<http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm>

**Lecture 6—Wednesday, September 15, 2010**

**Topics**

* [Obtaining the variances of treatment means from ANOVA models](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#obtaining)
  + [Expressing treatment means as dot products](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#treatmeans)
  + [The variance of a sum of random variables](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#treatvars)
* [Negative binomial distribution](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#NB)
  + [Basic characteristics](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#basic)
  + [Derivation of the negative binomial mass function](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#derivation)
  + [Mean and variance](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#mean)
* [Ecological parameterization of the negative binomial distribution](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#ecological)
  + [Gamma function](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#gammafunc)
  + [The variance of the negative binomial distribution in terms of μ and θ](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#variance)
* [Mechanisms that give rise to a negative binomial distribution](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#mechanisms)
  + [Negative binomial as a model of a nonhomogeneous Poisson process](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#nonhomo)
  + [Zero inflation](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#zero)
  + [Polya-Eggenberger urn model](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#polya)
  + [The generalized Poisson model](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#generalizedpoisson)
* [The use of the negative binomial distribution in practice](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#rationalizing)
* [Illustrations of negative binomial distributions](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#examples)
* [Methods of estimation](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#methods)
* [Maximum likelihood estimation](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#MLE) 
  + [The random sample](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#random)
  + [The likelihood](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#likelihood)
  + [The log-likelihood](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#loglikelihood)
  + [The maximum likelihood principle](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#principle)
  + [Maximizing the likelihood: the Newton-Raphson method](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#Newton)

**Obtaining the variances of treatment means from ANOVA models**

In [lecture 5](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture5.htm#obtain) we created mean profile plots to display estimated treatment means and their 95% confidence intervals from ANOVA models. To obtain the estimates we used the **effect** function of the **effects** package. These calculations can also be easily done by hand. To illustrate this I re-fit the best model from lecture 5. The R code for doing this is contained in an external file that I access with the **source** function.

source( 'http://www.unc.edu/courses/2010fall/ecol/563/001/notes/lecture6%20code%20prelim.txt', echo=T)

**Expressing treatment means as dot products**

The model we chose to display in lecture 5 was out3 whose coefficient estimates are shown below.

coef(out3)

(Intercept) fac1No fac1Ru fac2S fac32 fac1No:fac2S   
3.38323831 0.54729873 0.53698650 0.01714866 0.10757945 0.01341375   
fac1Ru:fac2S fac2S:fac32   
0.16350360 0.16322994

The model that was fit is the following. (Terms are displayed in the same order as they are shown in the **coef** output above.)

out3

Here *Z*1, *Z*2, *W*, and *V* are dummy variables that are defined as follows.

|  |  |
| --- | --- |
| **Z1,** | **Z2,** |
| W, | V |

Table 1 displays the corresponding expressions for each of the twelve treatment means.

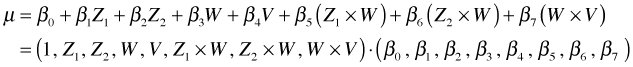
|  |
| --- |
| **Table 1** Treatment means predicted by ANOVA model out3 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **fac1** | **fac2** | **fac3** | **Z1** | **Z2** | **W** | **V** | **Treatment mean** |
| Co | D | 1 | 0 | 0 | 0 | 0 | β0 |
| No | D | 1 | 1 | 0 | 0 | 0 | β0 + β1 |
| Ru | D | 1 | 0 | 1 | 0 | 0 | β0 + β2 |
| Co | S | 1 | 0 | 0 | 1 | 0 | β0 + β3 |
| No | S | 1 | 1 | 0 | 1 | 0 | β0 + β1 + β3 + β5 |
| Ru | S | 1 | 0 | 1 | 1 | 0 | β0 + β2 + β3 + β6 |
| Co | D | 2 | 0 | 0 | 0 | 1 | β0 + β4 |
| No | D | 2 | 1 | 0 | 0 | 1 | β0 + β1 + β4 |
| Ru | D | 2 | 0 | 1 | 0 | 1 | β0 + β2 + β4 |
| Co | S | 2 | 0 | 0 | 1 | 1 | β0 + β3 + β4 + β7 |
| No | S | 2 | 1 | 0 | 1 | 1 | β0 + β1 + β3 + β4 + β5 + β7 |
| Ru | S | 2 | 0 | 1 | 1 | 1 | β0 + β2 + β3 + β4 + β6 + β7 |

Recall from matrix algebra (or physics) that the dot product of two vectors is obtained by multiplying the corresponding entries together and then adding.

dot product example

Our regression model can also be written as a vector dot product.



The first vector in the dot product expression for the mean is just the coefficient vector of the model, obtained from coef(out3). The second vector can be generated with a user-defined function in R that takes as its arguments the values of fac1, fac2, and fac3.

myvec <- function(fac1,fac2,fac3) c(1,fac1=='No',fac1=='Ru', fac2=='S', fac3==2, (fac1=='No')\*(fac2=='S'), (fac1=='Ru')\*(fac2=='S'), (fac2=='S')\*(fac3==2))

To obtain the treatment mean for a specific combination of factors I proceed as follows. Here **%\*%** is the matrix multiplication operator in R.

**#first vector of regressor values**

myvec('Co','D',1)

[1] 1 0 0 0 0 0 0 0

**#mean**

myvec('Co','D',1)%\*%coef(out3)

[,1]  
[1,] 3.383238

**#second vector of regressor values**

myvec('Ru','S',1)

[1] 1 0 1 1 0 0 1 0

**#mean**

myvec('Ru','S',1)%\*%coef(out3)

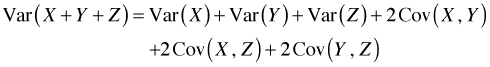
[,1]  
[1,] 4.100877

**The variance of a sum of random variables**

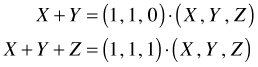
To obtain the standard errors of these treatment means we need to use some basic results from elementary probability theory. If *X* and *Y* are two random variables then the variance of their sum is the sum of their individual variances plus two times their covariance.

http://www.unc.edu/courses/2010fall/ecol/563/001/images/lectures/lecture6/sumvariance.gif

(For independent random variables the covariance is zero and the variance of their sum is the sum of their variances.) With three random variables the variance of a sum formula is more complicated.



Fortunately there is a simple formula underlying these different results. Each sum can be written as a dot product as was explained above. If we are working with three random variables then each of the above sums can be written as a dot product as follows.



In general then any of these sums can be written as a vector dot product of the form dot productwhere **c** is a vector of zeros and ones and **x** is a vector of random variables.

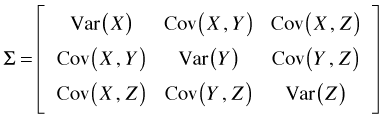
It's a basic result of linear algebra that any vector dot product can be written as matrix multiplication where the first vector is treated as a row matrix (a matrix with a single row) and the second vector is treated as a column matrix (a matrix with a single column). Because the convention in matrix algebra is to treat vectors as column matrices, row vectors are written as vector transposes. Hence we have the equivalence

general dot product

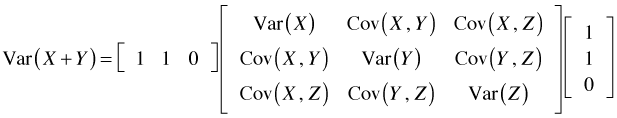
This leads to the basic result. The variance of a sum of random variables, when the sum is written in the form dot product, is the following.

variance  of dot product

The expression quadratic formis an example of a mathematical object called a **quadratic form** and Σ is the variance-covariance matrix of the elements of **x**. For our three variable example



In this case Var(*X* + *Y*) would be computed as follows.



R doesn't treat vectors as column matrices so in creating a quadratic form in R we don't need to explicitly transpose the first vector of the quadratic form. The matrix multiplication operator in R is **%\*%**. The **vcov** function is used to extract the variance-covariance matrix of the regression coefficients. Finally to obtain the standard error we take the square root of the quadratic form.

Here is how we can use this formula to calculate the standard errors of the treatment means in our ANOVA model.

sqrt(myvec('Co','D',1)%\*%vcov(out3)%\*%myvec('Co','D',1))

[,1]  
[1,] 0.05245374

sqrt(myvec('Ru','S',1)%\*%vcov(out3)%\*%myvec('Ru','S',1))

[,1]  
[1,] 0.05022518

To efficiently calculate all of the treatment means and their standard errors I first create a data frame that contains the factor level combinations corresponding to the different treatments. The **expand.grid** function can be used to generate all possible combinations of the levels of the three factors.

**#generate all combinations of factors**

fac.vals2 <- expand.grid(fac1=c('Co','No','Ru'), fac2=c('D','S'), fac3=1:2)

fac.vals2

fac1 fac2 fac3  
1 Co D 1  
2 No D 1  
3 Ru D 1  
4 Co S 1  
5 No S 1  
6 Ru S 1  
7 Co D 2  
8 No D 2  
9 Ru D 2  
10 Co S 2  
11 No S 2  
12 Ru S 2

To get the treatment mean that corresponds to the factor levels in row 1 of fac.vals2 I would need to enter the following.

myvec(fac.vals2[1,"fac1"], fac.vals2[1,"fac2"], fac.vals2[1,"fac3"])%\*%coef(out3)

[,1]  
[1,] 3.383238

Next I rewrite my vector function myvec so that it takes as input a specified row of the data frame fac.vals2.

myvec2 <- function(x) myvec(fac.vals2[x,1], fac.vals2[x,2], fac.vals2[x,3])

myvec2(1)

[1] 1 0 0 0 0 0 0 0

Using this function and the **sapply** function of R I obtain the mean and standard error of the mean for each treatment. We can obtain confidence intervals by noting that if the response variable is normally distributed then the regression parameter estimates have a joint normal distribution. Sums of jointly normally distributed random variables are also normally distributed with the standard error calculated above.

fac.vals2$est <- sapply(1:12, function(x) myvec2(x) %\*% coef(out3))

fac.vals2$se <- sapply(1:12, function(x) sqrt(myvec2(x) %\*% vcov(out3) %\*% myvec2(x)))

fac.vals2$low95 <- fac.vals2$est + qnorm(.025)\*fac.vals2$se

fac.vals2$up95 <- fac.vals2$est + qnorm(.975)\*fac.vals2$se

fac.vals2

fac1 fac2 fac3 est se low95 up95  
1 Co D 1 3.383238 0.05245374 3.280431 3.486046  
2 No D 1 3.930537 0.04606953 3.840242 4.020832  
3 Ru D 1 3.920225 0.05198867 3.818329 4.022121  
4 Co S 1 3.400387 0.04162371 3.318806 3.481968  
5 No S 1 3.961099 0.04277330 3.877265 4.044934  
6 Ru S 1 4.100877 0.05022518 4.002438 4.199317  
7 Co D 2 3.490818 0.04941952 3.393957 3.587678  
8 No D 2 4.038116 0.04304455 3.953751 4.122482  
9 Ru D 2 4.027804 0.04393244 3.941698 4.113910  
10 Co S 2 3.671196 0.04162371 3.589615 3.752777  
11 No S 2 4.231909 0.04178987 4.150002 4.313815  
12 Ru S 2 4.371686 0.04341654 4.286592 4.456781

When we compare these means and standard errors to those produced by the **effect** function of the **effects** package we see that they agree. On the other hand the upper and lower 95% confidence interval endpoints are slightly different at the third or fourth decimal place.

fac.vals

fac1 fac2 fac3 est se low95 up95  
2401 Co D 1 3.383238 0.05245374 3.279889 3.486587  
2411 No D 1 3.930537 0.04606953 3.839767 4.021307  
2421 Ru D 1 3.920225 0.05198867 3.817792 4.022657  
2431 Co S 1 3.400387 0.04162371 3.318376 3.482398  
2441 No S 1 3.961099 0.04277330 3.876824 4.045375  
2451 Ru S 1 4.100877 0.05022518 4.001919 4.199835  
24011 Co D 2 3.490818 0.04941952 3.393447 3.588188  
24111 No D 2 4.038116 0.04304455 3.953306 4.122927  
24211 Ru D 2 4.027804 0.04393244 3.941245 4.114364  
24311 Co S 2 3.671196 0.04162371 3.589186 3.753207  
24411 No S 2 4.231909 0.04178987 4.149571 4.314247  
24511 Ru S 2 4.371686 0.04341654 4.286143 4.457229

The source of the difference is that the **effect** function uses a t-quantile in calculating the confidence interval rather than a normal quantile. The correct degrees of freedom for the t-distribution is contained in the **$df.residual** component of the model. To match the results from **effect** we need to modify our calculation of the confidence intervals as follows.

names(out3)

