

Pandemic Disease Spread in Elementary Class

Authors: Mohammed Siddiqui Liam O'Donnell

Abstract

For the ISYE 6644 Final Project, the group of Liam O'Donnell and Mohammed Siddiqui selected Topic 3: *Pandemic Flu Spread* which simulated a flu-like disease that spread through a class of 31 elementary school students. Patient 0 (Tommy) attends school every day, including the day he is infected (day 0). Tommy is contagious until the end of day 3, for 3 continuous days infected. The probability that Tommy infects another student in his class was given as $p = 0.02$ and all students and days are assumed to be independent. Through simulation runs using (2) different simulation approaches, the group's major findings included that the number of days for the pandemic to end (i.e., no more new infections for 3 continuous days) differed depending on (a) the spread rate and (b) immunization rate.

Background

The group was asked to simulate the same disease mentioned above with two different classroom characteristics. In the first case (referred to as the base case) no immunized students were present. In the second case (referred to as case 2), a random number of students were immunized with $p = 0.50$. Due to the group's multi-tooled approach to modeling the spread of the disease, the *main findings* portion of the paper is divided into (4) sections. The first section discusses the key assumptions used for both simulation languages along with the applications area. Following key assumptions, the group discusses its findings from using Excel and Python to model the spread of the disease. The final portion of the *main findings* section discusses any differences between the two groups of results. All code/files used in this project are included in the zip file submitted.

Main Findings – Assumptions

There were several key assumptions that the group carried throughout the project's duration. These assumptions were driven by group discussion and instructions given by the ISYE 6644 staff. The group assumed a continuous school year with the only off days being Saturday and Sunday. Including 3-day weekends and extended breaks would cause the spread to terminate due to the contagious duration of 3 days. The second assumption the group made was that any student who became infected continued to attend school for the proceeding days. For example, if student A became sick on day 1, that student would

attend school on days 2 and 3. Another assumption was that students could not be infected more than once.

Main Findings – *Excel Simulation*

The group’s initial simulation package was Excel due to the group’s familiarity with its functions. The simulation was built out using a 220x31 matrix. 220 rows to represent the number of days in a school year and 31 students who attended school on each of those days. Given the number of students infected with the disease, s , each non-infected student, i , had a 0.02 chance of being infected. The excel RAND() function was utilized to determine if a student was infected or not. If the RAND() function returned a value $0 < \text{RAND} \leq 0.02$, the student became “infected”. This brought up one of the key limitations of using excel. By using RAND() for each i , the infection rate was independent of the number of students that had the infection. To put it a different way, the rate at which students were infected if there was 1 student infected, was no different than if there were 10 students infected at once. As a result, the average pandemic length for 20 simulation runs continued for the entire school year. This is shown in **Figure 1**.

