Epidemic Modelling

Mini Project

Debanshu Das

20-04-2022

—

Epidemic Modelling

—

Project's Report

Collected Dataset

```python
# Here, we load the data directly from the website.

current_data = pd.read_csv('https://covid.ourworldindata.org/data/ecd
current_data_array = np.array(current_data)

# Now, we want to get a list of the countries in the spreadsheet to m

locations = np.array(current_data['location'])
country_list = list(dict.fromkeys(locations))
print(country_list)

# To visualize the data from a country of our choice, we need to find

country_ind = country_list.index('United States') # Type in the name
new_ind = current_data_array[:,1] == country_list[country_ind]

new_cases = current_data_array[new_ind,2]
new_deaths = current_data_array[new_ind,3]
total_cases = current_data_array[new_ind,4]
total_deaths = current_data_array[new_ind,5]

dates = np.linspace(1, len(new_cases), len(new_cases)) # Because the
# dates = current_data_array[new_ind,0] For other countries where upd
```

Here, we load the data directly from the website. \\

#https://ourworldindata.org/coronavirus-source-data

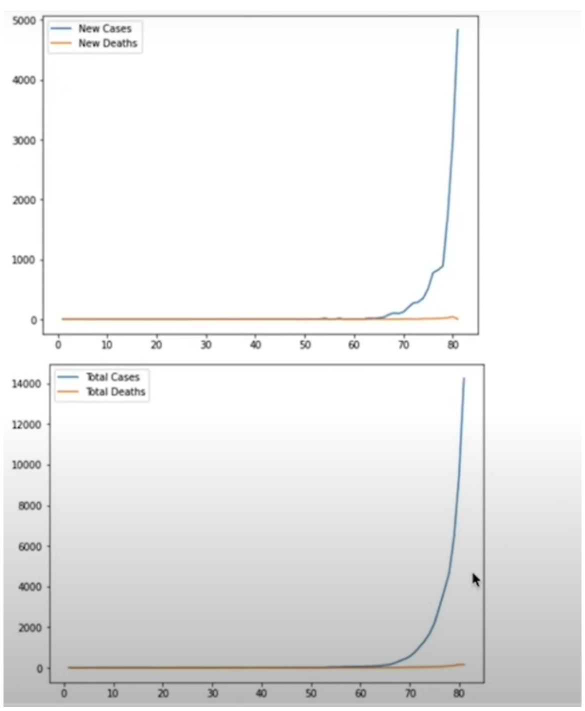
List of the countries in the spreadsheet to make it easier to visualize our data. The dataset is originally from European CDC but has been collected and organized by OWID.

01-11-2021	Armenia	1071	51	309397	6379	12845	324	26214
02-11-2021	Armenia	1232	62	310629	6441	12560	329	26392
03-11-2021	Armenia	2045	50	312674	6491	12531	340	26371
04-11-2021	Armenia	2330	41	315004	6532	12554	343	26098
05-11-2021	Armenia	1835	50	316839	6582	12293	350	25787
06-11-2021	Armenia	2177	46	319016	6628	12277	344	26002
07-11-2021	Armenia	1417	42	320433	6670	12107	342	25065
08-11-2021	Armenia	810	30	321243	6700	11846	321	24691
09-11-2021	Armenia	1121	62	322364	6762	11735	321	24295
10-11-2021	Armenia	1675	69	324039	6831	11365	340	23896
11-11-2021	Armenia	1482	36	325521	6867	10517	335	23071
12-11-2021	Armenia	1309	70	326830	6937	9991	355	22284
13-11-2021	Armenia	1251	39	328081	6976	9065	348	21342
14-11-2021	Armenia	882	40	328963	7016	8530	346	20637
15-11-2021	Armenia	378	39	329341	7055	8098	355	19944
16-11-2021	Armenia	572	52	329913	7107	7549	345	19284
17-11-2021	Armenia	982	46	330895	7153	6856	322	18221
18-11-2021	Armenia	1019	30	331914	7183	6393	316	16910
19-11-2021	Armenia	799	29	332713	7212	5883	275	15874
20-11-2021	Armenia	870	41	333583	7253	5502	277	14567
21-11-2021	Armenia	492	25	334075	7278	5112	262	13642
22-11-2021	Armenia	272	30	334347	7308	5006	253	13104
23-11-2021	Armenia	531	48	334878	7356	4965	249	12514
24-11-2021	Armenia	860	23	335738	7379	4843	226	11699
25-11-2021	Armenia	592	40	336330	7419	4416	236	10809
26-11-2021	Armenia	675	40	337005	7459	4292	247	10175
27-11-2021	Armenia	517	26	337522	7485	3939	232	9441
28-11-2021	Armenia	409	29	337931	7514	3856	236	8968
29-11-2021	Armenia	189	21	338120	7535	3773	227	8779
30-11-2021	Armenia	398	32	338518	7567	3640	211	8605
01-12-2021	Armenia	502	43	339020	7610	3282	231	8125
02-12-2021	Armenia	558	21	339578	7631	3248	212	7664
03-12-2021	Armenia	399	13	339977	7644	2972	185	7264
04-12-2021	Armenia	419	25	340396	7669	2874	184	6813
05-12-2021	Armenia	327	14	340723	7683	2792	169	6648
06-12-2021	Armenia	95	8	340818	7691	2698	156	6471

