

Pandemic Flu Spread

Team 65

December 6, 2022

Team Members -

Daozhenle Li

Meher Bhaskar Madiraju

Kristin Lynn Georgi

Contents

| | |
|-------------------------------------|----|
| Introduction | 3 |
| Background and Assumptions..... | 4 |
| Analysis..... | 5 |
| Analytical Approach..... | 5 |
| Excel | 7 |
| R Code | 8 |
| Simulation using Python | 10 |
| Project Questions and Answers | 12 |
| Conclusion | 15 |
| Future Scope of Work..... | 15 |
| Appendix..... | 16 |
| Resources..... | 16 |

Introduction

This report aims to examine what would happen when a child unknowingly brings the flu into class. To analyze the disease propagation, we utilize the following independent approaches -

- **Analytical:** Disease propagation is a well-studied phenomenon with established analytical solution. One such solution is the Susceptible-Infected-Recovered (SIR) Model developed in the 1920's by William O. Kermack and A. G. McKendrick. We utilize existing packages in R to simulate disease propagation using the SIR Model.
- **Simulation:** Using Python, we simulated 10,000 replications of the first 50 days since Tommy got infected.
- **Hand Simulation:** Wherever possible, we tried to use hand-calculations using Excel or pen-and-paper.

We then compared the results from the 3 approaches highlighted above and concluded that the results align with each other, hence confirming the validity of our analysis.

Background and Assumptions

It's wintertime and young Tommy goes to elementary school not feeling his very best. Unbeknownst to him he has the flu. There are 21 students in Tommy's class. What does this mean for the others that Tommy encounters? How many kids will get sick and how long will the flu pester Tommy's class?

We treat this problem as a discrete event, finite length simulation with the following key assumptions -

- There are only 21 students in the class (looks like there's no teacher – maybe the students teach themselves?)
- None of the students got a flu shot – so, there are all equally likely to catch a flu
- Every student interacts with every other student every day in the classroom
- None of the students take a sick day (they must really love their school). So, there are 21 students present in school every single day.
- Tommy is the only infected student on day 0
- The remaining 20 students are susceptible on day 0
- Once infected, the student is infectious for 3 days
- The probability that an infected student infects another is 0.02
- The happenings of a day are independent of the happenings of every other day
- Once infected, a student cannot be reinfected by the flu

Analysis

Analytical Approach

The SIR model divides individuals into three groups.

- The first are the **susceptible** who are students who can become sick from infection. At the beginning of our study there were 20 susceptible students in Tommy's classroom.
- Those who have the flu are **infected**, in our case we begin with 1 infected – Tommy.
- Those who have **recovered** and are no longer sick. This study assumes that once you are recovered you have completed the process and are no longer eligible to become susceptible or infected a second time.

Everyone is either ***Susceptible, Infected, or Recovered.***

In our analysis, time “t” measured in days. $t = 1$ (day 1), $t = 2$ (day 2) and so on. The dependent variables for the study are based on the total population “N” (which in our case is 21 students). Of the 21 students, at any given time t, “S” students are susceptible to the flu, “I” students are currently infected, and “R” students have been infected in the past and have already recovered. The sum of these 3 quantities at any time = “N” - the total number of students in the class. In other words,

$$S(t) + I(t) + R(t) = N$$

The initialization condition is as follows. **The same initial conditions were used in our python simulation-**

- $t = 0$
- Number of susceptible classmates: $S(0) = 20$
- Number of infected classmates (Tommy): $I(0) = 1$
- Number of students who have already recovered: $R(0) = 0$
- Total number of students in the class: $N = 20 + 1 + 0$

Students who are initially susceptible will flow through the model as follows:

$$S(t) \rightarrow I(t) \rightarrow R(t)$$

To perform the analytical analysis, we must first understand the math behind the madness. The SIR model will use a system of differential equations to determine the rate of change of S, I, and R with respect to time, t.

- dS/dt represents the rate of change of **susceptible people over time**
- dI/dt represents the rate of change of **infected people over time**
- dR/dt represents the rate of change of **recovered people over time**

At the beginning of the problem, we were told that the probability that one infected student infects any other susceptible kid is $p = 0.02$. It follows from the assumption that the probability of an infected student infecting any of his/her susceptible classmates can be modelled by an iid Bern(p) trial. We will have an infection probability $\beta = 0.02$. Since an infected student takes 3 days to recover, the recovery rate or γ is 0.33/day.

