COVID-19 Data Analysis

Rishav Bhagat

April 25, 2020

Contents

1 Dataset

I used the dataset that was collected and used by John Hopkins University for this project. This dataset was collected from many international health organizations and all compiled into onto github repository. It contains information on daily reports (including new cases, deaths, and recoveries), and global trends.

Here is the link to the datasets I used:

https://github.com/CSSEGISandData/COVID-19

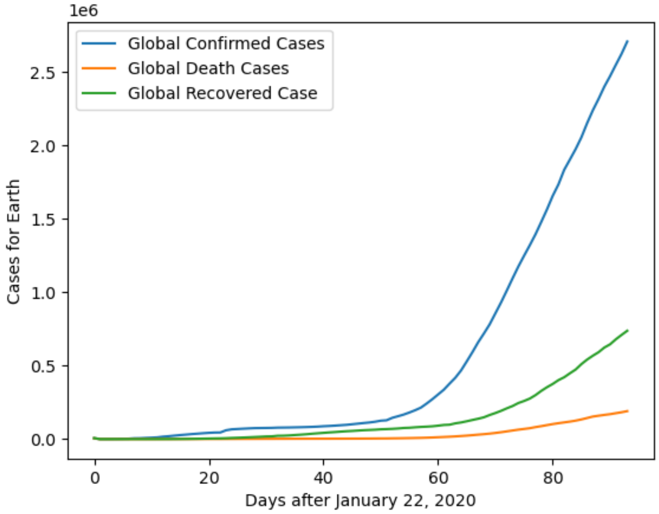
1.1 Downloading the Data

I created a batch script to download all this data using git, delete files that are not needed for my analysis, and move the files into more convienient locations.

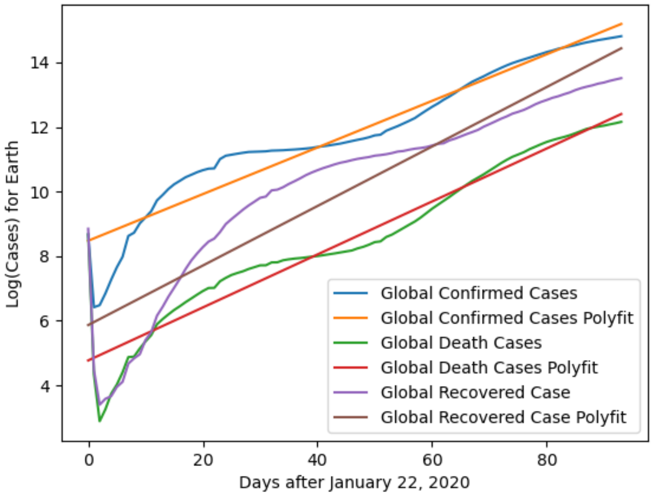
```
@echo off
rmdir data /s /q
git clone https://github.com/CSSEGISandData/COVID-19.git
rename COVID-19 data
cd data
rmdir .git /s /q
rmdir archived_data /s /q
rmdir who_covid_19_situation_reports /s /q
del README.md
mv csse_covid_19_data/* .
rmdir csse_covid_19_data /s /q
cd ..
git add data
git commit -m "updated data from John Hopkins github repo"
git push
```

1.2 Initial Glance

First, I plotted the global coronavirus data (cases vs days) in a similar fashion to the way it was plotting on the John Hopkins project. This is to reveal any obvious trends.



At a first glance, this looks like exponential growth, so I then plotting a logarithmic graph with a line of best-fit.



A linear relationship in the logarithmic graph corresponds to an exponential relationship in the original graph. As we can see, the logarithmic graph is approximately linear, which confirms that so far, the coronavirus is exponentially growing. But we also see that the curve is beginning to flatten out, corresponding to the true logistic growth of the original graph. The beginning of all logarithmic graphs appear to be exponential, which explains why it apears as it does.

Now, we know that logistic growth will eventually level out, and it is in our interest to see when and where it will do this globally. We can do this by tracking the growth rate of the graph, as we shall see in the next section.

