### *Lab Assignment 3*

# Discrete & Stochastic Beverton-Holt Model

*Written by*

Prof. Erin N. Bodine

*Prepared for*

Math 214: Discrete Math Modeling (Fall 2014)

Rhodes College

PreLab Assignment Due: Thursday, 9/18/2014 before class begins

Lab Assignment Due: Tuesday, 9/23/2014 before class begins

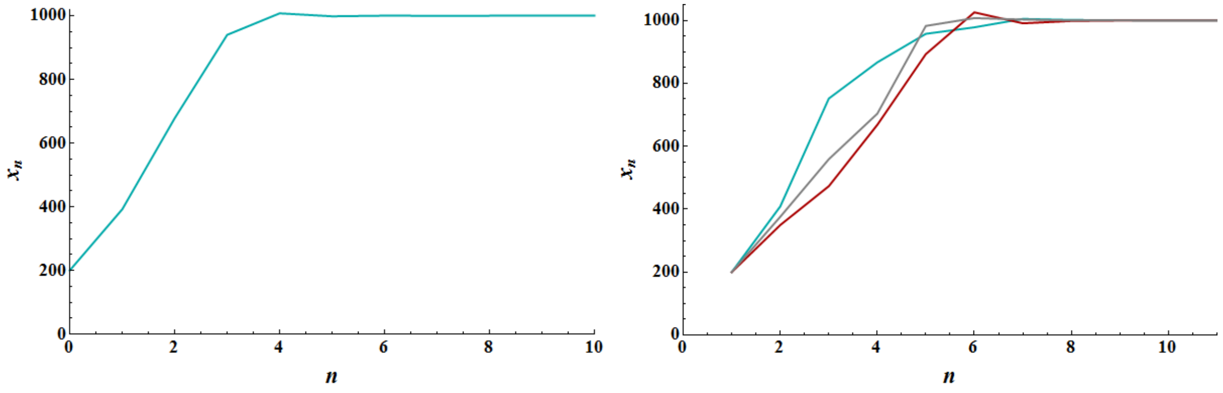
Material from the prelab is covered on the quiz on 9/18/2014.

In mathematical modeling, we can divide most models into one of two categories: deterministic or stochastic. ***Determnistic models*** are one in which if you simulate the model over and over again using the same initial conditions, you will get the same solution in each simulation. For example, the discrete logistic growth model,

$$x_{n+1} = x_n + 1.2 x_n \left(1 - \frac{x_n}{1000}\right)$$

is deterministic. If we choose to $x_0 = 200$, each time we simulate the model, we will get the same result (see Figure 1a).

***Stochastic models***, on the other hand, are ones in which there is an element of randomness such that each simulation of the model produces different results. For example, suppose we changed the discrete logistic growth model such that the intrinsic growth rate at each time step was a random number drawn from a uniform distribution from 0.5 to 1.5. In this case, the value of one of the parameters in the model changes randomly at each time step, thus causing a different result for each simulation even when every simulation has the same initial condition (see Figure 1b).



**(a)** Deterministic: $r = 1.2$, $K = 1000$, and $x_0 = 200$. **(b)** Stochastic: $r_n \in \mathcal{U}[0.5, 1.5]$, $K = 1000$, and $x_0 = 200$.

**Figure 1:** Simulations of $x_{n+1} = x_n + r x_n \left(1 - {}^{x_n}/_{K}\right)$.

We denote the stochastic discrete logistic growth model as

$$x_{n+1} = x_n + r_n x_n \left(1 - \frac{x_n}{K}\right),$$

where $r$ is a sequence of random numbers drawn from a uniform distribution between 0.5 and 1.5, denoted $\mathcal{U}[0.5, 1.5]$. Notice that each of the three simulations shown in Figure 1b eventually approach the carrying capacity and then stay there. Recall, that the carrying capacity, $K$, is the only stable equilibrium for the discrete logistic growth model if $x_0 > 0$. Thus, even though we added an element of randomness into the model, each simulation still approaches the same equilibrium.

In this lab we will be exploring the effects of making the Beverton-Holt model stochastic by changing its intrinsic growth rate, $r$, to a random number drawn from a uniform distribution. Recall that the Beverton-Holt model has two equilibria whose stability depends on the value of the intrinsic growth rate (please review Exercise 9 from the Reference Reading). If the value of $r$ changes with each time step, which equilibria is approached within a simulation? This is what we explore in this lab.