Changes over time are computed as follows:

- $dS/dt = -\beta * S(t) * I(t)$

This number is determined by multiplying beta by the previous day's count of susceptible students and previous day's infected student count. We add a negative sign to it as this number is subtracted from the previous day's susceptible student count.

- $dI/dt = \beta * S(t) * I(t) - (\gamma * I(t))$

The first term of the above equation is the negative of dS/dt . The second term of the equation represents an individual moving from the infected category to the recovered category.

- $dR/dt = \gamma * I(t)$

The rate of recoveries is gamma multiplied by the number infected students

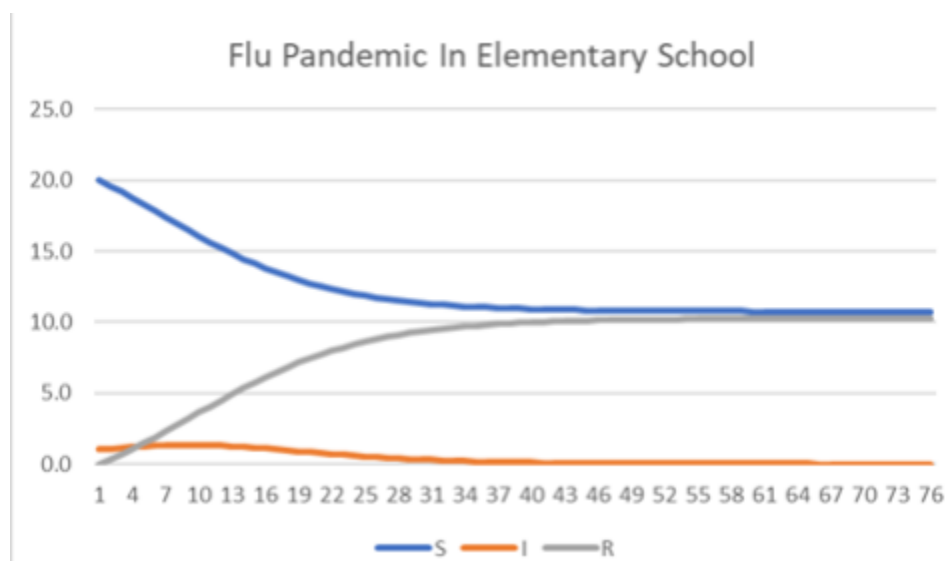
It is expected that the susceptible count is large, leading to an increase in infected count at the beginning of the simulation. As time passes, the susceptible value drops, due to "immunity" developed after infection, and the count of infected people goes down as they transition to recovered more quickly than new cases of infection. This marks the decline/end of the flu pandemic in Tommy's class.

Excel

To begin we took the equations we learned from Dr. Bazett's YouTube video and created an Excel spreadsheet to perform hand calculations. The table below lists the information for the first five days of the outbreak. As explained above on day 0, Tommy is the only student infected. The column $S \rightarrow I$ is the rate of infection that is subtracted from the susceptible group and added to the infected group. $I \rightarrow R$ is the recovery rate (subtracted from the infected group). On **day 1** the number of **infected** students is **1.4** which is computed by adding $I(0) + S \rightarrow I$. We see $I(1) = 1.1$ in the table because the recovery rate of 0.3 has been subtracted from number of infected - 1.4.

| DAY | S | I | R | N | S → I | I → R |
|-----|------|-----|-----|------|-------|-------|
| 0 | 20 | 1 | 0 | 21 | | |
| 1 | 19.6 | 1.1 | 0.3 | 21.0 | 0.4 | 0.3 |
| 2 | 19.2 | 1.1 | 0.7 | 21.0 | 0.4 | 0.4 |
| 3 | 18.7 | 1.2 | 1.1 | 21.0 | 0.4 | 0.4 |
| 4 | 18.3 | 1.2 | 1.5 | 21.0 | 0.4 | 0.4 |
| 5 | 17.9 | 1.3 | 1.9 | 21.0 | 0.5 | 0.4 |

This is our chart of the hand calculated output which indicates that about half of the class will get sick and recover.