[1] "coefficients" "residuals" "effects" "rank" "fitted.values"  
[6] "assign" "qr" "df.residual" "na.action" "contrasts"   
[11] "xlevels" "call" "terms" "model"

out3$df.residual

[1] 231

**#redo using t-distribution**

fac.vals2$up95 <- fac.vals2$est + qt(.975,out3$df.residual)\*fac.vals2$se

fac.vals2$low95 <- fac.vals2$est + qt(.025,out3$df.residual)\*fac.vals2$se

fac.vals2

fac1 fac2 fac3 est se low95 up95  
1 Co D 1 3.383238 0.05245374 3.279889 3.486587  
2 No D 1 3.930537 0.04606953 3.839767 4.021307  
3 Ru D 1 3.920225 0.05198867 3.817792 4.022657  
4 Co S 1 3.400387 0.04162371 3.318376 3.482398  
5 No S 1 3.961099 0.04277330 3.876824 4.045375  
6 Ru S 1 4.100877 0.05022518 4.001919 4.199835  
7 Co D 2 3.490818 0.04941952 3.393447 3.588188  
8 No D 2 4.038116 0.04304455 3.953306 4.122927  
9 Ru D 2 4.027804 0.04393244 3.941245 4.114364  
10 Co S 2 3.671196 0.04162371 3.589186 3.753207  
11 No S 2 4.231909 0.04178987 4.149571 4.314247  
12 Ru S 2 4.371686 0.04341654 4.286143 4.457229

It's worth noting that the approach described here will work for any regression model. For example, the simple regression model

lm.model <- lm(y ~ x)

corresponds to the following model for the mean: μ = β0 + β1*x*. The standard error of this mean that can then be used to construct confidence bands for the regression line is the following.

sqrt(c(1,x) %\*% vcov(lm.model) %\*% c(1,x))

**Negative Binomial Distribution**

I finish our survey of probability distributions useful in ecology with a discussion of the negative binomial distribution.

**Basic characteristics**

* A negative binomial (NB) random variable is discrete. A typical use of the negative binomial distribution is as a model for count data.
* Like the Poisson distribution the negative binomial distribution is bounded on one side. It is bounded below by 0, but is theoretically unbounded above.
* Suppose we have a sequence of independent Bernoulli trials in which the probability of a success on any given trial is a constant *p*. Let *Xr* denote the number of failures that are endured before *r* successes are achieved. Then *Xr* is said to have a negative binomial distribution with parameter *p* (and *r*).
* The negative binomial is a two-parameter distribution, but like the ordinary binomial one of the parameters, in this case *r*, is usually treated as known.

**Derivation of the negative binomial probability mass function**

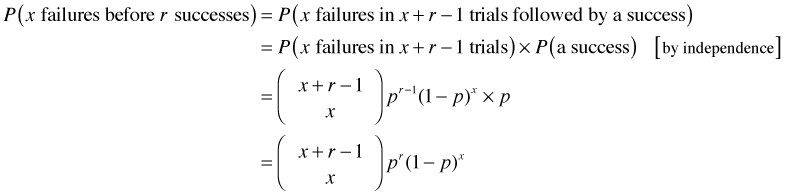
The probability mass function of the negative binomial distribution comes in two distinct versions. The first one is the one that appears in every introductory probability textbook; the second is the one that appears in books and articles in ecology. Although the ecological definition is just a reparameterization of the mathematical definition, the reparameterization has a profound impact on the way the negative binomial distribution gets used. We'll begin with the mathematical definition. From an ecological standpoint the mathematical definition is rather bizarre and except for perhaps modeling the number of rejections one has to suffer before getting a manuscript submission accepted for publication, it's hard to see how this distribution is useful.

Let *Xr* be a negative binomial random variable with parameter *p*. Using the definition given above let's calculate prob, the probability of experiencing *x* failures before *r* successes are observed. Note: The change in notation from *k* to *x* is deliberate. Unfortunately in a number of ecological textbooks the symbol *k* means something very specific for the negative binomial distribution so I don't want to use it in a generic sense here.

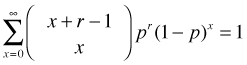
If we experience *x* failures and *r* successes, then it must be the case that we had a total of *x* + *r* Bernoulli trials. Furthermore, we know that the last Bernoulli trial resulted in a success, the rth success, because that's when the experiment stops.



What we don't know is where in the first *x* + *r* – 1 Bernoulli trials the *x* failures and *r* – 1 successes occurred. Since the probability of a success is a constant *p* on each of these trials, we have a binomial experiment in which the number of trials is *x* + *r* – 1. Thus we have the following.

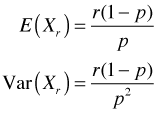


So we're done. Note: it's a nontrivial exercise to show that this is a true probability distribution, i.e.,



**Mean and variance**

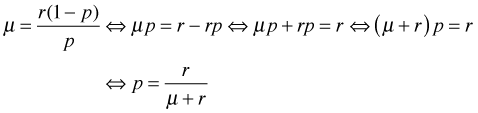
I'll just state the results.



**Ecological parameterization of the negative binomial distribution**

The ecological definition of the negative binomial is essentially a reparameterization of the formula for the probability mass function that we just derived.

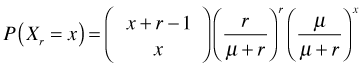
**Step 1:** The first step in the reparameterization is to express *p* in terms of the mean *μ* and use this expression to replace *p*. Using the formula for the mean of the negative binomial distribution above, I solve for *p*.



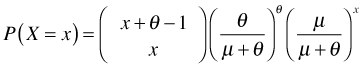
From which it immediately follows that

one minus p

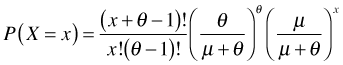
Plugging these two expressions into the expression for the probability mass function above yields the following.



**Step 2:** This step is purely cosmetic. Replace the symbol *r*. There is no universal convention as to what symbol should be used as the replacement. Venables and Ripley (2002) use *θ*. Krebs (1999) uses *k*. SAS makes the substitution alpha. I will use the symbol *θ*.



**Step 3:** Write the binomial coefficient using factorials.



**Step 4:** Rewrite the factorials using gamma functions. This step requires a little bit of explanation.

**Gamma Function**

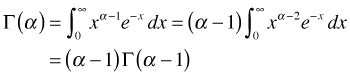
The **gamma function** is defined as follows.

gamma function

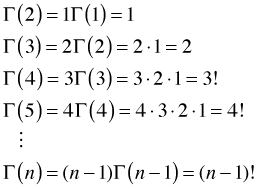
Although the integrand contains two variables, *x* and *α*, *x* is the variable of integration and will disappear once the integral is evaluated. So the gamma function is solely a function of *α*. The integral defining the gamma function is called an improper integral because infinity appears as an endpoint of integration. Formally such an improper integral is defined as a limit.

Now if http://www.unc.edu/courses/2010fall/ecol/563/001/images/lectures/lecture6/alphagt1.gif, but still an integer, the integral in the gamma function will be a polynomial times an exponential function. The standard approach for integrating such integrands is to use integration by parts. Integration by parts is essentially a reduction of order technique—after a finite number of steps the degree of the polynomial is reduced to 0 and the integral that remains to be computed turns out to be http://www.unc.edu/courses/2010fall/ecol/563/001/images/lectures/lecture6/Gof1.gif(but multiplied by a number of constants). It is a simple exercise in calculus to show that http://www.unc.edu/courses/2010fall/ecol/563/001/images/lectures/lecture6/Gof1.gif= 1.

