# Lecture 17: BinomialRegression

*This lecture is based on chapter 10 of Statistical Rethinking by Richard McElreath.*

```
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")
```

Let's consider the practical analysis of counts. This is especially poignant as a large portion of biological data are comprised of counts, especially in ecology.

Traditionally, binomially distributed data—count data with an upper bound representing the maximum number of trials—have been analyzed as proportions or percentages.

- ex: percent cover, proportion eaten, percent survival.

Converting data to proportions is not ideal, as it throws away data. How? Because 100/200 and 1/2 both yield the same proportion: 0.5.

- The former has a much larger sample size though, and therefore a lot more information or confidence about the proportion than the latter.
  - When we convert to proportions, that information is tossed away.

Instead, it is much more powerful to retain information by using the counts by using a generalized linear model (GLM).

As we just discussed, GLMs are regression models that use likelihood functions that are non-Gaussian. They do so by assigning a *link* function to the regression model, so that it constrains the model to the bounds required by the non-normal distribution we want to use.

However, we need to be careful in our interpretations because the parameters are no longer on the same scale as the outcome variable.

## Binomial regression of beautiful butterfly babies

As a motivating example we will use data from Dimarco & Fordyce (2013)'s paper in *Ecol. Entomology*. This study focused on a most handsome butterfly, *Battus philenor*, the pipevine swallowtail.

- Note, I am a world authority on this fine animal.

Pipevine swallowtails sequester toxic aristolochic acids from their host plants, which they then endow their offspring with as chemical defenses.

- They lay their eggs in clutches of different sizes, from singly to large clusters. The authors wanted to determine whether there were benefits to clutch size aggregations in deterring egg predators.
- Eggs were laid on plants by females. A subset of clutches were surrounded by Tanglefoot, a sticky barrier, to prevent crawling insects from predating on eggs. After 7 days, egg mortality was assessed.

Below are the data from a `.csv` called `ClutchData.csv`:

```
clutch <- read.csv("ClutchData.csv")
clutch[1:5,]
```

```
  Clutch Mort Treat
1     15    0     0
2     16    0     0
3     12    0     0
4      7    0     0
5     13    0     0
```

```
o <- order(clutch$Clutch)
clutch <- clutch[o,]
```

There are three variables:

1. `Clutch`: The total clutch size
2. `Mort`: the number of eggs in each clutch killed
3. `Treat`: An indicator variable for whether the eggs were accessible to crawling predators

Given the experimental design, there are a few different questions we might ask:

1. Because *Battus* eggs are toxic, do predators affect egg survivorship at all?
2. Does clutch size have an effect on survival (mortality)?
3. Is there an effect of clutch size after having taken into account predator presence?
4. Is there a nonlinear effect of predation and clutch size?

Models 1 & 2 are simple regressions with an intercept and one predictor variable. Model 3 is a multiple regression with both predictors, while model 4 includes an interaction term. Model 5 is an intercept-only model.

All of these models will have the same basic structure:

$$\begin{aligned} y_i &\sim \text{Binomial}(N, p_i) \\ logit(p_i) &= X_i\beta \\ \beta &\sim \text{Normal}(0, 5). \end{aligned}$$

The Stan model is surprisingly easy to fit. The model will be essentially identical to all of the multiple regression models we have built before. Only now, we can use the built-in `binomial_logit` likelihood function.

- This automatically applies the logit link to the $p_i$ variable, and is more numerically stable than coding the logit function yourself.

For ease, we will code one model and manipulate the design matrix $X$ to include or exclude the parameters of interest. The full model is below:

```
data {
  int<lower=0> nObs;
  int<lower=0> nVar;      // no. vars
  int<lower=0> obs[nObs]; // no. successes (or failures)
  int<lower=0> N[nObs];   // Total no. trials
  matrix[nObs, nVar] X;   // design matrix
  real<lower=0> bMu;      // mean of prior betas
  real<lower=0> bSD;      // SD of prior beta
}

parameters {
  vector[nVar] beta;
}

transformed parameters {
  vector[nObs] p;
  p = X * beta;
}

model {
  beta ~ normal(bMu, bSD);
  obs ~ binomial_logit(N, p);
}

generated quantities {
  vector[nObs] log_lik;

  for(n in 1:nObs)
    log_lik[n] = binomial_logit_lpmf(obs[n] | N[n], p[n]);
}
```

Below we will set up the data and run the four models.

```
obs <- clutch$Mort
N <- clutch$Clutch
nObs <- length(obs)
treat <- clutch$Treat
predMat <- as.matrix(model.matrix(~Clutch*Treat, data=clutch))
bMu <- 0; bSD<-5
```