The goals of the lab assignment are to:

- Simulate a stochastic Beverton-Holt difference equation in MATLAB.

- Utilize some basic statistics to analyze how adding a stochastic element to the Beverton-Holt difference equation changes the long-term dynamics of the model.

In this lab, we will be using MATLAB to help achieve these goals. Thus, as a part of this lab you will learn how to do the following in MATLAB:

- Generate a random number from a uniform distribution.
- Write a set of nested for-loops.
- Calculate summary statistics for an array of data.

## MATLAB Skills

There are a couple of new MATLAB skills you will need in order to complete this lab. As a part of the prelab Assignment, read through the following sections. **Note, all example files are posted on Moodle.**

### Generating Random Numbers

1. Open MATLAB and navigate the current directory to the location where you will save your work.
2. The command to generate a random real number between 0 and 1 (i.e. to sample from the uniform distribution $\mathcal{U}[0,1]$) in MATLAB is `rand`. In the command window type `rand` and press enter. Do this a few times and see the different numbers you receive as output.
3. If you want to sample from the uniform distribution $\mathcal{U}[0,10]$, you would use the command `10*rand`. In the command window type `10*rand` and press enter. Do this a few times and see the different numbers you receive as output. If you want to sample from the uniform distribution $\mathcal{U}[10,25]$, you would use the command `15*rand+10`. More generally, if you want to sample from the uniform distribution $\mathcal{U}[a,b]$, you would use the command `(b-a)*rand+b`. In the command window type `15*rand+10` and press enter. Do this a few times and see the different numbers you receive as output.

   Uniform Distribution: $\mathcal{U}[a,b]$ ⇒ MATLAB Command: `(b-a)*rand+b`

4. In Lab 1, we learned how to use a for-loop to generate the terms in a sequence generated by a difference equation. For example, if we wanted to generate the terms of the sequence

$$x_{n+1} = x_n + rx_n\left(1 - \frac{x_n}{K}\right),$$

   where $r = 1.2$, $K = 1000$, and $x_0 = 200$, we could use the script `L03Ex01.m`. Everything in the script should look familiar from the previous labs with the exception of line 19. The function `ylim` is used to define the lower and upper bounds of the vertical axis. The function takes in an array with two elements: the first element is the lower bound, and the second element is the upper bound. In this case we are defining the lower bound as 0 and the upper bound as 1050.

   Execute `L03Ex01.m`. Try changing the value of $r$ and executing again. How does the resulting graph change?

L03Ex01.m

```
% Lab 1 Example 1 by Prof Bodine
% Generate terms of the sequence generated by the
%   Discrete Logistic Growth Model

% Parameter Values & Initial Condition
r = 1.2;
K = 1000;
x0 = 200;

% For-loop to generate sequence terms
x(1) = x0;
for n = 1:15
   x(n+1) = x(n) + r*x(n)*(1-x(n)/K);
end

% Generate Plot of Sequence
plot(0:15,x,'r.-')
xlabel('Time'); ylabel('Population Size');
ylim([0 1050])
```

5. The script in `L03Ex01.m` will suit our purposes if we are simulating a deterministic model, but now we wish to simulate the stochastic logistic difference equation model. In the stochastic version, we need to generate a random value for $r$ during each pass through the loop. The modified script is shown in `L03Ex02.m`. Notice that the value of $r$ need to be defined within the loop in order to change with each pass through the loop.

   Execute `L03Ex02.m` several times and observe the differences in the graphical output of each simulation.

L03Ex02.m

```
% Lab 1 Example 2 by Prof Bodine
% Generate terms of the sequence generated by the
%   Stochastic Discrete Logistic Growth Model

% Parameter Values & Initial Condition
K = 1000;
x0 = 200;

% For-loop to generate sequence terms
x(1) = x0;
for n = 1:15
   r = (1.5-0.5)*rand + 0.5;
   x(n+1) = x(n) + r*x(n)*(1-x(n)/K);
end
```

```

% Generate Plot of Sequence
plot(0:15,x,'r.-')
xlabel('Time'); ylabel('Population Size');
ylim([0 1050])
```

## Nested For-Loops

Each time you execute the script file `L03Ex02.m` it runs **one simulation** of the stochastic logistic difference equation model. Each simulation will be slightly different due to the stochastic parameter $r$. Since every simulation produces a different sequence it would be nice if we had a way to describe the "average" or "expected" behavior of the model. We will ultimately use summary statistics (think "calculating mean and standard deviation") to do this, but in order to calculate summary statistics we first need to run many simulations. When there is a process we want MATLAB to repeat over and over again, we put it in a loop. Thus, to generate many solutions we will use a for-loop where each pass through the loop generates one solution. To generate one solution, we are already using a for-loop. This means we will have one for-loop nested inside another for-loop.

