

Week 9: Resampling and empirical estimation SOLUTIONS

(a) To simulate, just generate a list of binomial 0/1 events, using ϕ as the probability of each. This is as simple as:

```
p <- rbinom( 9999 , prob=0.4 , size=1 )
```

That generates 9999 0/1 events, each with probability 0.4 (an arbitrary value of ϕ). Now feed these 0/1 events into another line that generates random Poisson counts. The trick, however, is we only want to generate a Poisson count when $p=1$. Otherwise, we didn't see the birds and we record a zero. This code will work:

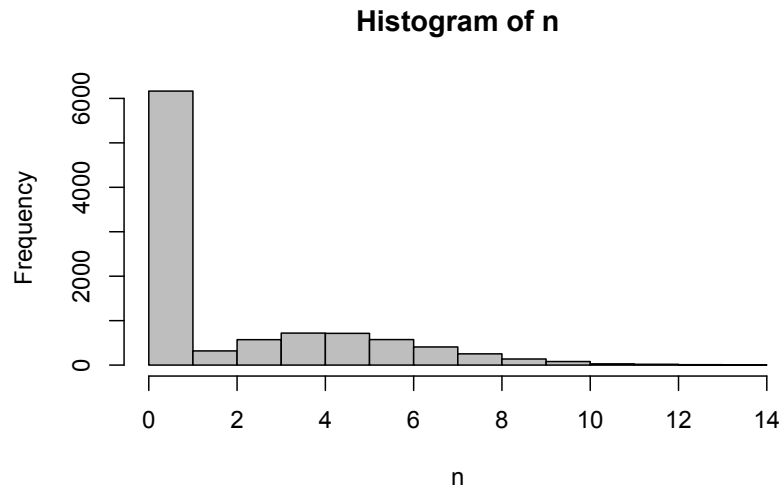
```
n <- ifelse( p==0 , 0 , rpois( 9999 , lambda=5 ) )
```

Or you could have used something like:

```
n <- p * rpois( 9999 , lambda=5 )
```

Either way, you end up with a list of zero-inflated counts. Plotting them for the parameters above, $\phi = 0.4$ and $\lambda = 5$:

```
hist( n , breaks=max(n)+1 , col="gray" )
```



(b) We now use the simulation code inside a density function. This function will return likelihoods of observed counts. Here's how I did it:

```
# empirical likelihood function
dsimzeropois <- function( x , phi , lambda , log=FALSE , R=999 ) {
  p <- rbinom( R , prob=phi , size=1 )
  n <- ifelse( p==0 , 0 , rpois( R , lambda=lambda ) )
  p <- sapply( x , function(z) length(n[n==z])/R )
  p <- ifelse( p==0 , 1e-100 , p )
}
```

```

    if (log==TRUE) p <- log(p)
  p
}

```

The first line inside the function generates the list of 0/1 events for finding any birds. The second line then generates counts from the zero-inflated Poisson, as before. The third line then computes the proportion of the simulated samples that match each datum. The line `p <- ifelse(p==0 , 1e-100 , p)` just ensures there are no exact zeros in the list of proportions. You can get away with not having this line, but it is helpful. Finally, the function logs the likelihoods and returns them.

Now to estimate the parameters from the data. Here's the code I used:

```

library(bbmle)
logit <- function(x) 1/(1+exp(x))
m1 <- mle2( d$n ~ dsimzeropois( phi=logit(phi) , lambda=exp(a) ) ,
  start=list(a=log(7),phi=0) , skip.hessian=TRUE , method="SANN" )

> summary(m1)
Maximum likelihood estimation

Call:
mle2(minuslogl = d$n ~ dsimzeropois(phi = logit(phi), lambda = exp(a)),
  start = list(a = log(7), phi = 0), method = "SANN", skip.hessian = TRUE)

Coefficients:
      Estimate Std. Error z value Pr(z)
a      2.19360      NA      NA      NA
phi    0.62085      NA      NA      NA

-2 log L: 1846.287

```

Note that we don't get any estimated standard errors! This is because the Hessian wasn't computed, so we don't have any estimate of the curvature of the likelihood surface. To get standard errors, you could run the fitting again, this time with `skip.hessian=FALSE`, or you could bootstrap.

We need to convert both estimates to the un-transformed scale:

```

> exp(coef(m1)[1])
      a
8.967404
> logit(coef(m1)[2])
      phi
0.3495888

```

So the MLE values of ϕ and λ are about 0.35 and 8.97, respectively. Note that the overall mean of `d$n` is about 2.4, so by accounting for zero-inflation, we have upped the estimate of density quite a lot.

(c) Fitting an old-fashioned Poisson model:

```
m0 <- mle2( d$n ~ dpois( lambda=exp(a) ) , start=list(a=log(mean(d$n))) )
```

The estimate of λ is now

```
> exp(coef(m0))
      a
2.433083
```

which is just the mean of $d\$n$. The zero-inflated model estimates a greater density, because it doesn't count all of the zeros as "real" zeros.

