Lecture 19: Zero-Inflated Models

** This lecture is based on chapter 11 of Statistical Rethinking by Richard McElreath.*

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

To date, we have covered simple and multiple regression using both normal and non-normal likelihoods. But up to now, we have been concerned with just ONE likelihood function, not several.

But many natural properties are hybrids of multiple processes that may not fit nicely within the assumptions of one particular distribution.

- What we want to do is piece together the simpler components of some of our previous models.
- By endowing a single model with properties of each component, we can statistically model outcome variables with inconvenient, but common properties.

Unfortunately, some of these models are tricky to formulate and even more tricky to understand. We will start with *zero-inflated* models that mix a binary event with an ordinary GLM likelihood such as a Poisson.

Mixtures

As mentioned above, often biological processes are not emissions from any one pure process, but are instead mixtures of different processes.

- When there are different causes for the same observation, mixture models can be useful.

Mixture modeling uses more than one probability distribution to model an amalgamation of causes by using more than one likelihood for the same outcome variable.

Counts are especially prone to belonging to a mixture of processes because zeros can arise by more than one cause.

- Zeros mean nothing has occurred, and this can happen because:
    1. the rate of events is low;
    2. The process that generated the event failed to get started in the first place.

For example, if we are surveying a rare bird species in the woods, we may record a zero because the bird was never in the woods in the first place, or because it was there but we scared them off when we started looking.

Zero-Inflated Poisson (ZIP)

When we discussed regular Poisson GLMs, I used the example of trying to estimate the abundance of a rare plant species to motivate our modeling.

- The process was essentially binomial, but with a large number of trials and a small probability of finding the plant, the distribution was better approximated as a Poisson.

Let us revisit this process: we are still interested in modeling the abundance of plants, but now let us acknowledge that, because the plant is so rare, we do not have a good idea of it's natural history and habitat preferences.

- Thus, we might get zeros because 1) we (or our undergrad minions) are surveying the wrong habitat, or 2) because we are in the correct habitat, but we simply do not observe plants that are present.

How can we model this process? It might be nice to know the probability that a plant is absent (θ), as well as the abundance of the plant when it does occur (λ).

To do this, we need an appropriate likelihood function that mixes these two processes. The likelihood of observing a zero is, therefore:

$$\begin{aligned}\Pr(0|\theta,\lambda) &= \Pr(-|\theta) + Pr(+|\theta) \times \Pr(0|\lambda) \\ &= \theta + (1-\theta)e^{\lambda}.\end{aligned}$$

The Poisson likelihood of y is:

$$\Pr(y|\lambda) = \frac{\lambda^y e^{\lambda}}{y!}$$

so the likelihood that $y = 0$ is just e^{λ}.

We can interpret the above equation as: *The probability of observing a zero is the probability that the plant was never present(-) OR the probability the plant was present but we failed to observe it.*

The likelihood of a non-zero value y is:

$$\begin{aligned}\Pr(y|\theta,\lambda) &= \Pr(-|\theta)(0) + Pr(+|\theta) \times \Pr(y|\lambda) \\ &= (1-\theta)\frac{\lambda^y e^{\lambda}}{y!}.\end{aligned}$$

.

Because we never find $y > 0$ if the plant never occurs at all, the above expression is the chance the plant both occurs AND we find it.

Our full Bayesian model would take a form like the following:

$$\begin{aligned}
y_i &\sim \text{ZIPoisson}(\theta_i, \lambda_i)\\
&= \begin{cases} \theta_i + (1-\theta_i) \times \text{Poisson}(0|\lambda_i) & \text{if } y_i = 0,\\ (1-\theta) \times \text{Poisson}(y_i|\lambda_i) & \text{if } y_i > 0 \end{cases}\\
\text{logit}(\theta_i) &= \alpha_\theta + \beta_\theta x_i\\
\log(\lambda_i) &= \alpha_\lambda + \beta_\lambda x_i\\
\alpha_\theta &\sim \text{Normal}(0, \sigma_{\theta a})\\
\alpha_\lambda &\sim \text{Normal}(0, \sigma_{\lambda a})\\
\beta\theta &\sim \text{Normal}(0, \sigma_{\theta b})\\
\beta\lambda &\sim \text{Normal}(0, \sigma_{\lambda b}).
\end{aligned}$$

Notice that there are two linear models and two link functions, one for each process. The parameters differ, because any predictor x may have different associations with each part of the mixture.