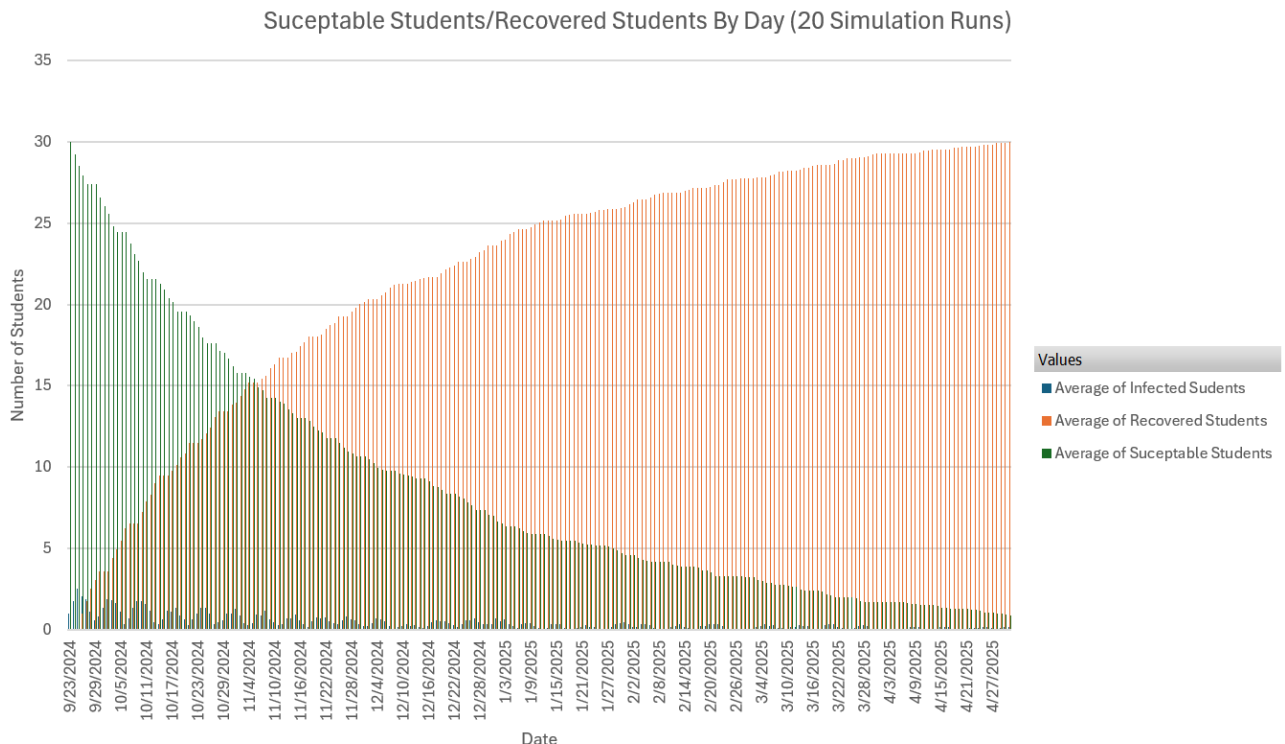


Figure 1: Average Number of Infected/Recovered Students over a school year using Excel simulation

Another indication that the Excel simulation did not fully capture the problem statement was that the average value of individuals infected on day 1 & day 2 were 1.75 and 2.5 respectively. The day 1 average value was significantly higher than the calculated expected value of:

$$E[X]_{day1} = 30 \cdot ((1) \cdot 0.02) = 0.6$$

The group was unable to simulate the test case where each student had a 50% chance of being immunized. This test-case will be discussed in the Python main findings portion of the report.

Main Findings – *Python Simulation*

Given the shortfalls discussed using Excel to simulate the spread of the disease in the classroom, the group elected to use Python. Python gave the group greater flexibility with several aspects of the project including:

- (i) Allowing the $p = 0.02$ infection rate to be dependent on the number of students infected
- (ii) Running a large number n simulations ($n \geq 100,000$)
- (iii) Greater flexibility with regards to immunization rate
- (iv) Potential exploration with higher infection-rate probabilities.
- (v) Graphing the Cumulative Density Function (CDF) and Probability Density Function (PDF) of the simulation results using the distfit¹ Library

In the base case (no students immunized), the group utilized NumPy's² array capabilities to keep track of several aspects of the disease spread to analyze following the simulation runs. These aspects included: Number of students infected at the start of each day, the “new” students infected during the school day, the number of students who were “newly” not infected (i.e. day 4 after being infected on day 0), a running total of students who were no longer infected, and the total number of “susceptible” students (students who had not yet been infected). An important factor of the group’s algorithm is that of random number generation and reproducibility. By using the random seed s as a function input parameter and increasing the seed $s + 1$ times for each independent simulation, the group could obtain reproduceable results. After building out the algorithm, the group ran the base case

¹

simulation 100,000 times. **Figure 2** shows a histogram of the “length” of the pandemic for the 100,000 simulation runs, that is, the number of days the pandemic lasted until there were no longer any infected students. **Figure 3** is a histogram of the number of students who were infected on day 1, and **Figure 4** is a histogram of the number of students infected on day 2. The average number of new students infected on day 1 and day 2, based on the 100,000 simulation runs were 0.6 and 0.92, respectively. Finally, **Figure 5** shows the average number of infected students for a given day. These charts were created using the Matplotlib³ library.

