Class 8: Partial pooling and zero-inflation

Andrew Parnell
andrew.parnell@mu.ie

Maynooth University
National University of Ireland Maynooth

Learning outcomes:

- Be able to describe the advantages of partial pooling
- Be able to fit some basic zero inflation and hurdle models
- Be able to understand and fit some multinomial modelling examples

A false dichotomy: fixed vs random effects

- We've been fitting a model with varying intercepts and slopes to the earnings data:

$$y_i \sim N(\alpha_{\mathrm{eth}_i} + \beta_{\mathrm{eth}_i} x_i, \sigma^2)$$

  where:

$$\alpha_j \sim N(\mu_\alpha, \sigma_\alpha^2) \text{ and } \beta_j \sim N(\mu_\beta, \sigma_\beta^2)$$

- In traditional parlance this is a random effects model
- When we fit our model we are learning about the values of the slopes and intercepts, and also the values of their means and standard deviations

The extremes of varying vs fixed parameters

- Now consider what happens when σ_α and σ_β get smaller and smaller. What will happen to the values of the slopes and the intercepts?
- Alternatively, consider what happens as σ_α and σ_β get larger and larger?
- Are these still random effects models?

The advantages of borrowing strength

- The process of σ_α and σ_β getting smaller or larger will control the degree to which the slopes and intercepts are similar to each other
- If they are similar to each other we say they are *borrowing strength* as data in the other groups is influencing the intercept/slope. This is a powerful idea
- Mathematically you can write out the estimated mean of the parameters as a weighted average of the group mean and the overall mean where the weights are dependent on the group and overall variance and sample sizes.
- Because of the weighted nature of the estimate this is often called *partial pooling*

Zero-inflation and hurdle models

- Let's introduce some new data. This is data from an experiment on whiteflies:

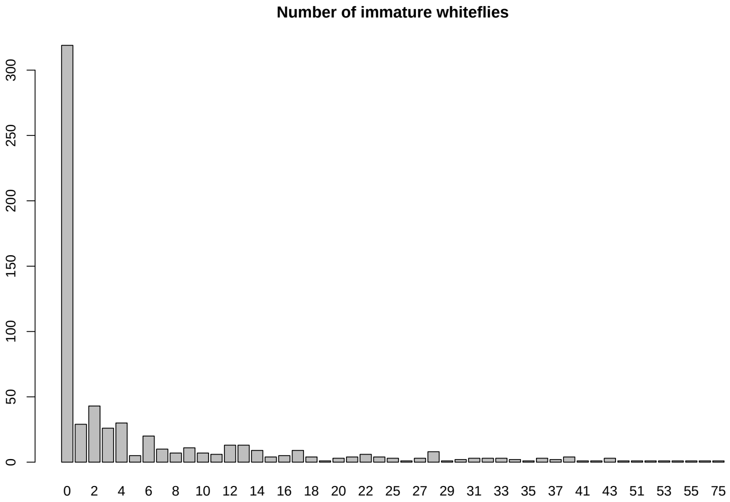
```
wf = read.csv('../data/whitefly.csv')
head(wf)
```

```
##   imm week block trt  n live plantid
## 1  15    1     3   5 12   11       1
## 2  16    2     3   5  8    6       1
## 3  28    3     3   5 10   10       1
## 4  17    4     3   5 10    8       1
## 5   9    5     3   5 10   10       1
## 6  28    6     3   5 10   10       1
```

The response variable here is the count `imm` of immature whiteflies, and the explanatory variables are `block` (plant number), `week`, and treatment `treat`.

Look at those zeros!

```r
barplot(table(wf$imm),
        main = 'Number of immature whiteflies')
```



A first model

- These are count data so a Poisson distribution is a good start
- Let's consider a basic Poisson distribution model for Y_i, $i = 1, \ldots, N$ observations:

$$Y_i \sim Po(\lambda_i)$$

$$\log(\lambda_i) = \beta_{\mathsf{trt}_i}$$

- We'll only consider the treatment effect but we could run much more complicated models with e.g. other covariates and interactions

Fitting the model in JAGS

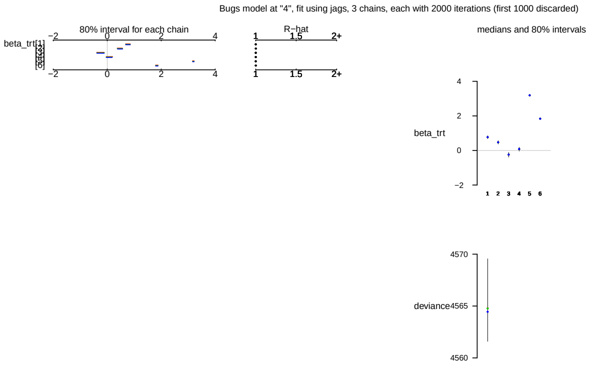
```
model_code = '
model
{
  # Likelihood
  for (i in 1:N) {
    y[i] ~ dpois(lambda[i])
    log(lambda[i]) <- beta_trt[trt[i]]
  }
  # Priors
  for (j in 1:N_trt) {
    beta_trt[j] ~ dnorm(0, 100^-2)
  }
}
'
```

Running the model

```
jags_run = jags(data = list(N = nrow(wf),
                            N_trt = length(unique(wf$trt)),
                            y = wf$imm,
                            trt = wf$trt),
                parameters.to.save = 'beta_trt',
                model.file = textConnection(model_code))
```

Results

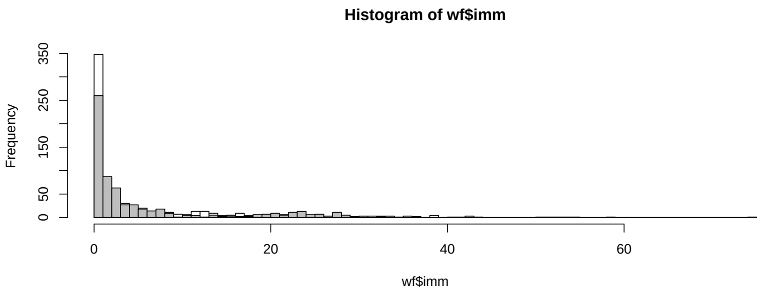
```r
plot(jags_run)
```



Some clear treatment effects - treatment 5 in particular

Did the model actually fit well?

```
beta_means = jags_run$BUGSoutput$mean$beta_trt
y_sim_mean = exp(beta_means[wf$trt])
y_sim = rpois(nrow(wf), y_sim_mean)
hist(wf$imm, breaks = seq(0,max(wf$imm)))
hist(y_sim, breaks = seq(0,max(wf$imm)),
     add = TRUE, col = 'gray')
```



What about the zeros?

- One way of broadening the distribution is through over-dispersion which we have already met:

$$\log(\lambda_i) \sim N(\beta_{\text{trt}_i}, \sigma^2)$$

- However this doesn't really solve the problem of excess zeros
- Instead there are a specific class of models called *zero-inflation* models which use a specific probability distribution. The zero-inflated Poisson (ZIP) with ZI parameter q_0 is written as:

$$p(y|\lambda) = \begin{cases} q_0 + (1 - q_0) \times \mathit{Poisson}(0, \lambda) & \text{if } y = 0 \\ (1 - q_0) \times \mathit{Poisson}(y, \lambda) & \text{if } y \neq 0 \end{cases}$$

Fitting models with custom probability distributions

- The Zero-inflated Poisson distribution is not included in Stan or JAGS by default. We have to create it
- It's possible to create new probability distributions in Stan
- It's a little bit fiddly to do so in JAGS, we have to use some tricks
- We will use JAGS to create a mixture of Poisson distributions; A Poisson(0) distribution for the zeros, and a Poisson(λ) distribution for the rest

Fitting the ZIP in JAGS

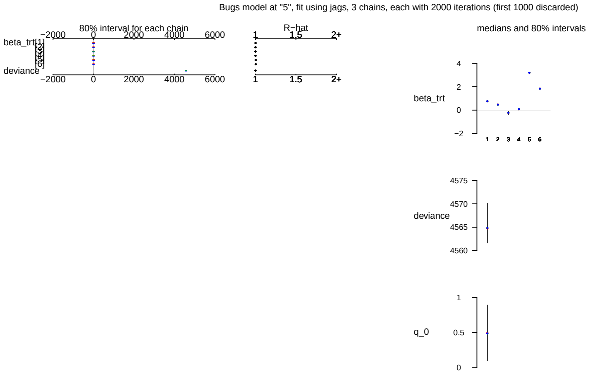
```
model_code = '
model
{
  # Likelihood
  for (i in 1:N) {
    y[i] ~ dpois(lambda[i] * z[i] + 0.0001)
    log(lambda[i]) <- beta_trt[trt[i]]
    z[i] ~ dcat(q_1)
  }
  # Priors
  for (j in 1:N_trt) {
    beta_trt[j] ~ dnorm(0, 100^-2)
  }
  q_1 <- 1 - q_0
  q_0 ~ dunif(0, 1)
}
'
```

Running the model

```
jags_run = jags(data = list(N = nrow(wf),
                            N_trt = length(unique(wf$trt)),
                            y = wf$imm,
                            trt = wf$trt),
                parameters.to.save = c('beta_trt','q_0'),
                model.file = textConnection(model_code))
```