- There is also no requirement that the same predictors are used in both models.

For our example, suppose that we sampled across a broad array of habitats in search of our plant by randomly selecting points on a map, visiting said points, and estimating the number of individuals found as well as abiotic factors.

- Based on previous studies and what little we know about the plant's natural history, we settled on two variables—soil moisture (`moist` and soil nitrogen (`nit`)—that we think are importance in determining the distribution and abundance of this species.
- Specifically, we hypothesize that moisture is more important in determining whether we find the plant at all (α_θ & β_θ).
- We also hypothesize that soil nitrogen is another resource that governs the local carrying capacity (i.e, maximum number of individuals) of a patch (α_λ & β_λ).

```r
dat <- read.csv("plantCount.csv")
names(dat)
```

```
[1] "moist" "nit"   "count"
```

The Stan model (`zipMod.stan`) will look familiar up until the `model` block:

```stan
data {
  int<lower=0> nObs;       // no. obs.
  int<lower=1> nVar;       // no. variables in each pred matrix
  int<lower=0> obs[nObs]; // obs. counts
  matrix[nObs, nVar] XT;  // design mat. including moisture
  matrix[nObs, nVar] XL;  // design mat. including nit.
  real<lower=0> thetaSD;  // value for theta sds
  real<lower=0> lambdaSD; // value for lambda sds
}

parameters {
  vector[nVar] betaT;     // logit betas for absence
  vector[nVar] betaL;     // log betas for abundance
}

transformed parameters {
  vector[nObs] logitTheta;
  vector[nObs] logLambda;

  logitTheta = XT * betaT;  // regression for thetas
  logLambda = XL * betaL;   // reg. for lambdas
}

model {
  betaT ~ normal(0, thetaSD); // priors for thetas
  betaL ~ normal(0, lambdaSD); // priors for lambdas

  for(n in 1:nObs) {
    if(obs[n] == 0)
      target += log_sum_exp(
        bernoulli_logit_lpmf(1 | logitTheta[n]),
        bernoulli_logit_lpmf(0 | logitTheta[n])
          + poisson_log_lpmf(obs[n]|logLambda[n]));
      else
        target += bernoulli_logit_lpmf(0 | logitTheta[n])
          + poisson_log_lpmf(obs[n] | logLambda[n]);
  }
}
```

In the `model` block, the first thing to note is that we have to do a loop. Currently there is no way to vectorize mixtures.

The next thing to notice is the `target +=` which increments the log probability of the likelihood.

Then there is the following statement:

```
if(obs[n] == 0)
      target += log_sum_exp(
        bernoulli_logit_lpmf(1 | logitTheta[n]),
        bernoulli_logit_lpmf(0 | logitTheta[n])
          + poisson_log_lpmf(obs[n]|logLambda[n]));
```

This says, if $obs_n = 0$, we want to add two log probabilities on a linear scale (`log_sum_exp`). It is defined to equal $\log(e^{lp1} + e^{lp2})$.

- $lp1$ then equals the Bernoulli logit probability of a true absence: `bernoulli_logit_lpmf(1 | logitTheta[n]))`. The `(1 | logitTheta)` indicates that θ should model the probability of an absence and $(1 - \theta)$ is the probability of presence.
- $lp2$ models the probability of the plant actually being present but unobserved $(1 - \theta)$ added to the expected abundance estimate, `bernoulli_logit_lpmf(0 | logitTheta[n]) + poisson_log_lpmf(obs[n]|logLambda[n]))`.

The second part of the likelihood function:

```
else
  target += bernoulli_logit_lpmf(0 | logitTheta[n])
          + poisson_log_lpmf(obs[n] | logLambda[n]);
```

models the Poisson abundance given that we did observe at least one individual: $(1 - \theta) \times \lambda_{\mathrm{n}}$.