R Code

Our next task was to solve the differential equations mentioned above. We chose R to do this. We created a function containing the sequence of differential equations.

```
# The SIR model takes input arguments: time, state a
sir_model <- function(time, state, parameters) {

  with(as.list(c(state, parameters)), {

    # The differential equations
    # Students move out of (-) susceptible at a rate
    dS <- -beta * S * I

    # Students move into (+) infected from susceptible
    # S * I and move out of (-) infected at a rate c
    dI <- beta * S * I - gamma * I

    # Students move into (+) recovered from infected
    dR <- gamma * I

    # Return the number of students in who are suscep
    return(list(c(dS, dI, dR)))
  })
}
```

We then set our initial parameters - beta, and gamma, and created a time sequence of 50 days.

```
# Create initial values on day 0
# S - Tommy's classmates are susceptible to infection
# I - Tommy - the single infected student
# R - No one has recovered
initial_vals <- c(S = 20, I = 1, R = 0)

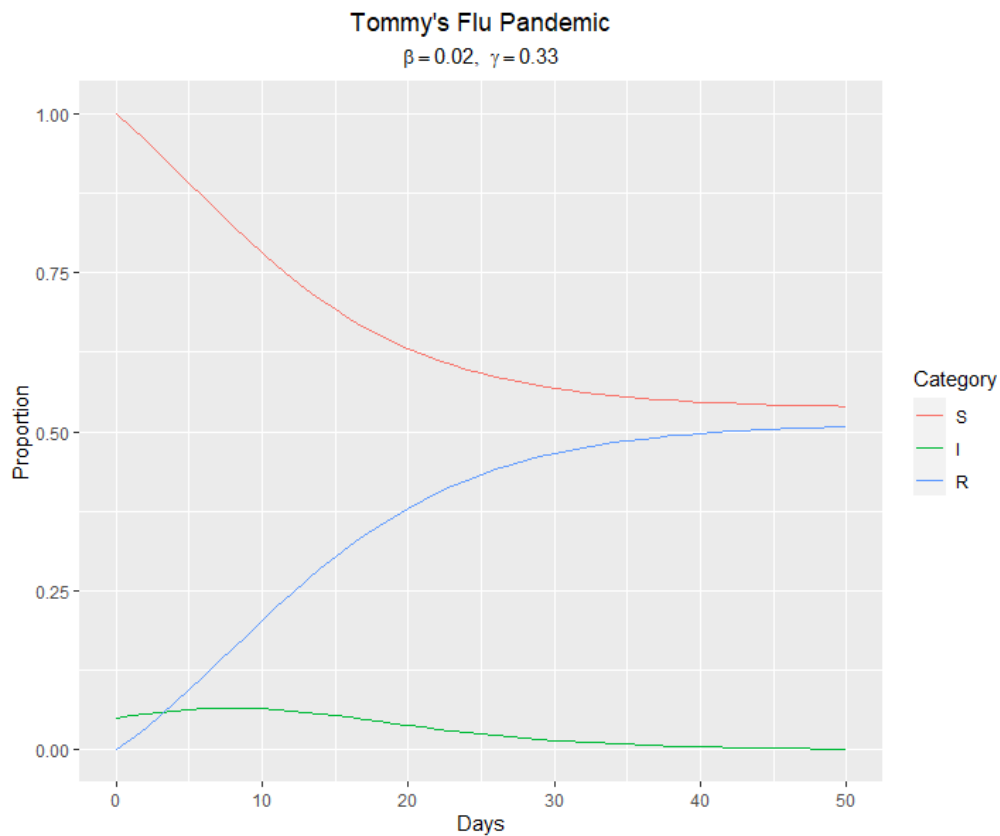
# Transmission Rate vector
# beta = probability of susceptible becoming infected
# gamma = the rate of recovery for infected students
parameters <- c(beta = 0.02, gamma = 0.33)

# Simulation Length vector
days <- seq(from = 0, to = 50, by = 1)
```

We utilized the R package “ode” to solve the ordinary differential equations.

```
# Use "ode" R package to solve the system of diff
sir_output <- as.data.frame(ode(y = initial_vals,
                               times = days,
                               func = sir_model,
                               parms = parameters))
```


The output from the ode solver is given below. As expected, this chart looks very similar to the graph we obtained from the Excel math (shown above).



The Summary Results:

| time | S | I | R |
|--------------|---------------|-----------------|----------------|
| Min. : 0.0 | Min. :10.80 | Min. :0.03418 | Min. : 0.000 |
| 1st Qu.:12.5 | 1st Qu.:11.01 | 1st Qu.:0.13947 | 1st Qu.: 5.109 |
| Median :25.0 | Median :11.83 | Median :0.50730 | Median : 8.659 |
| Mean :25.0 | Mean :13.17 | Mean :0.61391 | Mean : 7.217 |
| 3rd Qu.:37.5 | 3rd Qu.:14.68 | 3rd Qu.:1.10924 | 3rd Qu.: 9.853 |
| Max. :50.0 | Max. :20.00 | Max. :1.32584 | Max. :10.163 |

This quickly told us that out of our total population of 21 students, a little more than 1/2 did not get sick at all. Another key finding from the summary was that on average at most 1.32584 (<2) students were infected on any given day during the flu outbreak.

Simulation using Python

With the above analysis handy, we are well equipped to write some code to confirm if the simulation results align with the expected results. We chose Python to simulate the outbreak.

We created the following class (student) that generalizes a student.

Every student has the following methods and attributes –

- `curr_infected`: Indicator attribute to show if the student is currently infected.
- `prev_infected`: Indicator attribute to show if the student has previously been infected.
- `is_infectious`: Indicator attribute to show if the student is currently infectious.
- `num_days_infectious`: Variable holding the number of days from today that the student will be infectious for. This variable starts at 3 (when the student is first infected) and drops by 1 each passing day until the student fully recovers after 3 days.

Initialization conditions – Initially, all students (except Tommy) have `curr_infected`, `prev_infected`, `is_infectious` set to `False` and `num_days_infectious` set to 0.

Methods `get_infected` (to infect the student), `fully_recover` (>3 days after infection), `partial_recover` (0-3 days after infection) update the attributes mentioned above appropriately (as shown below)

```
class student:
    def __init__(self):
        self.curr_infected = False
        self.prev_infected = False
        self.is_infectious = False
        self.num_days_infectious = 0

    def get_infected(self):
        # Can be infected only if not infected in the past
        if self.prev_infected == False:
            self.curr_infected = True
            self.is_infectious = True
            self.num_days_infectious = 3

    def fully_recover(self):
        self.curr_infected = False
        self.is_infectious = False
        self.prev_infected = True

    def partial_recover(self):
        if self.curr_infected == True:
            self.num_days_infectious -= 1
            if self.num_days_infectious == 0:
                self.fully_recover()
        else :
            return
```

We also wrote some helper functions to collect the following statistics daily -

- Infected during the day
- Infected as on the end of the day (includes all infections in the last 3 days)
- Recovered as on the end of the day
- Susceptible at the end of the day

```
# Stats collected at end of each day
def update_daily_new_infected():
    daily_new_infected.append(sum([1 if x.curr_infected == True and x.num_days_infectious == 3 else 0 for x in students]))

def update_daily_total_infected():
    daily_infected.append(sum([1 if x.curr_infected == True else 0 for x in students]))

def update_daily_recoveries():
    daily_recovered.append(sum([1 if (x.prev_infected == True ) else 0 for x in students]))

def update_daily_susceptible():
    daily_susceptible.append(sum([1 if (x.prev_infected == False and x.curr_infected == False) else 0 for x in students]))
```

The simulation for a day is as follows – every day, student groups are affected as follows –

- If the student is currently infected, he is on the course to recover in 3 days.
- If the student has previously been infected, he is unaffected by the events of the day.
- If the student has never been infected, each of his “interactions” with an infected student is treated as a Bernoulli trial. If the outcome of the trial is 1, the uninfected student gets infected.

```
def simulate_day():
    # Assumption - All 21 students attend school everyday
    # Every student interacts with every other student
    # Cannot catch infection after recovery - [Change student.prev_infected == True block if that's not the case]

    curr_infected = daily_infected[-1]
    student_ctr = 0
    for student in students:
        student_ctr+=1
        # print('With Student {}'.format(student_ctr))
        if student.curr_infected == True:
            student.partial_recover()

        elif student.prev_infected == True:
            continue

        else:
            # print('~~~~~ Student {} is currently uninfected ~~~~~'.format(student_ctr))
            for i in range(curr_infected): # Bernoulli trial for getting infected from any of the affected students
                trial_prob = np.random.random(1)
                # print('~~ Intreaction with student {} gave trial prob {}'.format(i, trial_prob))
                if trial_prob < probab_infect:
                    student.get_infected()

    update_daily_new_infected()
    update_daily_total_infected()
    update_daily_recoveries()
    update_daily_susceptible()
```

At the end of each day, we collect logs for the events that occurred on that day. Simulation of 50 consecutive days constitutes one replication. We **randomly** simulated 10,000 replications and aggregated the statistics.

