Population Model (Coyotes vs. Rabbits)

(*Adapted from a course by the Fractal Foundation*)

Exponential Growth

Recurrence relations can be used to model population growth. For example, say we have a population of rabbits and for each new generation, the rabbits produce at a rate of double the current population. This would mean that if we started with 2 rabbits, they would reproduce to have 4 rabbits, giving us a total population of 6 rabbits.

The recurrence relation for this sequence would be given by $p_n = p_{n-1} + 2p_{n-1}$. We would have to provide an initial population, p_1.

1) Using the rule given above, find the number of rabbits in the 8th generation, p_8, if the initial population $p_1 = 2$.
   4374. They should write out the sequence: 2, 6, 18, 54, 162, 486, 1458, 4374

2) Copy your Fibonacci Mathematica program and save it as "Population." Modify your Fibonacci program to see what the population will be in the 20th generation, p_{20}. When you're modifying your program, think about the following things:
   a) What should **nmax** be? 8
   b) Do you need to define both **F[[1]]** and **F[[2]]** before the for loop? Why or why not? No, we only need **F[[1]]** because the first term is the only one given to us. The second term is defined by the rule.
   c) What needs to be changed in the for loop? The sequence definition needs to be changed in the for loop. Also, the starting value for **i** is 2, not 3.

3) Plot the population of rabbits using **ListLinePlot** in Mathematica. You can use the command **ListLinePlot[***name of sequence***]**.

This type of growth is called *Exponential Growth*, and as you can see, the population will get very large, very, very quickly.

4) Does this type of population growth make sense? Would you expect to see it in the real world? Why or why not? No, this does not make sense. In the real world, we normally don't see an explosion in the rabbit population because there are things that will decrease that population: 1) predators, 2) natural death, 3) lack of food or limited food.

The Verhulst Equation

In order to more realistically model population growth, the *Verhulst equation* was introduced in the 19th century. There are several factors that would limit the growth of a population of rabbits in the real world. First, there might not be enough food to support an enormous population. Secondly, there would be predators that would also limit the growth of the population. And rabbits have a limited lifespan, so some of them would die in later generations.

The Verhulst equation takes into account the first two factors by limiting the amount of growth and is given by the following recurrence relation:

$$p_n = p_{n-1} + cp_{n-1}\left(1 - \tfrac{p_{n-1}}{K}\right)$$

The first term, p_{n-1}, gives the current population. The second term, $cp_{n-1}\left(1 - \tfrac{p_{n-1}}{K}\right)$, gives the growth or the amount of reproduction given by the current population.

The Verhulst Equation includes something called a *carrying capacity*, given by K, which limits the amount of growth and sets a maximum total population for the rabbits. The growth is also determined by, c, the fertility factor, which would determine how much the rabbits reproduce. Each of these "parameters" would be assigned numeric values for a model.

Example: Let's say the fertility factor is $c = 0.5$ and the carrying capacity is $K = 200$, and the initial population of rabbits is $p_1 = 100$.

1) The model becomes $p_n = p_{n-1} + 0.5p_{n-1}\left(1 - \tfrac{p_{n-1}}{200}\right)$. Copy the bracket with your exponential growth code from above. Then revise it to find the population in the 10th generation, p_{20}.

2) Plot the population of rabbits using a **ListLinePlot** in Mathematica. You can use the command **ListLinePlot[***name of sequence***]**.

3) What is the difference between this plot and the plot from the previous model? This plot eventually levels off and approaches 200, but does not rise above it. The exponential model grew without limit.

The Logistic Map

The Verhulst Equation can exhibit some very complex behavior depending on the fertility factor, the carrying capacity, and the initial population (c, K, and p_1). Sometimes the behavior has a recognizable pattern, and at other times, it can be completely chaotic.

We will take a look at some of this behavior, but we will look at a simplified version of the equation that is called the "Logistic Map." The Logistic Map is given below:

$$p_n = cp_{n-1}(1 - p_{n-1})$$

1) Describe any differences you notice between the Verhulst Equation and the Logistic Map. The initial current population term is gone. K is set to 1.

2) Describe what the Logistic Map would represent in the real world. What is it modeling? Use the changes in the equation to come to a conclusion. This represents ONLY the growth in each generation because we are no longer adding the current population size. Also the carrying capacity would be 1.