Thus, θ serves both as the probability of absence **AND** as a parameter defining the mixing proportions for the two likelihoods.

Now, let's run the complete model:

```
XT <- model.matrix(~moist, data=dat)
XL <- model.matrix(~nit, data=dat)
nObs <- nrow(dat)
obs <- dat$count

d <- list(nObs=nObs, nVar=2, obs=obs, XT=XT, XL=XL, thetaSD=1,
  lambdaSD=1)
m1 <- stan(file="zipMod.stan", data=d, iter=2000, chains=4,
  seed=867.5309)
```

Lets look at the parameters for the probability of true absences:

```
print(m1, pars="betaT")
```

```
Inference for Stan model: zipMod.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

          mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
betaT[1]  1.37    0.01 0.45  0.52  1.06  1.35  1.66  2.29  1680    1
betaT[2] -0.23    0.00 0.08 -0.38 -0.28 -0.23 -0.18 -0.09  1743    1

Samples were drawn using NUTS(diag_e) at Tue Apr 24 11:14:16 2018.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

On the natural scale, these estimates are

```
logistic(1.35)
```

```
[1] 0.7941296
```

```
logistic(1.35-0.23)
```

```
[1] 0.7539887
```

Thus, when `moisture` $= 0$, there is an 80% probability that we will never find our plant, but that probability decreases by $\approx 5\%$ when moisture increases by one unit.

- Remember parameters in logistic regressions interact with themselves and all other parameters so it is often better to plot the results (see below).

We can also look at the results for the Poisson part of the model:

```r
print(m1, pars="betaL")
```

```
Inference for Stan model: zipMod.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

         mean se_mean   sd 2.5%  25%  50%  75% 97.5% n_eff Rhat
betaL[1] 0.75    0.01 0.26 0.21 0.58 0.75 0.92  1.26  1745    1
betaL[2] 0.04    0.00 0.02 0.01 0.03 0.04 0.06  0.08  1752    1

Samples were drawn using NUTS(diag_e) at Tue Apr 24 11:14:16 2018.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

On the natural scale,

```r
exp(0.75)
```

```
[1] 2.117
```

```r
exp(0.04)
```

```
[1] 1.040811
```