After the first round of integration by parts is applied to the gamma function we obtain the following.



where in the last step I recognize that the integral is just the gamma function in which *α* has been replaced by alpha minus 1. This is an example of a recurrence relation; it allows us to calculate one term in a sequence using the value of a previous term. We can use this recurrence relation to build up a catalog of values for the gamma function.



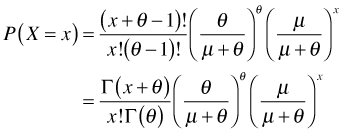
So when *α* is a positive integer, the gamma function is just the factorial function. But http://www.unc.edu/courses/2010fall/ecol/563/001/images/lectures/lecture6/gammaofalpha.gifis defined for all positive *α*. For example, it can be shown that

gamma of one half

and then using our recurrence relation we can evaluate others, such as

gamma of threehalf

**Step 4 (continued):** So using the gamma function we can rewrite the negative binomial probability mass function as follows.



where I've chosen to leave *x*! alone just to remind us that *x* is the value whose probability we are computing.

So what's been accomplished in all this? It would seem not very much, but that's not true. The formula we're left with bears little resemblance to the one with which we started. In particular, all reference to *r*, the number of successes, has been lost having been replaced by the symbol *θ*. Having come this far, ecologists then take the next logical step. Since the gamma function does not require integer arguments, why not let *θ* be any positive number? And so *θ* is treated solely as a fitting parameter, it's original meaning having been lost (but see below).

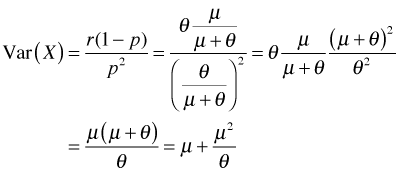
* Engineers sometimes follow the convention of reserving the term "negative binomial distribution" for only the first parameterization we've described, the one in which the parameter *r* takes on only positive integer values. In contrast they refer to the ecologist's parameterization with the positive continuous parameter *θ* as the Polya distribution.
* As if this were not confusing enough the engineer's "true" negative binomial distribution is sometimes called the Pascal distribution. Then the two parameterizations we've described are called the Pascal and Polya distributions respectively, and the term negative binomial distribution is not used at all.

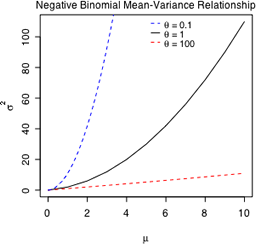
So, what we're left with is a pure, two-parameter distribution, i.e., X distr NB, where the only restriction on *μ* and *θ* is that they are positive. With this last change, the original interpretation of the negative binomial distribution has more or less been lost and it is best perhaps to think of the negative binomial as a probability distribution that can be flexibly used to model discrete data.

**The variance of the negative binomial distribution in terms of μ and θ**

It turns out that a Poisson random variable can be viewed as a special limiting case of a negative binomial random variable in which the parameter *θ* is allowed to become infinite. Given that there are infinitely many other choices for *θ*, this is further evidence of the flexibility of the negative binomial distribution over the Poisson distribution. In a sense *θ* is a measure of deviation from a Poisson distribution. For that reason *θ* is sometimes called the **inverse index of aggregation** (Krebs 1999)—inverse because small values of *θ* correspond to more clumping than is typically seen in the Poisson distribution. It is also called the **size** parameter (documentation for R), but most commonly of all, it is called the **dispersion parameter** (or **overdispersion parameter**).

The relationship between the negative binomial distribution and the Poisson can also be described in terms of the variances of the two distributions. To see this I express the variance of the negative binomial distribution using the parameters of the ecologist's parameterization.





**Fig. 1** Mean-variance relationship as a function of θ

Observe that the variance is quadratic in the mean. Since parabola, this represents a parabola opening up that crosses the *μ*-axis at the origin and at the point parabola root. *θ* controls how fast the parabola climbs. As http://www.unc.edu/courses/2010fall/ecol/563/001/images/lectures/lecture6/thetagoestoinfinity.gif, http://www.unc.edu/courses/2010fall/ecol/563/001/images/lectures/lecture6/poisson%20variance2.gif, and we have the variance of a Poisson random variable. For large *θ*, the parabola is very flat while for small *θ* the parabola is narrow. Thus *θ* can be used to describe a whole range of heteroscedastic behavior.

Note: In the parameterization of the negative binomial distribution used by SAS, SAS parameterization. Thus the Poisson distribution corresponds to *α* = 0 and values of *α* > 0 correspond to overdispersion. This is perhaps the more natural parameterization.

**Mechanisms that give rise to a negative binomial distribution**

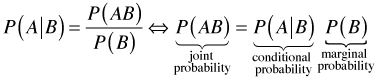
**Negative binomial as a model of a nonhomogeneous Poisson process**

The extreme flexibility of the negative binomial distribution in fitting heteroscedastic discrete data would be enough to recommend it but it turns out that it can also be motivated on purely ecological grounds. Recall that two of the assumptions of the homogeneous Poisson process, homogeneity and independence, are unlikely to hold for most ecological data. If either one of these assumptions is relaxed, then under certain circumstances the distribution that we observe, rather than being Poisson, turns out to be negative binomial. I next try to make this connection more precise.

In a homogeneous Poisson process the rate constant λ is the same for all observational units. In a **nonhomogeneous Poisson process**, the rate constant λ is allowed to vary according to some distribution. Given a particular realization from this distribution, say lambdai, the resulting random variable *X* will have a Poisson distribution with lambdai. We can express this formally using the notion of conditional probability. We write

http://www.unc.edu/courses/2010fall/ecol/563/001/images/lectures/lecture6/conditional%20poisson.gif

The fact that λ has a distribution is a bit of a nuisance because what we want is the unconditional (or marginal) probability marginal. But if we knew what the distribution of λ was, we could obtain this marginal probability as follows. Recall from the definition of conditional probability that



If our interest is in http://www.unc.edu/courses/2010fall/ecol/563/001/images/lectures/lecture6/P(A).gifwe can find it by summing out B in the joint distribution.

joint marginal

For continuous distributions the same effect is accomplished by integration.

http://www.unc.edu/courses/2010fall/ecol/563/001/images/lectures/lecture6/continuous.gif

Thus for the nonhomogeneous Poisson process, if we knew what the distribution of λ was, f of lambdasay, we could calculate marginalas follows.

marginal poisson

The function mixing distributionthat appears in the above integral is called a **mixing distribution** for the Poisson. What should we choose as the mixing distribution? Let's list the obvious requirements for such a density.