3) Copy and modify your Verhulst Equation code to model the Logistic Map and find all the terms up to p_{65}. (You will need to add an extra statement at the beginning of your program to define c). Start with **c = 0.2;** and $p_1 = 0.5$. <u>Make sure to add the name of the sequence at the end of the code, so you can see the list of terms.</u>

4) Plot the population of rabbits using a **ListLinePlot** in Mathematica. You can use the command **ListLinePlot[***name of sequence,* **PlotRange ->All]**.

5) Also create a histogram of the values in the sequence by using the command **Histogram[***name of sequence*, **50]**. (Here the number 50 tells Mathematica how many bars the number range should be split into.)

6) Fill in the following table with your observations by changing the **c** value to those given in the table. Use the sequence, the list plot, and the histogram to make observations. Some of the table has already been filled in to give you some hints of what to look for.

   Pay attention to what happens to the terms in the sequence. Do they approach a certain number or numbers? You can use the histogram to see if there is more than one number that appears in the sequence frequently, and you can use the sequence to see what those numbers are.

   Are these numbers getting larger or smaller as you increase **c**?

Is the behavior of the numbers in the sequence periodic (meaning: is there a repeated pattern) or is it chaotic? How does the graph behave?

c	How Many Repeated Numbers? Values?	Repeated Numbers Increasing or Decreasing?	Graph Behavior?
0.2	1, value = 0	First example, can't say	Approaches 0
0.4	1, value = 0	Increasing a very little bit	Approaches 0
0.6	1, value = 0	Increasing a very little bit	Approaches 0
0.8	1, value = 0	Increasing	Approaches 0
1	1, value about 0.014	Increasing	Approaches 0.014
1.2	1, value 0.166667	Increasing	Approaches 0.166667
1.4	1, value 0.285714	Increasing	Approaches 0.285714
1.6	1, value 0.375	Increasing	Approaches 0.375
1.8	1, value 0.444444	Increasing	Approaches 0.444444
2	1, value 0.5	Increasing	Is always 0.5
2.2	1, value 0.545455	Increasing	Small oscillation, then approaches 0.545455

2.4	1, value 0.583333	Increasing	Oscillates, then approaches 0.583333
2.6	1, value 0.615385	Increasing	Oscillates, dies to one value
2.8	1, value 0.642857	Increasing	Oscillates, dies to one value
3	2, value about 0.7 and about 0.63 (this one hard to tell)	Splits (so increase and decrease)	Oscillates
3.2	2 repeated numbers: 0.799 and 0.513	Greater increases, lesser decreases	Oscillates
3.4	2, values around 0.84215 and 0.45196	The greater number increased, and the lesser number decreased.	Oscillates
3.6	Chaotic, no particular values repeated	--	Chaotic
3.8	Chaotic, no particular values repeated	--	Chaotic

7) Summarize what you have observed for different values of **c**.
   If c between 0 and 1, the value dies to 0. If between 1 and 2, it approaches a number greater than 0. If between 2 and 3, it first oscillates, then approaches a number greater than 0. If between 3 and 3.4, it oscillates between 2 numbers. If greater than 3.5, the values are chaotic.

8) Predict the behavior you would expect to see for **c = 3.6625** and **c = 3.8375**.
   We would expect see chaotic behavior for both of these values.

9) Try inputting the two values of **c**. What do you observe?
   For 3.6625, we see the sequence oscillating between 8 different values: 0.915572,

0.283112, 0.743339, 0.698754, 0.770945, 0.646756, 0.836744, 0.50031. For 3.8375, the sequence is oscillating between 3 different values: 0.150652, 0.491032, 0.95906.

As we can see, the Logistic Map produces some very complex, beautiful, and at times, unexpected, behavior. In fact, if we were to make a scatterplot using the values of **c** as our x-values and the values from the second column in our table as the y-values, it would produce a figure known as a *bifurcation map*.

10) Using the data in your table, make a rough scatterplot by hand.

11) Imagine we had the data for all values of **c** and could connect the points in the scatterplot. Predict what this would look like. You might want to discuss this with your group members before coming to a conclusion.

12) Look up “Logistic Map” on Wikipedia and find the actual *bifurcation map*. Is it what you expected? Why or why not?

    What do you notice? Does the map agree with the observations you made in your table? Why or why not?
    These questions and the scatterplot are pretty open-ended, but the scatterplot should be vaguely similar to the bifurcation map, which looks like this: