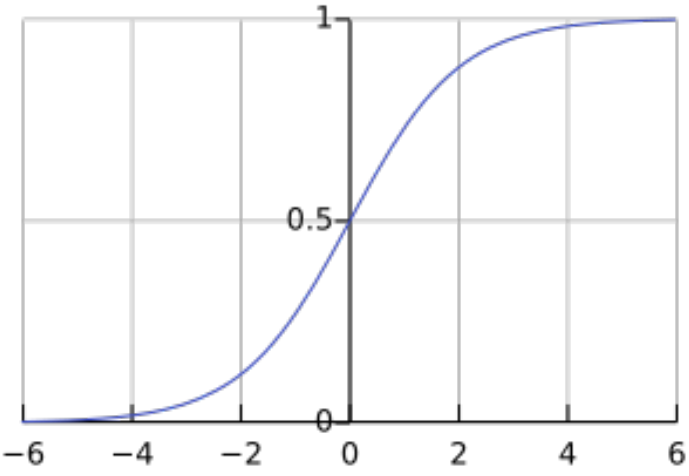
Mini Project 2 - Interpolation and Curve Fitting

Trevor Wylezik

December 11, 2024

Abstract

In 2020, the world was struck with COVID-19, a highly contagious virus. Even though this was a fatal virus for some, it made for some interesting discoveries in the sciences as we suddenly had one of the most in-depth virus data collections alongside the technological advancements of the twenty-first century. Scientists predicted that we would see the data represent a logistic curve that is typically used to predict population growth (Figure 1). In this report, we will investigate state COVID case data to see if we can extract logistic curves that fit to the data. We will also focus on the delta variant of COVID-19 as it was even more infectious and shows a clear spike in the data. The fitting for this data will be using California data as it is most credible (high population).



Curve.png

Figure 1: A generic logistic curve with inflection point $x = 0$ and capacity $L = 1$.

1 Introduction

In this report, we will investigate COVID-19 case data and analyze the behavior with respect to time. In particular, each state has recorded daily data on the number of total covid cases, recoveries, deaths, and other metrics. This report will focus on the COVID-19 cases that follow a logistic curve (Figure 1). This project starts with collecting the COVID-19 data from a known source and cleaning this data in excel. Then, this cleaned data set will be pulled in to python for further analysis with Pennsylvania, California, Texas, Florida, and New York. The curve fitting will be with California data and uses concepts from linear algebra such as linearization, normal equations, and Vandermonde matrices. Next, statistics concepts like the minimization of least squares will be used to model the data and produce legible charts.

This project is organized as follows: In Section 2, we will highlight the problem statement. In section 3, we will go in-depth in to the cleaning of the data set, pulling it in to python, and performing the curve fitting. Section 4 and 5 will be the final results of the project and a discussion about the tool and results. Finally, Section 6 will be conclusions of the project.

2 Problem Statement

The goal of this project is to model data using linear algebra concepts and built in python functions from NumPy, SciPy, and Matplotlib. These will be used to solve both interpolation and linear least squares formulations.

3 Methodology

In this section, we will dive in to the data collection/cleaning and model building.

3.1 Data Collection and Cleaning

The data for this project comes from a COVID-19 data repository from the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. The data is in the format of CSV files for each day of data collected. This was then imported in to one excel file for further investigation.

At first, the data was messy and had a lot of unnecessary information as seen below.