1. Since the Poisson distribution is a model of counts and λ is the mean of that distribution, we must have λ > 0. Thus a distribution such as the normal distribution that allows both positive and negative values is clearly out.
2. Without any specific knowledge about how λ might vary across subjects we should probably choose a function that is flexible, that can describe a wide range of possible distributions for λ.
3. We should probably choose a function that will allow us to actually compute the integral above. Hence it needs to be "complementary" to the Poisson mass function that it multiplies in the integral. (Note: this last point is less important today with the availability of MCMC for estimating such integrals. We'll return to this point when we discuss Bayesian estimation.)

One distribution that satisfies all three requirements is the [**gamma distribution**](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture3.htm#gamma). The upstart is that if we carry out the above integration with a gamma distribution as the mixing distribution we obtain the ecologist's parameterization of the negative binomial distribution with α playing the role of θ. Thus the marginal density of a nonhomogeneous Poisson process when the gamma distribution is used as a mixing distribution is negative binomial with parameters μ and α. A negative binomial distribution constructed in this way is sometimes called a mixture distribution. In the spatial statistics literature, particularly in geography, the negative binomial is called a **compound Poisson distribution**.

This provides another ecological rationalization for fitting a negative binomial model to count data. Essentially, if you suspect heterogeneity may be at play the negative binomial should be a good choice.

* For example, if you have count data for a given species on different transects where the habitat quality of the transects varies, then you might suspect that the fitness of the species will also vary causing parameter heterogeneity across the transects.
* If you have count data for many species collectively and the species composition varies across quadrats, then the average rate parameter in a quadrat would be expected to vary once again yielding heterogeneity.
* If you are monitoring a disease for which hosts have a varying susceptibility, you might expect heterogeneity to be present.

While heterogeneity may initially be present, it's possible that by including useful predictors in a regression model you can end up "explaining" much of it. The negative binomial error distribution then could still be used to account for the lingering heterogeneity that was not accounted for by the model.

The connection between the gamma mixing distribution and the final negative binomial model opens up some interesting possibilities. Suppose you suspect heterogeneity in your population but have no idea what form it might take. When you fit a negative binomial model you obtain μ and θ from which you can then obtain α and β of the gamma mixing distribution. The gamma probability distribution with these values of α and β then can be used to explicitly characterize, as a distribution, the way in which the heterogeneity manifests itself in your sample.

**Zero Inflation**

**Zero inflation** refers to observing more zeros than is predicted by a particular probability model, typically, more zeros than is predicted by a Poisson model. Zero inflation often occurs in applications and is particularly common in ecology, especially when the event being tracked is rare or sporadic.

A number of so-called zero-inflated models have been proposed in which the zeros are treated as heterogeneous, some of them coming from the same probability generating mechanism that generated the observed counts (the true zeros) and some coming from another process (the false zeros). Such an interpretation can make sense in habitat suitability studies in which, say, the abundance of a colonizing species is being tracked. In this situation one might suspect that there exists habitat that although suitable for habitation is still uncolonized because it's inaccessible (false zeros), as well as other habitat that although accessible has not been colonized because it is not suitable (true zeros). A recent proponent of zero-inflated models in ecology is Martin *et al.* (2005). It is my experience, also see Warton (2005), that negative binomial models are very good at accounting for excess zeros thus making it unnecessary to model the zero category separately.

I have typically found that for heteroscedastic count data with few or no zeros, log-transformed count models often do as well or better than negative binomial models in describing the data. But as soon as the zero category becomes prominent, log-transformed models become nonsensical. A log-transformation works by converting an asymmetric distribution of counts into a symmetric distribution of log counts. But a pile of zeros at one end of the distribution can never be transformed away. After the transformation is applied that pile will still be at the end of the distribution resulting in a transformed distribution that is neither bell-shaped nor symmetric.

**Polya-Eggenberger Urn Model**

A completely different way in which a negative binomial distribution can arise in theory is via what is called the Polya-Eggenberger urn model. This is one of the many so-called Polya urn models that have been proposed. The description of this one is as follows.

* Suppose we have an urn with *N* balls in it of which a fraction *p* are red and the remaining fraction 1 – *p* are black.
* At each trial we draw one ball from the urn, observe its color, and return it to the urn along with *θ × N* balls of the same color (where usually we take 0 < θ < 1 although this is not required). By adding more balls of the same color we make the occurrence of the same event at the next trial more likely.
* Let *Xr* = number of red balls in the urn after *r* trials of this sort. It is not hard to write down the formula for Prob Xr = kbut it is rather involved, particularly the simplification that is required to proceed further, so we'll skip it.
* It can be shown that if we take the limit of this expression in a certain way, r goes to infinitywhile p and theta, that the limiting distribution is negative binomial.

It's hard to know how useful this result is in practice. What is important is that the Polya-Eggenberger scheme clearly violates the independent increments hypothesis that was one of the three basic assumptions of the Poisson model. Thus we see that if either one of the major assumptions of the Poisson model, homogeneity or independence, is violated we can be led, under certain circumstances, to a negative binomial model.

**A Generalized Poisson Model**

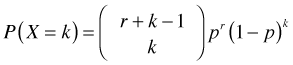
There is an even more direct way in which violation of the independence hypothesis can lead from a Poisson model to a negative binomial model. This is through what's sometimes called a generalized Poisson model.

Suppose events come in clusters. The clusters themselves are distributed spatially according to a Poisson distribution with parameter λ. Within each cluster suppose the number of events observed follows a logarithmic distribution (also called a log series or Fisher log series distribution) with parameter *p*, 0 < *p* < 1. Then in any cluster *i* we have

http://www.unc.edu/courses/2010fall/ecol/563/001/images/lectures/lecture6/logseries.gif

This is a particular long-tailed distribution that has been used to model species abundances ([Dewdney 2000](http://www.biolbull.org/cgi/reprint/198/1/152.pdf)).

Now suppose we're not aware of the clustering that is going on and we just lay down quadrats on the terrain. In each quadrat we count up how many events we observe, irrespective of whether or not they occur in clusters or not. Let *X* be the number of events observed in a quadrat. Then it turns out that



where r parameterizationand *p* is the *p* of the log series distribution. You should recognize this as the original parameterization of the negative binomial distribution.

So we see that clustered events, if the clusters and the events in a cluster arise in the way described here, can also give rise to a negative binomial distribution. Note: the log series distribution is required for this to happen. The log series distribution has a long history in ecology of being used to model species abundance relations.

**The use of the negative binomial distribution in practice**

We have now seen three distinct more or less biologically relevant mechanisms that can lead to a negative binomial distribution.

