Lesson 14

Clark

Recall that a Poisson distribution is a limiting case of the Binomial. That is, we would want to use a Poisson when n (or the number of trials, and we use this term really loosely here) is really large and the probability of success is really small. Typically, in practice a Poisson is used for count data and is the default choice due to its simplicity in formulation and interpretation.

However, we did notice an issue with the Poisson, if you recall, last class we attempted to fit data from the Galapagos Islands using a Poisson GLM and we had some issues

```
library(faraway)
library(tidyverse)
data(gala)
gala_df <- gala %>%
  select(-Endemics)


gala_glm <- glm(Species ~ Scruz +
                Elevation, data=gala_df,
                family=poisson)

1-pchisq(deviance(gala_glm),df.residual(gala_glm))
```

```
## [1] 0
```

```
deviance(gala_glm) / df.residual(gala_glm)
```

```
## [1] 61.28803
```

One of the reasons that we may have a poor fit is due to *over dispersion*, that is, if we recall we had one serious limitation of a Poisson:

Now, to motivate this next type of model, I first want to introduce the idea of a hierarchical, or mixture, model. Let's consider the following situation, let's say I am interested in determining whether crime or weather impacts the number of people that show up for voting and I collect data across a wide range of counties in American for a wide range of days. Perhaps my model becomes:

However, there are undoubtedly some unique characteristics of a county that cause some counties to have higher or lower voting turnout that aren't accounted for in the covariates. We can think of this as our mechanism of interest ($\log(\lambda)$) manifests itself differently for different situations. We write:

Now, the field of mixture models is a subject in of itself, so today we're only going to talk about one specific mixture model and in later classes we will have a more general discussion about the role mixture models play in GLMs.

For now, we will assume that $g(\lambda|\mu, \kappa)$ follows a Gamma distribution. That is we can write a joint distribution for $h(Y, \lambda)$ as:

However, we don't every observe λ so we need to marginalize over λ to get a distribution just for Y. That is, we must calculate:

Once we do this, we now have $f(Y|\mu, \kappa)$, which it turns out is the density of a Negative Binomial distribution.

So, the key here is, when we are using a negative binomial, we are assuming that our mechanism of interest

(the expected count) varies according to a Gamma distribution. The second order effect is we can calculate the expected value and the variance of this new distribution:

And, as it turns out, for a fixed value of κ we can write the negative binomial in exponential dispersion family form with a canonical parameter of $\frac{\mu}{\mu+\kappa}$, however most of the time negative binomial GLMs use the log link. Now, since a NB GLM isn't technically a member of the exponential dispersion family, we have to use a different function in R to fit the models, we can use `library(MASS)` which should come standard in R (meaning you don't need to install it.)

We can do:

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
gala_glm2<-glm.nb(Species~ Scruz +
                Elevation,data=gala_df)

summary(gala_glm2)
```

```
##
## Call:
## glm.nb(formula = Species ~ Scruz + Elevation, data = gala_df,
##     init.theta = 1.249551502, link = log)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.1715906  0.2649365  11.971  < 2e-16 ***
## Scruz       -0.0020360  0.0024987  -0.815    0.415
## Elevation    0.0025404  0.0003965   6.407 1.48e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.2496) family taken to be 1)
##
##     Null deviance: 66.957  on 29  degrees of freedom
## Residual deviance: 33.413  on 27  degrees of freedom
## AIC: 307.34
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.250
##           Std. Err.:  0.309
##
##  2 x log-likelihood:  -299.339
```

```
logLik(gala_glm2)
```

```
## 'log Lik.' -149.6694 (df=4)
```

```
logLik(gala_glm) #We have to be careful here
```

```
## 'log Lik.' -907.8039 (df=3)
```

Now, unfortunately because we're estimating κ we cannot use our deviance to conduct a goodness of fit. We could use AIC, but to me it's really obvious the negative binomial fits better:

What's the implications of using the wrong model?

