Lecture 6 HW: Hierarchical modeling with discrete groups

As before, we need to load some packages and set some options prior to running any models:

```
library(rstan)
library(shinystan)
library(car)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
source("../utilityFunctions.R")
```

Hierarchical binomial models with discrete groups

For the homework, I asked you to use the newt data I simulated (`newtDat.csv`) to model ranavirus infection rates in eastern newts within each pond sample size class.

```
newtDat <- read.csv("newtDat.csv")
head(newtDat)
```

```
  site N trueInfect infect propInf
1    1 5  0.2266825      1     0.2
2    2 5  0.5166810      1     0.2
3    3 5  0.1843030      1     0.2
4    4 5  0.3283834      2     0.4
5    5 5  0.5750651      3     0.6
6    6 5  0.2519853      2     0.4
```

In this data example,

- Newts are censused across $N = 60$ ponds and scored as to whether they are infected or not.
- In some ponds, nary a newt is noticed because the density is low or the pond is hard to sample. Other ponds might be beside themselves with newts (the best ponds).
- Our question is what the average infection rate is within each pond density class ($N =$ 5, 10, 25, 40) while accounting for the nonindependence of ponds in the region.

The Stan model we need for this is identical to the model we wrote last time. The only different is now we will have four $\theta's$ with 15 observations pooled within each θ.

$$\begin{aligned}
infected &\sim \text{Binomial}(N, \theta_i) \\
\theta_i &\sim \text{Beta}(a, b) \\
a &= \omega(\kappa - 2) + 1 \\
b &= (1 - \omega)(\kappa - 2) + 1 \\
\omega &\sim \text{Beta}(\alpha_\omega, \beta_\omega) \\
\kappa &\sim \text{Normal}^+(2, \sigma_\kappa)
\end{aligned} \tag{1}$$

where θ_i is the infection probability for the ith density class and α_ω, β_ω, and σ_κ are constants.

- Here ω will be given a weak Beta$(2, 2)$ prior
- Because κ must be positive and > 2, it has a folded normal distribution and will be given a regularizing prior of $\sigma = 2.5$.

We will code the model (`hierDens.stan`) as follows:

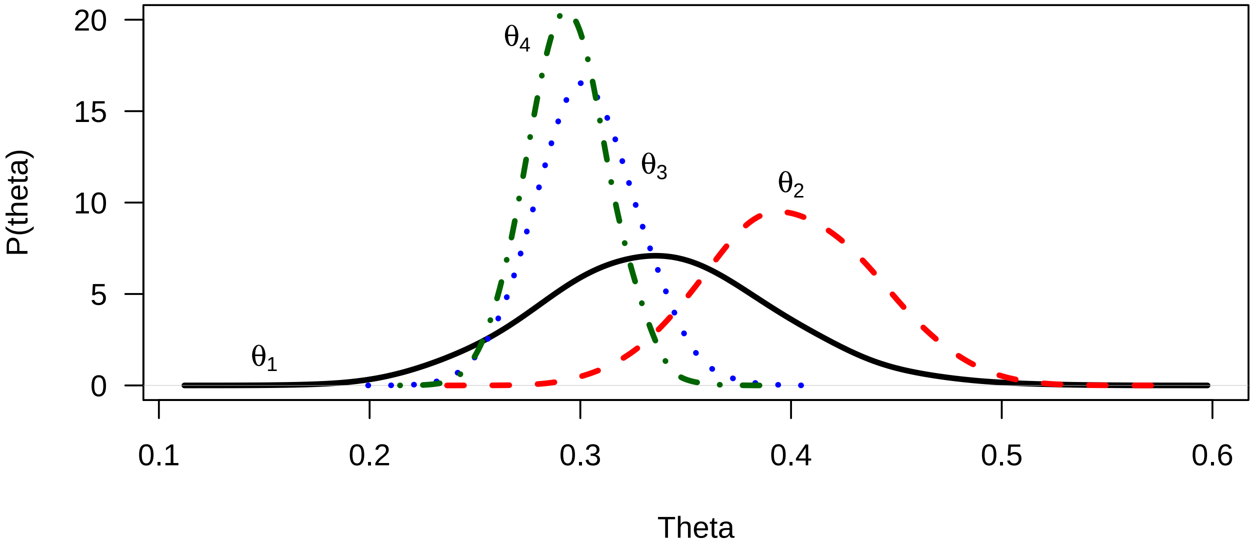
```
data {
  int<lower=0> nObs;          // No. obs.
  int<lower=0> nDens;        // No. groupings
  int<lower=0> density[nObs];  // Indicator values for groupings
  int<lower=0> N[nObs];       // No. sampled newts
  int<lower=0> obs[nObs];     // No. infected newts
  real<lower=0> alpha;        // priors on Omega
  real<lower=0> beta;         // priors on Omega
  real<lower=0> sigma;        // prior on kappa scale
}

parameters {
  real<lower=0, upper=1> omega;     // avg. amg-density infect prob.
  real<lower=2> kappa;              // similarity of sites
  vector<lower=0, upper=1>[nDens] theta; // w/in-density infect prob.
}

transformed parameters {
  real<lower=0> a;
  real<lower=0> b;
    a = omega * (kappa - 2) +1;
    b = (1 - omega) * (kappa - 2) + 1;
}

model {
  omega ~ beta(alpha,beta);              // prior on omega
  kappa ~ normal(2, sigma);              // prior on kappa
  theta ~ beta(a,b);                     // prior for thetas
  for(n in 1:nObs)
  obs[n] ~ binomial(N[n], theta[density[n]]);
}
```