2 Data Analysis and Implications

First let's define some variables:

- Let us refer to the region we are talking about as X. For now X refers to the entire globe.
- Let N^X be the final number of people who have COVID-19 within X. I may sometimes just use N, which will refer to the region in question at the time.
- Let N_i^X be the current number of people who have COVID-19 within X on the i-th day after January 22nd (which is when John Hopkins began collecting data).
- Let $\Delta N_i^X = N_i^X - N_{i-1}^X$ be the number of new cases on a given day, defined by i.
- In a similar way, define $D^X, D_i^X, \Delta D_i^X$ as the people who died due to COVID-19 in the respective time frames.
- In a similar way, define $R^X, R_i^X, \Delta R_i^X$ as the people who recovered from COVID-19 in the respective time frames.

While doing the analysis on the data, I will focus on the number of cases, but the number of deaths and recoveries also follow very similar patterns.

2.1 Growth Ratio

Another intrinsic value that can give us insight into the data is the growth ratio. Let define this as

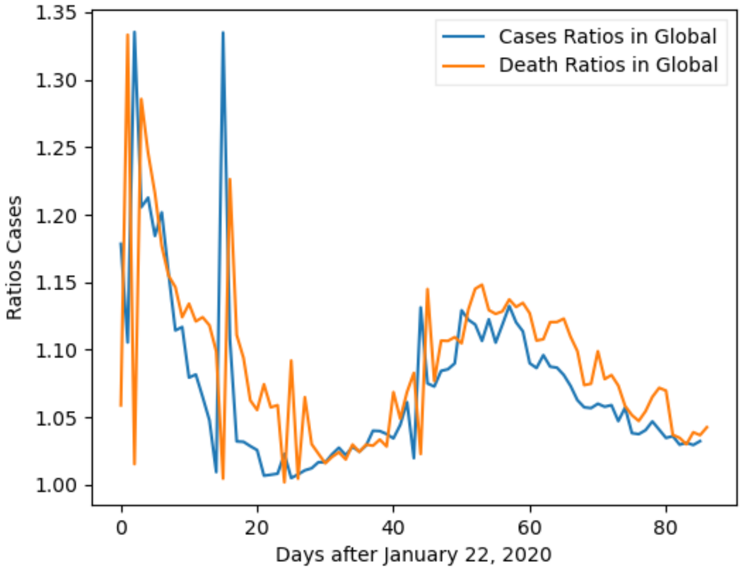
$$R_i^X = \frac{N_i^X}{N_{i-1}^X} \tag{1}$$

For pure exponential growth the ratio should be a constant number throughout the entire domain, but since the ratio is decaying with time (as we see in the graph below), there is more evidence pointing to a more logistic approach. We see that the growth ratio is approaching one for both the cases and deaths, showing that eventually $N_i^X \to N^X$.

Another interesting observation that is made apparant from this graph is the relationship between the deaths and cases. In other graphs this relationship is also evident, but here they are both plotted on the same scale so it is even more clear. The deaths are following a similar pattern as the cases, but they are slightly time-shifted (forward in time). This makes sense since we would expect a ratio of cases to become death in a certain time.

We can also estimate the effectiveness of our treatment by checking if the death ratios start to become less that the cases ratios, but it is clear that this is not the cases currently, implying that we have not yet found a proper treatment.

A further and more rigorous analysis of this graph and more detailed data could also give us insight to how long it takes the coronavirus to cause death on average.

