**Modeling Logistic Growth**

Nonlinear Least Squares Estimation of the Logistic Growth Function Using Scipy in Python — Using China’s Coronavirus data

In a previous article, I have explained how to model the spread of the Coronavirus outbreak using [Exponential Growth](https://towardsdatascience.com/modeling-exponential-growth-49a2b6f22e1f). It was limited to the first phase of the outbreak since the big limitation of Exponential Growth is that it never stops growing.

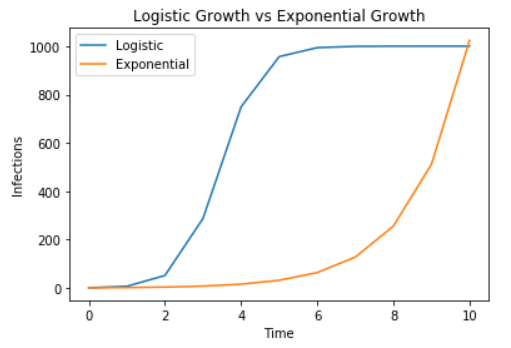
In this article, I present the next step for this model: the Logistic Growth model. If you want to follow along, you can get the [notebook here](https://jooskorstanje.com/modeling-logistic-growth-corona.html) and the [data from here](http://jooskorstanje.com/full_data_logistic.csv).

Why Logistic Growth?

Logistic Growth is a mathematical function that can be used in several situations. Logistic Growth is characterized by increasing growth in the beginning period, but a decreasing growth at a later stage, as you get closer to a maximum. For example in the Coronavirus case, this maximum limit would be the total number of people in the world, because when everybody is sick, the growth will necessarily diminish.

In other use cases of logistic growth, this number could be the size of an animal population that grows exponentially until the moment where their environment does not provide enough food for all animals and hence the growth becomes slower until a maximum capacity of the environment is reached.

The reason to use Logistic Growth for modeling the Coronavirus outbreak is that epidemiologists have studied those types of outbreaks and it is well known that the first period of an epidemic follows Exponential Growth and that the total period can be modeled with a Logistic Growth.

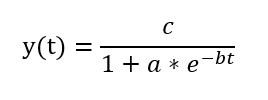


Logistic Growth vs Exponential Growth

The formula of Logistic Growth

Logistic Growth is characterized by the following formula:

Logistic Growth



The Logistic Growth Formula

In which:

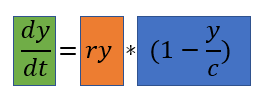
* *y(t)*is the number of cases at any given *time t*
* *c*is the limiting value, the maximum capacity for *y*
* *b* has to be larger than 0

I also list two very other interesting points about this formula:

* the number of cases at the beginning, also called *initial value*is*: c / (1 + a)*
* the maximum growth rate is at t =*ln(a)* / b and y(t) =*c / 2*

Deep-dive of the Logistic Growth formula for math lovers

For those who want to have a better mathematical understanding of the formula: the above function is actually derived from the below differential formula discovered by Pierre François Verhulst. I will explain it in small steps:



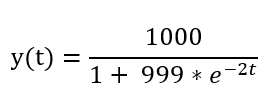
* The green part dy / dt indicates that this formula is not for the population size, but for the growth of the population.
* We can see that y and c are in the formula, so we understand that the growth of the population depends on the value of y (the population size) and the value of c (the maximum capacity)
* When y is equal to c (that is, the population is at maximum size), y / c will be 1. Therefore, the blue part will be 0 and hence the growth will be 0.
* When y is much smaller than c (the population is far away from the limit) the blue part will be almost 1. Therefore, the growth is defined by the orange part. The orange part is actually the formula for exponential growth.

A simple case of Logistic Growth

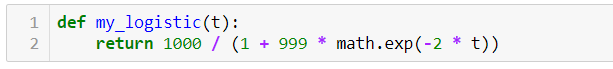
To make this more clear, I will make a hypothetical case in which:

* the maximum number of sick people, *c*, is 1000
* we start with an *initial value* of 1 infected person, so *c / (1 + a)*= 1, giving *1000 / (1 + a) = 1*, giving*a = 999*
* at the beginning of the infection, each sick person infects 2 other people, so the *growth rate* b = 2
* we will inspect the development of the epidemic from time 0 to time 10