This model is essentially identical to what we did before. I have just changed some variable names for clarity. All that is going to differ is how the data are constructed:

```
# set up the data
nObs    <- nrow(newtDat)
nDens   <- length(unique(newtDat$N))
density <- as.integer(as.factor(newtDat$N))
N       <- newtDat$N
obs     <- newtDat$infect
alpha   <- 2                    # weak beta
beta    <- 2                    # priors
sigma   <- 2.5                  # regularizing scale prior

dat <- list(nObs=nObs, nDens=nDens, density=density, N=N, obs=obs,
            alpha=alpha, beta=beta, sigma=sigma)

modPP <- stan(file="hierDens.stan", data=dat, iter=2000,
              chains=4, seed=3, verbose = FALSE)

thetaPP <- as.matrix(modPP, pars="theta")
```

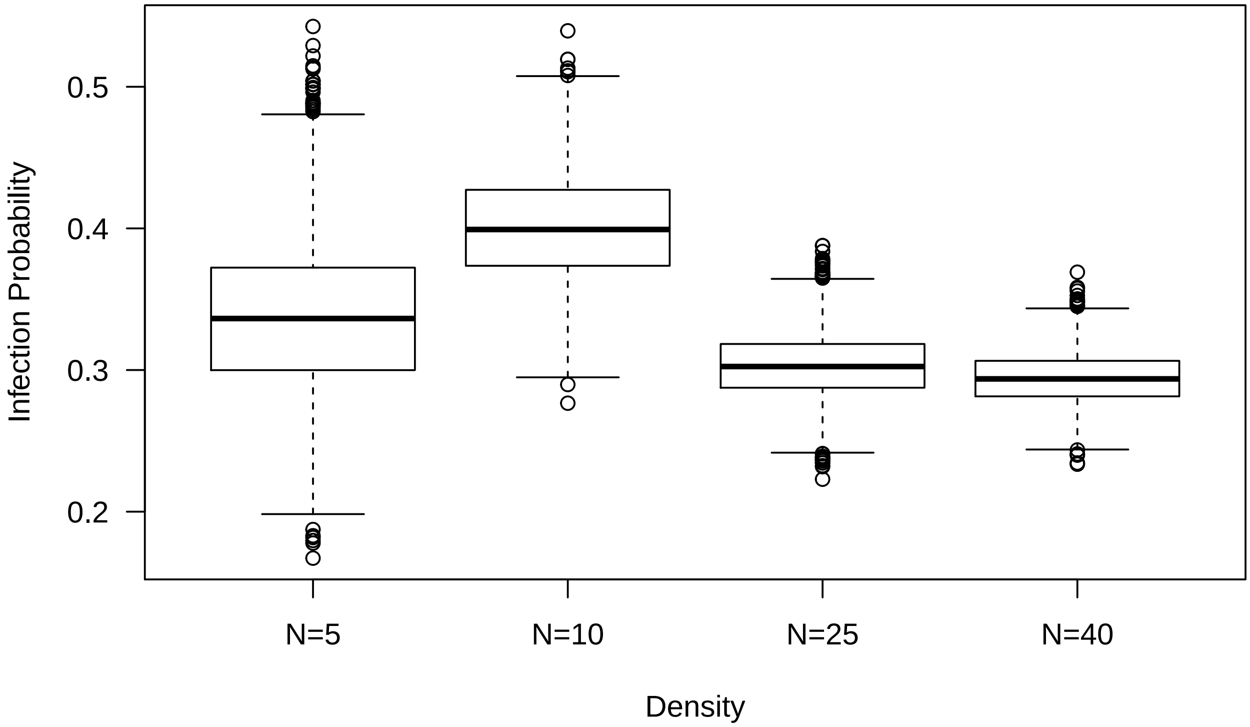
Obviously, given the exercise, we are probably interested in the different $\theta's$ for each density class. We can plot the densities:

```
cols <- c("red", "blue", "darkgreen")
par(mar=c(4,4,0.1,0.5))
plot(density(thetaPP[,1],adj=2), las=1, xlab="Theta",
  ylab="P(theta)", ylim=c(0,20), main="", lwd=3)
for(i in 2:ncol(thetaPP)) {
  lines(density(thetaPP[,i],adj=2), lty=i, col=cols[i-1], lwd=3)
}

text(0.15, 1.5, expression(paste(bold(theta[1]))))
text(0.4, 11, expression(paste(bold(theta[2]))))
text(0.335, 12, expression(paste(bold(theta[3]))))
text(0.27, 19, expression(paste(bold(theta[4]))))
```