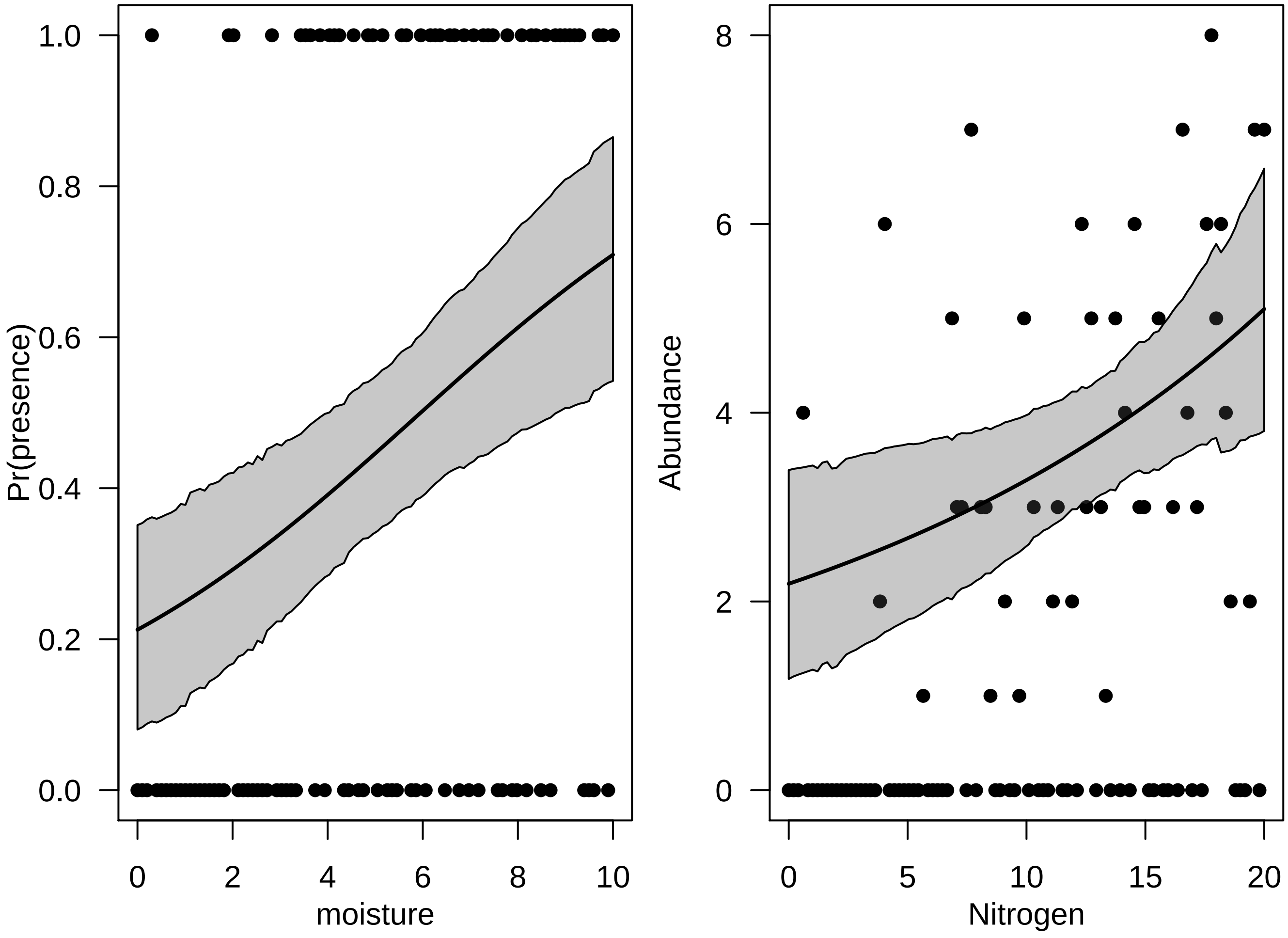
```r
exp(0.75+0.04)
```

```
[1] 2.203396
```

Thus, when we do find a plant, we would normally expect 1–3 plants when the nitrogen content is 0. Increasing nitrogen by one unit increases the average abundance by about 4%.

- Again, it is usually better to plot out the results rather than to rely too much on interpreting tables:

```r
par(mar=c(3,3.2,0.1,0.5))
par(mfrow=c(1,2))

# Plot theta
theta <- logistic(as.matrix(m1,"logitTheta"))
avTheta <- 1-colMeans(theta)
hdiTheta <- 1 - apply(theta, 2, HDI, credMass=0.95)

x <-  dat$moist
y <- ifelse(dat$count == 0, 0, 1)
plot(x, y,  type="p", ann=FALSE, las=1, pch=16)
mtext("moisture", side=1, line=2)
mtext("Pr(presence)", side=2, line=2.2)

polygon(c(x, rev(x)), c(hdiTheta[1, ], rev(hdiTheta[2,])),
  col="#50505050")
lines(x,avTheta, lwd=2)

# plot lambda
lambda <- exp(as.matrix(m1, "logLambda"))
avLam <- colMeans(lambda)
hdiLam <- apply(lambda, 2, HDI, credMass=0.95)

x <-  dat$nit
y <- dat$count
plot(x, y,  type="p", ann=FALSE, las=1, pch=16)
mtext("Nitrogen", side=1, line=2)
mtext("Abundance", side=2, line=2.2)

polygon(c(x, rev(x)), c(hdiLam[1, ], rev(hdiLam[2,])),
  col="#50505050")
lines(x,avLam, lwd=2)
```



The only thing to note in these plots is that I have changed the probability of absence (θ) to the probability of presence ($1 - \theta$) because I find it more intuitive.

Note that if we still thought there was overdispersion, say in the counts > 0, we could have replaced the Poisson likelihood with a negative binomial distribution.

There is also no reason why we can't have zero-inflated binomial distributions or zero-inflated beta-binomial distributions. Setting them up in Stan would use the general syntax.