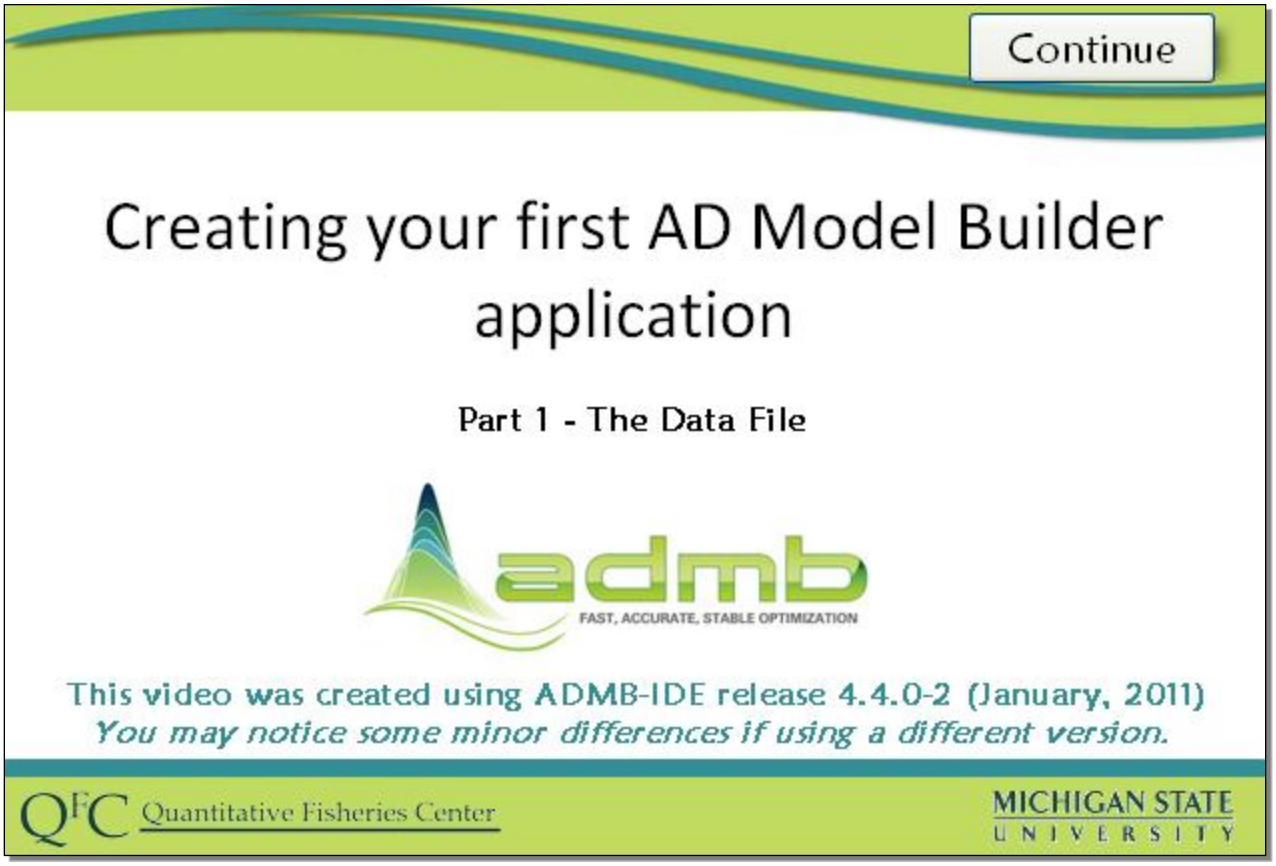
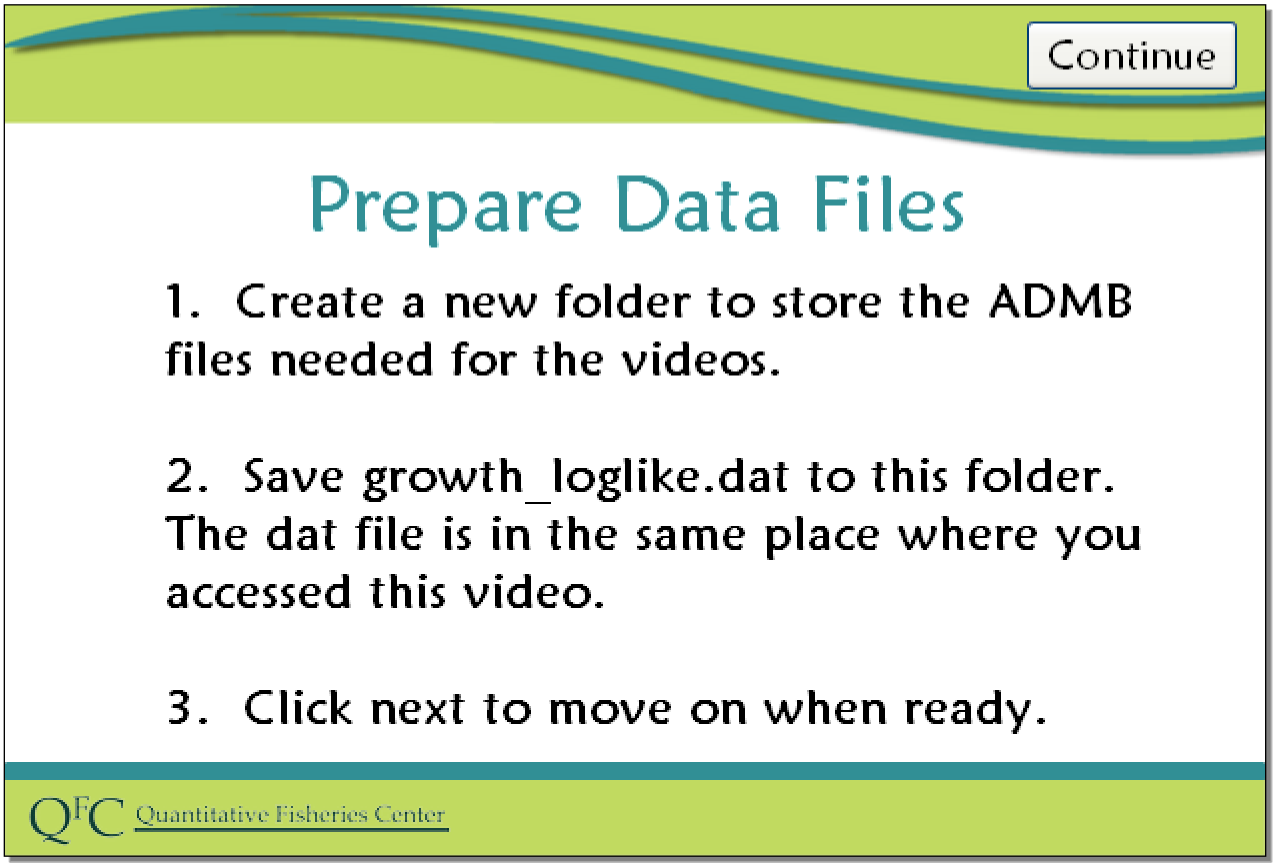
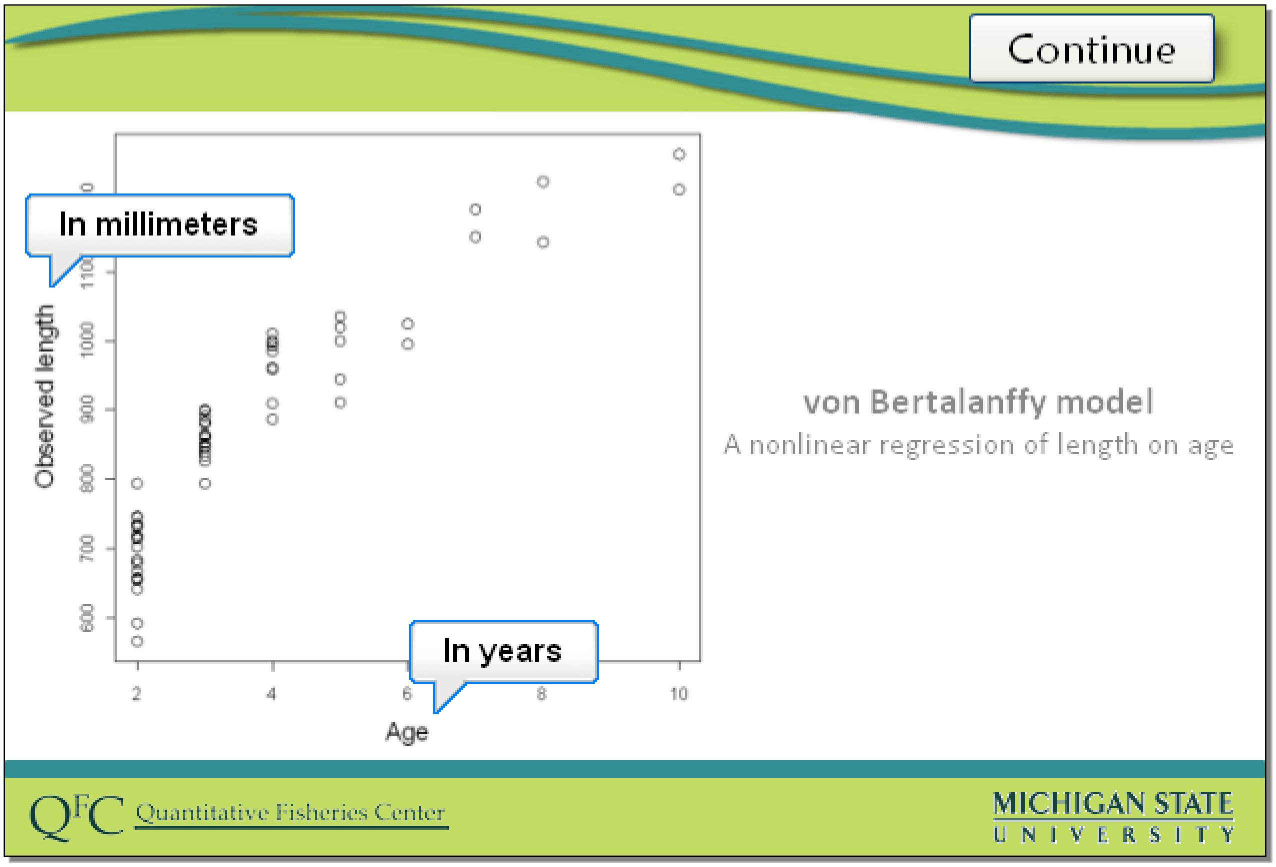



This is the first in a series of videos intended to illustrate how to build an ADMB application. In this series of videos we assume you already have ADMB installed on your computer. We also assume that you understand the fundamentals of maximum likelihood estimation and want to learn how to apply the approach using ADMB. If you do not already have ADMB installed you can refer to our installation video for ADMB-IDE for windows, an emacs integrated development environment for ADMB. We will use this environment in these videos.

# The Data

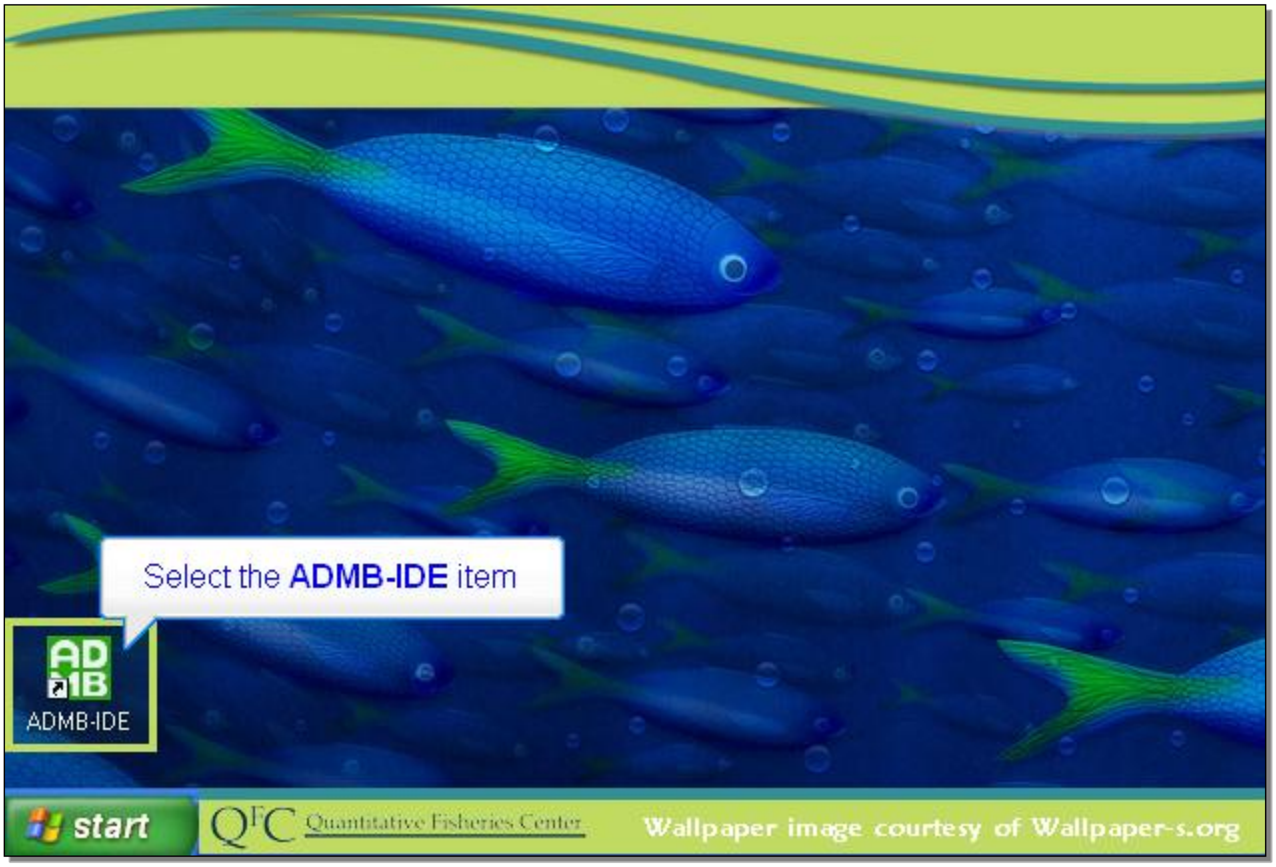



1. Create a new folder
2. Locate and save growth_loglike.dat to this folder.



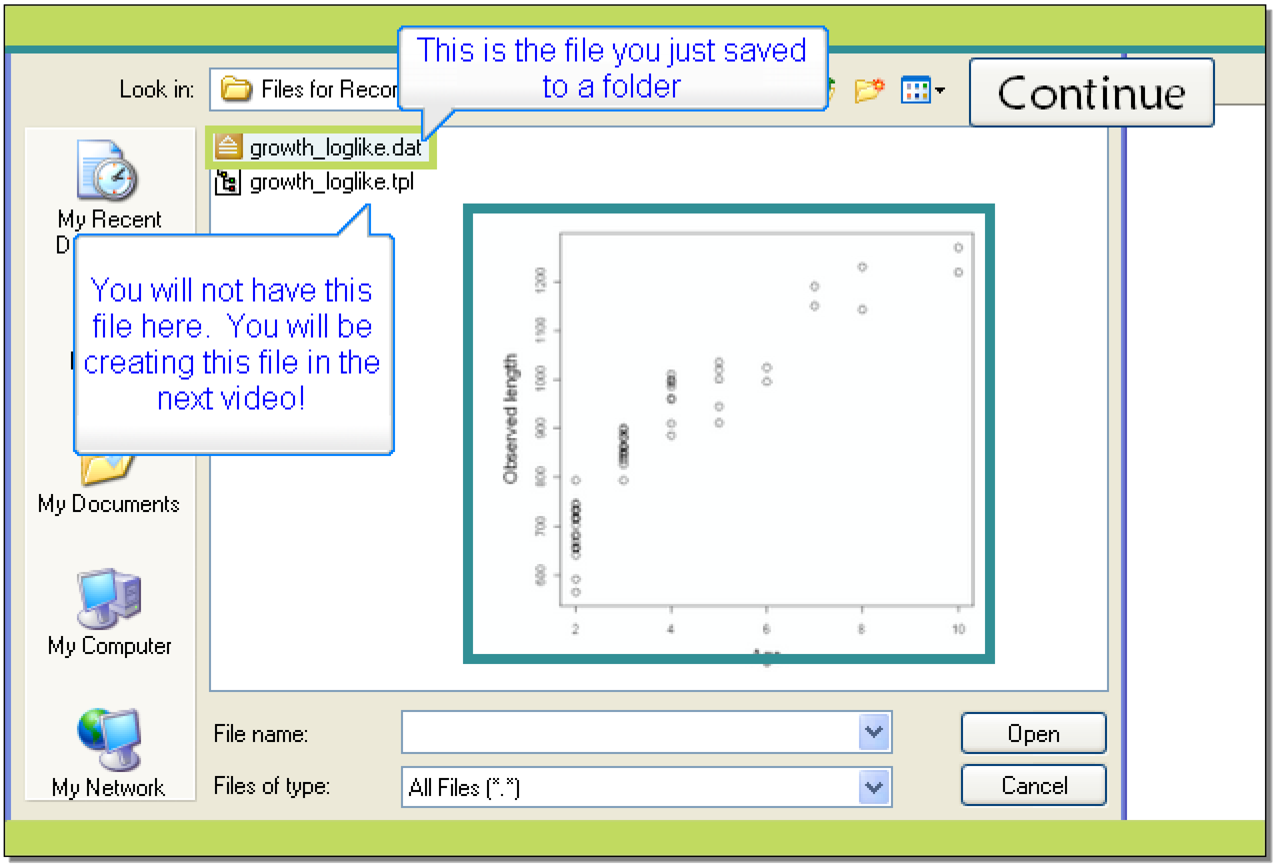