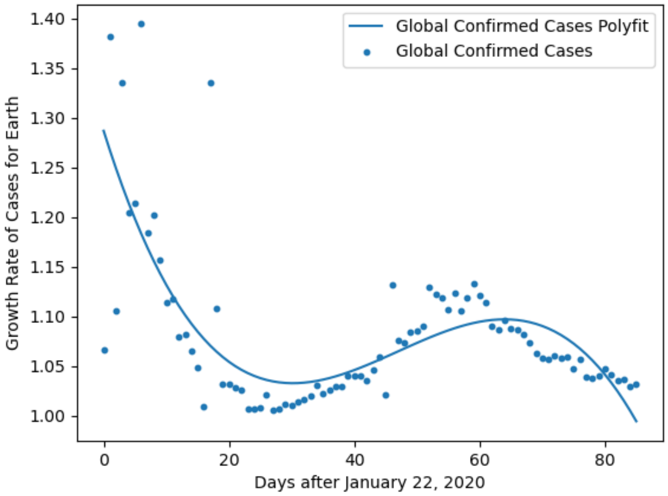


2.2 Growth Rate

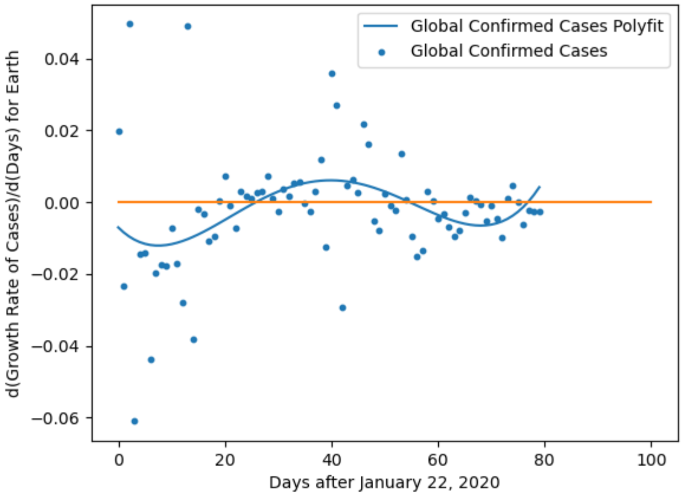
Now we can define our growth rate as

$$G_i^X = \frac{\Delta N_i^X}{\Delta N_{i-1}^X} \tag{2}$$

We are interested when this value crosses from $G > 1$ to $G < 1$, which corresponds to a point of infection. It is when the numbers of new cases each day begins to decrease rather than increase. Or in terms of calculus, when the second derivative becomes negative. The reason we do not just use the second derivative is that this data is noisy and approximating a high-order derivative would not work too well. Now plotting the growth rate:



If the graph continues to follow the polyfit curve drawn, then we expect the inflection point to occur soon, but we do not know if it will go back up as it did at around $i = 25$. In fact, we can expect it to go up, which is evident by plotting the approximated derivative [1] of the growth function.



[1] When I refer to the derivative I am using the central difference approximation give by:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

with $h = 1$ since we have a discrete input space (each day).

The derivative of the growth rate at the current time ($i \approx 80$) is following the same pattern as at $i = 25$ and it has gone above the zero-line, implying that the growth rate will increase again.

This is a dissapointing result since finding the true point of inflection will let us estimate where the number of cases will max out. If the point of inflection occurs at i, then it is reasonable to guess that the number of cases will max at $2 * N_i^X$ by the way logistic growth works.

2.2.1 Modelling with the Growth Rate

The curve generated by `np.polyfit` fits the growth rate pretty well, so it may be possible to create a model based on the growth rate. To do so, we just have to follow the definitions of the growth rate and the solve for N_i:

$$G_i^X = \frac{\Delta N_i^X}{\Delta N_{i-1}^X} = \frac{N_i^X - N_{i-1}^X}{N_{i-1}^X - N_{i-2}^X}$$
$$G_i^X(N_{i-1}^X - N_{i-2}^X) = N_i^X - N_{i-1}^X$$

and finally

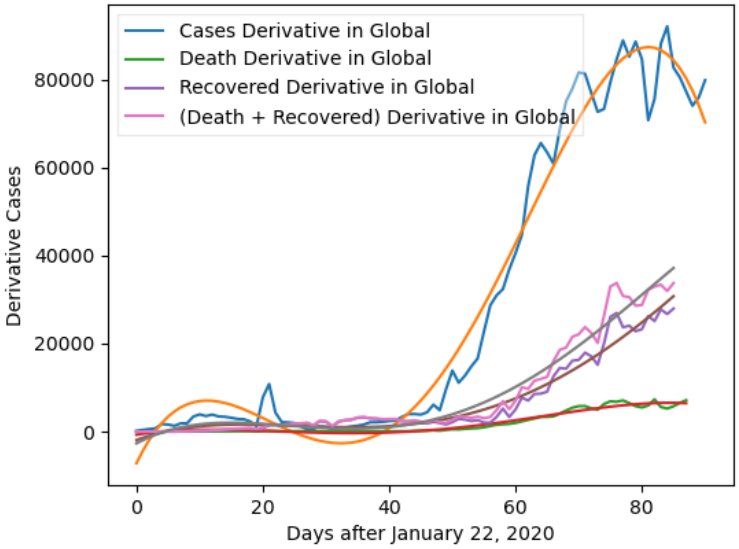
$$N_i^X = N_{i-1}^X + G_i^X(N_{i-1}^X - N_{i-2}^X) \tag{3}$$