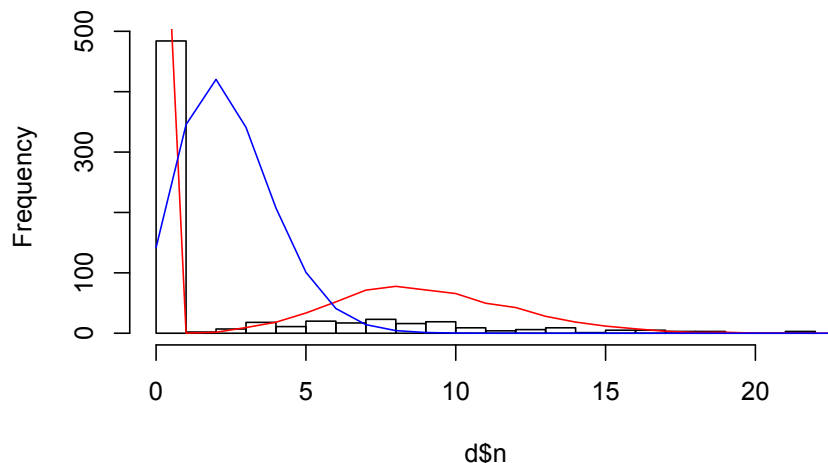
Comparing the fit of the two models:

```
> AICtab( m0 , m1 , base=TRUE , weights=TRUE )
      AIC    df dAIC  weight
m1 1850.3    2   0.0     1
m0 5274.6    1 3424.3 <0.001
```

Pretty convincing win for the zero-inflated model. Here are the raw data, with predictions of each model overlain. Red curve is the zero-inflated model. Blue curve is the basic Poisson model.

```
hist( d$n , breaks=30 )
k <- coef(m1)
ls <- sapply( 0:30 , function(z) dsimzeropois(z,phi=logit(k[2]),lambda=exp(k[1]),R=9999) )
lines( 0:30 , ls*sum(d$n) , lty=1 , col="red" )
k <- coef(m0)
ls <- sapply( 0:30 , function(z) dpois(z,lambda=exp(k[1])) )
lines( 0:30 , ls*sum(d$n) , lty=1 , col="blue" )
```

Histogram of $d\$n$



Notice that the zero-inflated model is still not fitting very well—it isn't sufficiently over-dispersed! For fun, you can fit and plot the zero-inflated negative-binomial, also. Here's the density function:

```

# zero-inflated neg-binom
dsimzeronbinom <- function( x , phi , mu , size , log=FALSE , R=999 ) {
  p <- rbinom( R , prob=phi , size=1 )
  n <- ifelse( p==0 , 0 , rnbinom( R , mu=mu , size=size ) )
  p <- sapply( x , function(z) length(n[n==z])/R )
  p <- ifelse( p==0 , 1e-100 , p )
  if (log==TRUE) p <- log(p)
  p
}

> summary(mznb)
Maximum likelihood estimation

Call:
mle2(minuslogl = d$n ~ dsimzeronbinom(phi = logit(phi), mu = exp(a),
  size = exp(s)), start = list(a = log(7), phi = 0, s = 0),
  method = "SANN", skip.hessian = TRUE)

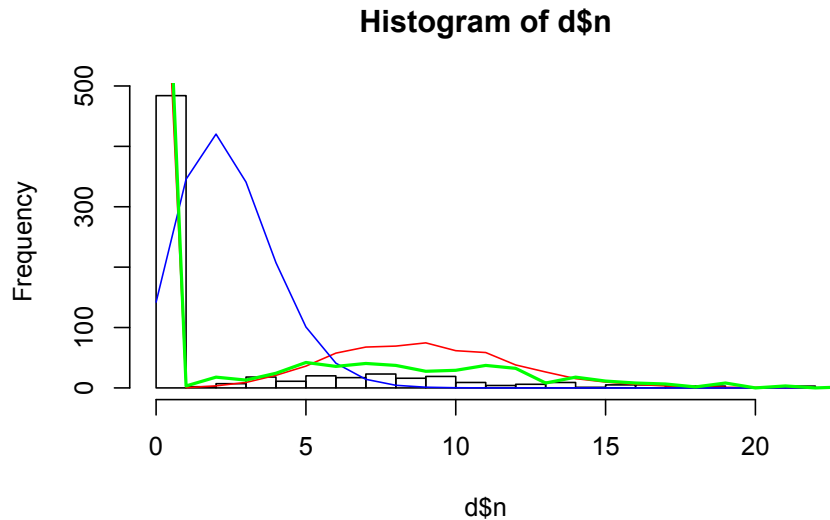
Coefficients:
      Estimate Std. Error z value Pr(z)
a          2.1955         NA      NA   NA
phi        1.0050         NA      NA   NA
s          2.0260         NA      NA   NA

-2 log L: 1787.757
> AICtab( m0 , m1 , mznb , base=TRUE , weights=TRUE )
      AIC   df dAIC  weight
mznb 1793.8  3    0.0    1
m1    1850.3  2   56.5 <0.001
m0    5274.6  1  3480.8 <0.001

We can add the predictions to our plot:

k <- coef(mznb)
ls <- sapply( 0:30 , function(z) dsimzeronbinom(z,mu=exp(k[1]),size=exp(k[3]),
  phi=logit(k[2])) )
lines( 0:30 , ls*sum(d$n) , lty=1 , col="green" , lwd=2 )