```
#Profile CIs

confint(gala_glm)
```

```
## Waiting for profiling to be done...
```

```
##                    2.5 %       97.5 %
## (Intercept)  3.856801254  3.998974350
## Scruz       -0.006495040 -0.004651991
## Elevation    0.001377193  0.001501654
```

```
confint(gala_glm2)
```

```
## Waiting for profiling to be done...
```

```
##                    2.5 %      97.5 %
## (Intercept)  2.543275482 3.844970257
## Scruz       -0.007317684 0.004357105
## Elevation    0.001534437 0.003667347
```

Another type of mixture model that gets used a bit is what's called a zero-inflated Poisson or ZIP model. These get used when the number of zeros in our data set is greater than what is to be expected under a Poisson. To view my motivation for teaching you, I'm going to quote from a 600 level Stats course I took at Iowa State from Prof Mark Kaiser:

What are known as zero-inflated models have applications in situations for which one might naturally consider using a binomial or a Poisson model, but in which the observed frequency of zeros in data are in excess of what would reasonably be expected in a model with positive mean and the specified distribution. Some statisticians are perfectly happy to simply specify a data model with an inflated probability of zero in such cases, just as they are to use a gamma-Poisson mixture and call it a negative binomial distribution. I greatly prefer if a *reasonable conceptualization is available that can represent a mechanism that could lead to increased frequency of zero values*, just as I prefer a gamma-Poisson to be thought of as a hierarchical model in which the relative frequencies with which a mechanism manifests itself are reflected in the gamma mixing distribution.

To me, this means, think about the situation you are modeling and if you can conceive of two different processes (one that makes zeros and one that makes a Poisson) use a ZIP model.

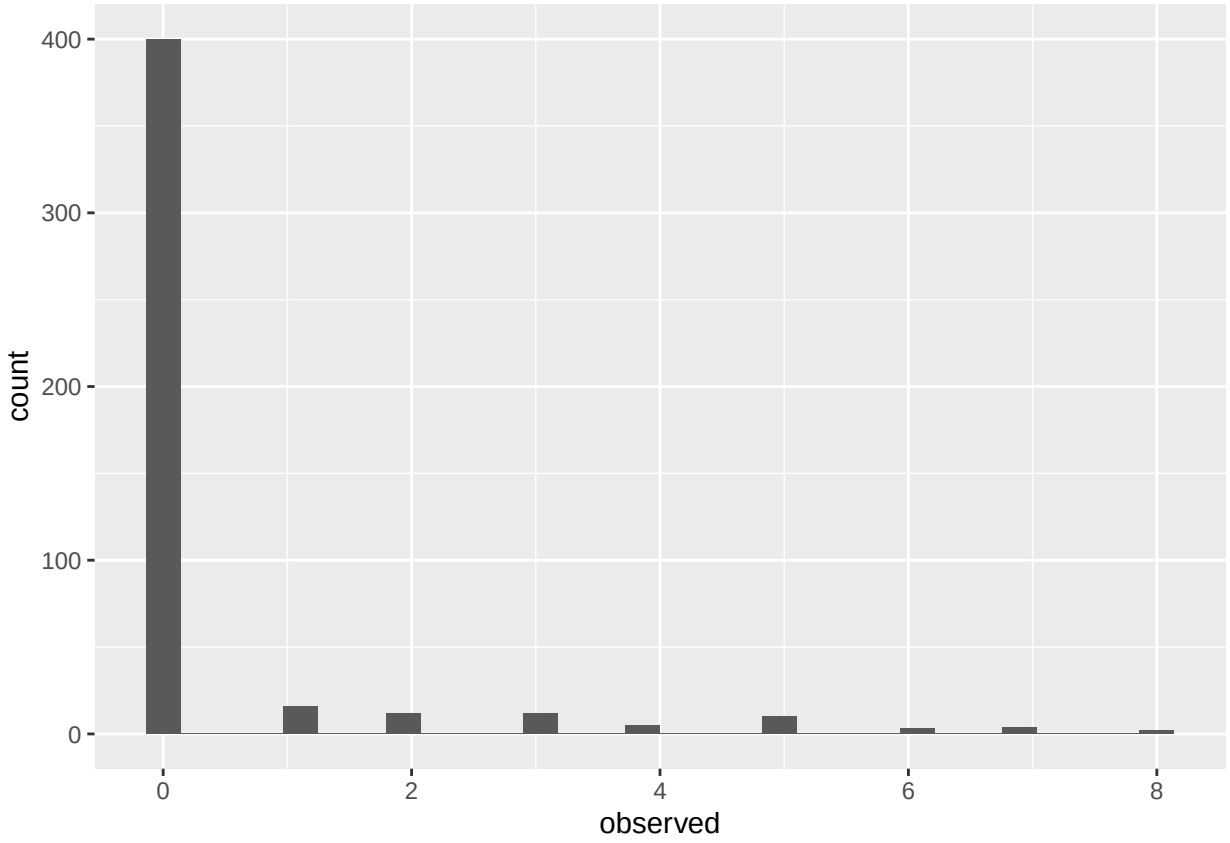
A ZIP model is structured as a two component mixture

We can then parameterize the mixing probability ϕ_i and/or $\log(\lambda_i)$. Let's give an example motivated from a study done by Royle and Dorazio (2008). Here they wanted to determine the number of weasels that lived in a variety of of sampling locations. They visited 464 locations and collected data that looked like:

```
weasels <- data.frame(observed=c(rep(0,400),rep(1,16),rep(2,12),rep(3,12),rep(4,5),
            rep(5,10),rep(6,3),rep(7,4),rep(8,2)))

weasels %>% ggplot(aes(x=observed))+
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We could fit this with a Poisson

```
pois<-glm(observed~1,data=weasels,family=poisson)
1-pchisq(902,df.residual(pois))
```

```
## [1] 0
```

Clearly doesn't fit well and, under this model, the probability of observing a zero is:

```
dpois(0,exp(-.8023))
```

```
## [1] 0.6387152
```

Meaning, we would expect to have observed $.64(464) = 300$ zeros instead of the 400 we observed. So, let's think about the problem. If we go into an area and we observe zero Weasels what might have happened?

So, maybe there's two different mechanisms we have to consider.

```
library(pscl)
zip_glm <- zeroinfl(observed~1,data=weasels,family="poisson")
summary(zip_glm)
```

```
##
## Call:
## zeroinfl(formula = observed ~ 1, data = weasels, family = "poisson")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -0.3502 -0.3502 -0.3502 -0.3502  5.8989
##
## Count model coefficients (poisson with log link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.13276    0.07502    15.1   <2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.7791     0.1362   13.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 9
## Log-likelihood: -314.9 on 2 Df
```

Under this model, the probability of being degeneratively zero is:

```
exp(1.78)/(1+exp(1.78))
```

```
## [1] 0.8556969
```

And the overall probability of observing a zero is:

Which we can calculate:

```
Pr0<-0.86+(1-0.86)*dpois(0,exp(1.13))
```

Therefore, our expected number of each count is:

```
counts<-seq(1,8)
obs<-464
zip_est<-c(obs*Pr0,obs*dpois(counts,exp(1.13))*(1-0.86))
pois_est<-obs*dpois(seq(0,8),exp(-.8023))
actual<-c(400,16,12,12,5,10,3,4,2)
rbind(pois_est,zip_est,actual)
```

```
##              [,1]       [,2]     [,3]      [,4]       [,5]        [,6]
## pois_est 296.3638 132.858926 29.78011  4.450108  0.4987422  0.04471689
## zip_est  401.9791   9.098551 14.08299 14.532038 11.2465494  6.96309078
## actual   400.0000  16.000000 12.00000 12.000000  5.0000000 10.00000000
##                 [,7]         [,8]         [,9]
## pois_est 0.003341073 0.0002139703 1.199027e-05
## zip_est  3.592556204 1.5887599951 6.147819e-01
## actual   3.000000000 4.0000000000 2.000000e+00
```

Not a perfect fit:

```
ch_sq <- sum((zip_est-actual)^2/zip_est)
1-pchisq(ch_sq,8)
```

```
## [1] 0.02387231
```

But close.

So, again, think of a NB GLM as a gamma-Poisson mixture and a ZIP model as a mixture of a bernoulli and a Poisson. While we didn't do so here, we could parameterize either component of a ZIP model. Take a look at the Details of the zeroinfl function. There also are zero inflated negative binomial models, I kind of shrug at those as I have a hard time thinking about what those models mean and I much prefer using a model that I can explain vs a model that might fit the data a little better but I really don't know what the model means.