`L03Ex03.m`

```
% Lab 1 Example 3 by Prof Bodine
% Runs N simulations of the Stochastic Logistic Growth Model
N = 100;

% Parameter Values & Initial Condition
K = 1000;
x0 = 200;

% For-loop to run N simulations
for i = 1:N
  x(i,1) = x0;

  % For-loop to generate sequence terms
  for n = 1:15
    r = (1.5-0.5)*rand + 0.5;
    x(i,n+1) = x(i,n) + r*x(i,n)*(1-x(i,n)/K);
  end
end

% Generate Plot of Sequence
plot(0:15,x,'r.-')
xlabel('Time'); ylabel('Population Size');
ylim([0 1050])
```

1. The script `L03Ex03.m` runs $N = 100$ simulations of the stochastic logistic difference equation model. Execute `L03Ex03.m` several times and observe the difference in the graphical output of each simulation.

   Let us take a moment to understand everything contained within this script.

   (a) The outer loop starts on line 9 and ends on line 18. Notice the index for the outer loop is $i$, and the loop run $N$ times. Since $N = 100$, the outer loop runs 100 times.

   The inner loop, which is nested within the outer loop, starts on line 13 and ends on line 17. Notice the index for the inner loop is $n$, and the loop runs 15 times.

   (b) In `L03Ex01.m` and `L03Ex02.m` the array `x` was one-dimensional; that is, it only had one row of elements. The array `x` in this script is two-dimensional. Here, each row contains the sequence of values generated from one simulation. Each column in the array corresponds to one time step. Since there are 15 time steps, each simulation generates a sequence with 16 terms ($x_0,\ x_1,\ \ldots,\ x_{15}$), and thus there are 16 columns in the array `x`.

   To access the element of `x` in generated in the $n^{th}$ time step of simulation $i$, we need to access the $i^{th}$ row and $n^{th}$ column of `x`. In Matlab the code to do this is `x(i,n)`; the first argument indicates the row number, and the second argument indicates the column number.

   Compare line 11 in `L03Ex03.m` (this code) with line 10 in `L03Ex02.m` (the previous example code). Both lines are doing the same thing; each line is setting the first value in of the sequence for the current simulation to `x0`. The difference is that in the previous code there is only one simulation and thus `x` only has one row, but in this code there are 100 simulations and thus we must save the initial condition in the row corresponding to the current simulation, `i`.

   Likewise, compare line 16 in `L03Ex03.m` (this code) with line 13 in `L03Ex02.m` (the previous example code). Do you understand how both lines of code are essentially preforming the same task, but in this code we must account for the which simulation we are currently running? If you do not understand, now is the time to stop and ask your instructor.

   (c) Notice that the plot command on line 21 in `L03Ex03.m` (this code) is no different from the plot command on line 17 in `L03Ex02.m` (the previous example code). However, when the plot command is executed in this code, it generates more than one curve on the plot. In this code, Matlab is creating one curve for each row in the array `x`.

## Summary Statistics

Now that we have many simulations of our model, we would like to summarize what all these different simulations are telling us. We will use some summary statistics to help us. Basic summary statistics can be grouped as

- Measures of central tendency (giving some sense of the central value of a data set).
- Measures of dispersion (giving a measure of spread out the data set is).

As a measure of central tendency we will calculate the mean value at each time step over all the simulations. Thus, we wish to take the mean value of each column of the array `x`. In Matlab , if you have a two-dimensional array `x` with $N$ rows and $M$ columns, the command `mean(x,1)` calculates the mean value of the elements in each column

resulting in an array with 1 row and $M$ columns, and the command

`mean(x,2)` calculates the mean value of the elements in each row

resulting in an array with $N$ rows and 1 column.

