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 succesful 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/agiles231/shinyBio. Or to see the app itself, visit https://obewanjacobi.shinyapps.io/logistic.

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. 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. 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 numbr 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.

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.

Model Description

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 $r = b - d$.

Thanks For Using Our App!