But there are a lot of problems with a model like this. For one it is based on previous estimates, so any error in early estimates will propagate through to the later estimates, making the model extremely inaccurate for anything too far out of the domain of the data.

Also, as we showed before, the growth rate will likely not follow the polyfit curve outside the domain of the data.

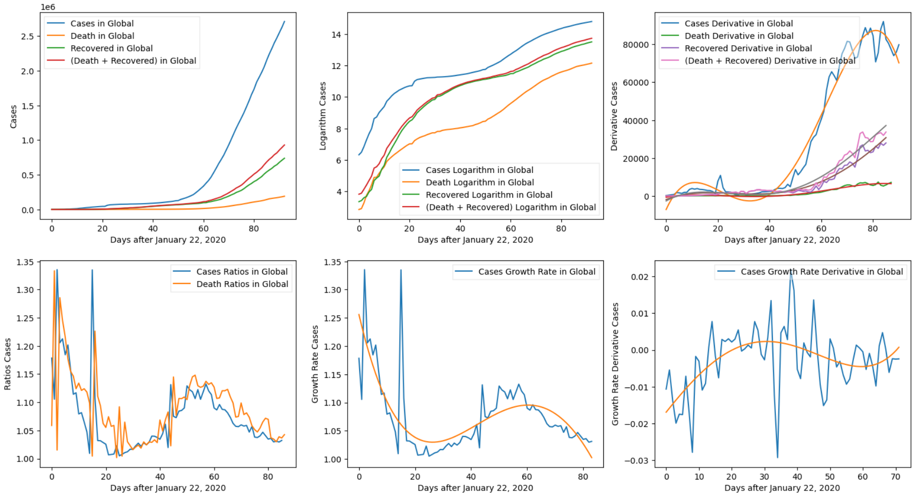
2.3 Derivative

The derivative (approximated) of N_i^X with respect to i, or $\frac{dN_i^X}{di}$, is an important value to look at since it can also help us classify the type of model the spread of coronavirus is following. In an exponential model, we would see $\frac{dN_i^X}{di} \propto N_i^X$, while in a logistic model we would expect $\frac{dN_i^X}{di} \propto (1 - N_i^X/N^X)N_i^X$. Now plotting the derivative:



As expected the derivative is never negative since that would meann the number of cases are going down, which is impossible (since the number of cases is defined as the number of all cases recorded and does not decrement from recoveries).

Putting this graph next to one of the number of cases, we can compare them to see proportionality and see what type of model the spread is following. I will do this with the Analysis Dashboard for Global Cases, which I talk about more in Section 2.5



The derivative is initially approximately proportion to the number of cases, which is seen by looking at the first and third graphs. But then the derivative starts to level off as expected by logistic growth. The derivative is starting to decrease implying a critical point. Now if we assume that the graph will continue to fall in this manner we can expect the derivative to go back to 0 at around $i = 120$, or about 4 months after January 22, which puts us at around the end of April. Once the derivative is back at zero there will be no new cases and the virus will have stopped spreading. But a crude prediction such as that one does not account for skew, since chances are that the graph will be right skewed, since that is how logistic graphs are. Accounting for this we can expect the curve to flatten out a bit later than the end of April.

Another option is that we can try to use an initial subset of the data to try to estimate the proportionality constant c by

$$\frac{dN_i^X}{di} = c(1 - N_i^X/N^X)N_i^X \tag{4}$$

where $(1 - N_i^X/N^X) \approx 1$ (in the initial subset), so $\dfrac{dN_i^X}{di} \approx cN_i^X$. After finding c, we can use the graph for the derivative to estimate N^X, assuming a perfectly logistic model. I will actually implement this in Section 2.6.