As a measure of dispersion we will calculate the standard deviation at each time step over all the simulations. The higher the standard deviation, the more dispersed the data around the mean. In Matlab , calculating the standard deviation is similar to calculating the mean. For a two-dimensional array `x` with $N$ rows and $M$ column the command

`std(x,1)` calculates the standard deviation of the elements in each column

resulting in an array with 1 row and $M$ columns, and the command

`std(x,2)` calculates the standard deviation of the elements in each row

resulting in an array with $N$ rows and 1 column. Since we would like to calculate the standard deviation at each time step, we will calculate the standard deviation of the elements in each column.

`L03Ex04.m`

```
% Lab 1 Example 4 by Prof Bodine
% Runs N simulations of the Stochastic Logistic Growth Model
N = 100;

% Parameter Values & Initial Condition
K = 1000;
x0 = 200;

% For-loop to run N simulations
for i = 1:N
  x(i,1) = x0;

  % For-loop to generate sequence terms
  for n = 1:15
    r = (1.5-0.5)*rand + 0.5;
    x(i,n+1) = x(i,n) + r*x(i,n)*(1-x(i,n)/K);
  end
end

% Calculate the mean value at each time step
y = mean(x,1);
z = std(x,1);

% Generate Plots
```

```
25 subplot(2,1,1);
26 plot(0:15,x,'r.-'); hold on; % Plot of sequence
27 plot(0:15,y,'k-'); hold off; % Plot of average sequence
28 xlabel('Time'); ylabel('Population Size');
29 ylim([0 1050])
30
31 subplot(2,1,2);
32 plot(0:15,z,'k.-'); % Plot of standard deviation
33 xlabel('Time'); ylabel('Standard Deviation');
```

1. The script `L03Ex04.m` runs $N = 100$ simulation of the stochastic logistic difference equation model, plots each simulation and the "mean" of the simulations, and plots of standard deviation of the simulations over time.

   Let us take a moment to understand everything contained within this script.

   (a) Lines 1–18 are the same as in `L03Ex03.m`.

   (b) In line 21 we calculate the mean value at each time step over all the simulations. This means we are calculating the mean value of each column of `x`.

   In line 22 we calculate the standard deviation at each time step over all the simulations. This means we are calculating the standard deviation of each column of `x`.

   (c) In lines 25–31 we have are creating two subplots. The first plots each of the 100 simulations in red (as in `L03Ex03.m`, the previous example code), and the mean of the simulations in black. The second plots the value of the standard deviation of the time steps of the simulation.

2. Run the script file. Each time you execute the script file it will produce a slightly different result (see Figure 2 for some example output). However, notice that the mean simulation curve looks almost identical each time the script file is executed.

   When we look at the simulation curves (in red), we can see that the most variation (or dispersion) occurs between time steps 1 and 5. There is no variation at the initial time step since all simulations start with $x_0 = 200$. Eventually all the simulation approach the carrying capacity $K = 1000$. Most simulations are very near carrying capacity by time step 10, and thus we see that the standard deviation is very close to zero by time step 10.

## PreLab Assignment

Suppose you want to model the gene flow of alleles from a continent population to an island population. You decide to use the continent-island model,

$$q_{t+1} = (1 - m)q_t + m\hat{q},$$

but the migration rate ($m$) each year is random. After some observation, you determine that the migration rate each year can be drawn from the uniform distribution $\mathcal{U}[0.01, 0.5]$.