1. A gamma mixing distribution for a Poisson distribution with heterogeneous rate parameters.
2. A Polya-Eggenberger urn model.
3. A generalized Poisson model with Poisson distributed clusters and log series counts in a cluster.

In a classic paper, Boswell and Patil (1970) outlined 12 distinct ways in which a negative binomial distribution can arise in practice, of which we've described four (if you include the original definition which is essentially that of a waiting time distribution). Because the negative binomial can arise in so many distinct ways it is probably not surprising that it often fits ecological data really well. Its appeal for use as a probability generating mechanism in ecology derives from the following.

* It offers a model for heteroscedasticity.
* It can deal with zero inflation without auxiliary assumptions.
* It respects the discreteness of the data. (It doesn't insult the data by transforming them and then pretending they're continuous.)
* It can be motivated biologically—see the three mechanisms described above.

Here are some concrete illustrations of how we might rationalize the use of the negative binomial distribution as a probability model.

* Suppose seeds are randomly distributed in an environment (governed by a Poisson model) and that each seed germinates and gives rise to a number of stems via vegetative reproduction, the number of stems being governed largely by how long the plant has been present in the environment. If the details of the generalized Poisson model apply we might expect the number of plants present to follow a negative binomial distribution.
* Suppose seeds are heterogeneous in their propensity to germinate. This would imply a nonconstant λ in a Poisson model yielding a nonhomogeneous Poisson process. This may in turn yield a negative binomial distribution for the number of plants.
* In the case of a contagious disease where initial inoculates were distributed randomly according to a Poisson process, the Polya-Eggenberger urn model suggests we might end up with a distribution that is negative binomial.

In nature we often see count data that are zero-inflated relative to a Poisson (meaning we see more zeros than the fitted Poisson model can predict). Typically, a negative binomial distribution can be successfully fit to zero-inflated data.

The R negative binomial probability functions are **dnbinom**, **pnbinom**, **qnbinom**, **rnbinom**. The R functions allow both the mathematical and ecological parameterizations of the negative binomial we've described. For the ecological parameterization R calls **size** the dispersion parameter while **mu** is the mean.

WinBUGS only offers the mathematical parameterization of the negative binomial distribution in its **dnegbin** function. This is not adequate for ecological work because in this parameterization the "dispersion" parameter is required to be a positive integer. So when we use WinBUGS to fit negative binomial models we'll have to formulate the ecological parameterization ourselves by combining a Poisson probability model with a gamma mixing distribution. There are actually two ways in which this can be done giving rise to two kinds of negative binomial distributions. In the econometrics literature they are referred to as the NB1 and NB2 models. The basic model we've been describing corresponds to the NB2 model.

In conclusion the negative binomial distribution is useful as a model for count data whenever we suspect heterogeneous rates, lack of independence, or clustering may be occurring. We will study the negative binomial model again in a regression setting later in this course.

**Illustrations of negative binomial distributions**

As a first illustration I fix the mean of the negative binomial distribution to be 1 (μ = 1) and I vary the dispersion parameter from small (θ = 0.1) to relatively large (θ = 100).

par(mfrow=c(1,4))

par(lend=2)

**#fix the mean, vary the dispersion parameter**

plot(0:20, dnbinom(0:20, mu=1, size=.1), type='h',lwd=4, xlab='count category', ylab='probability', ylim=c(0,.8))

mtext(side=3, line=.5, expression(list(mu==1, theta==.1)), cex=.9)

plot(0:20, dnbinom(0:20, mu=1, size=1), type='h',lwd=4, xlab='count category', ylab='probability', ylim=c(0,.8))

mtext(side=3, line=.5, expression(list(mu==1, theta==1)), cex=.9)

plot(0:20, dnbinom(0:20, mu=1, size=10),type='h', lwd=4, xlab='count category', ylab='probability', ylim=c(0,.8))

mtext(side=3, line=.5, expression(list(mu==1, theta==10)), cex=.9)

plot(0:20, dnbinom(0:20,mu=1,size=100), type='h',lwd=4, xlab='count category', ylab='probability', ylim=c(0,.8))

mtext(side=3, line=.5, expression(list(mu==1, theta==100)), cex=.9)

par(mfrow=c(1,1))

|  |
| --- |
| **fig 2** |
| **Fig. 2** Negative binomial distributions with the same mean but different dispersions |

What we see is that as θ gets small the fraction of zeros increases and the distribution becomes more spread out. Observe that with θ = 0.1 almost 80% of the observations are zero yet there is a non-negligible probability of obtaining a count of 20 or more. On the other hand when θ = 100 less than 40% of the observations are zero and the probability of obtaining a count beyond 10 is negligible.

When θ = 100 in Fig. 2 the distribution looks very Poisson-like. I investigate this by comparing it to a Poisson distribution with the same mean.

**#compare Poisson and NB**

par(mfrow=c(1,2))

plot(0:12, dnbinom(0:12, mu=1, size=100), type='h', lwd=6, xlab='count category', ylab='probability', ylim=c(0,.5))

mtext(side=3, line=.5, expression(paste('NB: ',list(mu==1, theta==100))))

plot(0:12,dpois(0:12,1), type='h', lwd=6, xlab='count category', ylab='probability', ylim=c(0,.5))

mtext(side=3, line=.5, expression(paste('Poisson: ', lambda==1)))

par(mfrow=c(1,1))

|  |
| --- |
| **fig 3** |
| **Fig. 3** Comparing a negative binomial and Poisson distribution with the same mean (μ = 1) and θ relatively large (θ = 100) |

The distributions are nearly identical. In retrospect this should be expected. As θ → ∞, the negative binomial distribution with mean μ becomes a Poisson distribution with mean μ.

Finally I fix the dispersion parameter of the negative binomial distribution at a small value and vary the mean.

**#fix the dispersion parameter, vary the mean**

par(mfrow=c(1,3))

plot(0:40, dnbinom(0:40, mu=1, size=.1), type='h', lwd=3, xlab='count category', ylab='probability', ylim=c(0,.8))

mtext(side=3, line=.5, expression(paste('NB: ', list(mu==1, theta==.1))), cex=.9)

plot(0:40, dnbinom(0:40, mu=10, size=.1), type='h', lwd=3, xlab='count category', ylab='probability', ylim=c(0,.8))

mtext(side=3, line=.5, expression(paste('NB: ', list(mu==10, theta==.1))), cex=.9)

plot(0:40, dnbinom(0:40, mu=20, size=.1), type='h', lwd=3, xlab='count category', ylab='probability', ylim=c(0,.8))

mtext(side=3, line=.5, expression(paste('NB: ', list(mu==20, theta==.1))), cex=.9)

par(mfrow=c(1,1))

|  |
| --- |
| **fig 4** |
| **Fig. 4** Negative binomial distributions with the same dispersion but different means |

What's striking about Fig. 4 is that other than a slight shrinkage in the zero fraction, the distributions look about the same. In truth the right tails of the distribution also become more prominent as the mean increases. Still the primary consequence of increasing the mean is to reduce the number of zeros. A negative binomial distribution with a small dispersion parameter will still have a sizeable zero fraction even if it has a very large mean.

**Methods of Estimation**

A probability model is used to describe the behavior of a response variable. As I explained in [lecture 4](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture4.htm#extending), a useful way to think of a probability model in a regression setting is as a data-generating mechanism. The role of predictor variables is to constrain the probability model by determining the values of the model's parameters. While I've argued that data analysis in ecology should use probability models that respect the natural characteristics of ecological data, I've not explained how the information contained in the predictors can be used to determine the values of the parameters of our probability models. We begin this topic today.

While there are many estimation methods available, there are three broad classes of estimation techniques that are widely popular. They are

* least squares estimation
* maximum likelihood estimation
* Bayesian estimation

Least squares estimation is extremely popular because it's easy to understand, easy to implement, and has good properties. In ordinary linear regression the regression model takes the form

ordinary regression

Geometrically the problem amounts to fitting a line to a scatter plot. The method of least squares obtains values for β0 and β1 that minimize the sum of squared deviations about that line.

http://www.unc.edu/courses/2010fall/ecol/563/001/images/lectures/lecture6/OLS.gif

The sum of squared deviations is the objective function of least squares and it readily generalizes to more complicated scenarios. The standard tools of calculus are used to minimize the objective function and well-defined algorithms exist for finding solutions in both the linear and nonlinear setting.

Observe that least squares makes no assumption about an underlying probability model for the response. Typically, a normal model is assumed, but that assumption is tacked on at the end and is not formally part of the least squares solution. A normal probability model is a natural choice for least squares because the basic formulation involving squared deviations necessarily assumes that the errors are symmetrically disposed about the regression surface. Symmetry is also a characteristic of a normal distribution.

For asymmetric probability models, which include most of the ones we've discussed, least squares solutions don't make much sense. The methodologies we consider as an alternative to least squares incorporate the probability model directly into the objective function, rather than tacking it on at the end as an afterthought. Two popular approaches are maximum likelihood estimation and Bayesian estimation. We begin with a discussion of maximum likelihood and turn later to Bayesian methods.

**Maximum Likelihood Estimation**

**The random sample**

Suppose we obtain a random sample of size *m*. For example, suppose we select *m* shoots at random from a field. On each shoot we count the number of aphids present. The number of aphids observed on a given shoot is a random variable (it has a probability distribution). Denote this random variable by the symbol *X*. In our random sample then we observe the values of *m* random variables, *X*1, *X*2, ... , *Xm*, one for each shoot in our sample.

What is the probability of obtaining the data we collected? The notation for this probability is the following.

joint probability

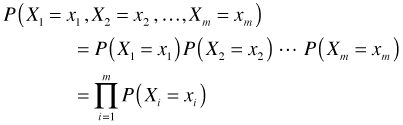
This is called a joint probability, the probability of simultaneously observing all *m* events. (I follow the standard statistical convention of using capital letters for random variables and lower case variables for their values.) Another way of writing this is

joint  probability

Since we have a random sample, each of these events is independent of the other. From elementary probability theory, if events *A* and *B* are independent then we have

independence

Applying this to our data we have

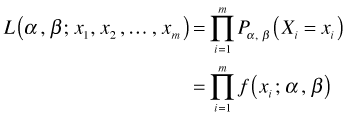


where I use Π to indicate the product (analogous to Σ for sums).

**The likelihood**

Now suppose we have a probability model for these data. For count data this might be a Poisson or negative binomial model, but for the moment we'll treat the probability model generically. Let prob modeldenote the probability model. Here *f* in general can be a density or a probability mass function while α and β are its parameters. For a Poisson model there is a single parameter, λ, while for a negative binomial model there are two parameters usually denoted μ and θ. When we extend the problem to regression models the parameter list will also include the set of regression coefficients in our model.

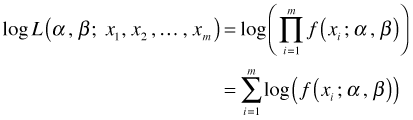
Using this generic probability model we can rewrite the expression for the probability of our data. This new formulation is called the **likelihood** of our data with respect to the model *f*.



As the notation indicates, in the likelihood perspective the probability of our data is a function of the parameters α and β. After we've collected the data it's the data that are known, observed values and the parameter values that are unknown. (Note: if *f* is a density function then the above expression is not a probability. We need to integrate a density in order to obtain a probability. We can get around this by including an increment delta xiin the expression for the product and then arguing that the product of a density times an increment is an approximation to the true probability.)

**The log-likelihood**

For both practical and theoretical reasons, it is preferable to work with the natural logarithm of the likelihood function, i.e., the **log-likelihood**. For our generic probability model the log-likelihood takes the following form.



In the last step I used one of the basic properties of the logarithm function. For positive numbers *a* and *b* we have that log product, so the logarithm function turns multiplication into addition.

**The maximum likelihood principle**

One of R. A. Fisher's major contributions (one of many) to statistics was to realize that the likelihood function perspective is a vehicle for obtaining parameter estimates. He proposed what has become known as the maximum likelihood principle for parameter estimation.

**Maximum Likelihood Principle**: Choose as your estimates of the parameters those values that make the data we actually obtained the most probable. In other words, we should choose the values that maximize the value of the likelihood. For the following reasons we typically find the parameter estimates that maximize the log-likelihood instead.

1. Because the logarithm is a monotone increasing function, the likelihood and the log-likelihood will achieve their maximum at exactly the same place. So, nothing is lost is using the log-likelihood.
2. For hand calculations the log-likelihood is far easier to work with since it converts products into sums.
3. All of the theoretical results concerning maximum likelihood estimators are based on the log-likelihood.
4. Using log-likelihoods increases the numerical stability of parameter estimates. Because likelihoods arise from joint probabilities (at least in the discrete setting) that, under independence, factor into a product of marginal probabilities, the magnitude of the likelihood can be quite small, often very close to zero. With a large number of observations this value can even approach the machine zero of the computing device being used, leading to numerical problems. Log-transforming the likelihood converts these tiny probabilities into moderately large negative numbers thus eliminating numerical instability.

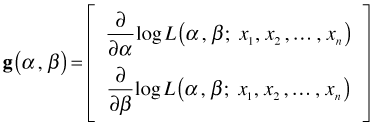
Another argument in favor of maximum likelihood estimates (MLEs) is that the maximum likelihood estimates of a model's parameters give that model the best chance of fitting the data. If after using these "best" estimates the model is deemed inadequate we can be sure that it is truly inadequate.