Project Questions and Answers

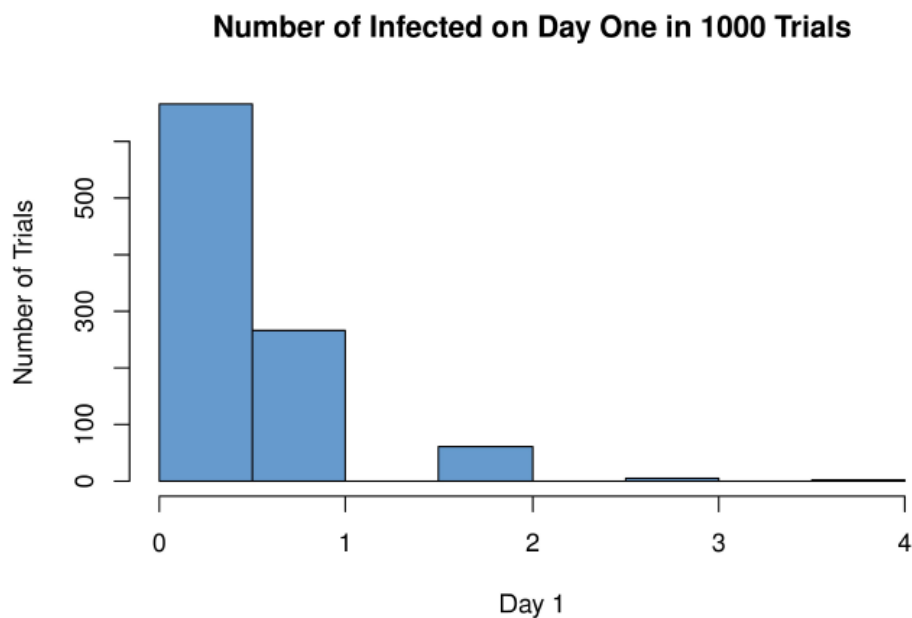
(a) What is the distribution of the number of kids that Tommy infects on Day 1?

$Bin(20,0.02)$

(b) What is the expected number of kids that Tommy infects on Day 1?

Number of students infected on day 1 - We included Tommy as infected and estimated the following:

Based on 1000 runs of the simulation of Day 1,
it is expected that 1.4 students are infected. (Based on SIR model)



This result matched both our Excel simulation math and our output below.

$$E[t = 1] = 1 + (0.02 * 20 * 1) = 1.4 \text{ (Hand Calculation)}$$

Our analysis showed that the results of the simulation align with those from the SIR model. For example, the number infected on day 1, we see the same conclusion as we calculated previously:

~~~~~ Day 1 Infected: 1.4058 ~~~~~ (From python simulation)

All 3 analyses indicate that the expected number of people infected on Day 1 is 1.4. This indicates that Tommy infects an average of **0.4** (1.4 - 1) students.

**(c) What is the expected number of kids that are infected by Day 2 (you can count Tommy if you want)?** We counted Tommy (did not want to exclude him just because he's sick 😞)

$$E[2] = 1.4 + (0.02 * 19.6 * 1.4) = 1.9488 \text{ (Hand Calculation)}$$

The above calculation considers the count of susceptible and infected students

We can see that this number aligns with the outcome from the simulation.

~~~~~ Day 2 Infected: 1.9306 ~~~~~ (Python Simulation)

This indicates that the total number of students infected at the end of Day 2 is ~1.9.

(d) Simulate the number of kids that are infected on Days 1,2,... Do this many times. What are the (estimated) expected numbers of kids that are infected by Day i , $i = 1; 2; \dots$? Produce a histogram detailing how long the "epidemic" will last.

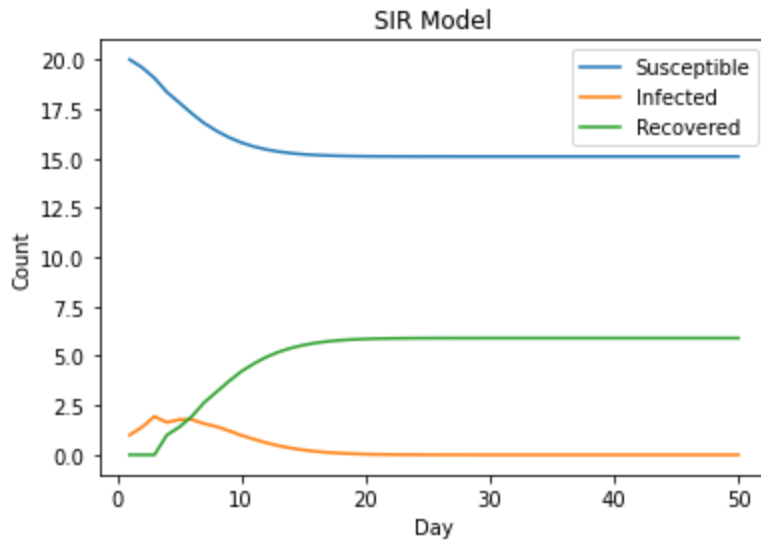
The count of daily infections from our simulation for the first 5 days is as given below-

```
~~~~~ Day 1 Infected: 1.4058 ~~~~~
~~~~~ Day 2 Infected: 1.9306 ~~~~~
~~~~~ Day 3 Infected: 1.6409 ~~~~~
~~~~~ Day 4 Infected: 1.7621 ~~~~~
~~~~~ Day 5 Infected: 1.7813 ~~~~~
```

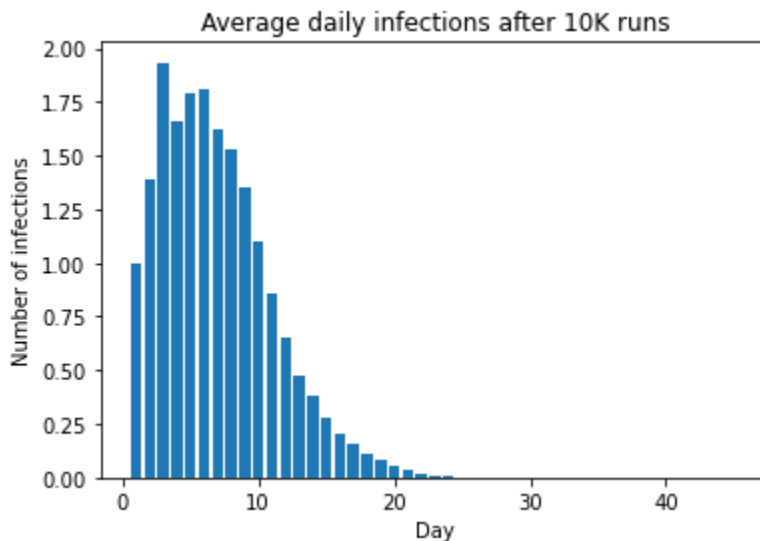
We also determined that approximately 1-2 kids would have the flu on any given day.

~~~~~ Max Infected on any single day: 1.9306 ~~~~~

The source file for the python code attached to the project. The graph of the output is very similar to our original Excel and R plots showing that not everyone gets sick, a significant portion of the students do not even get infected.



From the histogram of daily infections (averaged from 10,000 runs), we can see that the expected length is ~25 days and appears to be completely gone after this time. This is closely reflected in the Excel and hand calculations. So that leads us to predict that Tommy's Flu Pandemic will last about a month.



# Conclusion

In thinking about when we were in school it makes sense that the flu runs through a class in about a month. That generally happens every year and this example helped us to see and use the skills we learned in class this fall. Specifically, we learned that the study of diseases is a large field which people study to help both understand and use to determine ways to curb or prevent further spread of diseases as small as the flu, but the models can be extended to huge pandemics such as COVID-19. It was interesting to find out how scientists determine what constitutes a “pandemic” and methods that can be used to try to speed up recovery and methods used to curb susceptibility. Things like “wash your hands” and “wear a mask” that we got so used to during the COVID shut down translate into variables that can be plugged into the pandemic equation to help lower the spread of disease.

Starting out we wanted to calculate things with Excel to better understand the process and what we should be looking for in our code. This was a good baseline for us, and we used it as a reference when working on the rest of the code. This simple example of a small-scale flu and how it affects a classroom was an excellent example of a real-life situation that can be simulated with results that can be both informative and helpful in planning and dealing with the spread of disease.