The data we will be working with are ages and corresponding lengths of Musky collected in the New River of Virginia by Travis Brenden during 2000 through 2003. Observed lengths for individual fish are in millimeters and corresponding ages are in years. We will be fitting a model that relates length to age. In particular we will be fitting the von Bertalanffy model. Now lets look at the data file using the admb-IDE emacs interface.

# Open ADMB




Begin by opening ADMB. Check to see if you have a desktop icon. If you do, you can double-click it to open ADMB. Otherwise click the start button and find it in your programs.

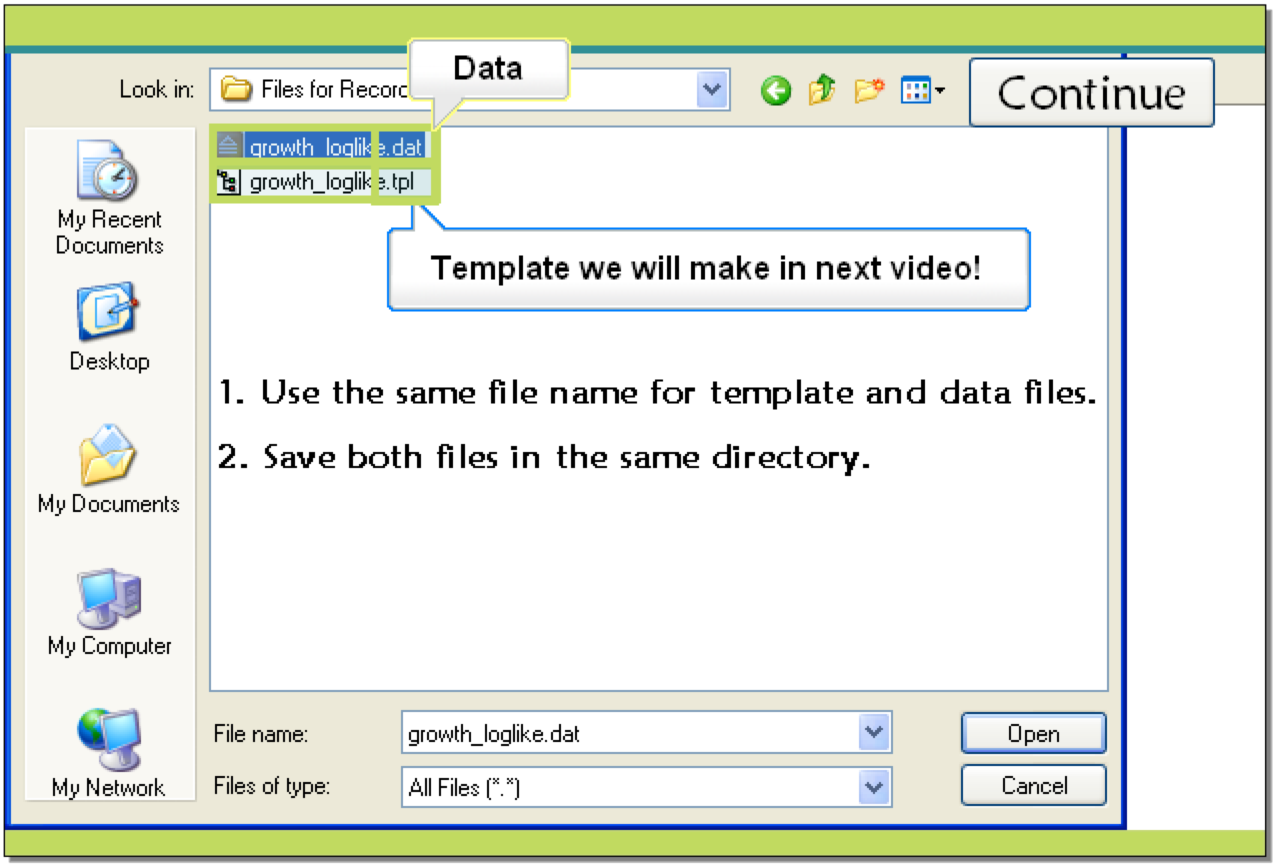
# Open Dat File




The ADMB emacs interface opens. Now we will open our data file.

1. Begin by selecting file
2. to open file.
3. and locate your files.

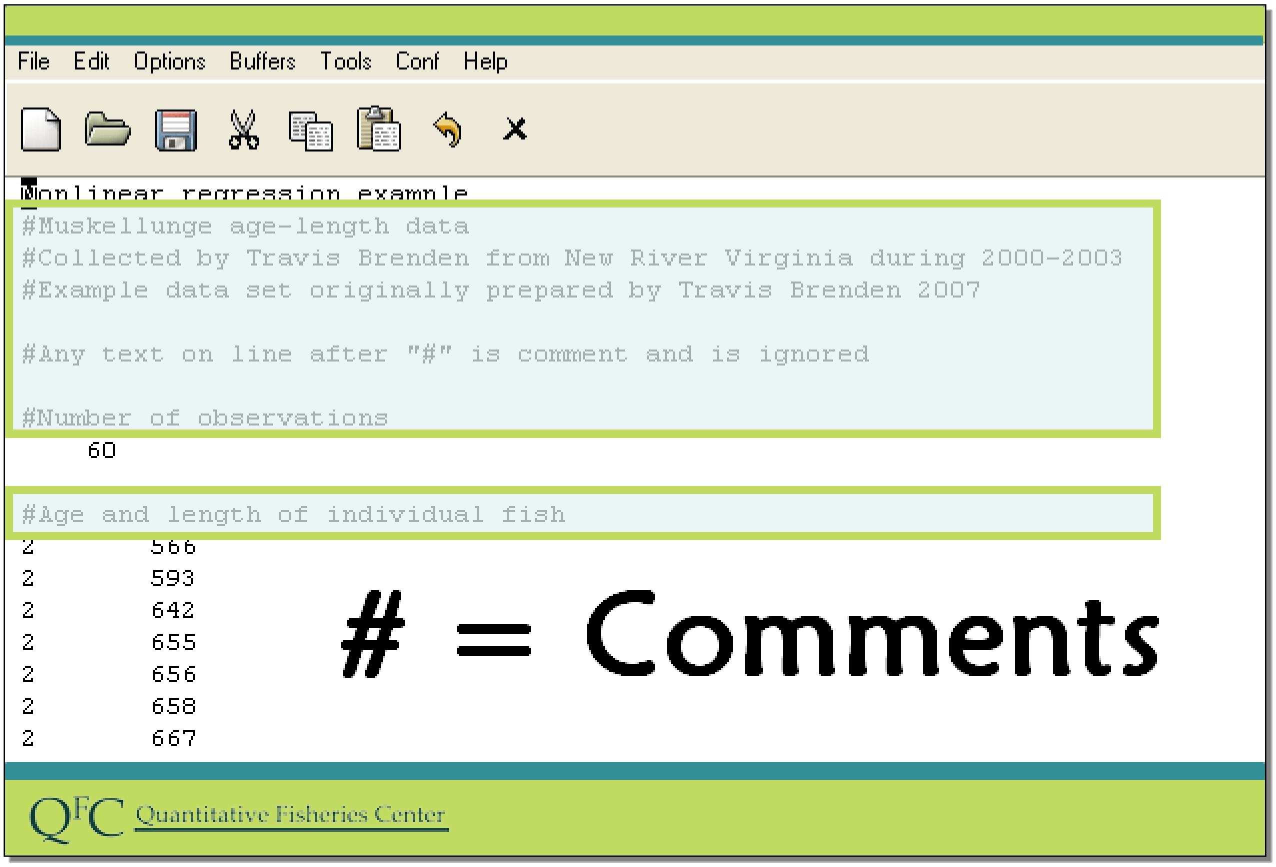
This file, growth_loglike.dat, contains the data you just saw plotted.