2.4 Fitting Polynomials to Subsets of the Data

The numpy module has a `polyfit` method that can take a bunch of data and fit a polynomial of a given degree to it. In theory, this can be used to create a model for the COVID-19 data, but there are many problems with this[2]. But we can still use a technique like this to gain insights into our data.

2.4.1 Finding the Best Degree for Polyfit

Just like any other model, we can measure how well the generated polynomial fits the data using some sort of loss function. I used mean squared error, defined by

$$MSE(\vec{p}, (N_i^X)) = \frac{1}{N}\sum_{i=0}^{N-1}(N_i^X - \vec{p}\cdot\vec{x}(i))^2 \tag{5}$$

for our polynomial model, where $\vec{p}$ is a vector containing the weights for our polynomial model, N is the number of samples. The function $\vec{x}(i)$ returns a vector defined by

$$x(i) = [i^m, i^{m-1}, i^{m-2}, \ldots, i^2, i, 1] \tag{6}$$

where m is the polynomial's degree. The vectors $\vec{p}$ and $\vec{x}(i)$ have length $m + 1$. Here x acts as a feature transformer.

Now to find the smallest m that has a good fit to our data, I track the ratio between the impovement since $m - 1$ and the original MSE with $m = 1$ and increment m until it is less than a theshold ϵ. Essentially I find the smallest m, where

$$\frac{MSE(\vec{p_m}, (N_i^X)) - MSE(\vec{p_{m-1}}, (N_i^X))}{MSE(\vec{p_1}, (N_i^X))} < \epsilon \tag{7}$$

[2] I go further in depth into these problems in Section 3.1

where the MSE for $m = 1$ is used as a scaling factor. After this we know that after $m - 1$ there is not going to be much improvement, so we do our polyfits on $m - 1$.

2.4.2 Correlations Between the Best Degree and Subsets of the Data

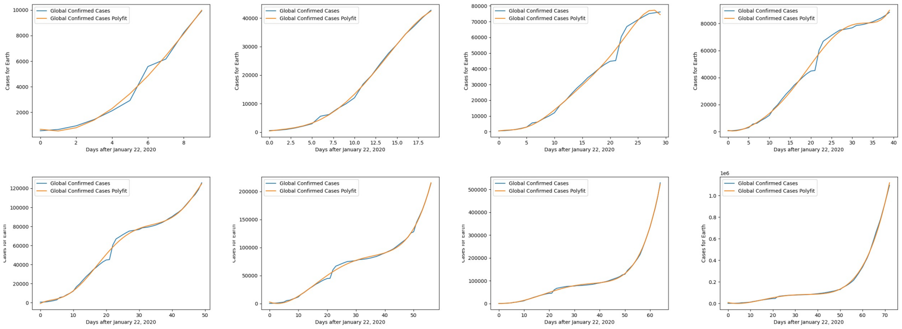
From now on, I will refer to m as the best degree to use `np.polyfit` with. I will use python subindexing, which is defined like this: If

$$x = [1, 2, 3, 4, 5]$$

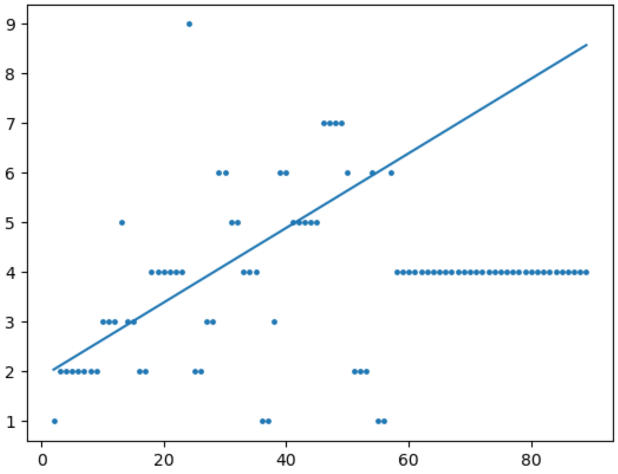
then

$$x[0 : 2] = [1, 2]$$

And it is [inclusive, exclusive]. Now define, $m(j) = \text{findBestDegree}((N_i^X)[0 : j])$. Incrementing up j and calculating $m(j)$, we are essentially finding a best fit polynomial of optimal degree for more and more of the data. Here is some of the polyfits for some subsets (incrementing by 10):



Now, there are two main factors in play as we increase j: there is more data to model, and the rate of change keeps increasing more as we get further in (N_i^X), requiring a higher polynomial degree to model. Both of these require a higher degree to model as j increases.