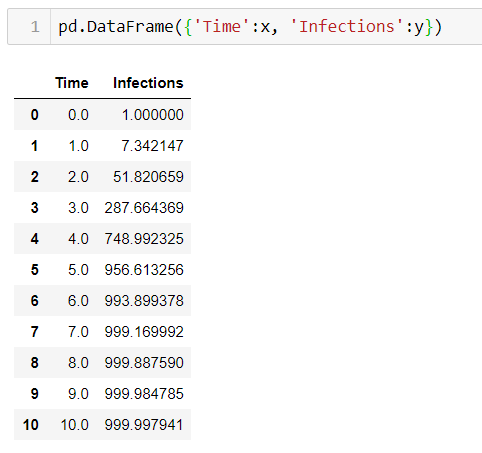
We first need to plug the values for a and b in the formula to obtain the formula for our specific epidemic:



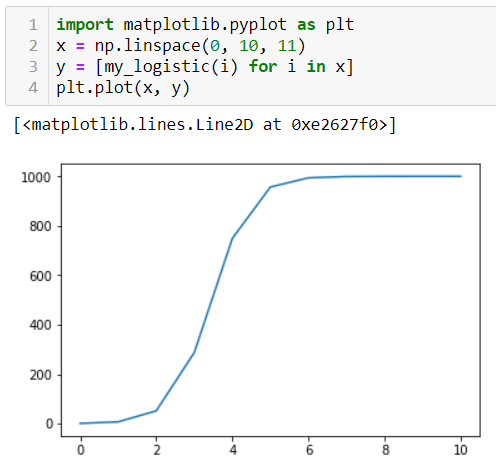
In Python this looks as follows:



Then we can use this formula to compute the value of y for each value of t from 0 to 10. When we do this, we obtain the following numbers of Infected people at every time step, as seen in the below table. This shows that in the beginning, there is fast growth of the number of infections, but then it slows down a lot and ends at the maximum capacity.



If we want to represent this graphically, we start to see a graph that looks a lot like the very alarming curves that we see concerning the Coronavirus:

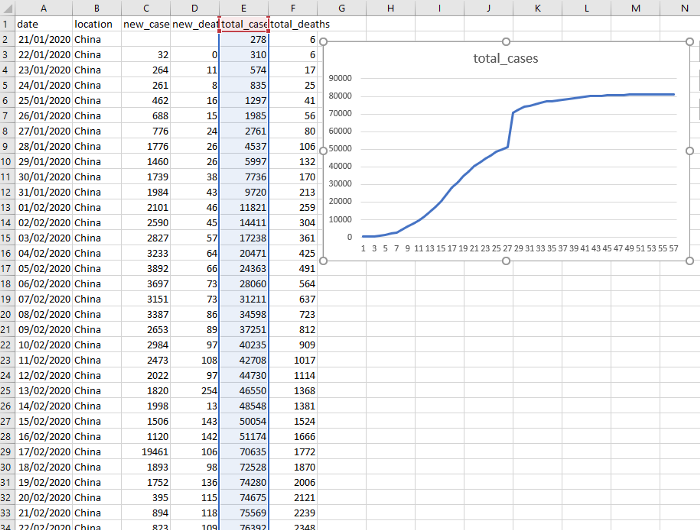


Graph of Logistic Growth

From Coronavirus data to Logistic Growth formula

Now, we know that this graph has more or less the right shape, but we need to make an additional step to make our analysis useful. We need to find *the real curve of the Corona epidemic*, by looking at the data from the epidemic spread.

In this case, I will take the data from China: since the growth has already strongly declined over there, the logistic curve will be quite a good match.



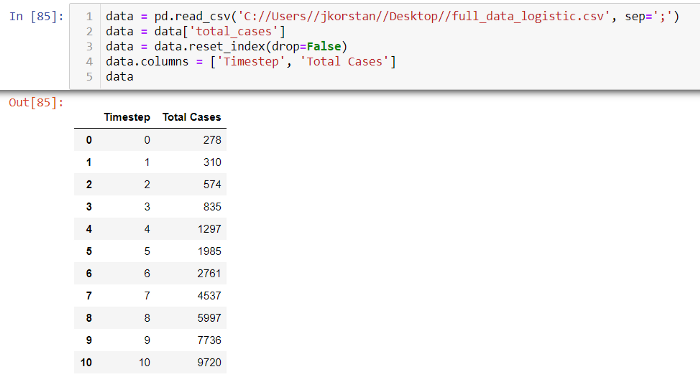
Extract of the Coronavirus Data for China. Source: <https://covid.ourworldindata.org/data/full_data.csv>

Nonlinear Least Squares for China’s Logistic Growth

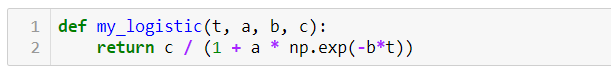
When looking at the data, we only have the number of cases per day. We also have the formula that we want to apply, but we do not yet have the correct values of the parameters a, b and c in the formula.