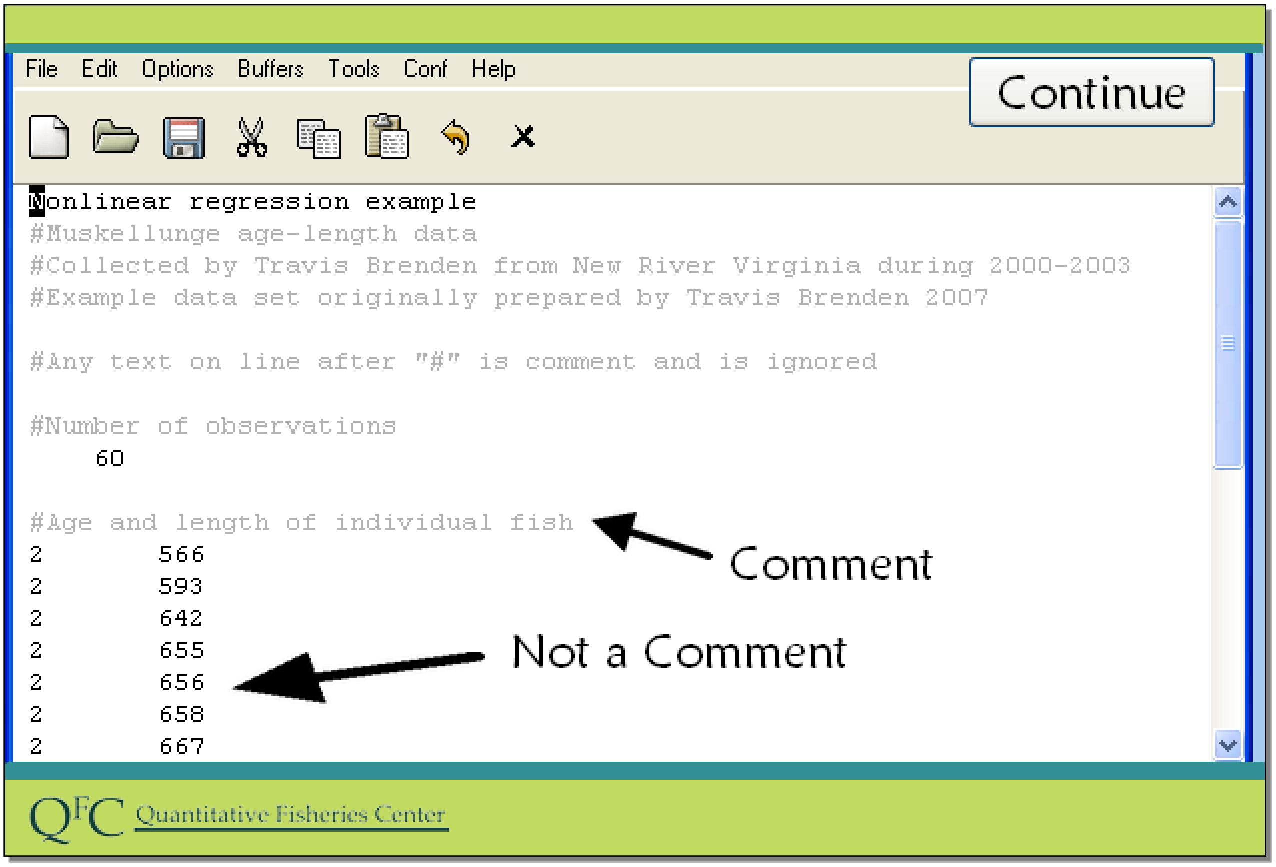

The filename here is the same name we will use for our admb template file but has an extension of “dat” rather than “tpl”. Use of the same file name and the “dat” extension are the default expectations made by admb. The data file is stored in the same directory or folder we will keep the template file in, which is also a default expectation if you have a standard installation of admb-IDE.

Double click the dat file.

# The Dat File




Notice that some lines in this file start with a pound sign. In an admb data file the pound sign indicates that the text that follows is a comment. When admb reads the data file it will skip right over the lines starting with the pound sign. These comments are just here for us humans.

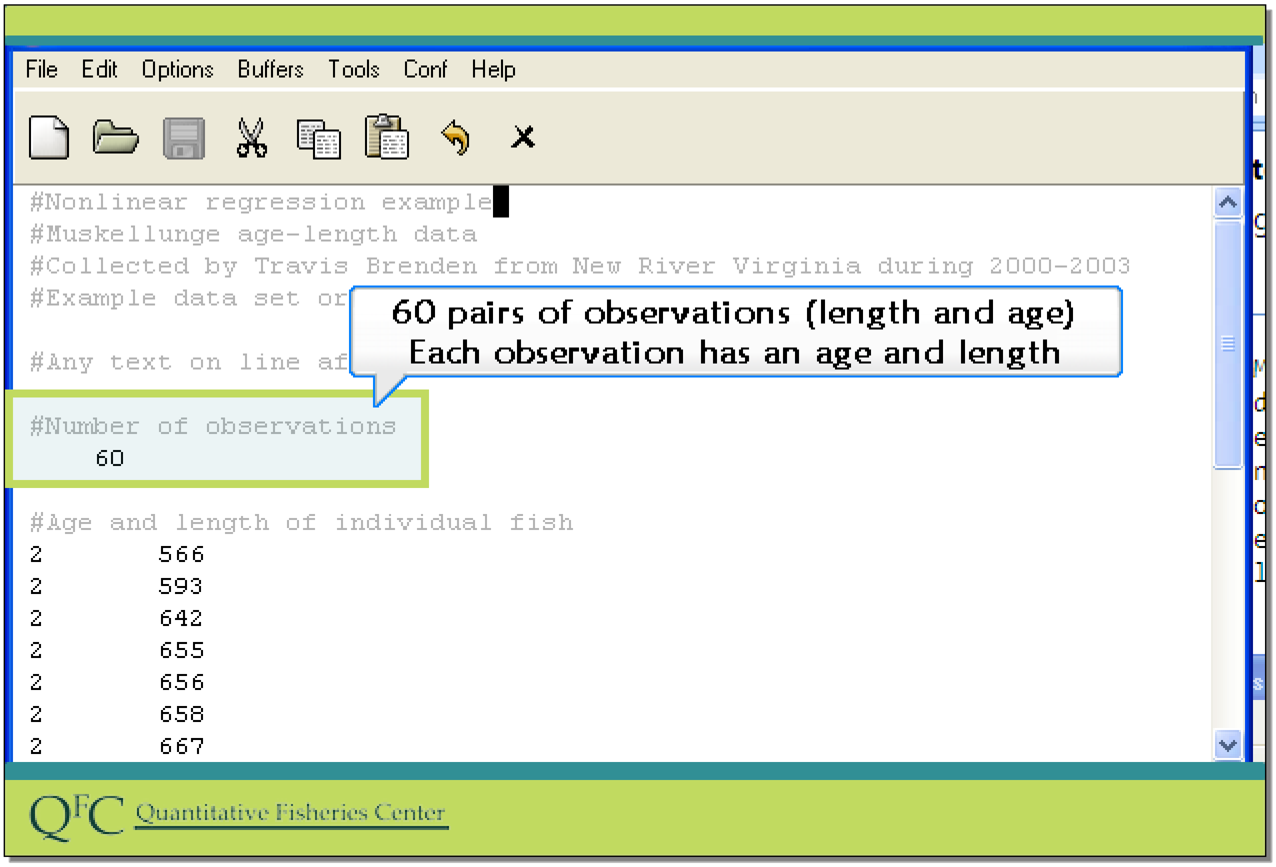



Notice that first line, which is intended to be a comment, does not have a # sign at the start. This is a mistake and will cause errors when admb attempts to read the file. After fixing the problem notice that the text on that line that is now a comment and has become shaded.

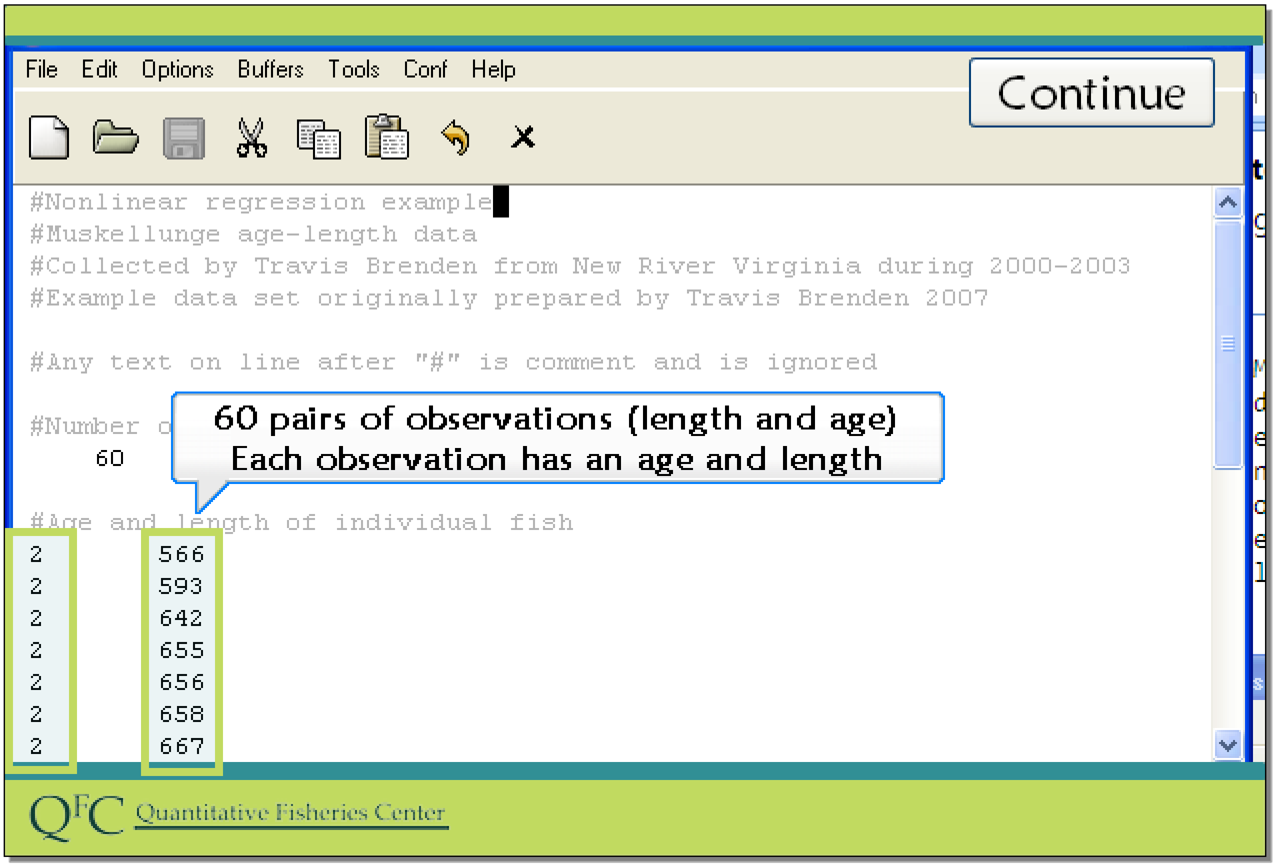
Comment lines are shaded which makes it easy to see when a line is a comment or not.