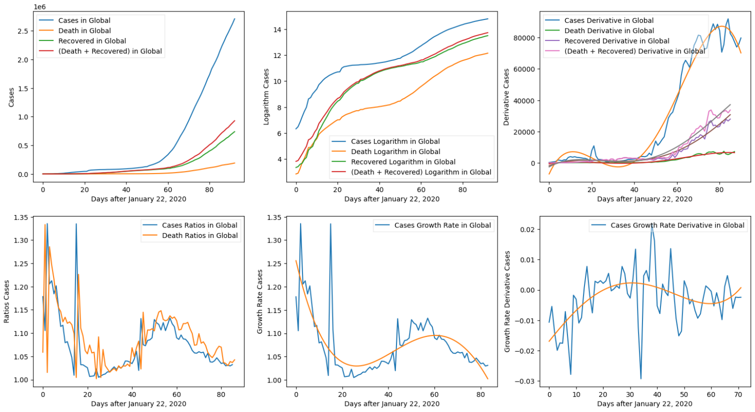
In this plot the linear regression line is drawn based on the only the $j < 50$. And the graph exihibits the predicted pattern while $j < 50$, increasing m with j, but after around $j = 55$, the m stays constant at 4. Cross-referencing this with our previous finding that there is a point of inflection around that point (since $dG_i^X/di = 0$), I am guessing that the rate of change of the function slowed down enough for a polynomial of order $m = 4$ to model the graph.

Another interesting thing to see is that the points on the graph are clustered, which makes sense since adding one point will likely not change m by at least one everytime, and the output space is discrete, causing clusters to from.

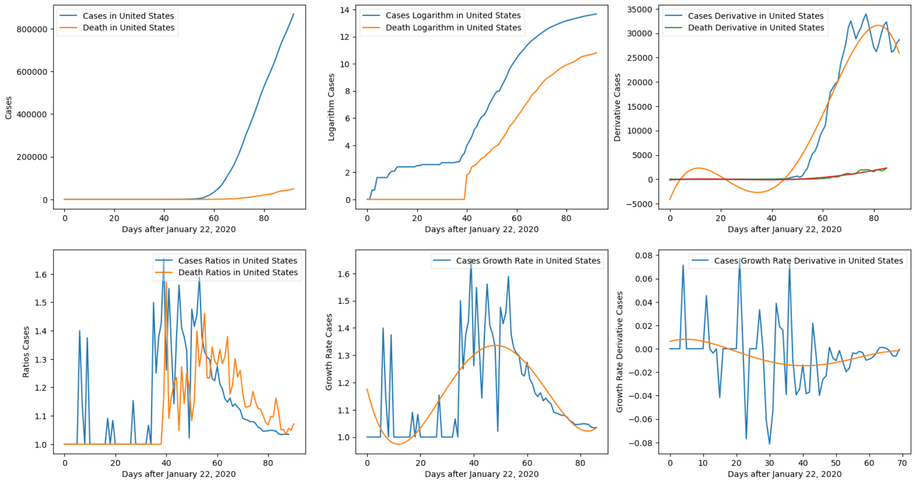
2.5 Country Analysis

To do an analysis of the countries, I will be using a ”dashboard” with a bunch of graphs that each convey different information. Many of the graphs are talked about in previous sections talking about the global data. But now we will focus in on other regions (different values for X).