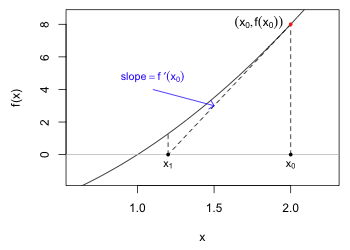
**Maximizing the log-likelihood: the Newton-Raphson method**

Maximizing a log-likelihood can be done in various ways.

* Graphically by plotting the log-likelihood and estimating where the peak occurs.
* Algebraically by using calculus. This is a viable option only for fairly simple problems.
* Numerically using special optimization routines.

The derivative of the log-likelihood is called the **score** or **gradient** function. For log-likelihoods that are functions of more than one parameter obtaining the gradient means taking first partial derivatives with respect to each parameter in turn. The result is then organized in a vector.





**Fig. 5** An illustration of Newton's method ([R code](http://www.unc.edu/courses/2010fall/ecol/563/001/notes/lecture6%20fig5.html))

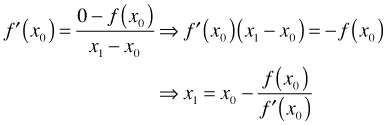
From calculus we know that all local maxima occur at points where the first derivative is equal to zero, the so-called critical points. In all but the simplest of problems finding a critical point has to be done numerically. Numerical optimization is a rich and active area of research in mathematics, and there are many different approaches available. The Newton-Raphson method is often a good choice.

The Newton-Raphson method is just the multivariate generalization of Newton's method, a method which you should have learned in calculus. Suppose we're trying to find a root of the function *f*, i.e., the value of *x* such that *f*(*x*) = 0. Fig. 5 illustrates this scenario where it appears that *x* = 1 is a root of the function *f*. Newton's method allows us to start with an initial guess for the root and then if our guess is good enough and the function is sufficiently well-behaved, it will after a finite number of steps return a reasonable estimate of the root.

We start with a guess for the root. Fig. 5 shows the guess x0 = 2. We evaluate the function at the guess to obtain the point on the curve labeled (*x*0, *f*(*x*0)). The derivative of *f* yields a formula for the slope of the tangent line to the curve at any point on the curve. Thus *f* ′(*x*0) is the slope of the tangent line to the curve at (*x*0, *f*(*x*0)). A portion of the tangent line is shown in Fig. 5 as the line segment that connects (*x*0, *f*(*x*0)) to the point (*x*1, 0), the place where the tangent line crosses the *x*-axis. Using the usual formula for the slope of a line segment we obtain two different expressions for the slope of the displayed tangent line.

point slope

I solve this expression for *x*1 (where at the second step below I assume that http://www.unc.edu/courses/2010fall/ecol/563/001/images/lectures/lecture6/fprimenotequalzero.gif).



Observe from Fig. 5 that *x*1 is closer to the point where *f* crosses the *x*-axis than was our initial guess *x*0. We could now use *x*1 as our new initial guess and repeat this process yielding a value *x*2 that will be closer to the root than was *x*1.

The formula given above is an example of a recurrence relation and it allows us to use the current value to obtain an updated value. This is the basis of Newton's method. Written more generally the estimate of the root at the (*k*+1)st step of the algorithm is given in terms of the *k*th step as follows.

newton

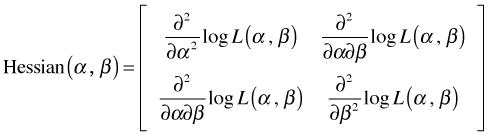
In the maximum likelihood problem we desire the roots of the score function, the derivative of the log-likelihood. To simplify notation let loglikelihood. Suppose the log-likelihood function is a function of the single parameter θ. Then to find the maximum likelihood estimate of θ we need to find the roots of the equation

score eqn

Thus we need the roots of l prime. In Newton's formula let *x* = θ, l prime, and l  primeprime. Newton's method formula applied to the maximum likelihood problem becomes the following.

newton-raphson

When there are multiple parameters to be estimated θ becomes a vector of parameters, l primebecomes the gradient or score vector (as was depicted above for two parameters), and ldoubleprimebecomes a matrix of second partials called the Hessian matrix **H**. The Hessian matrix when there are two parameters, α and β, is the following.



In the formula for Newton's method ldoubleprimeoccurs in the denominator. This is equivalent to multiplying by its reciprocal. The analogous operation for matrices is matrix inversion. Thus the Newton-Raphson method as implemented for finding the MLEs of a log-likelihood with multiple parameters is the following.

newton

Here, for example, **θ** might be the vector , **g** and **H** are as shown above, and **H**-1 denotes the inverse of the Hessian matrix.

**Cited References**

* Boswell, M. T. and G. P. Patil. 1970. Chance mechanisms generating the negative binomial distributions. In G. P. Patil (editor), *Volume 1: Random Counts in Models and Structures*, Pennsylvania State University Press, University Park, PA, 1–22.
* Dewdney, A. K. 2000. A dynamical model of communities and a new species-abundance distribution. *Biological Bulletin* **198**: 152–165.
* Hilbe, Joseph M. 2007. *Negative Binomial Regression*. Cambridge University Press, New York.
* Krebs, Charles J. 1999. *Ecological Methodology*. Addison Wesley Longman: Menlo Park, CA.
* Martin, T. G., B. Wintle, J. Rhodes, P. M. Kuhnert, S. A. Field, S. Low Choy, A. J. Tyre, and H. P. Possingham. 2005. Zero tolerance ecology: improving ecological inference by modelling the source of zero-inflation. *Ecology Letters* **8**: 1235–1246.
* Venables, W. N. and B. D. Ripley. 2002. *Modern Applied Statistics with S, 4th edition*. Springer-Verlag: New York.
* Warton, David I. 2005. Many zeros does not mean zero inflation: comparing the goodness-of-fit of parametric models to multivariate abundance data. *Environmetrics* **16**(3): 275–289.

**R Code**

A compact collection of all the R code displayed in this document appears [here](http://www.unc.edu/courses/2010fall/ecol/563/001/notes/lecture6%20Rcode.html).

[Course Home Page](http://www.unc.edu/courses/2010fall/ecol/563/001/index.html)

|  |
| --- |
| Jack Weiss *Phone:* (919) 962-5930 *E-Mail:* jack\_weiss@unc.edu *Address:* Curriculum in Ecology, Box 3275, University of North Carolina, Chapel Hill, 27516 Copyright © 2010 Last Revised--Sept 28, 2010 URL: [http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm](http://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture6.htm#lecture6) |