta_{1,j} \overset{iid}{\sim} N(\mu_1, \sigma_1^2) \qquad j = 1, \dots, J$$

and then we'll use the prior

$$\sigma^2 \sim \text{Gamma}(a, b)$$

Under this prior specification, there is no information being shared across the J counties in terms of the $\beta_{0,j}$ or $\beta_{1,j}$ (except through σ^2). We'll choose/set the values of a, b, μ_0, μ_1, σ_0^2, and σ_1^2.

We can easily fit this model in JAGS. What we need to create is a variable/mechanism that tells us which county each observation corresponds to. Then we'll leverage indexing when coding up the model.

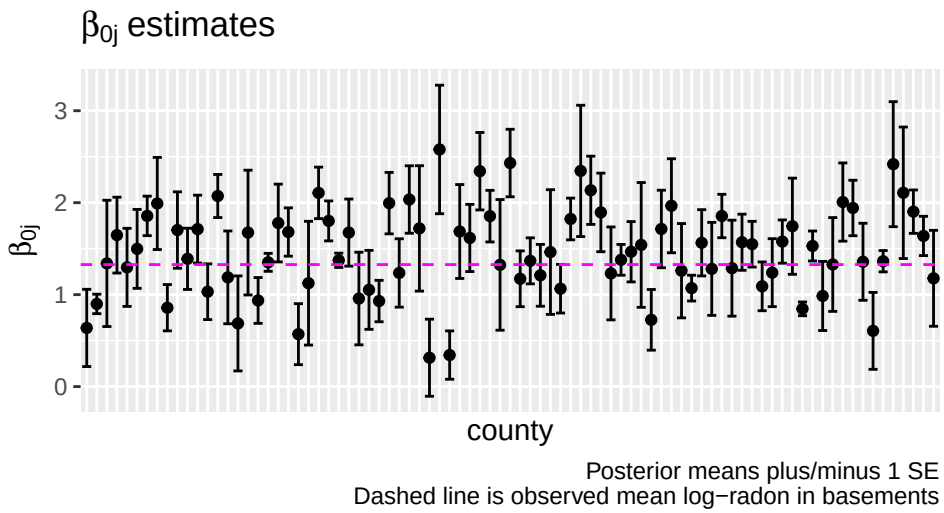
```r
# create variables as necessary
y <- mn_radon$log_radon
x <- model.matrix(~floor, mn_radon)[,2]
n <- length(y)

# need to index counties with a number
county_index <- as.numeric(as.factor(mn_radon$county))
J <- length(unique(county_index))
```

```r
{
sink("../handouts/lm_no_pool.R")
cat("
model {
## sampling model
for (i in 1:n) {
    y[i] ~ dnorm(beta0[county_id[i]] + beta1[county_id[i]]*x[i], 1/sigma2)
}

## priors
for(j in 1:J){
  beta0[j] ~ dnorm(mu0, 1/s20)
  beta1[j] ~ dnorm(mu1, 1/s21)
}
sigma2 ~ dgamma(a, b)
}
", fill = T)
sink()
}
data_ls <- list("y" = y, "n" = n, "x" = x,
                "county_id" = county_index, "J" = J,
                 "mu0" = 1, "s20" = 5,
                "mu1" = 0, "s21" = 5,
                 "a" = 10, "b" = 1)
n_chain <- 2
jm <- jags.model("../handouts/lm_no_pool.R",
                 data = data_ls,
                 n.chains = n_chain)
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
```

```
   Observed stochastic nodes: 919
   Unobserved stochastic nodes: 171
   Total graph size: 3229

Initializing model
```

```r
B <- 5000
update(jm, n.iter = B)
jags_out <- coda.samples(jm,
                         variable.names = c("beta0","beta1", "sigma2"),
                         n.iter = B)
# MCMCtrace(jags_out, pdf = F, params=c("beta0", "sigma2"))
# MCMCsummary(jags_out, params = "beta0")
# MCMCsummary(jags_out, params = "beta1")
```



Posterior means plus/minus 1 SE
Dashed line is observed mean log−radon in basements

Hierarchical model

Let's maintain the same sampling model, but now allow for borrowing of information about the average log-radon level in basements and first-floors of these homes across counties.

Stage 1:

$$\begin{aligned}\beta_{0,j} &\overset{iid}{\sim} N(\mu_{\beta_0}, \sigma^2_{\beta_0}) \qquad j = 1, \dots, J \\ \beta_{1,j} &\overset{iid}{\sim} N(\mu_{\beta_1}, \sigma^2_{\beta_1}) \qquad j = 1, \dots, J\end{aligned}$$

Stage 2:

$$\begin{aligned}\mu_{\beta_0} &\sim N(\mu_0, \sigma^2_0) \\ \mu_{\beta_1} &\sim N(\mu_1, \sigma^2_1) \qquad j = 1, \dots, J \\ \sigma^2_{\beta_0} &\sim \text{Gamma}(a_0, b_0) \\ \sigma^2_{\beta_1} &\sim \text{Gamma}(a_1, b_1)\end{aligned}$$

We'll use the same $\sigma^2 \sim \text{Gamma}(a, b)$ prior as before.

