Logistic Growth of a Population App

Jacob S. Townson

March 27, 2016

1 Abstract

The classroom is becoming more of a technological advanced place with each passing year. It is becoming increasingly more common to use apps and other electronic forms of learning to be present in a class. In order to try to see this in action, our group has made an app for an Evolution and Ecology course at our college. This app was made using different probability models and theoretical differential equations in a Shiny App made in R Studio. The app is used to show/simulate a population's growth when there is a specific carrying capacity, birth rate, and death rate. The app's purpose is to help teach the biology class this lesson of a population with a carrying capacity through the use of dynamic graphs and an interesting yet easy to use interface. The app has been successful in the implementation behind it, as it is fully functional and ready for the classroom. To test it's use in a class, we have given a survey to two separate classes to test their knowledge of the subject. One class used the app and the other did not. *Insert results here*

2 Introduction

2.1 Question

Over the summer of 2015, I worked in a research group on Georgetown College's campus with Dr. Homer White and fellow student Andrew Giles. Originally, we were working on learning about Data Science, and began taking classes in a Data Science Specialization Course online. After Andrew and I learned some of the material, Dr. White recommended we work on Shiny apps, and make a successful Shiny app by the end of the summer. We were approached by a biology professor, Dr. Timothy Griffith, with a project to make a shiny app for his Evolution and Ecology class at Georgetown. We chose to do this project and made a functional and easy to use app for the class. So the question arose, how well will this app work in the classroom setting with biology students? In other words, does using an app rather than learning the full and heavy math behind the model help students more, or does it mask too much of the details?

As mentioned, our app works with the math behind an evolution and ecology course. The app is used to show/simulate a population's growth when there is a specific carrying capacity, birth rate, and death rate. Below I will explain the design and methods we used to make the app. To see our code, and more details on how we did our work, feel free to visit our GitHub repository for this project at https://github.com/obewanjacobi/shinyBio/tree/master. Or to see the app itself, visit https://obewanjacobi.shinyapps.io/logistic.

2.2 Significance

Using technology in the classroom has been an up and coming subject in the past few years. Schools all around the US and even out of the country have begun bringing in new tech to try and help enhance the learning experience to make it easier for students. So, the reason that my question matters is because if apps like these are easy to make, if they are helpful, then we can start integrating them into classes more regularly. Students wouldn't be required to try and understand the gory details of a subject that don't apply to them, unless they really want to. The teacher can easily regulate what is shown at one certain time versus another. When asked, future teachers prefer having apps because it helps to get all of the students involved in the learning process, as well as organize thoughts to help manage the class and how they learn.

Studies show especially for the subject of evolution and ecology (the course that this app was made for), because understanding the theory behind many things requires the conceptualization of some fairly difficult mathematics, biology students have difficulty learning it. This is referenced in the article *Creative Education* (2011) when they say "...population dynamics is a complex branch of population ecology that has an essentially quantitative nature. The effective assimilation of this topic should consider basic aspects of population theory, which involves the conceptual understanding of mathematical models." People are looking for new ways to teach this subject, and many others as well. Using an app like this one could make a huge difference to students everywhere. Other studies, like Scott McDaniel's article in *EScholarship*, show that the entrance of apps and applets into the classroom have actually increased results on tests. So making these apps more integrated into the way we teach could actually help students learn more easily.

2.3 Population Biology Overview

Will work more on this after my conference with Dr. Griffith

3 Methods

3.1 Tools

Before going into the design and models used in the app, one must first understand what R and RStudio are. R is a coding language and environment, mostly used for statistical purposes. I won't go into the entire history of how the language was made, but it is based on of the S language. When I say that it's a language and environment, I mean that it is a language in that R is a type of coding language, and it's an environment because it is a system of tools rather than a group of seemingly unrelated tools as is the case with some other languages. It has a large range of graphical, and many different kinds of statistical abilities, thus why it is extremely useful for this project.

RStudio is the IDE, or integrated development environment for R. What that means is that RStudio can be used to make R code and run it, while also helping along the way by attempting to help correct errors when they are made. This is where our group did all of our coding with R to make the app.

The other tools that were essential to our project were git and GitHub. Git is a version control system, which means that it is a method used for keeping track of the versions made of software. We used git to handle when we would make changes to our app overtime. Having version control is important so that if something went wrong, we could trace it back to a certain version that we had made.

GitHub is used for similar things, however it has a nicer user interface, and is on the internet for everyone to see. So when we would make an updated version of the app and commit it using git, we would then "push" our changes to GitHub so that the rest of the group could get the newest version of the app. This helped so that the entire group was on the same page of where the app was in development. If you would like to see the GitHub repository with all of our work, please visit https://github.com/obewanjacobi/shinyBio/tree/master. Feel free to make an account as well, that way you can comment on any of our work by using the "Issues" button on GitHub. This helps us so that we can make any necessary changes as soon as possible.

3.2 Mathematical Modeling

Before one can understand the methods and reasoning behind each step of making this app, one must first understand where the math comes from that the app is based on. Our model relies on the use of differential equations and their solutions for the theoretical portion of the app, and probability and statistical models for the simulations made and run in the app. One last thing to understand before I delve into how we made the models used in the app is the notation and variables that we used for the models themselves. Let us establish the following notation:

- $n_0 =$ initial population
- $b =$ unconstrained birth rate (or the maximum birth rate)
- $d =$ unconstrained death rate (or the minimum death rate)
- $m =$ carrying capacity (max population that an environment can safely sustain)

This next set of variables are dependent on time. The reason we have need these variables is for the probability model and the simulation of the population that will be run. There is a divide between the previous and following notations because as the following have to do with the probability model, the previous are important to the deterministic model for the theoretical population equation.

- $n(t) =$ population at time t
- $L_t =$ number of litters at time t
- $S_t^i =$ size of the i^{th} litter born at time t. These are independent of each other and of L_t
- $B_t =$ number of births at time t
- $b_t =$ expected value of birthrate at time t. Defined as $b_t = \frac{E(B_t)}{n}$
- $d_t =$ expected death rate at time t. Defined as $d_t = d + max(0, b(\frac{n}{m} - 1))$

3.2.1 Differential Equations

This section will explain how we found the differential equation that was used in the theoretical calculation of the population and its growth. We will begin with what we know, and expand from there using our previous definitions given.

We know that the rate of population growth is going to be the the number of rabbits times the birth rate at that specific time minus the death rate at that specific time. To put this in terms of math, we would write it as $\frac{dn}{dt} = nb_t - d_t$ which simplifies to be

$$\frac{dn}{dt} = n(\frac{b-d}{m})(m-n) \tag{1}$$

by our model. To solve this theoretical equation, we must use knowledge of differential equations. From here we will split into 3 cases, when $n_0 = m$, $n_0 > m$, and $n_0 < m$. To start, take note of the almost trivial answer when $n_0 = m$. Here, based off of our differential equation we made, we can note that we get that $\frac{dn}{dt} = 0$. This tells us that the theoretical equation is a constant, which makes sense considering that if a population starts off at it's carrying capacity, theoretically it wouldn't grow (even though this isn't necessarily true, which we make note of with our simulation).

Next, we must look at our other two cases, $n_0 > m$ and $n_0 < m$. For each we want to move all of the n values to one side of the equation to make it easy to integrate. This leads us to get

$$\int \frac{dn}{n(m-n)} = \int \frac{b-d}{m} dt = \frac{b-d}{m} t + C \tag{2}$$

where C is some constant that we will find later. So now we want to integrate the left side of the equation. In order to do this, we will need to use partial fractions to simplify it to make it easy to integrate. After doing this, we get that

$$\int \frac{1}{mn} + \frac{1}{m^2 - mn}) dn = \frac{b-d}{m} t + C \tag{3}$$

When we integrate this simpler integral and apply some natural log rules, we get that

$$\frac{1}{m}\left(\ln\left(\frac{n}{m-n}\right)\right) = \frac{b-d}{m} t + C \tag{4}$$

Now we want to find what the C value is. In order to do this, let $t = 0$ so $n = n_0$. After solving equation number 4 for C, we get that $C = \frac{1}{m}(ln(\frac{n_0}{m-n_0}))$. This gives us that

$$\frac{1}{m} ln(\frac{n}{m-n}) = \frac{b-d}{m} t + \frac{1}{m} ln(\frac{n_0}{m-n_0}) \tag{5}$$

Now we can simply solve the equation for n. This mostly just involves some elementary algebra, which leads us to get that

$$n = \frac{m}{1 + (\frac{m-n_0}{n_0}) \mathrm{e}^{(d-b)t}} \tag{6}$$

if $m > n_0$. This is our equation for the theoretical model. However, we must realize that you cannot take the natural log of a negative number hence our third case of $n_0 > m$. To account for this we must go back to equation four. Here, we notice that because $n_0 > m$ we get a negative value inside the natural log. Thus we must take the absolute value of the number inside the natural log, then go through the steps as we did in the second case. After doing these algebra steps, we find that

$$n = \frac{-m}{\frac{n_0 - m}{n_0} \mathrm{e}^{(d-b)t} - 1} \tag{7}$$

if $n_0 > m$. This is our model for the theoretical graph.

After some research, our group found that this model follows the theoretical equation for biologists as well. Their equation was

$$N(t) = \frac{K}{1 + (\frac{K-N(0)}{N(0)})e^{-rt}} \tag{8}$$

found by Roughgarden(1979), Emlen(1984), and Neuhauser(2000). This relates to our equation because $N(t) = n$, $K = m$, $N(0) = n_0$, and $r = b - d$.

3.2.2 Probability Model

In order to run simulations to convey to the students that the deterministic model isn't always true, we need a probability model for the simulation to follow. This is where the variables defined before as being reliant on time come into play. They are reliant on time is because at each given time interval, the app will run a simulation based on what the previous time values were. We will get more into the programming of this later, but for now, here is the math behind our model.

To start, let's make two more variables just to help the overall calculations: let

$$s_t = E(S_t^i)$$

and

$$l_t = \frac{E(L_t)}{n} \tag{9}$$

We know that the number of births at time t will be the sum of the sizes of all of the litters, so $B_t = \sum_{i=1}^{L_t} S_t^i$. We call the expected birth rate $b_t = \frac{E(B_t)}{n}$, as we defined before. We can assume then that the expected birth rate is the expected litter rate times the expected size of the litters, or $E(B_t) = E(S_t^i) \times E(L_t)$. This can be proven as well using what we know about each variable and simple algebra. When this is simplified becomes

$$b_t = s_t \times l_t \tag{10}$$

The s_t and the l_t are what we simulate using the rpois() function in R, which as previously stated, we will address more later. After some research, we found that the average litter sizes for rabbits was 8, so we let the

$$s_t = \frac{8\sqrt{b_t}}{\sqrt{b}} \tag{11}$$

and

$$l_t = \frac{\sqrt{b_t}}{8} \times \sqrt{b} \tag{12}$$

in order to be able to calculate the the number of births at time t for the simulated numbers. Then to calculate the number of deaths for each time interval for the simulation, we have our d_t function that we defined previously that depends on how close the population is to the carrying capacity,

$$d_t = d + max(0, b(\frac{n}{m} - 1)) \tag{13}$$

By simply adding the values of b_t and d_t together we obtain our simulated data.