Name: Abdullah Shahzad
Course: CS 152
Section: B
Date: 9/29/2021

Project 4 Report:

Simulations Mapping the Probability of Extinction for Penguin in the Galapagos.

Abstract:

This project is about penguins and how likely they are to survive 201 years (an arbitrary amount) if certain factors were to change (or not). Penguins in the galapagos have a relatively stable environment to live in except for an irregular event that occurs once every few (3, 5 or, at most, 7) years called El Nino. If a year is an El Nino year, the penguin population decreases significantly. If it is not an El Nino year, the penguin population increases slightly.

For this project, I used functions, which are basically blocks of code, to map the population growth of penguins by a growth factor *rho.* The function was given an initial amount of penguins which it randomly assigned male and female genders to (with there being an equal probability of either) and put into a list. It added more penguins to that list for every year that the simulation ran, removing penguins by a negative growth factor *rho2* if it was an El Nino year. If the population in the list ever became unviable (a population of either less than 10 penguins, or a population with only male or female penguins), the function stopped the simulation and printed out the year of extinction.

Using a for loop, which is a piece of code that allows me to repeat processes, I was able to perform the same simulations thousands of times to see if my results were stable. After this, by compiling the values of extinction years in a list, I was able to graph a cumulative distribution, which is the output of this report—an easy-to-read way of seeing how likely the penguin population is to survive 201 years.

Usage Statement:

Users enter three command-line arguments to run the program. The program name, the number of simulations to run, and the difference between El Nino Years. Since the year being an El Nino year decreases the population significantly, if the third argument is a large number, more simulations will make it to 201 years and vice versa.

results on the next page

Results:

Required plots 1: Generate three plots of the CEPD for three runs of 1000 simulations for 201 years with the default parameters:

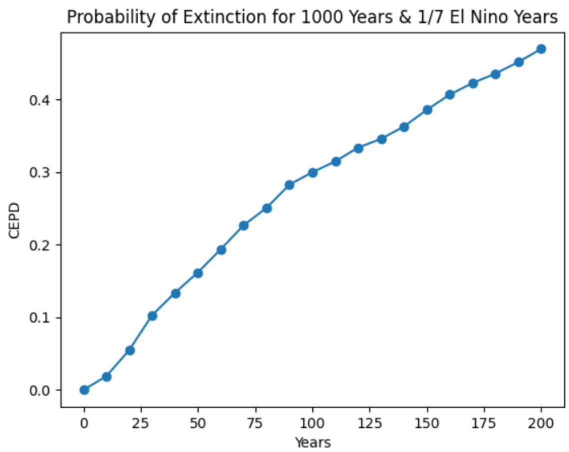

Fig 1.1

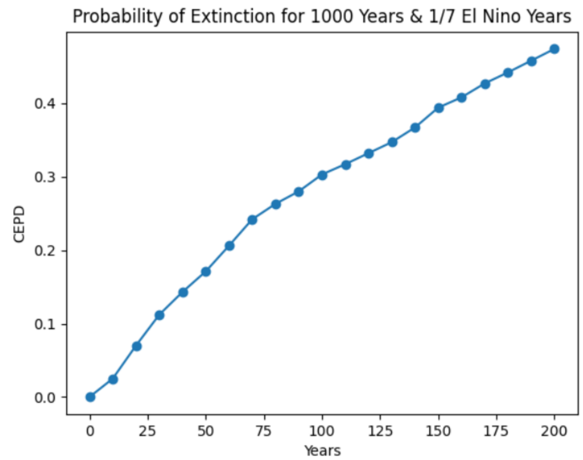

Fig 1.2

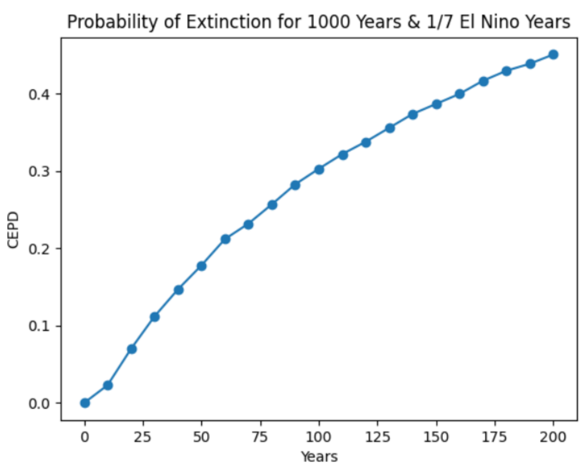

Fig 1.3

Figures 1.1, 1.2, and 1.3 are three different simulations run by the same program with identical parameters (values for calculations). They all take on an extremely similar parabolic shape with only minor differences in what generally seems to be a smooth curve with a small indent between 100 and 150 simulated years. Having a relatively smooth graph indicates that the probability of extinction at any point was approximately the same and the exact probability of extinction can be found out by taking the derivative of this graph at any point.

Required plot 2: a plot of the CEPD for the three El Nino cycle options:

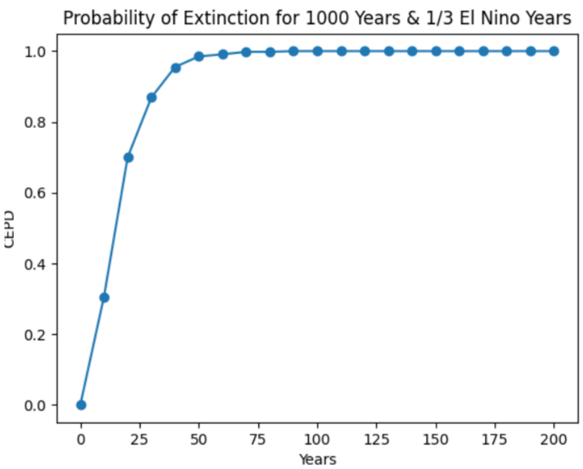

Fig 1.4. 3 year gap between El Nino Years.

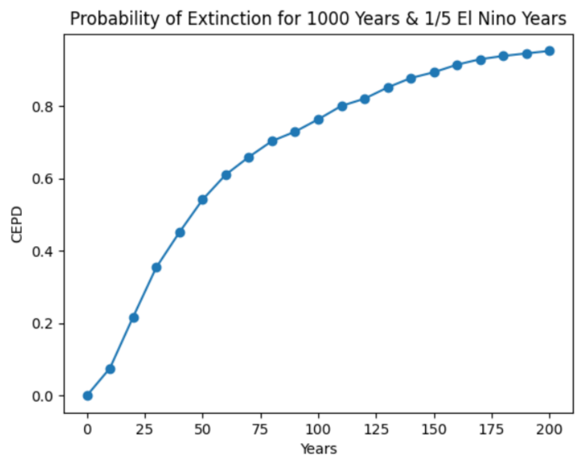

Fig 1.5. 5 year gap between El Nino Years.

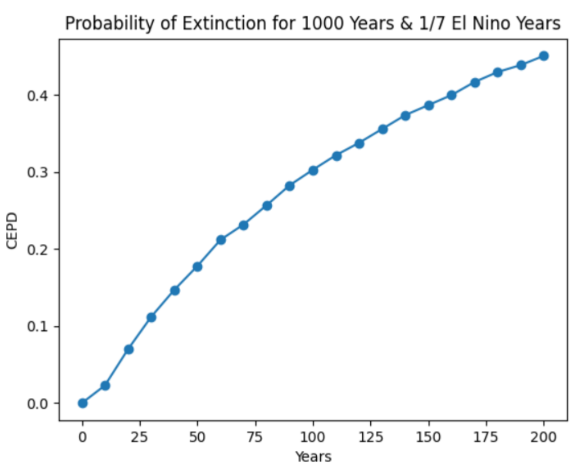

Fig 1.6 7 year gap between El Nino Years.

As you can see, the graphs of fig 1.6 and fig 1.5 are very similar. The only notable difference is that, at the start of the graph in fig 1.5, the slope is steeper. This means that the likelihood of extinction is higher when the difference between El Nino Years is 5 years compared to it being 7 years, which is intuitive. In fig 1,4 the slope is even steeper, meaning that the penguin population is extremely unlikely to survive; the graph even becomes flat at 1.0 after 50 years which means that no simulation made it past the 50 year mark before going extinct.