1. Linear regression with clutch size as a predictor.

```
X <- predMat[,1:2]
d1 <- list(nObs=nObs, nVar=ncol(X), obs=obs, N=N, X=X,
```

```
  bMu=bMu, bSD=bSD)
m1 <- stan(file="logitMod.stan", data=d1, iter=2500, chains=4,
  seed=867.5309)
```

2. Linear regression with predator access as a predictor.

```
X <- predMat[,c(1,3)]
d2 <- list(nObs=nObs, nVar=ncol(X), obs=obs, N=N, X=X,
  bMu=bMu, bSD=bSD)
m2 <- stan(file="logitMod.stan", data=d2, iter=2500, chains=4,
  seed=867.5309)
```

3. Linear regression with clutch size and predator access as predictors.

```
X <- predMat[,c(1:3)]
d3 <- list(nObs=nObs, nVar=ncol(X), obs=obs, N=N, X=X,
  bMu=bMu, bSD=bSD)
m3 <- stan(file="logitMod.stan", data=d3, iter=2500, chains=4,
  seed=867.5309)
```

4. Linear regression with an interaction between clutch size and predator access.

```
X <- predMat
d4 <- list(nObs=nObs, nVar=ncol(X), obs=obs, N=N, X=X,
  bMu=bMu, bSD=bSD)
m4 <- stan(file="logitMod.stan", data=d4, iter=2500, chains=4,
  seed=867.5309)
```

The priors we specified are weakly regularizing. Therefore we should use model selection to guard against overfitting.

```
(comp <- compare(m1,m2,m3,m4))
```

```
    WAIC pWAIC dWAIC weight    SE   dSE
m4 461.0  10.0   0.0   0.73 49.77    NA
m3 463.0   8.6   2.0   0.27 47.55  5.95
m2 471.8   5.6  10.7   0.00 44.69 13.99
m1 667.9   8.7 206.9   0.00 59.70 43.83
```

Model selection shows that `m4`, the model with the interaction term, is the best model, but not by much.

The multiple regression model without the interaction term is close behind with 36% of the weight.

Therefore, we probably should not discount `m3`. Instead, let's use model averaging to combine the two and then look at the results.

First we need to extract the weights and the posterior $\beta's$.

- We will add a column of zeros to the simpler model to account for the fact it does not have an interaction parameter.

```
wts <- round(comp@output$weight[1:2],2)
betaM3 <- as.matrix(m3, pars="beta")
betaM3 <- cbind(betaM3, rep(0, nrow(betaM3)))
betaM4 <- as.matrix(m4, pars="beta")
```

Now, we will multiply each posterior $\beta$ by the weight of it's model, and then add the weighted model together, producing a consensus model.

```
avgBeta <- betaM4*wts[1] + betaM3*wts[2]
Mean <- colMeans(avgBeta)
StDev <- apply(avgBeta, 2, sd)
hdi <- apply(avgBeta, 2, HDI, credMass = 0.95)
x <- as.matrix(round(cbind(Mean, StDev, t(hdi)),3))
colnames(x) <- c("Mean", "SD", "0.025","0.975")
x
```

```
          Mean    SD  0.025  0.975
beta[1] -2.953 0.406 -3.749 -2.173
beta[2]  0.002 0.024 -0.044  0.048
beta[3]  3.188 0.431  2.345  4.008
beta[4] -0.051 0.025 -0.099 -0.003
```

Initially, it doesn't look like there is much of an effect of clutch size. However, we have an interaction term, so we need to consider the joint effects of $\beta_2$ and $\beta_4$. We also need to consider the results on the scale of probabilities, not on the log-odds scale.

```
noPred <- avgBeta[,2] + avgBeta[,4]*0
pred <- avgBeta[,2] + avgBeta[,4]*1
quantile(logistic(noPred), probs=c(0.025,0.5,0.975))
```

```
     2.5%       50%     97.5%
0.4890529 0.5004383 0.5118952
```

```
quantile(logistic(pred), probs=c(0.025,0.5,0.975))
```

```
     2.5%       50%     97.5%
0.4826774 0.4876290 0.4923167
```

Initially, it doesn't look like there is much of a difference in the effect of clutch size when we include vs. exclude predators.

However, we need to consider the intercepts as well. This is because a 50% change will have a much greater impact if the intercept is far away from zero or one.

```
# No predators clutch of 1
(np1 <- round(mean(logistic((avgBeta[,1]))),3))
```

```
[1] 0.053
```

```
# No predators clutch of 2
(np2 <- (round(mean(logistic((avgBeta[,1]+noPred))),3)))
```

```
[1] 0.053
```

```
# with predators clutch of 1
(p1 <- round(mean(logistic(avgBeta[,1]+avgBeta[,3])),3))
```

```
[1] 0.558
```

```
# with predators clutch of 2
(p2 <- round(mean(logistic(avgBeta[,1]+avgBeta[,3] + pred)),4))
```

```
[1] 0.546
```

```
np2-np1 # no predators
```

```
[1] 0
```

```
p2-p1   # predators
```

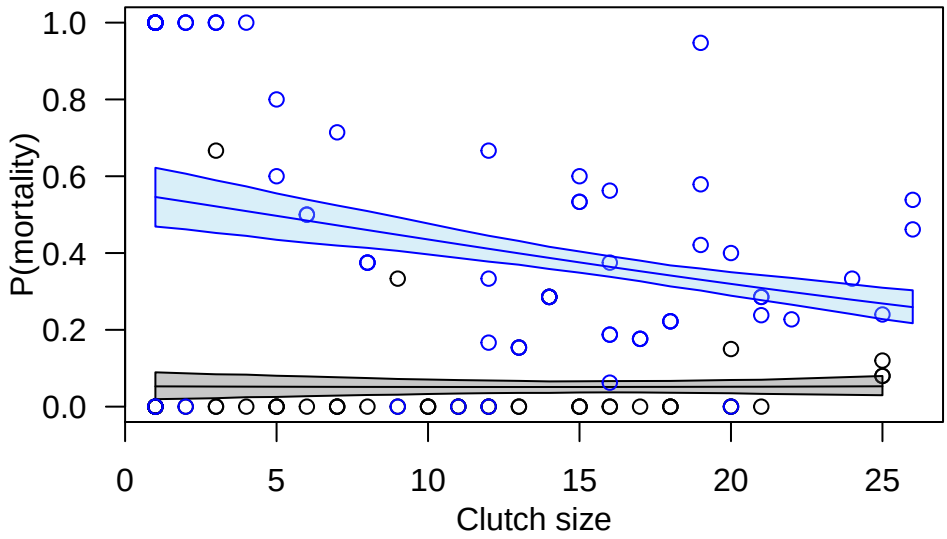
```
[1] -0.012
```

Thus even though the slope of clutch size between predator exclosure and predator access treatments differ very little, The decrease in mortality as a result of clutch size is an order of magnitude higher when predators are present.

- Because of this non-additivity in parameter estimates, it is usually better to plot the results rather than trying to interpret tables of numbers.

```
# extract the p's, the expected log-odds of the probabilities
#of each observation, and model average them.
propObs <- obs/N
logitP3 <- as.matrix(m3, "p")
logitP4 <- as.matrix(m4, "p")
avgLogitP <- logitP4*wts[1]+logitP3*wts[2]

pz <- logistic(avgLogitP)
mnP <- colMeans(pz)

hdiP <- apply(pz, 2, HDI, credMass=0.95)

plot((obs/N) ~ N, ylim=c(0,1), las=1, type="n", ann=FALSE)
mtext("P(mortality)", side=2, line=2.2)
mtext("Clutch size", side=1, line=2)
# no predators
x <- N[treat==0]
y <- propObs[treat==0]
mnNP <- mnP[treat==0]
hdiNP <- hdiP[,treat==0]

points(x, y)
polygon(c(x, rev(x)), c(hdiNP[1,], rev(hdiNP[2,])), col="#50505050")
lines(x, mnNP, col="black")
```

```
# predators
x <- N[treat==1]
y <- propObs[treat==1]
mnPr <- mnP[treat==1]
hdiPr <- hdiP[,treat==1]

points(x, y, col="blue")
polygon(c(x, rev(x)), c(hdiPr[1,], rev(hdiPr[2,])), col="#88CCEE50", border="blue")
lines(x, mnPr, col="blue")
```




In the figure, we can see that the impact of clutch size on mortality is much greater for the treatments with predator access (blue) than for the predator exclosure treatments (grey).

But, as we saw earlier, the slopes between predator and nonpredator treatments were almost identical.

- The difference we see has everything to do with the nonadditive nature of GLMs and the effect of the link function.
- Also note that the uncertainty shrinks as we move from left to right. This is because larger clutch sizes have more information than smaller clutch sizes.
- This means we need to be very careful when interpreting parameter estimates from GLMs. It also highlights the need for plotting if we really want to understand what is occurring.

After visually observing the results of the model and calculating the slopes of the different treatments by hand, the model selection results make more sense.

- There is an interaction happening between clutch size and predation treatment.
- However, much of that interaction is accounted for by the nonadditive nature of the model and the difference in intercepts.
  - The addition of an explicit interaction term has much less of an effect here.
- This is why the simpler model without the interaction term got over $\frac{1}{3}$ of the model weight, and why the WAIC values were so similar.