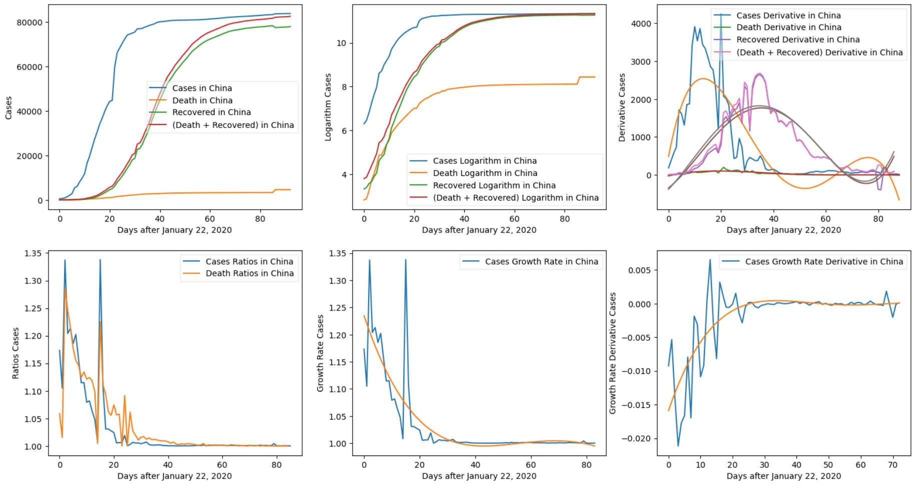
Many of the observations I make will be in relation to the global dashboard. One of the difference that apply to all countries is that the graphs are scaled down from the global scale, but since we are analyzing trends, the amplitude does not matter too much.