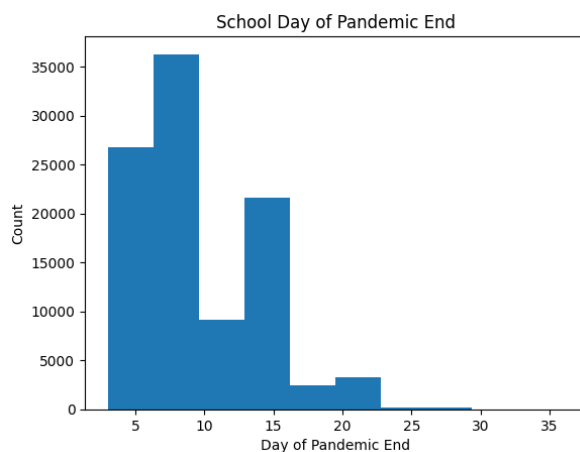


Figure 2: Histogram of Pandemic Length (base case)

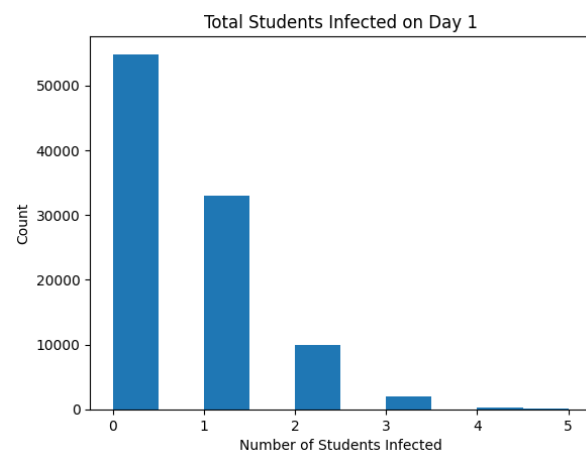


Figure 3: Histogram of Number of students infected on Day 1 (base case)

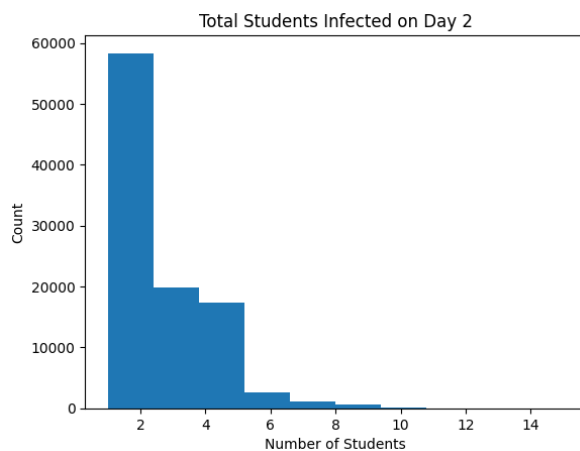


Figure 4: Histogram of Total Number of Students Infected on Day 2 (base case)

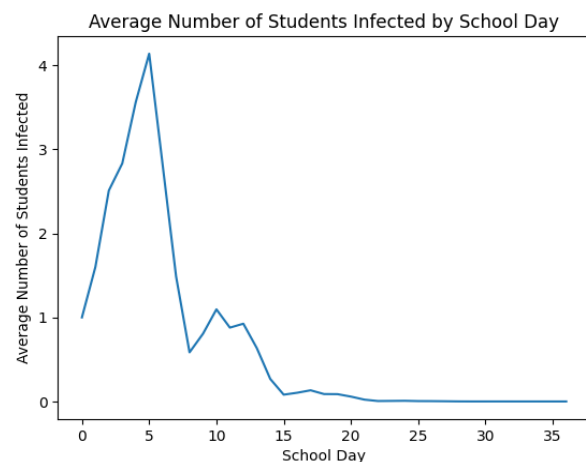


Figure 5: Average Number of Students Infected by School Day (base case)

As mentioned above, one of the advantages that Python’s distfit library offers is its ability to provide a best fit to several popular probability distributions. The pandemic length values were fit using the package and a *Weibull* distribution was determined to be the best

fit based on the Residual Sum of Squares (RSS) value. **Figure 6** shows several different probability distributions, ordered by their RSS values when fitting the pandemic length data. **Figure 7** graphs the PDF and CDF of the data when fit to the *Weibull* distribution.

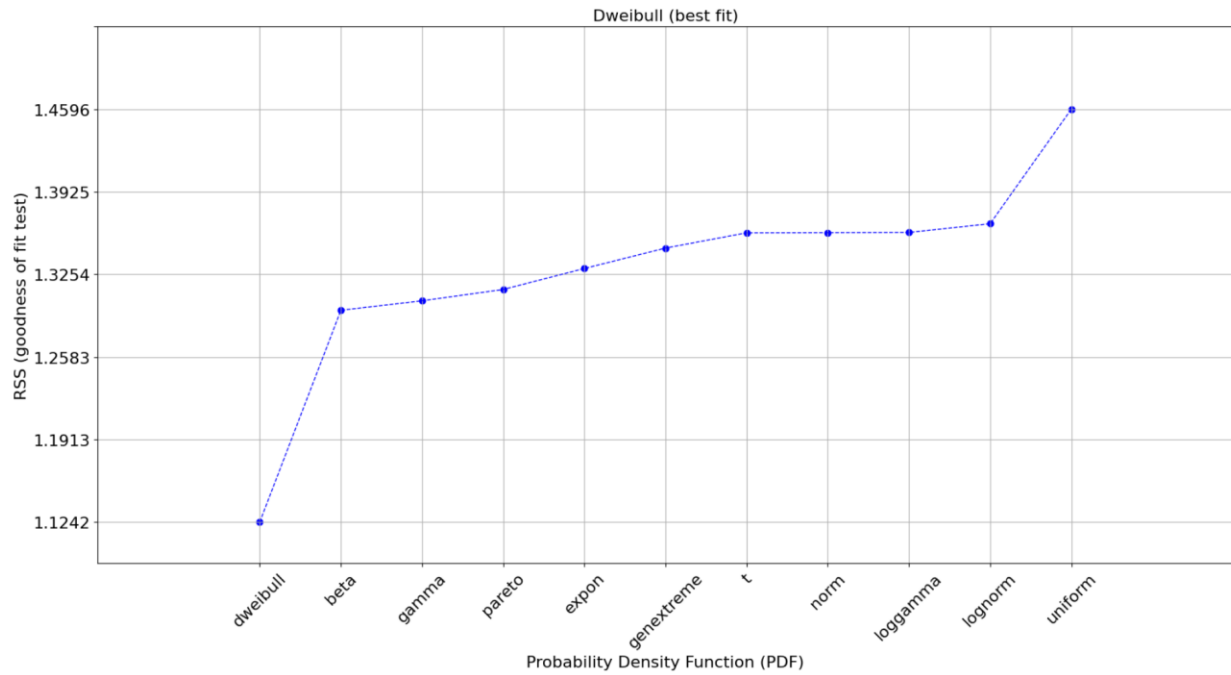


Figure 6: RSS Values of several probability distributions

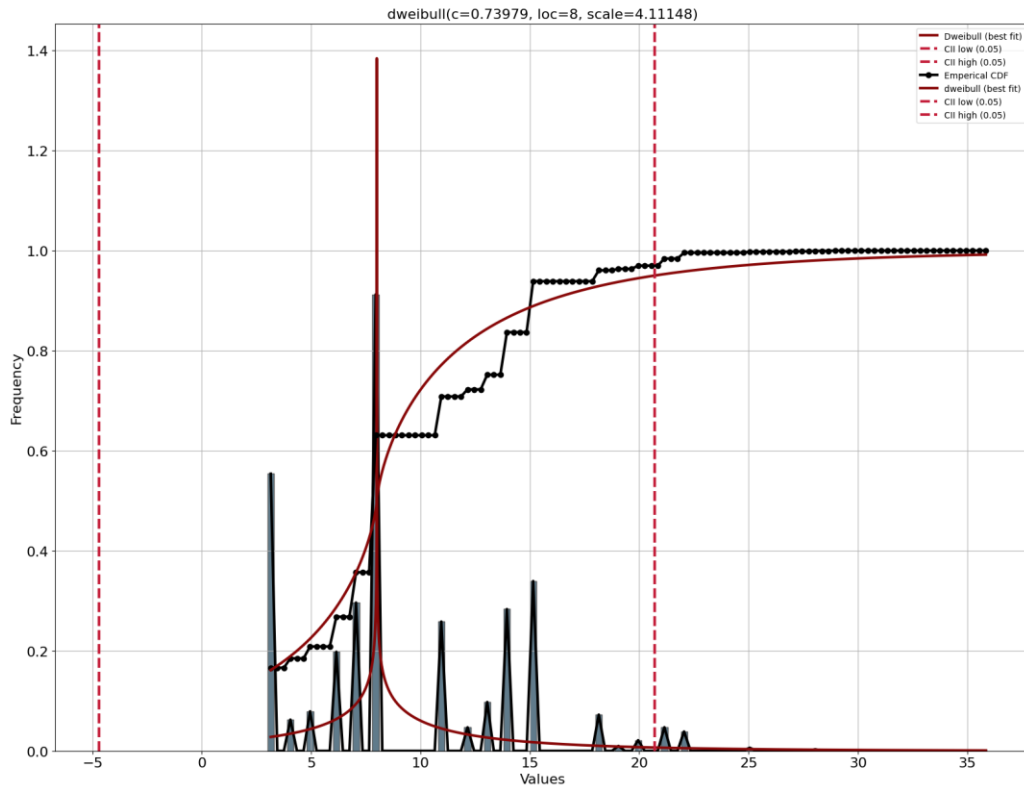


Figure 7: PDF/CDF of Pandemic Length (base case) fit to *Weibull* Distribution

In Case 2 (students have a 50% chance of being immunized), a similar approach was taken, this time subtracting a random number of students with $p = 0.50$ from the “number of students possible to infect” on day 0. The simulation was again run 100,000 times. A histogram of the number of students immunized is shown in **Figure 7**. The number of students infected on day 1 had an average value of 0.3 (see **Figure 8**). The total number of students infected after day 2 had an average value of 1.7 (see **Figure 9**). **Figure 10** shows a histogram of the pandemic length whereas **Figure 11** shows the average students infected at the end of each day. As expected, when a significant portion of the students are immunized, the pandemic ends “faster” than when no students are immunized. Based on our simulations, the average pandemic length with no immunizations was 9.2 school days compared to a pandemic length of 5.3 school days with immunizations.

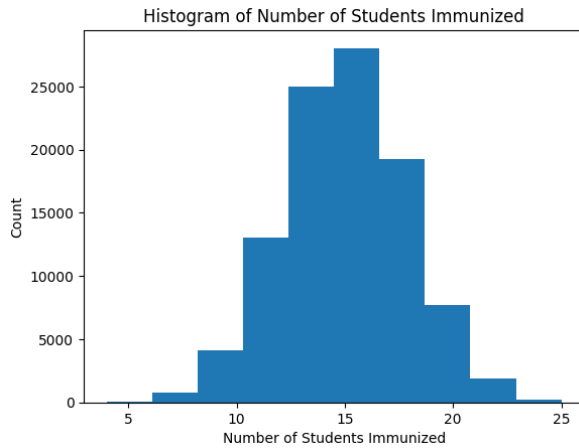


Figure 7: Histogram of Students Immunized $E[X] = 15$ (Case 2)

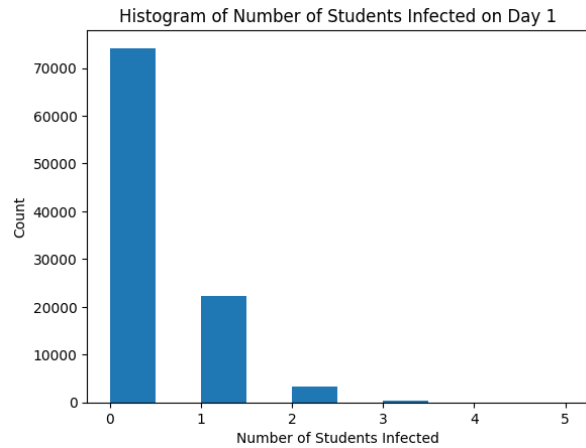


Figure 8: Histogram of number of students infected on day 1 $E[X] = (30 \cdot 0.5) \cdot 0.02 = 0.3$

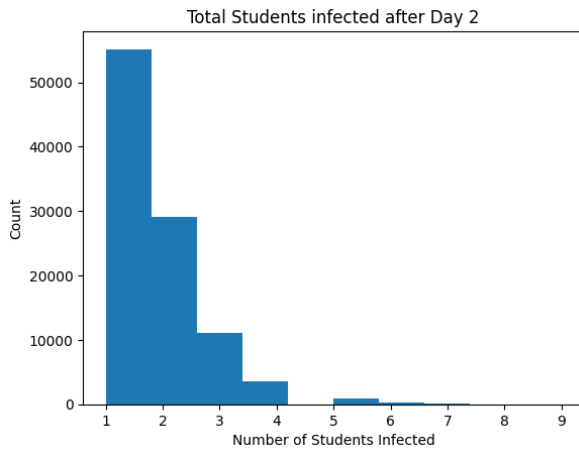


Figure 9: Histogram of Total Students Infected after Day 2 (Case 2)

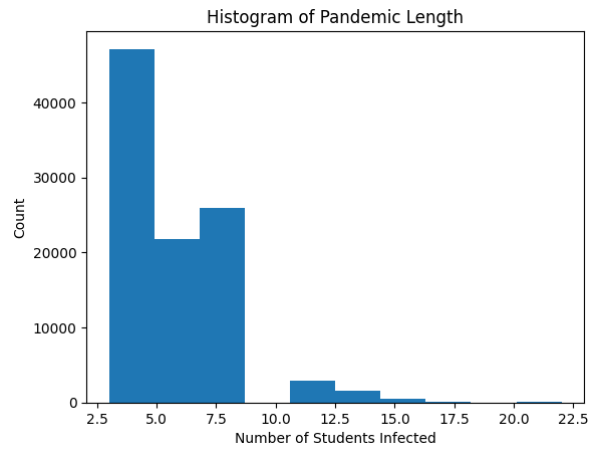


Figure 10: Histogram of Pandemic Length (Case 2)

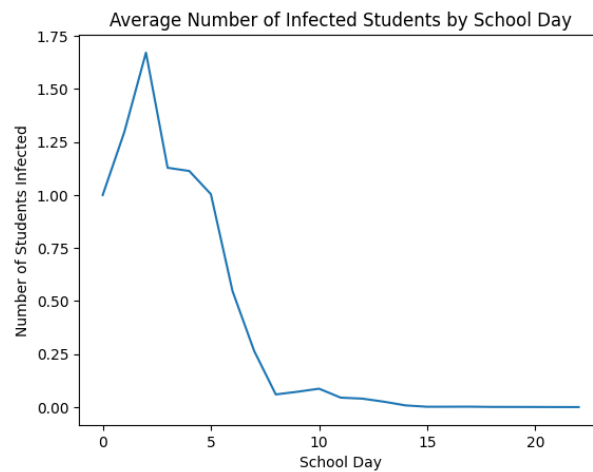


Figure 11: Average number of Students infected by school day (Case 2)

The distfit library was used in a similar manner to get a best-fit distribution from the data. The beta distribution was found to be the “best”, however, the RSS (4.28) was significantly higher than the base case fit.

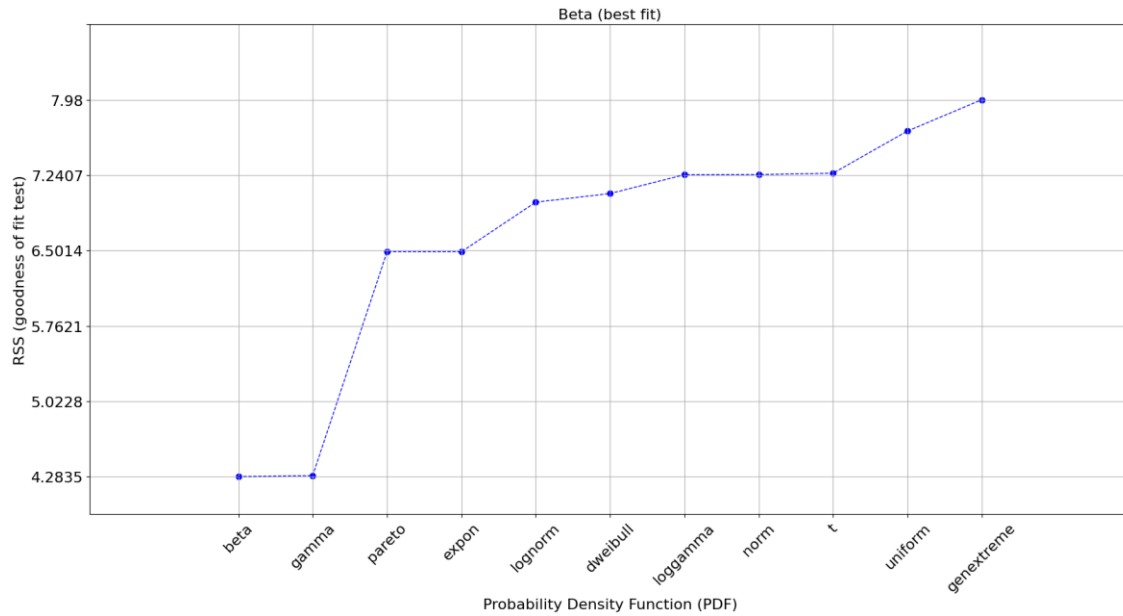


Figure 12: Best Fit Distribution based off RSS (Case 2)

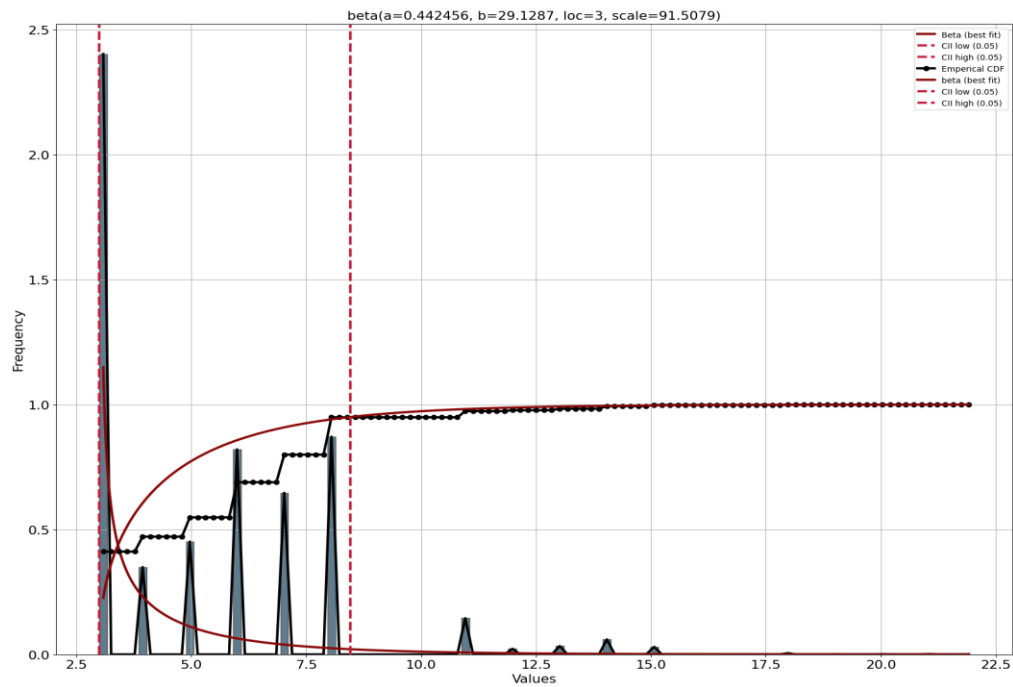


Figure 13: Beta Distribution PDF and CDF fit (Case 2)

An additional aspect that the group explored was graphing the Average pandemic length of several p values for the base classroom case. The group found that, after simulating each p value 10,000 times, a spread rate of approximately $p = 0.29$ led to the longest pandemic. See **Figure 14**.

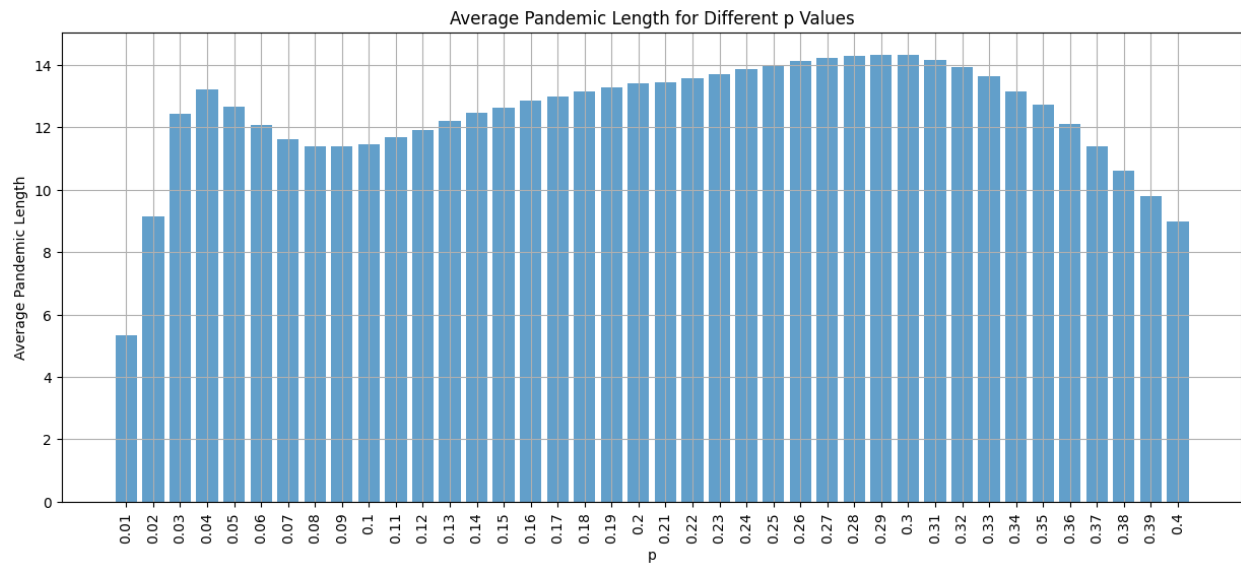


Figure 14: Average Pandemic Lengths for different values of p (base case)

Comparing the Two Simulation Methods for the Group's Use-Case

After using Excel and Python to model a pandemic spread in a classroom, Python clearly offers greater flexibility. Excel is limited by its lack of reproducibility with the `RAND()` function whereas Python allows a specific seed to be initiated. Additionally, the Excel simulation lacked dependency on the number of students infected. In comparison, the number of new infections in the Python simulation had a strong correlation to the number of students infected. Furthermore, Python's ability to run many simulations, along with its plotting/statistics packages, make it a far superior tool to run this level of simulation.

Conclusions

This project gave the group a unique opportunity to combine their programming knowledge along with the topics discussed in ISYE 6644 Lectures. The group witnessed firsthand the effect that pseudo random number generation can have within a multi-layered simulation. The program that the group developed in Python is flexible enough to

handle multiple different class sizes, p values, and immunization cases. Therefore, this script and these findings could be applied to a multitude of different disease spreads. For example, in a large college lecture of 200 students, if a new disease with a $p = 0.04$ contagious rate was introduced, 7% of the students were immunized, and students were contagious for 4 days on average how many students would get sick each day? (The group performed this simulation using the code generated during this project and the results are shown below in **Figure 15**). Simulating something as serious as disease spread will help officials make decisions to stop the spread of the disease and keep students and themselves safe. The models built in this project and ideas discussed regarding computer simulation in ISYE 6644 have far reaching implications that can improve the well-being of the general population.

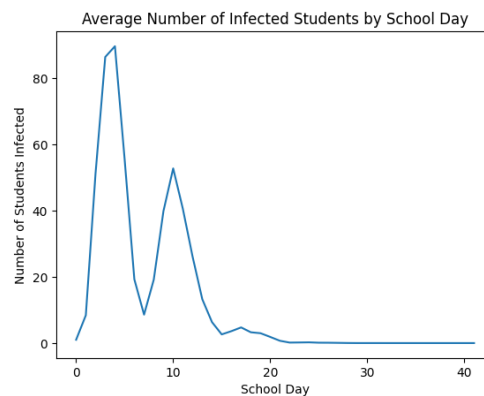


Figure 15: What if $p = 4\%$, immunization rate = 7%, class size = 200 students and students were contagious for 4 days? (Average for 10,000 simulations)

Works Cited

¹Taskesen, Erdogan. "distfit is a python library for probability density fitting." YAML metadata. Released January 2, 2020. Version 1.4.0. Available at: <https://erdogant.github.io/distfit>.

²Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.

³J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007