**Slide Code:**

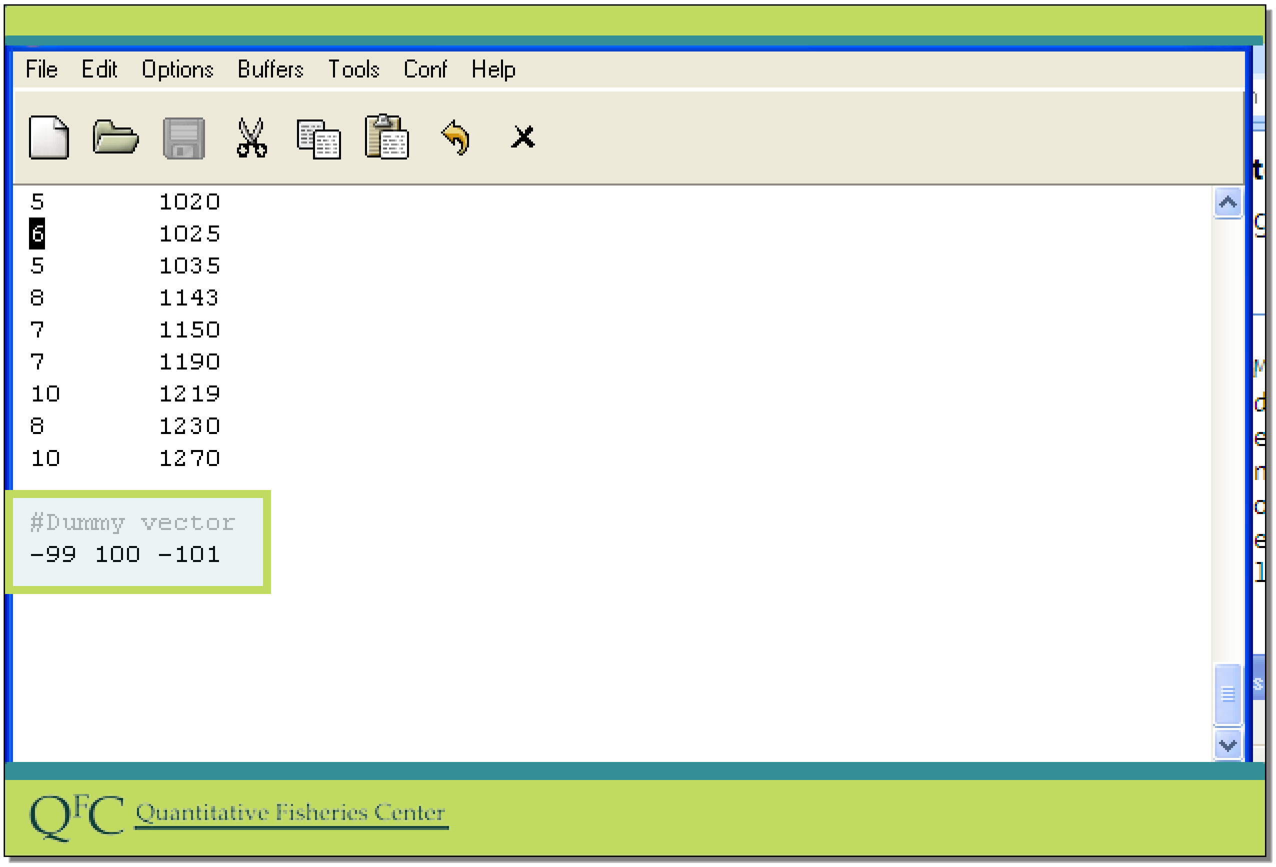
#



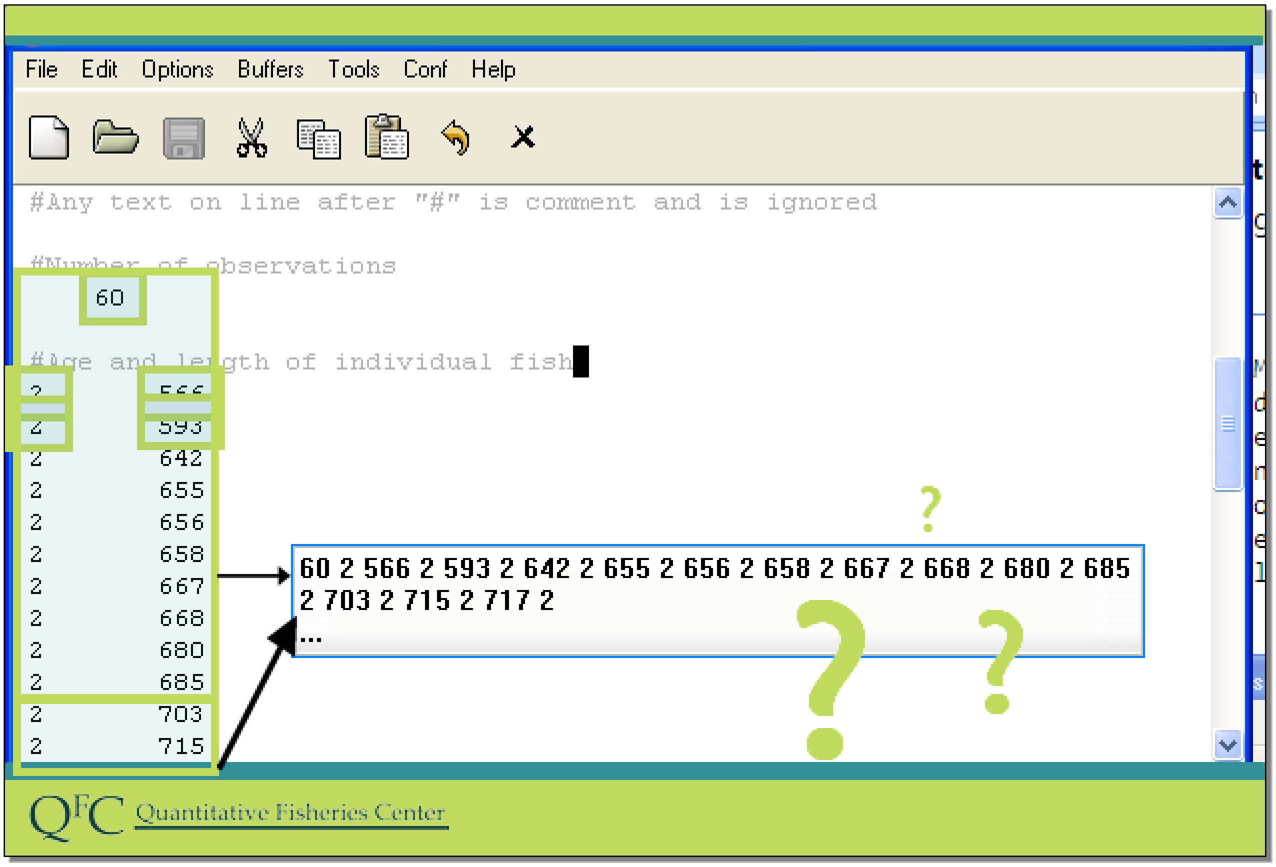

The first actual data value that your admb program will read in the file is 60, the number of observations. Because admb programs read the data file sequentially they need to be told in advance how many observations to read. Here 60 refers to the number of pairs of ages and lengths that follow.



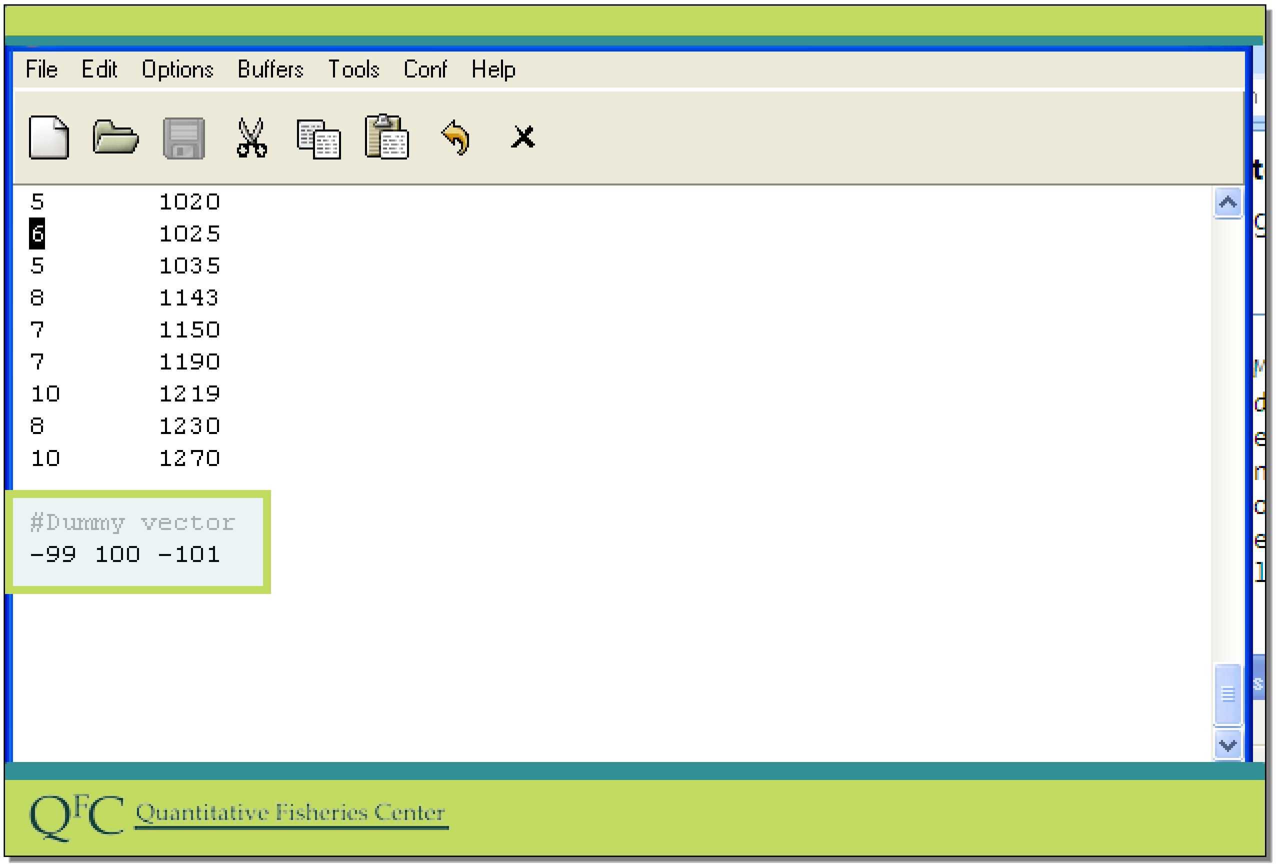

The data file includes ages and lengths organized in 60 rows with the first column being age and the second being length.




Finally at the bottom of the data files is a vector of three test numbers. These are included so that when admb is finished we can check if it is still reading in data to the right place.