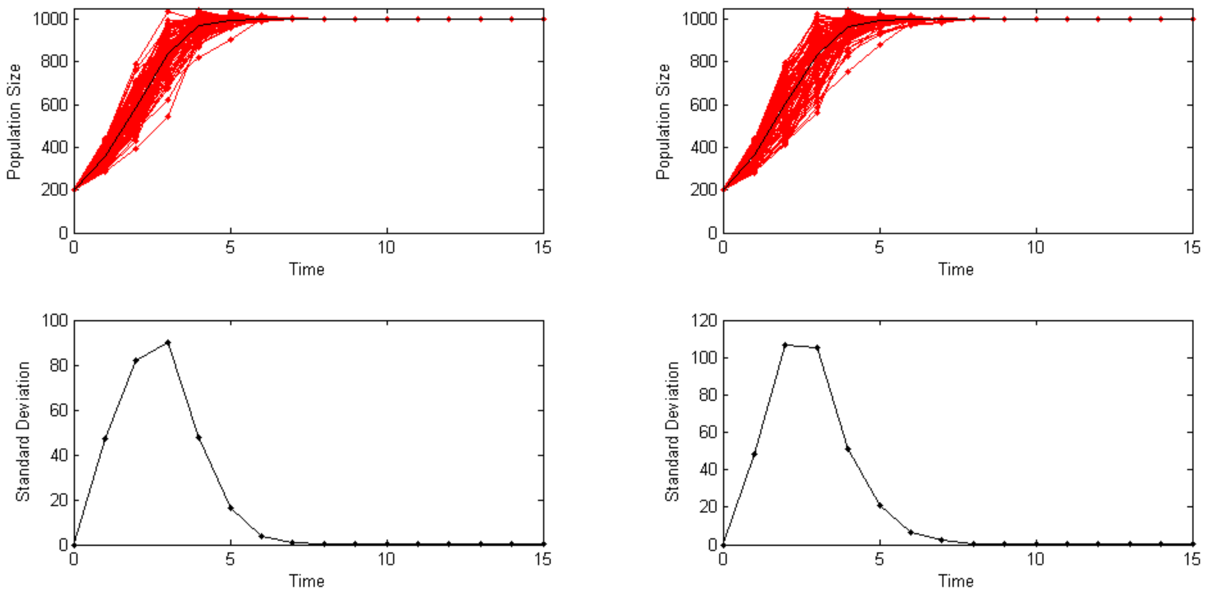


**Figure 2:** Plot produced from `L03Ex03.m` script.

1. Write a script file similar to `L03Ex04.m` which runs 100 simulations of the continent-island model with the stochastic parameter $m$ drawn from the uniform distribution $\mathcal{U}[0.01, 0.5]$, plots each simulation and the mean of the simulations, and plots the standard deviation of the simulations over time. Assume that the at the initial time 100% of the continent population as the allele, and 0% of the island has the allele. Generate solutions over 30 time steps. Save your script file as `PLab03_YourLastName.m`.
2. Execute your script file 3 times. Each time you run your code, save the resulting figure. Use the filenames `PLab03a_YourLastName.jpg`, `PLab03b_YourLastName.jpg`, and `PLab03c_YourLastName.jpg`.
3. Submit your script file and your 3 image files on Moodle. There is nothing you need to report in the textbox.

# Lab Assignment

Suppose you are modeling a population whose environmental carrying capacity is 1000 individuals. The current size of the population is 200 individuals. The population can be modeled with the discrete Beverton-Holt model, but due to varying environmental conditions, the intrinsic growth rate of the population can vary wildly from year to year. It was found that the yearly intrinsic growth rate can be drawn from the uniform distribution $\mathcal{U}[0.2, 3]$.

The discrete Beverton-Holt model is given by,

$$x_{n+1} = \frac{r_n x_n}{1 + \frac{r_n - 1}{K} x_n},$$

where $x_n$ is the population size in year $n$, $r_n$ is the intrinsic growth rate in year $n$, and $K$ is the carrying capacity.

1. What are the two equilibria of the discrete Beverton-Holt model, and under what conditions are each stable? *Hint: Review Exercise 9 in the Reference Readings.*

2. Write a script file similar to `L03Ex04.m` which runs 100 simulations of the discrete Beverton-Holt model with the stochastic parameter $r$ drawn from the uniform distribution $\mathcal{U}[0.2, 3]$, plots each simulation and the mean of the simulations over time, and plots the standard deviation of the simulations over time. Generate solutions over 100 years. Save your script file as `Lab03_YourLastName.m`.

3. Execute your script file several times. Save 3 of the resulting figures using the filenames `Lab03a_YourLastName.jpg`, `Lab03b_YourLastName.jpg`, and `Lab03c_YourLastName.jpg`.

4. As you run many simulations you will notice that though some simulations have the population getting close to extinction, in the end (by year 100) each population is thriving. Why do you think this happens? *Hint: Think about your answer to #1.*

5. Modify your script file so that the stochastic parameter $r$ is drawn from the uniform distribution $\mathcal{U}[0.2, 2]$. Run the script several times and save one of the resulting figures as `Lab03d_YourLastName.jpg`. What effect does this have on the simulations? Explain how what is occurring in the simulations is related to the equilibria values and the conditions under which they are stable.

6. **Submit:**
   - 1 script file (`Lab03_YourLastName.m`)
   - 4 image files (`Lab03a_YourLastName.jpg`, `Lab03b_YourLastName.jpg`, `Lab03c_YourLastName.jpg`, `Lab03d_YourLastName.jpg`)
   - Answers to questions posed in parts 1, 3, 4, and 5.