data.png

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Source.Name	Province_State	Country_Region	Last_Update	Lat	Long_	Confirmed	Deaths	Recovered	Active	FIPS	Incident_Rate	Total_Test_Results
2	01-01-2021.csv	Alabama	US	1/2/2021 5:30	32.3182	-86.9023	365747	4872	202137	158738	1	7459.375895	
3	01-01-2021.csv	Alaska	US	1/2/2021 5:30	61.3707	-152.4044	47019	206	7165	39648	2	6427.355802	1275750
4	01-01-2021.csv	American Samoa	US	1/2/2021 5:30	-14.271	-170.132	0	0			60	0	2140
5	01-01-2021.csv	Arizona	US	1/2/2021 5:30	33.7298	-111.4312	530267	9015	76934	444318	4	7285.171274	5155330
6	01-01-2021.csv	Arkansas	US	1/2/2021 5:30	34.9697	-92.3731	229442	3711	199247	26484	5	7602.945718	2051488
7	01-01-2021.csv	California	US	1/2/2021 5:30	36.1162	-119.6816	2434971	26298			6	6164.469663	33058311
8	01-01-2021.csv	Colorado	US	1/2/2021 5:30	39.0598	-105.3111	362438	5435	18102	314186	8	5854.774381	4444206
9	01-01-2021.csv	Connecticut	US	1/2/2021 5:30	41.5978	-72.7554	185708	5995	9800	169913	9	5208.781229	4320693
10	01-01-2021.csv	Delaware	US	1/2/2021 5:30	39.3185	-75.5071	58064	1065	18851	38148	10	5962.841099	1021582
11	01-01-2021.csv	Diamond Princess	US	1/2/2021 5:30			49	0			88888		
12	01-01-2021.csv	District of Columbia	US	1/2/2021 5:30	38.8974	-77.0268	29252	788	20941	7523	11	4144.816358	904302
13	01-01-2021.csv	Florida	US	1/2/2021 5:30	27.7663	-81.6868	1323315	21673			12	6161.333478	17392558
14	01-01-2021.csv	Georgia	US	1/2/2021 5:30	33.0406	-83.6431	677589	10958			13	6381.859327	5401054
15	01-01-2021.csv	Grand Princess	US	1/2/2021 5:30			103	3			99999		
16	01-01-2021.csv	Guam	US	1/2/2021 5:30	13.4443	144.7937	7326	122	7047	157	66	4460.844309	96201
17	01-01-2021.csv	Hawaii	US	1/2/2021 5:30	21.0943	-157.4983	22397	289	11958	10150	15	1581.852032	816811
18	01-01-2021.csv	Idaho	US	1/2/2021 5:30	44.2405	-114.4788	141077	1436	58649	80992	16	7894.340721	545485
19	01-01-2021.csv	Illinois	US	1/2/2021 5:30	40.3495	-88.9861	963389	17978			17	7602.60897	13374665
20	01-01-2021.csv	Indiana	US	1/2/2021 5:30	39.8494	-86.2583	517773	9468	345474	162831	18	7690.970837	5730043

Figure 2: The original excel file with state COVID-19 data (before cleaning).

This dataset was updated to make it easier to work with. Some of these updates are as follows: removing US territories and cruise ships, changing the date column to be a serial number, removing duplicate rows, removing latitude/longitude and other unnecessary columns. These changes can be seen below.

data.png

	B	C	D	E	F
1	**Province_State**	**Last_Update**	**Date (Serial)**	**Confirmed**	**Deaths**
2	Alabama	4/12/2020	43934	3667	93
3	Alaska	4/12/2020	43934	272	8
4	Arizona	4/12/2020	43934	3542	115
5	Arkansas	4/12/2020	43934	1280	27
6	California	4/12/2020	43934	22201	632
7	Colorado	4/12/2020	43934	11819	808
8	Connecticut	4/12/2020	43934	12035	554
9	Delaware	4/12/2020	43934	1625	49
10	Diamond Princess	4/12/2020	43934	49	0
11	District of Columbia	4/12/2020	43934	1875	50
12	Florida	4/12/2020	43934	19895	461
13	Georgia	4/12/2020	43934	12452	433

Figure 3: Cleaned COVID-19 state data set.

With a full data set of COVID-19 data that is cleaned easily accessible, we can now pull this data in to python in the next section.

Note: For all of the curve fitting the data has to be normalized to the interval $(0, 1)$ and then scaled back up for charting purposes.

3.2 Pulling and Organizing Data in Python

For the purposes of pulling excel data in to python, Pandas will be the main tool used. We first have to create a data frame and segment the data in to specific states that we want to see.

pulling1.png

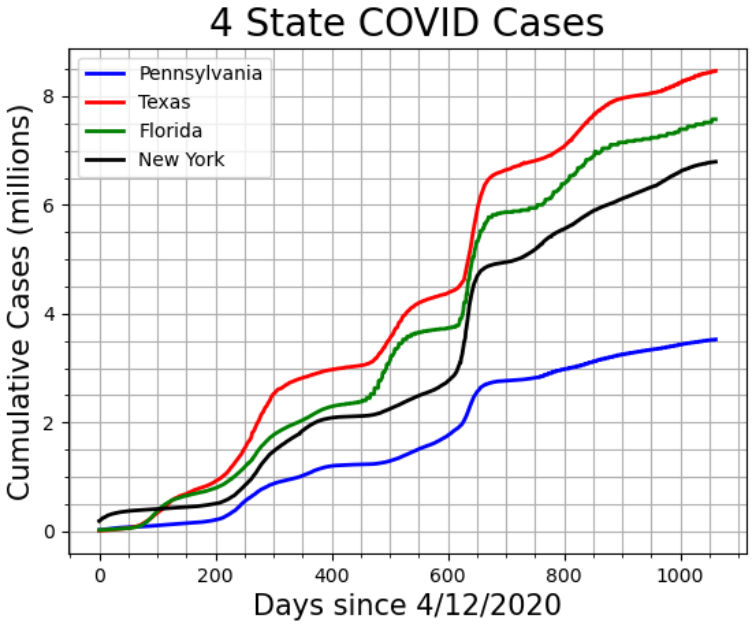
```
# ----------------------- DATA PULLING ----------------------- #

# Import the data (dataframe):
df = pd.read_excel('E:\Covid Data.xlsx', 'Cleaned Data Set') # Exce

# Grabbing all State rows:
Cat = 'Province_State'
Grouped = df.groupby(Cat)
df_CA = Grouped.get_group('California')
df_TX = Grouped.get_group('Texas')
df_FL = Grouped.get_group('Florida')
df_NY = Grouped.get_group('New York')
df_PA = Grouped.get_group('Pennsylvania')
```

Figure 4: Pulling an excel file in to Python and grouping rows by states.