Results

```r
plot(jags_run)
```

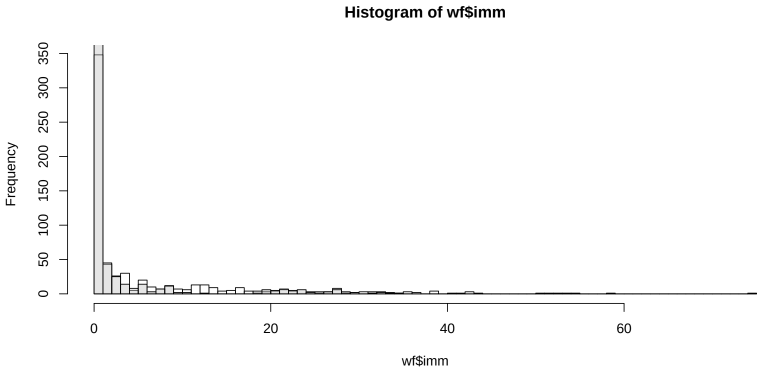


Did it work any better? - code

```
beta_means = jags_run$BUGSoutput$mean$beta_trt
q_0_mean = jags_run$BUGSoutput$mean$q_0[1]
y_sim_mean = exp(beta_means[wf$trt])
rZIP = function(mean, q_0) {
  pois = rpois(length(mean), mean)
  pois[runif(length(mean))<q_0] = 0
  return(pois)
}
y_sim = rZIP(y_sim_mean, q_0_mean)
```

Did it work any better? - picture

```r
hist(wf$imm, breaks = seq(0,max(wf$imm)))
hist(y_sim, breaks = seq(0,max(wf$imm)),
     add = TRUE, col = rgb(0.75,0.75,0.75,0.4))
```



Some more notes on Zero-inflated Poisson

- This model seems to predict the number of zeros pretty well. It would also be interesting to perhaps try having a different probability of zeros (q_0) for different treatments
- It might be that the other covariates explain some of the zero behaviour
- We could further add in both zero-inflation and over-dispersion

An alternative: hurdle models

- ZI models work by having a parameter (here q_0) which is the probability of getting a zero, and so the probability of getting a Poisson value (which could also be a zero) is 1 minus this value
- An alternative (which is slightly more complicated) is a hurdle model where q_0 represents the probability of the *only way* of getting a zero. With probability (1-q_0) we end up with a special Poisson random variable which has to take values 1 or more
- In some ways this is richer than a ZI model since zeros can be deflated or inflated
- Unfortunately this is much fiddlier to fit in JAGS

A hurdle-Poisson model in JAGS

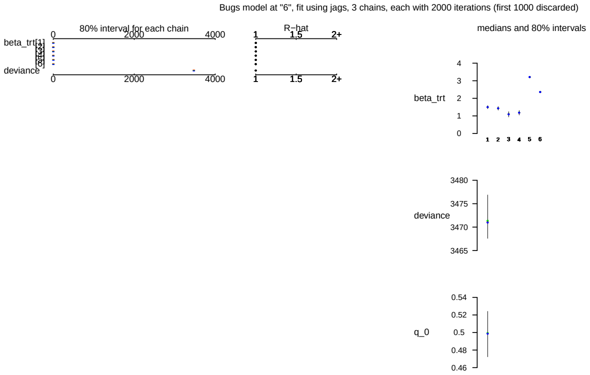
```
model_code = '
model
{
  # Likelihood
  for (i in 1:N) {
    y[i] ~ dpois(lambda[i])T(1,)
    log(lambda[i]) <- beta_trt[trt[i]]
  }
  for(i in 1:N_0) {
    y_0[i] ~ dbin(q_0, 1)
  }
  # Priors
  for (j in 1:N_trt) {
    beta_trt[j] ~ dnorm(0, 100^-2)
  }
  q_0 ~ dunif(0, 1)
}
'
```

Running the model

```
jags_run = jags(data = list(N = nrow(wf[wf$imm > 0,]),
                            N_trt = length(unique(wf$trt)),
                            y = wf$imm[wf$imm > 0],
                            y_0 = as.integer(wf$imm == 0),
                            N_0 = nrow(wf),
                            trt = wf$trt[wf$imm > 0]),
                parameters.to.save = c('beta_trt', 'q_0'),
                model.file = textConnection(model_code))
```

Results

```
plot(jags_run)
```



Some final notes on ZI models

- To complete the Poisson-Hurdle fit we would need to simulate from a truncated Poisson model. This starts to get very fiddly though - see the `jags_examples` repository for worked examples
- We can extend these models further by using a better count distribution such as the negative binomial which has an extra over-dispersion parameter
- We can also add covariates into the zero-inflation component, though it is not always clear whether this is desirable

Summary

- We have seen how partial pooling is a balance between a model of complete independence and complete dependence between groups
- We have fitted some zero inflated and hurdle Poisson models in JAGS