Computer Laboratory, Exercise 1 and Discussion.

Objective: Become familiar with the use of Computer Learning Laboratory, and comfortable with use of Excel or Quattro Pro spreadsheet program.

1. Program a spreadsheet to generate a density-independent population growth function with a difference equation model.

A. Set up the problem with two cells containing N_0 and R so that you can change these cells and have the entire model change.
B. Set up the first two rows of the population model.
C. Copy the second row down to project the population in time.
D. Graph the resulting population size.
E. Learn how to change the marker style and line width, and their colors.
F. Learn how to name cells so that you can use the name to program the value, not the cell address.
G. Learn how to format cell contents.

2. Change the value of *R*. How does the graph change? Hint: watch the scale of the *y* axis -- it may be changing, but the graph appears to not change!

3. Graph multiple populations with different values of R for each. What happens when the values of *R* are very different?

4. Add to the model another column that provides the solution of the differential equation model. Compare the differences in the predictions of these 2 models. What is the difference between difference and differential equation models in biological terms. Think of examples of populations where each is appropriate.

5. What happens if you make *R* a random variable that changes each year, e.g., make *R* a function of weather, where weather is totally random each year. A reasonable model is to generate *R* from a beta distribution. To generate a beta variable in Quattro Pro with parameters α and β, you use the `@betainv(@rand, alpha, beta)` function. The betainv function returns the value of a beta variate with parameters α and β given a probability between 0 and 1. By making the probability a random uniform variable, we are able to generate a random beta variate. I have provided an example in the `exponent.wb2` spreadsheet. For the beta distribution,

mean = $\frac{\alpha}{\alpha + \beta}$, variance = $s^2 = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}$, mode = $\frac{\alpha - 1}{\alpha + \beta - 2}$ for $\alpha \geq 1$.

Given a mean (μ) and standard deviation (s), $\alpha = \frac{-\mu(\mu^2 - \mu + s^2)}{s^2}$, and $\beta = \frac{(\mu - 1)(\mu^2 - \mu + s^2)}{s^2}$.

6. The Excel or Quattro Pro function @norminv(probability, mean, SD) can be used to generate a

normally distributed random variate. Modify the `exponent.wb2` spreadsheet to include 3 populations where *R* is modeled with the same mean as the deterministic populations. Explore what happens when you use different standard deviations. Is the expected population size for these stochastic models the same as the deterministic model? Is the variance of this simulated population approximated well by Equation 1.8 of Gotelli (1998:15) taken from May 1974 (Ecosystem patterns in randomly fluctuating environments. Progress in Theoretical Biology 3:1-50)?

7. If you've been reading ahead, can you program a logistic growth function?

For the advanced Excel user.

We will not do advanced Excel programming in this class, but for those of you that want to obtain some experience with this capability, I've provided the following example, taken from Bradley (2002a, b).

One of the limitations of Excel is that the package does not have an easy way to generate random Poisson variables. To generate a random Poisson variable with a mean μ, you just need to take the product of random uniform (0, 1) values until the product is $<\exp(-\mu)$, where exp is the exponential function. The following Visual Basic code will perform this task:

```
Function ranpoi(Mean)
' Generate and return a random Poisson variable
' with mean Mean
Dim Value As Double
Dim Prod As Double
Value = Exp(-Mean)
Prod = 1
Do Until (Prod < Value)
    ranpoi = ranpoi + 1
    Prod = Prod * Rnd()
Loop
ranpoi = ranpoi - 1
End Function
```

This function is easy to program with the Do Until loop, but is unwieldy to do in a spreadsheet. To enter this function into Excel and make it available to your spreadsheet, perform the following steps.

1. Open a new Excel workbook.
2. Click on *Tools | Macro | Visual Basic Editor*
3. Make sure the Project Explorer is visible (if not, choose *View | Project Explorer*) and locate and select the filename of your workbook.
4. Choose *Insert | Module*, and in the module window, type the code.
5. When you're done, choose *File | Close and Return to Microsoft Excel*.
6. Test the function by generating at least 1000 random Poisson values, and then computing their mean and variance. Both of the mean and the variance of the random values should

be pretty close to the true mean (the value specified as an argument to the `ranpoi` function.

7. To save your new function so that you can use it in other spreadsheets, choose *File | Save As*.
8. From the *Save as type* drop-down list, choose *Microsoft Excel Add-In (*.xla)* and type a name for the file. Also note what subdirectory the new file will be saved in. Excel automatically selects the add-in folder as the location for saving the file, so you may want to change this subdirectory if you want to copy your new Add-in to a different machine.
9. To use your new add-in on a different machine, copy the XLA file to the new machine. Then, enable the new addin-in by choosing *Tools | Add-ins*. If you don't see your add-in, click on *Browse* and locate and choose your XLA file, then click on *OK*. Click on the check box to the left of your file's description, and click on *OK*. You can now use the add-in in your Excel spreadsheet.

Literature Cited

Bradley, H. 2002a. Build your own Excel function. PC Magazine 21(21):68-69.

Bradley, H. 2002b. Create your own Excel add-in. PC Magazine 21(22):74-75.