We can now view each state's data and see the waves of COVID-19 spreading rapidly. This was done with Florida, Pennsylvania, New York, and Texas as examples to show the logistic curves.



state data.png

Figure 5: NY, CA, FL, and TX COVID-19 cumulative case data.

3.3 Linearly Solved Logistic Fit

The focus of this fit was to look at a specific time span when the delta variant of COVID-19 was spreading. This variant was less lethal than normal, but was more contagious leading to higher COVID-19 cases.

First, we can look at a generic logistic equation to base our model on:

$$y = \frac{1}{1 + ae^{-bx}}$$

This equation is able to rearranged and expressed linearly as follows:

$$\begin{aligned}
&\implies \quad \frac{1}{1 + ae^{-bx}} = y \\
&\implies \quad 1 + ae^{-bx} = \frac{1}{y} \\
&\implies \quad ae^{-bx} = \frac{1}{y} - 1 \\
&\implies \quad \ln\left(ae^{-bx}\right) = \ln\left(\tfrac{1}{y} - 1\right) \\
&\implies \quad \ln a + \ln e^{-bx} = \ln\left(\tfrac{1}{y} - 1\right) \\
&\implies \quad \ln\left(a\right) + \ln\left(e^{-bx}\right) = \ln\left(\tfrac{1}{y} - 1\right) \\
&\implies \quad \ln(a) - bx = \ln\left(\tfrac{1}{y} - 1\right)
\end{aligned}$$

This results in the following matrix equation:

$$\underbrace{\begin{bmatrix} 1 & -x_1 \\ 1 & -x_2 \\ \vdots & \vdots \\ 1 & -x_n \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \ln\left(a\right) \\ b \end{bmatrix}}_{\vec{c}} = \underbrace{\begin{bmatrix} \ln\left(\frac{1}{y_1} - 1\right) \\ \ln\left(\frac{1}{y_2} - 1\right) \\ \vdots \\ \ln\left(\frac{1}{y_n} - 1\right) \end{bmatrix}}_{\vec{y}}$$

The normal equation for regression can then be made and solved for $\vec{c}$.

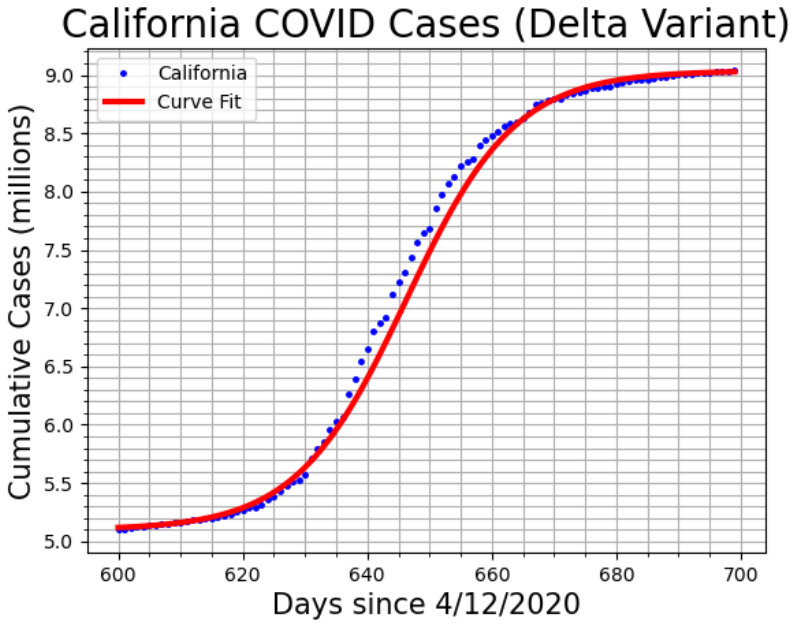
$$\left(A^T A\right)\vec{c} = A^T\vec{y}$$

Solving the above normal equation with numpy and plotting the generic logistic function with a and b produces the following chart and values:

$$\text{Logistic Equation: } y = \frac{1}{1 + ae^{-bx}}$$

$$a = 0.6507 \text{ and } b = 0.1138 \text{ (rounded to 2 decimal places)}$$

$$R^2 = 0.9846$$



delta.png

Figure 6: California COVID cases between 12/3/2021 and 3/13/2022 (delta variant).