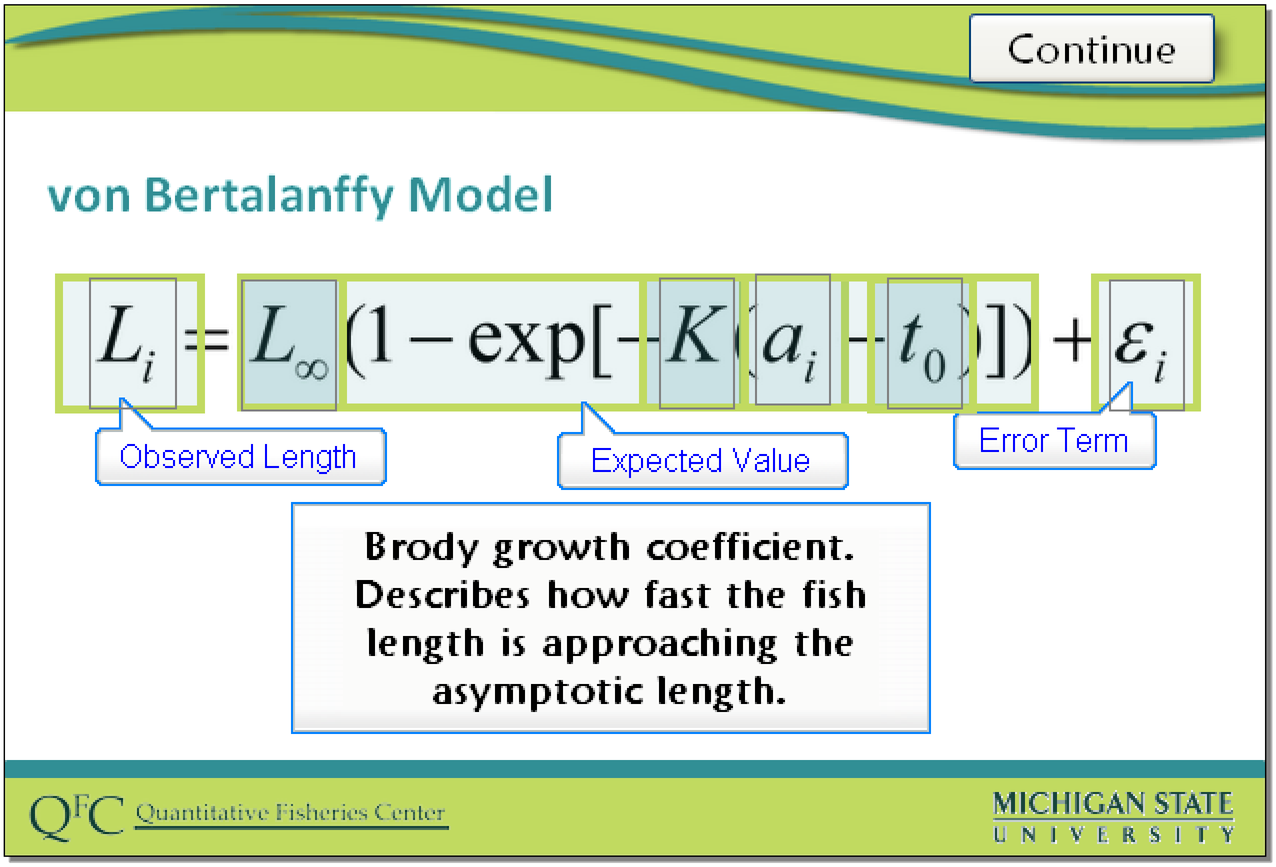



Note that Admb also does not care about the organized way the numbers are presented in the file. It just reads numbers sequentially in the order we will tell it to. Each time it encounters white space, i.e., spaces or tabs, it will move on to the next number. We could have put the number of observations and the first 10 pairs of age and length data on the first line, then followed this with a line with the next 3 age-length pairs and the age for the next age length pair and so on with line breaks when we feel like. Of course we don't do this because we would have trouble looking at the data file and understanding what was what. However it is important to remember that admb does not see our careful organization. If you tell it to read in a matrix containing 60 rows and two columns with the age and length data, but there were really 59 rows it would read right past the end of the data and end up with nonsense values for the 60th observation.

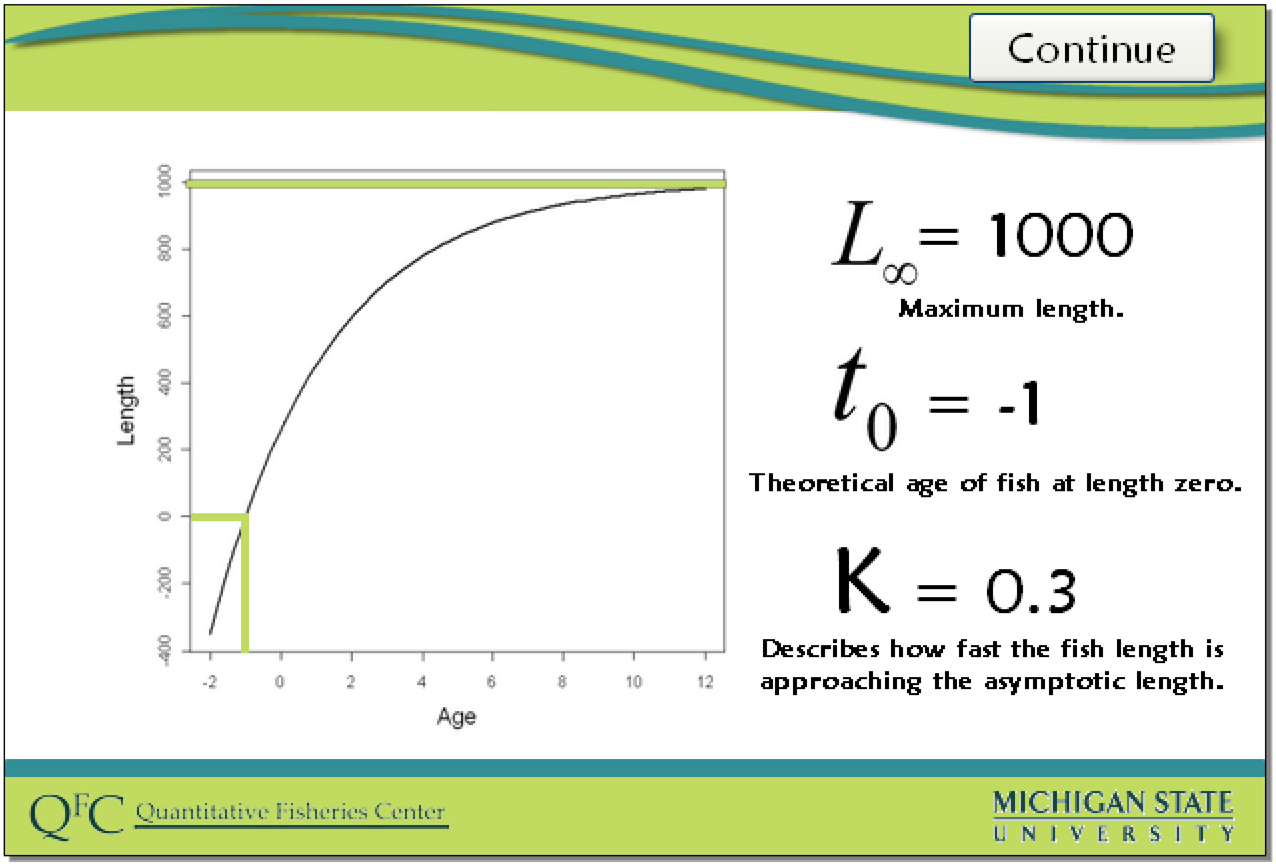



This is why having a test value or vector at the end of the data file is a useful check. If when your admb program reads in the test vector it gets the numbers you expect you know your data file had the same number of values before the test vector as your program assumed there would be. Before moving on I again want to make sure you are with me so again you will have a chance to answer a few simple questions.

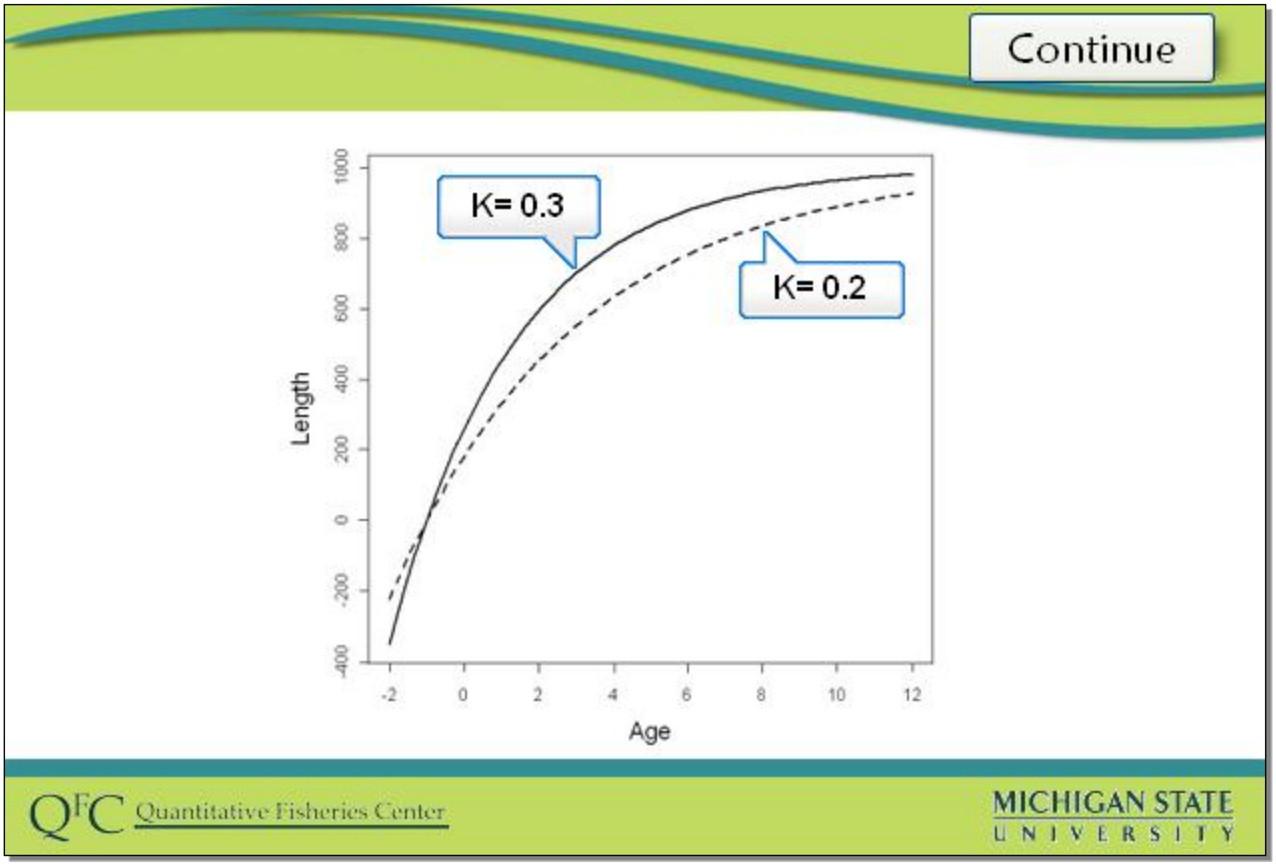
# The Model



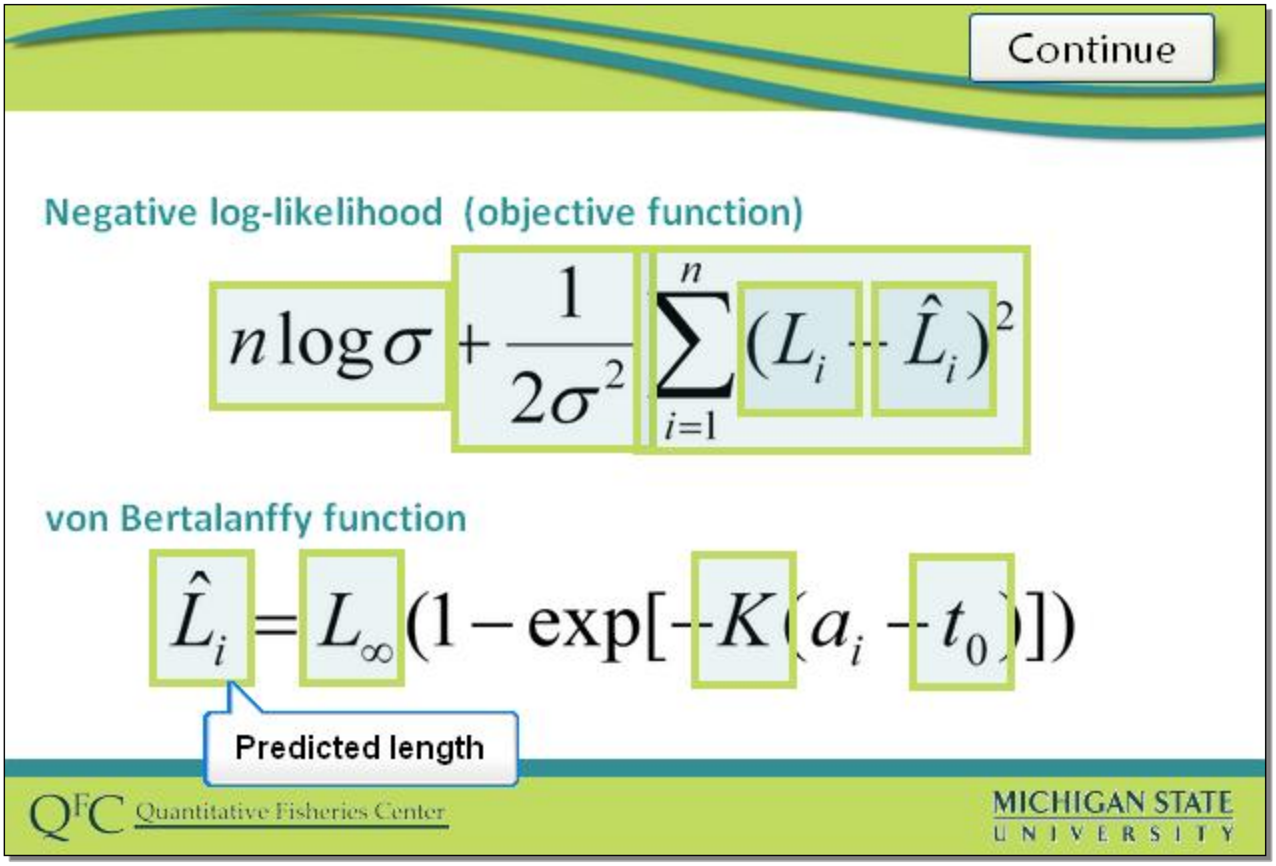

Shown here is the equation for the model. Here L sub i represents the observed length of the ith individual, and a sub i is the age of that individual. Three of the parameters of the model are L-infinity, K, and t-naught and these will need to be estimated. Epsilon sub i is the error term, which reflects how much the ith observed length deviates from the value expected for a fish of that age. The expected value of length increases as age increases. L-infinity is the asymptotic length or expected length for very old fish and for our application will have units of millimeters. The parameter t-naught is the so called theoretical age a fish would be of length zero and for our application will have units of years. This is referred to as “theoretical” because the von Bertalanffy model is typically applied to older fish and ages where fish would be near zero are outside the range we would expect the model to apply to. K is the Brody growth coefficient. This is not really a measure of growth because it has units of time to the minus 1 rather than size per unit time. It describes how fast the fish length is approaching the asymptotic length. To complete the description we also need to specify a distributional assumption for the error term. Here we assume a normal distribution with mean zero and variance sigma squared. This is equivalent to saying that the distribution of length given age is normal with expected value given by the vonBertalanffy function ignoring the error term, and with variance sigma squared.




Shown here is the von Bertalanffy function for expected length given age, with L-infinity equal to 1000, t naught equal to minus 1, and K equal to 0.3  As you can see predicted length is zero at an age of -1 or t0.  Length approaches L-infinity asymptotically as age increases.




Here we have added dashed curve to the same graph. This curve is for a von Bertalanffy function with the same L-infinity and t-naught. The difference is that we have decreased K from 0.3 to 0.2.




To use the maximum likelihood method to fit the von Bertalanffy model and estimate the parameters, we need to write out the negative log-likelihood function. AD model builder will require us to code this as our objective function. Based on our assumed normal distribution for the errors it follows that the negative log likelihood function, ignoring an additive constant, is n, the sample size time the log of sigma, plus 1 over 2 times sigma squared, times the sum of squared deviations, where each deviation is the difference between the observed length and predicted length. Expected lengths are from the von Bertalanffy function ignoring the error term, and are a function of a, the age. Note that here we use hat over L to denote expected length. This is the expected length given age and the parameters t-naught, L-infinity, and K. Because we don't know the values of the parameters and are trying different values or trial estimates in the equation, we often refer to L hat as a predicted length. When we obtain maximum likelihood estimates for the parameters the value we obtain from this equation is really an estimate of the expected value of length given age. We now have the background needed to implement this example in AD Model builder.