# Future Scope of Work

In researching the SIR model and the system of equations we found many applications, uses, studies, and far more variables that could be considered besides the simple model of susceptible, infected, and recovered. Although our research was based on Tommy and his classmates, it is interesting to see how many of the studies are related to the COVID pandemic. In addition, it would be interesting to see the impact of adding other variables such as becoming susceptible even after recovery, adjusting infected periods, and considering possibilities such as hospitalization, immunization, vaccinations, and death after infections to the study. What we found most interesting is how this type of modeling is used to determine the impact of the spread of disease on society and the efforts applied by mathematicians, scientists, and doctors to try to educate the public about health and hygiene to lower the impact of a pandemic.

# Appendix

Function used to solve differential equations in R:

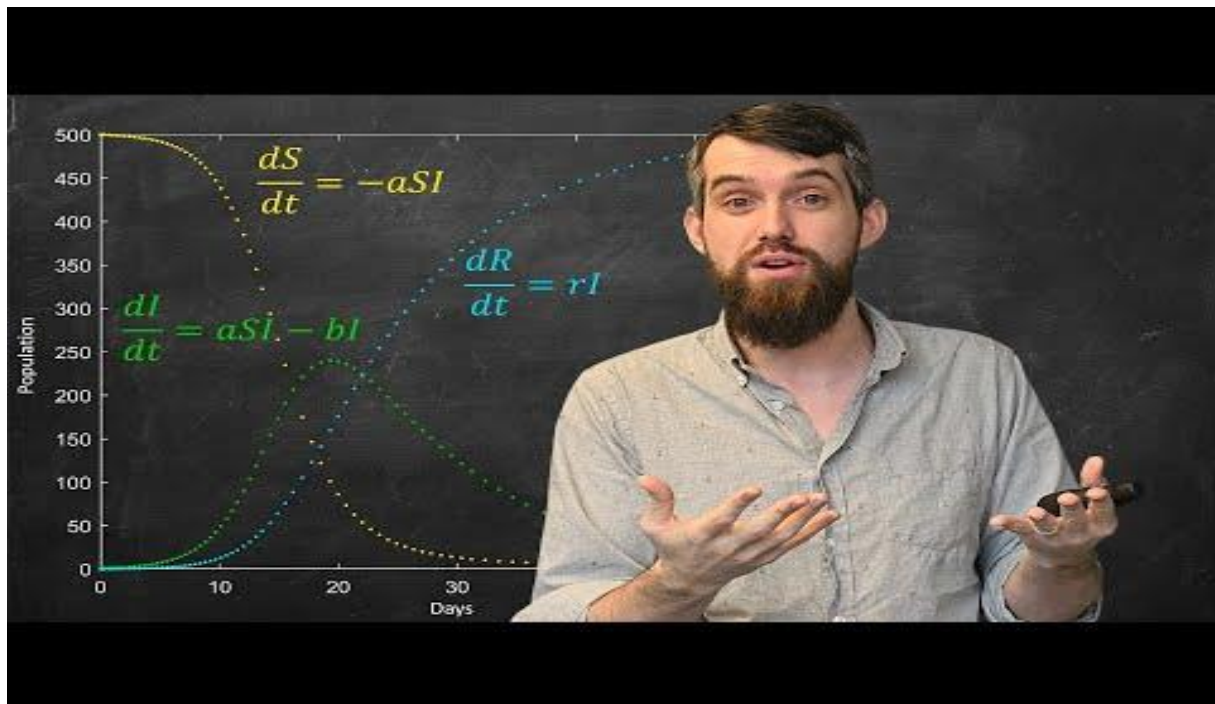
ode: General Solver for Ordinary Differential Equations, from RDocumentation located in deSolve package

Description: Solves a system of ordinary differential equations; a wrapper around the implemented ODE solvers

Value: A matrix of class deSolve with up to as many rows as elements in times and as many columns as elements in y plus the number of "global" values returned in the second element of the return from func, plus an additional column (the first) for the time value. There will be one row for each element in times unless the integrator returns with an unrecoverable error. If y has a names attribute, it will be used to label the columns of the output value.

## Resources

[The MATH of Pandemics | Intro to the SIR Model](#)





<https://www.maa.org/press/periodicals/loci/joma/the-sir-model-for-spread-of-disease-the-differential-equation-model>

<https://stockton.edu/sciences-math/ezone/fall2021/mathematical-models.html>

<https://www.rdocumentation.org/packages/deSolve/versions/1.34/topics/ode>

<https://www.programmingr.com/examples/neat-tricks/sample-r-function/r-rbinom/>