We'll choose/set the values of a, b, a_0, b_0, a_1, b_1, μ_0, μ_1, σ^2_0, and σ^2_1.

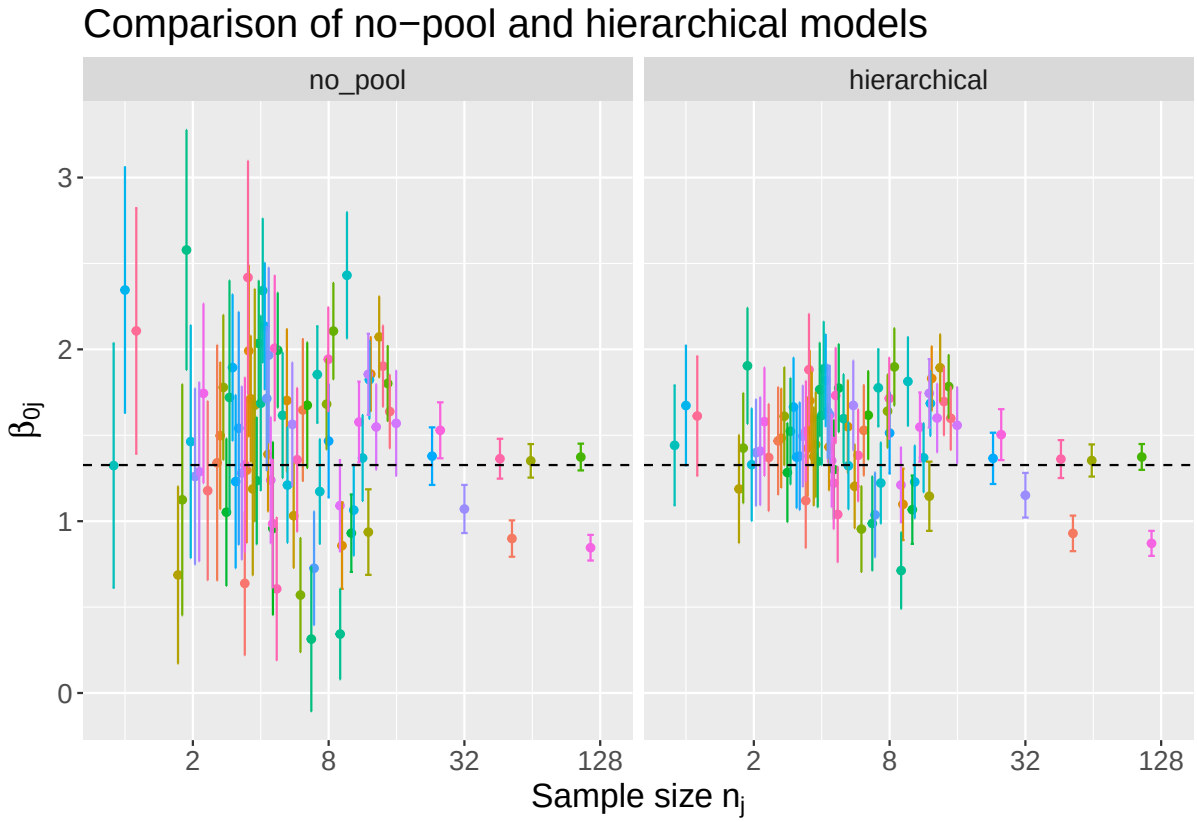
```
{
sink("../handouts/lm_hierarchical.R")
cat("
model {
## sampling model
for (i in 1:n) {
   y[i] ~ dnorm(beta0[county_id[i]] + beta1[county_id[i]]*x[i], 1/sigma2)
}

## priors
for(j in 1:J){
  beta0[j] ~ dnorm(mu_beta0, 1/s2_beta0)
  beta1[j] ~ dnorm(mu_beta1, 1/s2_beta1)
}
mu_beta0 ~ dnorm(mu0, 1/s20)
s2_beta0 ~ dgamma(a0, b0)

mu_beta1 ~ dnorm(mu1, 1/s21)
s2_beta1 ~ dgamma(a1, b1)
```

```
sigma2 ~ dgamma(a, b)
}
", fill = T)
sink()
}
data_ls <- list("y" = y, "n" = n,  "x" = x,
                "J" = J, "county_id" = county_index,
                "mu0" = 1, "s20" = 5,
                "mu1" = 0, "s21" = 5,
                "a0" = 4, "b0" = 1,
                "a1" = 4, "b1" = 1,
                "a" = 10, "b" = 1)
n_chain <- 2
jm2 <- jags.model("../handouts/lm_hierarchical.R",
                  data = data_ls,
                  n.chains = n_chain)
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
   Observed stochastic nodes: 919
   Unobserved stochastic nodes: 175
   Total graph size: 3239

Initializing model
```

```
B <- 5000
update(jm2, n.iter = B)
jags_out2 <- coda.samples(jm2, variable.names = c("beta0", "beta1", "mu_beta0",
                                                  "s2_beta0", "mu_beta1", "s2_beta1",
                                                  "sigma2"),
                          n.iter = B)
# MCMCtrace(jags_out2, pdf = F)
```

Let's see what we gain/learn differently from the hierarchical model compared to the no-pool model:



What do we notice?