Unfortunately, it is not possible to rewrite the Logistic Function as a Linear Regression, as was the case for the Exponential model. We will therefore need a more complex method: Nonlinear Least Squares estimation.

**STEP 1** — Read in the data

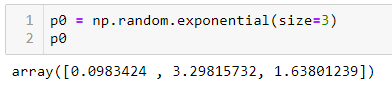


**STEP 2**— Define the logistic function that has to be fitted



**STEP 3** — Randomly initialize a, b and c

In this case I use np.random.exponential for this, but you can use anything you like. Attention though: your choice here may affect the performance of step 5.



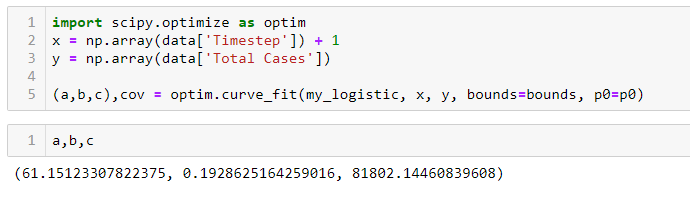
**STEP 4** — Set upper and lower bounds for a, b and c

Lower bounds for all parameters are 0. I set an upper bound for b on 3, because at my first try I let it free and it became way too high. a and c upper bounds did not negatively affect the curve fitting, so I let their bounds relatively high.

Logistic Growth

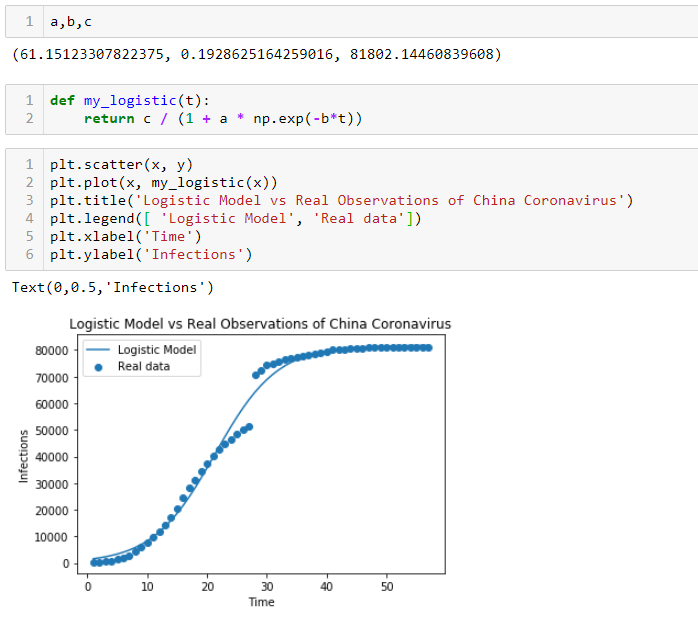
**STEP 5** — Use Scipy Curve Fit for Nonlinear Least Squares Estimation

In this step, Scipy does a Nonlinear Least Squares optimization and gives us the values for a, b and c that minimize the Least Square Error of our model.



**STEP 6**— Plot the fitted function vs the real data

As you can see in the graph below, the Logistic Model is really not that far from the actual Chinese Coronavirus data.



**STEP 7**— Conclusion

We have found a Logistic Function that is really quite close to the observed Coronavirus data from China. To really use this information in real life, it would be necessary to do a lot of model validation, compare accuracy and other performance metrics of different models and follow closely whether future trends follow the selected model.

This would go way beyond this article, which is merely trying to show how to fit a Logistic Growth curve, but still, using the theory, we can state some observations:

* According to this model, c is 81802, that would mean that the maximum limit for the number of infections of China would be 81,802.

We can also compute when the maximum growth rate occurred according to this model:

* the moment in time was: t =*ln(a)* / b = ln(61.5) / 0.1929 = day 21
* and the number of infections at that moment was y =*c / 2 = 40500*

With a thoroughly validated model, this type of information could for example be used by policymakers to estimate how to take the right measures.