What we can take away from these results is that, the lesser the gap between El Nino years, the more likely the Galapagos penguin population is to go extinct.

Reflection:

Lecture concepts utilized in this project include importing and using programs from a library so that I wouldn't have to reinvent the wheel (for matplotlib and sys), using the command line to get input from the user, writing for loops and nested for loops (a for loop inside a for loop) and incrementally writing code. All of these skills have multiple uses in the real world but the overall project, which was running simulations and making predictions intrigued me the most. The method can be extrapolated to calculate the odds for anything: sports games, fluctuations in markets, population growth for humans in any region, etc. Running simulations helps visualize real outcomes and plan for them, prevent them, or facilitate them, which seems like a powerful tool.

Follow-up Questions

1. **What is the difference between the following two code snippets?**

```
# example 1
a = [5, 10, 15, 20]
for i in range(len(a)):
   print(a[i])
# example 2
a = [5, 10, 15, 20]
for x in a
    print(x)
```

In example 1, saying i in range(len(a)) is equivalent to saying i in range(4) because there are 4 items in the list, while in example 2, x refers to each value in a, starting from 5 and ending at 20.

2. **Why do we test code incrementally? Why not write all of the code and test it once?**

To make it easier to debug. If we write all the code and then test it, we'd have to look over the entire program.

3. **Why do we use random numbers in this population simulation model?**

Populations hardly have any order to them, they're supposed to be random and using random numbers is an attempt to accurately mimic that behavior as best we can.

4. **What is your favorite wild (undomesticated) animal?**

Polar Bears

Extension (optional):

1) **Use matplotlib to automate the process of creating plots:**

As soon as the user plugs in arguments into the command line, after the computer processes the simulations, a plot is made with the given parameters.

2) **In addition to the required plots, show a plot of the population levels for a single simulation.**