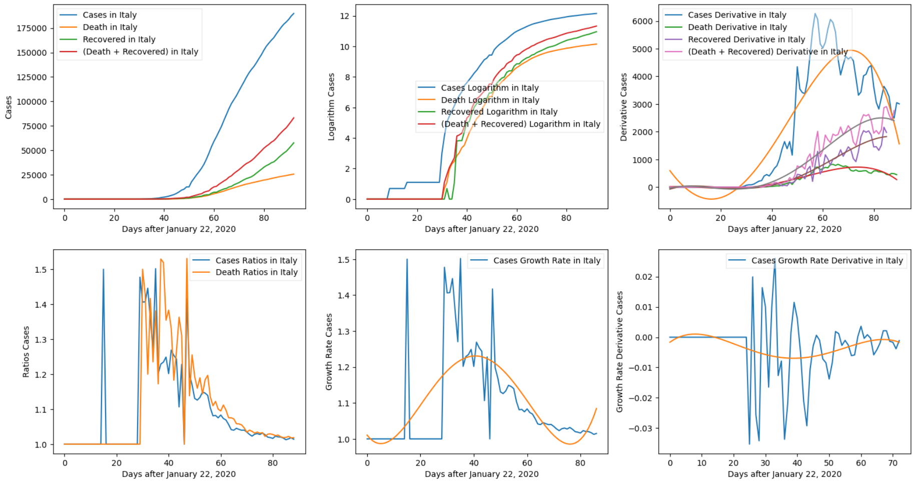
2.5.1 United States



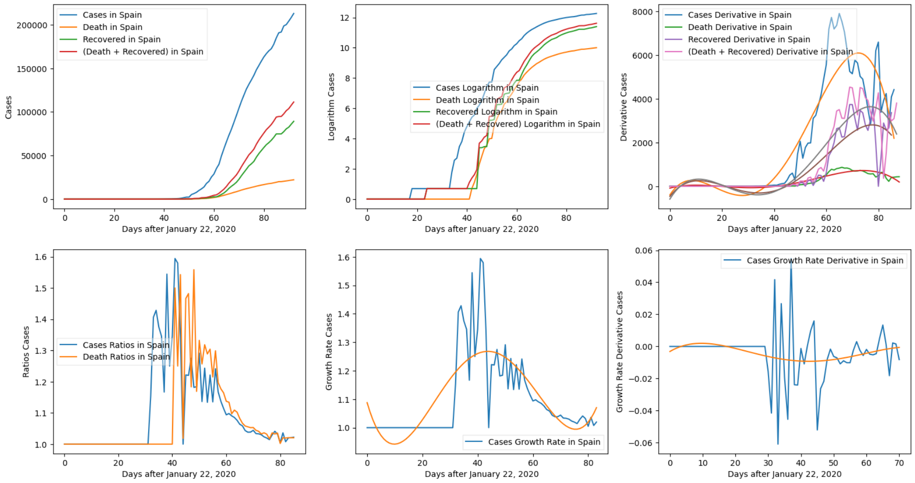
2.5.2 China



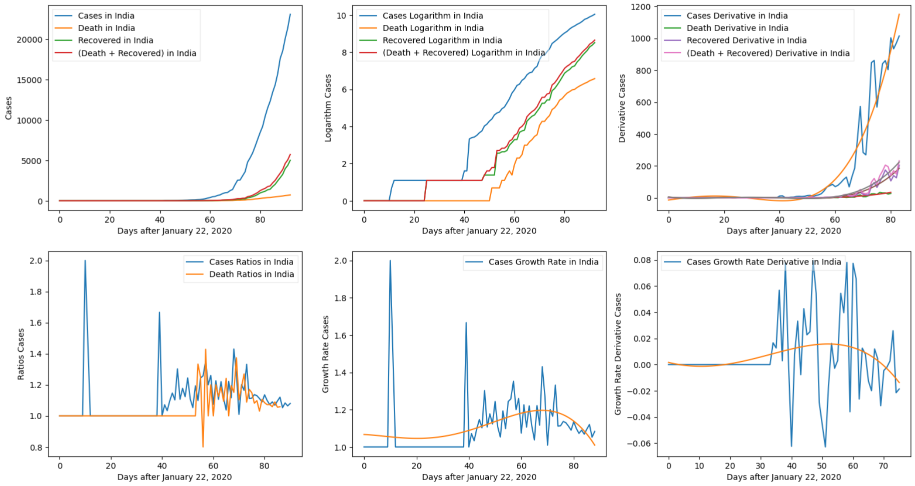
2.5.3 Italy



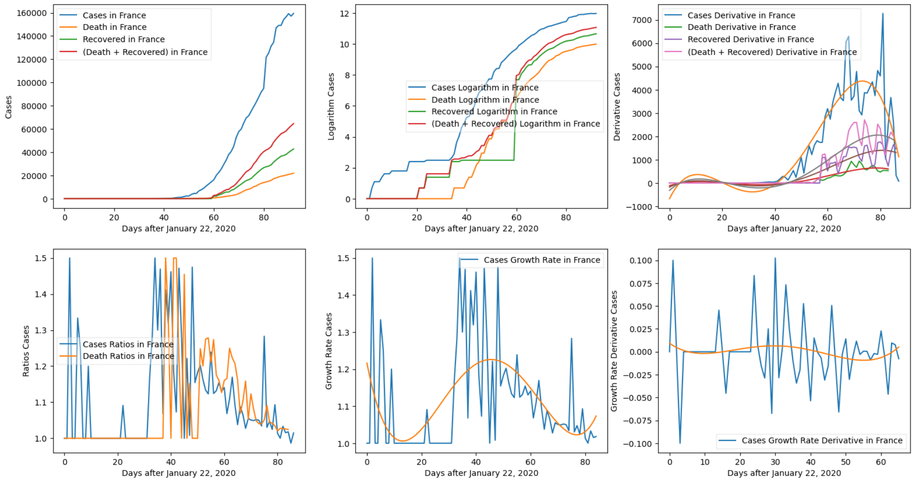
2.5.4 Spain



2.5.5 India



2.5.6 France



2.6 Auto Analysis

TODO: Code the auto analysis stuff

3 Modelling

3.1 Problems with Polynomial (Polyfit) Models

3.2 Gradient Descent

3.3 Logistic Growth Model

3.3.1 Using Gradient Descent

TODO: fix the negative sign and redo entire thing

3.3.2 Transforming the features and then performing Linear Regression

3.4 Markov Chain Model

TODO: Code Markov Chain Model

3.5 RLC Circuit Model

3.6 Neural Network

TODO: Code the neural network stuff

4 Conclusion

5 The Code

5.1 Libraries

5.2 Project Structure

5.3 RB Math Package

5.3.1 Gradient Descent and Models

5.3.2 Transforms

5.3.3 Plot Function

5.4 Plots