3.4 Multiple-Logistic Curve Fit

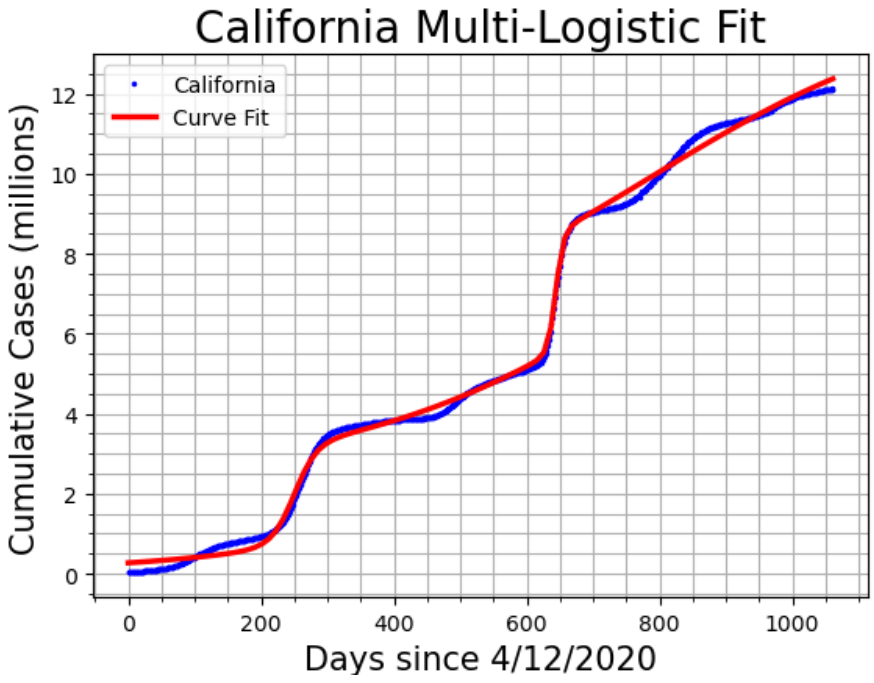
The COVID-19 cases seemed to follow a logistic curve, but repeated as COVID-19 infected people in waves. This is difficult to model from the start as the logistic function acts in such a way that it has one inflection point (x_0), maximum value (carrying capacity L), and growth rate (k). But, each jump in the COVID-19 case data is not at the same place with the same capacity, and does not have the same growth rate.

One way around this is to add multiple logistic functions together that are all offset from each other. This works since all values are effectively flat at the bottom and top of the curve. It seems like the data has 3 main growth points, so we will look at a multiple-logistic curve fit with 3 jumps.

$$y = \frac{L_1}{1 + e^{-k_1(x-x_1)}} + \frac{L_2}{1 + e^{-k_2(x-x_1-x_2)}} + \frac{L_3}{1 + e^{-k_3(x-x_1-x_2-x_3)}}$$

In simpler terms, L_i is additional the height of the carrying capacity from the section of the curve. k_i is the i^{th} curve's growth rate. And x_i is the distance from the previous inflection point. After using SciPy's optimize curve fit function, the chart and values were produced:

$$\begin{array}{lll} L_1 = 0.20518 & L_2 = 0.24404 & L_3 = 0.73664 \\ k_1 = 0.05945 & k_2 = 0.17356 & k_3 = 0.00452 \\ x_1 = 250.077 & x_2 = 393.792 & x_3 = 143.726 \\ & R^2 = 0.9984 & \end{array}$$



fit.png

Figure 7: California COVID cases with a multi-logistic curve fit.

4 Results

We have now been able to analyze COVID-19 data at the state level and show that it follows a classic logistic curve. This is due to the high R^2 values that represent the proportion of error due to the fit (which is related to minimal error).

5 Discussion

This project had many intricacies in the coding process that took a substantial amount of time. For example, the California case data is in the millions over a thousand day time period. This caused the curve fitting to struggle as it had to hit such high points in the data.

A way around this was by normalizing the data to make everything fit in between $(0, 1)$ (not inclusive). This caused the data to be easier to work with and then was able to scaled back up to its original size. Some values like L_i and k_i seem very small in this respect, and that is the reason why.

6 Conclusion

In this project, interpolation and curve fitting was used to find functions that represent COVID-19 data. This was done by first collecting and cleaning data, then pulling that data in to python to run least squares curve fitting, and finally producing optimized parameter values and legible charts.

7 Resources

For this project, the data was sourced from The Center for Systems Science and Engineering (CSSE) at Johns Hopkins University following hyperlink:

COVID-19 Data by state