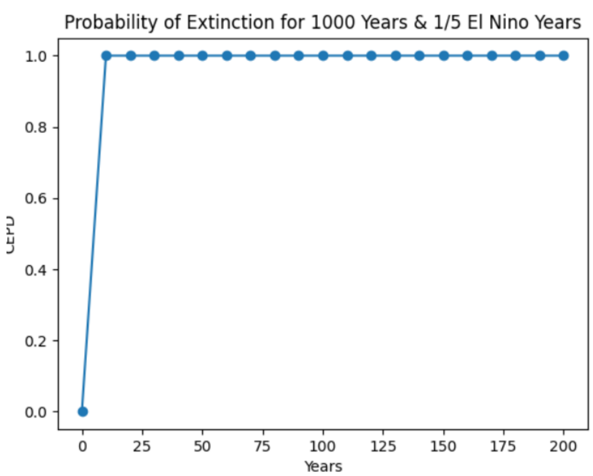

Fig 1.7

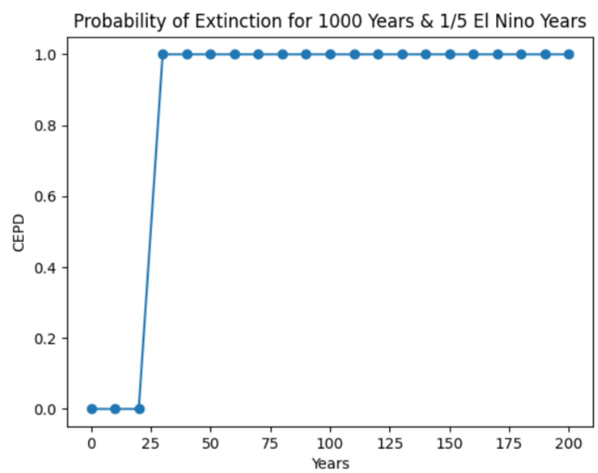

Fig 1.8

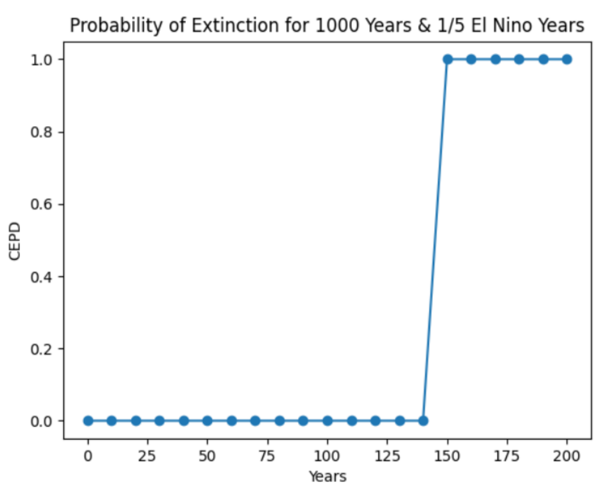

Fig 1.9

I ran single simulations 3 times: once with the difference between El Nino Years set to 3, then to 5, and then to 7. According to the simulations, if there was an El Nino year every 3 years, the galapagos penguins would go extinct in under 12.5 years. If there was an El Nino year every 5 years, the penguins would do a little better but still not make it to 25 years. If there was an El Nino year every 7 years, however, the penguins would do significantly better and survive for almost one and a half centuries.

3) **Display the CEPD data in a graph using matplotlib, or have your code write out the results as a CSV file and use a graphing program:**

The function plots the CEPD data on the y axis and the number of years on the x axis as soon as the program is run.

4) **Explore variations on the model. For example, what effect on the extinction probability do you see if you adjust the number of years in the El Nino cycle?**

I'm testing if changing the carrying capacity to 3000 modifies the 100-year extinction probability for the 5 year El Nino cycle?

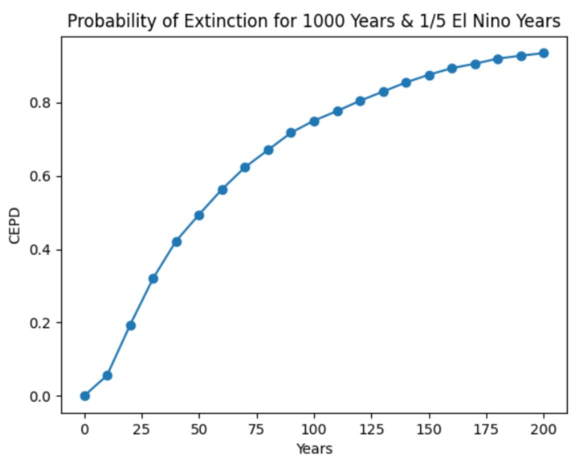

Fig 2.0 carrying capacity changed to 5000

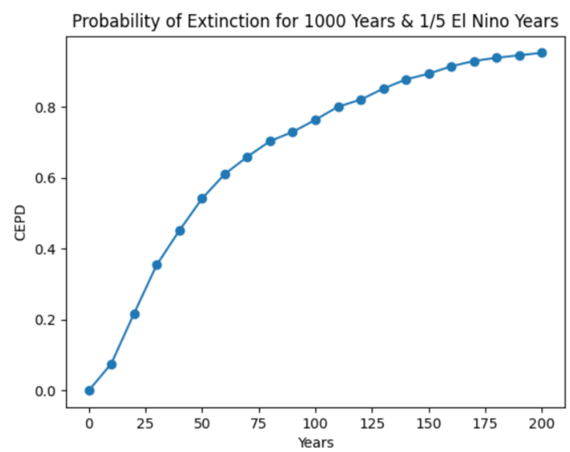

Fig 2.1 carrying capacity is 2000

Changing the carrying capacity didn't significantly change the graph for the CDEP when the difference between El Nino Years was 5. The graph did, however, become more smooth near the center (at around 100 years), which isn't a good thing because that means that more models went extinct in Fig 2.0 than in 2.1 but since that doesn't have an intuitive link with changing the max capacity, it can be ignored.

Acknowledgements: Professor Allen Harper, Giang Pham