Logistic Growth of a Population App

Jacob Townson

October 7, 2015

Research Question

Over the summer of 2015, I worked in a research group on Georgetown College's campus with Dr. White and Andrew Giles. Originally, we were working on learning about Data Science, and began taking classes in a Data Science Specialization Course online. After learning 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 found a project to do with the help of Dr. Griffith, and made a Shiny app for his evolution and ecology course at Georgetown. 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. The app's purpose is to help teach an evolution and ecology class. 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.

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. 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 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 easier.

Methodology

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 off 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.

App Design

Before going deep into the app design, I invite you once more to experience the app for yourself so that the design process makes more sense. Again, the app can be found at https://obewanjacobi.shinyapps.io/logistic. Also, this proposal will be connected to the app under a tab labeled "About", so if you would like, you may read the proposal and follow along in the app yourself!

Sidebar

- **Time**: This lets you choose the time value displayed on the x axis of the graph under the *Population Size* tab.
- **Initial Population**: This lets you choose the initial population, or the y intercept under the *Population Size* tab.
- **Birth Rate**: This lets you choose the maximum birth rate for the population. The effective birth rate will change according to how close the population gets to the carrying capacity.
- **Death Rate**: This lets you choose the minimum death rate for the population. The effective death rate will change according to how close the population gets to the carrying capacity.
- **Carrying Capacity**: This lets you choose the carrying capacity for a population. It could be thought of as the amount of resources a population has to survive/thrive. Note, carrying capacity option is removed if death rate is above birth rate because a carrying capacity doesn't affect this given situation.
- **Setting the Seed**: This check box gives the user the option to set the seed of the simulated values. By setting the seed, the user can get the same outcomes for every time the *Simulate* button is pushed while the seed is set to that specific value. When the box is unchecked, the option goes away and the simulation is random again.
- **Simulate**: This button runs the app with a new simulation with the given input parameters.

Population Graphs

This section is the main portion of the application. This is the only tab available before the **Simulate** button is hit. This tab displays the graphs used to convey information on the population growth. Before the **Simulate** button is hit, the graph displays only the theoretical results. These theoretical results are what would happen "on average". This means that if we can imagine an infinite amount of populations were observed, the average would look like our theoretical results. If the *Size* button is chosen, then the population size graph will be shown. This graph shows the growth of a population over an arbitrary amount of time as it approaches it's carrying capacity. Here the population is conveyed by the red line and the carrying capacity is shown in blue. The next option under this tab is to plot the growth rate. Here the growth rate is plotted against either time or the population itself. This switching between what is plotted on the x axis can help to teach students how the growth rate changes, and makes it easier to understand through a look at it from both perspectives. Similar to this structure, the next option under this tab is to plot the per capita growth rate. Again, the user can choose what to put on the x axis, either time or population to give the user an easier way of understanding depending on their perspective.

The only change that is made under this tab by hitting the **Simulate** button is that a simulated population is displayed on the *Size* graph. This is conveyed by a black line. This simulated line is made using our model and with the help of the rpois() function in R. So this creates a possible outcome for a single population to show what it could look like over a certain given time. This is not the same thing as the theoretical graph, as the theoretical graph shows the "average" situation, whereas the simulated shows a possible situation for one group. For more information on the model and how the simulation works, skip down to the **Model Description** Section of the Help File.

Field

This portion of the application displays a model for the field, which is a hypothetical situation for the simulated data shown in the *Population Graphs* tab. Here there are two plots next to each other, with an animation bar above them. The two plots are used to show the total population at the time that is shown in the animation slider. When the play button is hit below the slider, the slider will move from time value to the next time value for every second or so. Whatever the simulated population is at that time, that is the total number of dots on the plots. The plot on the left shows the adults in the field foraging for food for their children.

The plot on the right shows the newborn rabbits in what is cleverly called the "Warren". Here, the litters are grouped closely together using a plotting system we made involving keeping litters close together using the rnorm() function. In the field function, you may also notice the color of the graph changing over time. If there is a carrying capacity used in the simulation, then the color of the graph will change. The idea is that if the rabbits are far away from the carrying capacity, then the field will be more green. On the other hand, if the population is above, close to, or on the carrying capacity, they will naturally be running out of resources, making the field a more brown and barren color. This makes the app more interactive, and maybe fun for the students to learn the concepts, and actually get the feeling that they are seeing the changes in the environment.

Below both graphs, a table will be displayed if any rabbits were born and added to the Warren. The table will show the number of litters at that time, and also show the average size of the litters born.

One final feature of this tab happens when the simulated population dies out. You as the user will have to find out what happens then for yourself.

This section of the app is important because we were tasked with giving visual representations that could be fun for the students, but also educational. One thing we were assigned is to attempt to make the students learn how to read graphs better. Because of this, we not only made the different graph settings in the population section, but we also made this section to make the students read a graph in which they can watch the population grow and decrease. This makes the experience more interactive and much more interesting for students.

Graveyard

This portion of the application is similar to the field in that it is a model used to present hypothetical information. It uses the simulated population to show the deaths at each time. It is uses a similar format to the field in that it uses an animation slider to show the graveyard over time.

This tab also gives the rabbits a cause of death, again, to give the user a feeling of the whole situation being more real. The deaths are divided into three categories, death by lawnmower, death by foxes, and death by malnutrition. The death by lawnmowers is more likely to happen when the population is small, and the death by foxes and malnutrition's likelihood go up as the population grows. The app also has an info link to tell the user this explanation. There is also a table to show the population and the carrying capacity at the bottom, just to remind the user to give more of a reference on the amount of deaths at that certain time.

One final feature of this tab happens when the simulated population dies out again. You as the user will have to find out what happens then for yourself.

Similarly to how we made the *Field* section of the graph more interactive, we did the same with this section for the same purposes. We want these sections to make the user learn how to read graphs better, and to do so, we have tried to make a fun environment for the student to learn.

Model Description

For reference, this section is heavy in mathematics. I have attempted to simplify the wording and steps as much as possible, however there are some explanations that cannot change due to their importance in the topic. I advise the less technical readers to simply skim this section. Skimming this section won't take away too much from the main idea from the project. If you can attempt to follow the dense math parts though, it is actually quite interesting.

Variables and their Relations

Our group made this model hoping to make it in such a way that we could relate the simulated graph to the theoretical one in an abstract way. So to start, let us establish the following notation:

- $n(t) =$ population at time t
- $n_0 =$ 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)
- $L_t =$ 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

Now that these variables have been established, we can attempt to relate them together. To start, let's make two more variables just to help the overall calculations: let

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

and

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

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 can call the expected birth rate $b_t = \frac{E(B_t)}{n}$. 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. When this simplified becomes

$$b_t = s_t \times l_t$$

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

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

and

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

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 made it a function of how close the population was to the carrying capacity so that

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

When adding these two things together, we got our simulated data.

The Theoretical Differential Equation

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

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

by our model. To solve this theoretical equation, we must use knowledge of differential equations. To start, 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

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

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 $\int (\frac{1}{mn} + \frac{1}{m^2 - mn}) dn = \frac{b-d}{m} t + C$. When we integrate this simpler integral and apply some natural log rules, we get that

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

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

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

Finally we need to make n the outcome of the equation. This mostly just involved some elementary algebra, which lead us to get that

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

if $m > n_0$. This was our equation for the theoretical model. However, we realized later that you cannot take the natural log of a negative number, so to account for this, we put everything in the natural logs to be the absolute value. To account for this, we found that

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

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

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

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 $r = b - d$.

Measurement

The final part of this project will take place in the spring of 2016 with the help of Dr. Griffith. We are currently working on making a before and after survey for the students of the class to take. This survey will help to quantify how helpful the app actually was to the class. The first survey will be taken before using the app, and the students will quantify how well they feel that they learned the subject. Then Dr. Griffith will introduce the app to the class and teach them how to use it. After using the app in class, the students will be presented with one more survey to help quantify how well they understand the subject after using the app.

Using this measurement, we will be able to see if having technology in the classroom such as applications like this one can truly be helpful and create a better environment for learning, and thus answer the big question stated at the beginning of this proposal: "does using an app rather than learning the full and heavy math behind the model help students more?"

Bibliography

-Alstad, Don. *Basic Populus Models of Ecology.* Upper Saddle River, NJ: Prentice Hall, 2001. Print. This book helped us in making our model, and is reflected perfectly in our final model.

-Matloff, Norman S. *The Art of R Programming: A Tour of Statistical Software Design.* San Francisco, Calif: No Starch, 2011. Print. This book was used to help me learn about some of the coding in R.

-McDaniel, Scott N., and Lisa B. Green. "Using Applets and Video Instruction to Foster Students' Understanding of Sampling Variability." *EScholarship.* N.p., 2012. Web. 10 Oct. 2015. This article was used to help with the significance section.

-Serra, H. & Godoy, W. (2011). Using Ecological Modeling to Enhance Instruction in Population Dynamics and to Stimulate Scientific Thinking. *Creative Education*, 2, 83-90. doi: 10.4236/ce.2011.22012. This article was used to help with the significance section.

-Wickham, Hadley. *Advanced R.* N.p.: CRC, 2015. Print. This book was used to help me learn about some of the coding in R.