or look at the marginal $\theta's$ with something like a boxplot:

```
par(mar=c(4,4,0.1,0.5))
boxplot(thetaPP, las=1, xlab="Density",
  ylab="Infection Probability", xaxt="n")
axis(1, at=c(1, 2, 3, 4), labels=c("N=5", "N=10", "N=25", "N=40"))
```



But how can we quantitatively assess whether the thetas are different or not? Try this as an exercise.

Clearly it looks like there is a difference in infection between sites with the lowest two densities

($N = 5$ & 10). However, we need to be careful when making comparisons among parameters in Bayesian models when priors are shared.

This is because parameters are not independent in joint probability space. There can be correlation. For example, two HDI's may completely overlap marginally, but, at each iteration, one parameter may be greater than the other.

Therefore we need to consider the joint probability distribution, or the difference in two or more parameters. E.g.:

Calculating the HDI of the difference

```
d12 <- thetaPP[,1] - thetaPP[,2]
HDI(d12, credMass=0.95)
```

```
[1] -0.19  0.06
```

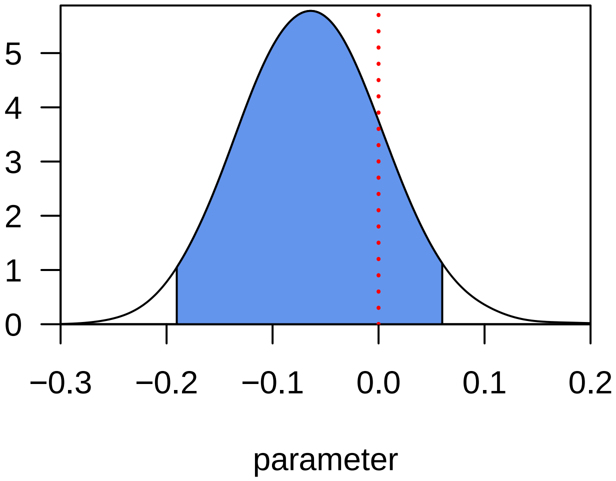
Or calculating the proportion or percentage of times $\theta_1 > \theta_2$.

```
sum(d12>0)/(length(d12))
```

```
[1] 0.17
```

As you can see, even though the marginal distribution boxplots of θ_1 & θ_2 are almost completely overlapping, The 95% HDI of the joint difference $\delta_{1,2}$ does overlap zero $\approx$ 17% of the time.

```
plotInterval(d12, HDI=TRUE, interval=0.95, col="cornflowerblue", xlims=c(-0.3,0.2))
abline(v=0, lwd=2, col="red", lty=3)
```


parameter