```



That looks much better. There is substantial over-dispersion in the counts, probably due to heterogeneity among the sites. Notice also that the MLE of ϕ has declined from about 0.35 to:

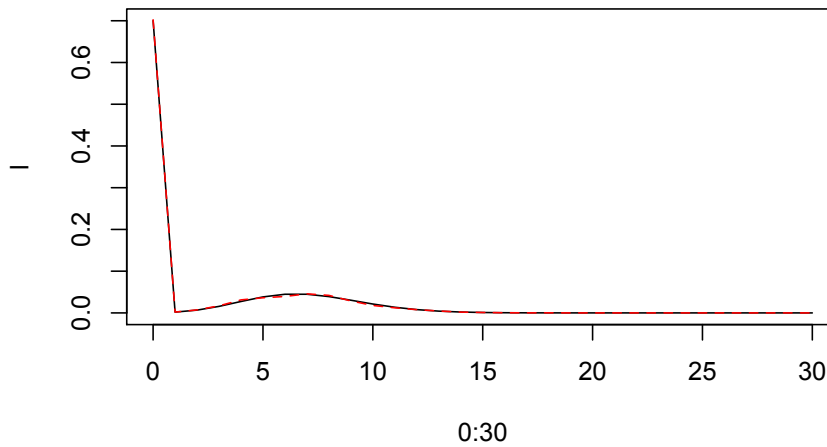
```
> logit(coef(mznb)[2])
      phi
0.2679589
```

(d) This is easier than it might seem. There are two kinds of events. First, if we find a group of birds, then the probability of an observed count x is just taken from the Poisson density. Second, if we don't find a group of birds, the probability of x is 0 when $x > 0$ or 1 when $x == 0$. The probability ϕ selects between these two kinds of events. So here's code that will do the job for us:

```
# analytical likelihood function for zero-inflated Poisson
dzeropois <- function( x , phi , lambda , log=FALSE ) {
  p <- (1-phi)*ifelse(x==0,1,0) + phi*dpois( x , lambda=lambda , log=FALSE )
  if ( log==TRUE ) p <- log(p)
  p
}
```

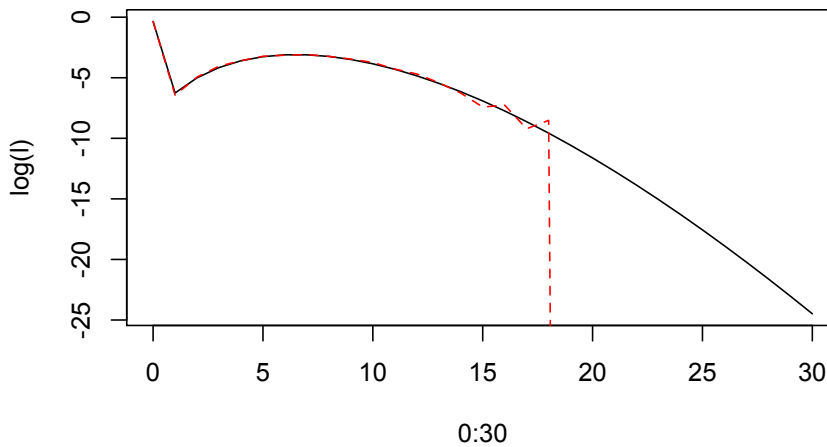
It's that easy. As long as ϕ isn't exactly 0 or 1, we won't get probabilities of zero, so this is still a kosher density that can be used in `mle2`. Verifying that this function returns nearly the same likelihoods as the simulation function:

```
kphi <- 0.3
kl <- 7
l <- sapply( 0:30 , function(z) dzeropois(z,phi=kphi,lambda=kl) )
plot( 0:30 , l , type="l" )
ls <- sapply( 0:30 , function(z) dsimzeropois(z,phi=kphi,lambda=kl,R=9999) )
lines( 0:30 , ls , lty=2 , col="red" )
```



The red dashed curve (the simulation function) and the black curve (the analytical function) are almost exactly the same. This is easier to appreciate in log-likelihood space:

```
kphi <- 0.3
kl <- 7
l <- sapply( 0:30 , function(z) dzeropois(z,phi=kphi,lambda=kl) )
plot( 0:30 , log(l) , type="l" )
ls <- sapply( 0:30 , function(z) dsimzeropois(z,phi=kphi,lambda=kl,R=9999) )
lines( 0:30 , log(ls) , lty=2 , col="red" )
```



Notice that the red curve drops off the bottom at some point. Those are high counts that were never observed in the simulations, so they got assigned arbitrary very low probabilities. The analytical

function, in contrast, has a smooth change in probability across all counts. An important lesson here is that empirical likelihood estimation works best for likely events. For very rare events, under a given model, it is quite prone to stochastic error.