Cases and Deaths

-Once, we have the data loaded we want to collect the list of the countries that will be represented in the data, to visualize the data more easily, by country or by the whole world.
-For now, taking up the United States, Also printed out the list of the various listed countries, So you can visualize other countries as well other than the United States.
-Then, let's look into our Data, So analysing newly recorded cases and deaths each day and the total cases and deaths.

Plot



we'll be indirectly using an approach called logistic regression to model exponential growth. Using population data from the US, we'll look at how coronavirus might spread without any intervention. To do this, we'll focus on the data on new cases, and will look at how the number of new cases compares to the available medical resources. We didn't model deaths here. In order to figure out the exact exponential formula, we apply the natural logirithm to the number of new cases over time to get a fairly linear line. This makes it easier to find our formula - all we need to find is the slope!

From there, we can model the actual growth by multplying the number of days it's been since we started recording outbreaks by the slope, and then take the exponential of that number. You can see all this below!

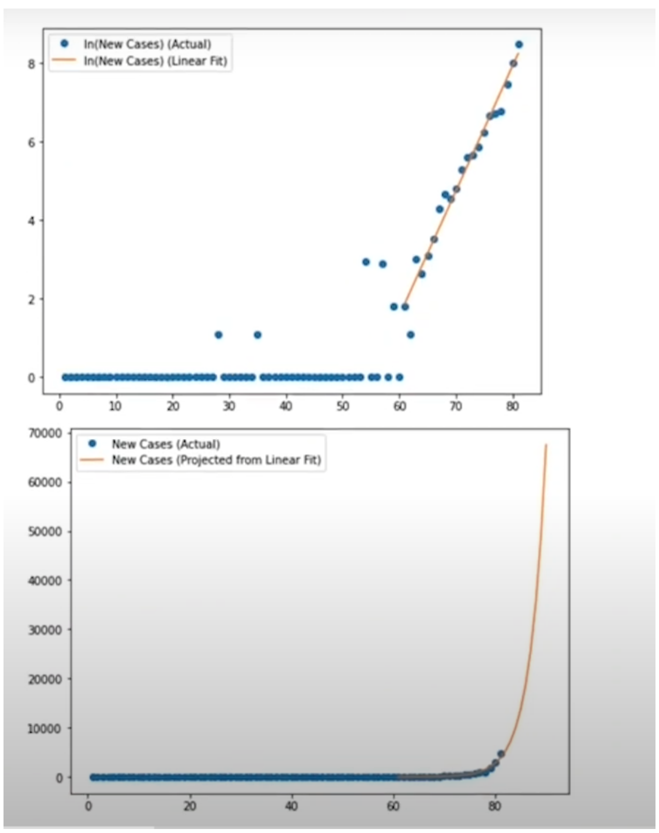
Now, this approach applies to a point but has several limitations - as fewer people remain healthy, there are fewer people to infect, and the disease spread plateaus.

We're going to start by actually taking the natural log of this data because we think that the data is exponential they should turn our dataset into a relatively flat line, something that we can more easily calculate the slope of.

So, this slope using linear regression frames the cluster of growing new cases, using sklearn. (certain region of data, that because that first of the data, particularly useful to us , we want to model starting with the first day of cases and go from there....)

```
dates_into_future = np.linspace(1, 90, 90) # Now, we want to look at future dates
fut_pred = model.predict(dates_into_future[60:, np.newaxis]) # Making predictions in future dates
```

Now, from the above we will be knowing, there will be how many cases in the next ten days. So, we have the first 81 days, and we want to look through day 90. to do this, we just multiply themselves the dates themselves the number of days by that slope, we originally found from the Linear Regression, that will give us the log of the actual no of cases which continues that straight line from the first part. For the actual no.of cases we take the exponential of that , through these days in the second graph it just grows exponentially.



SIR Model

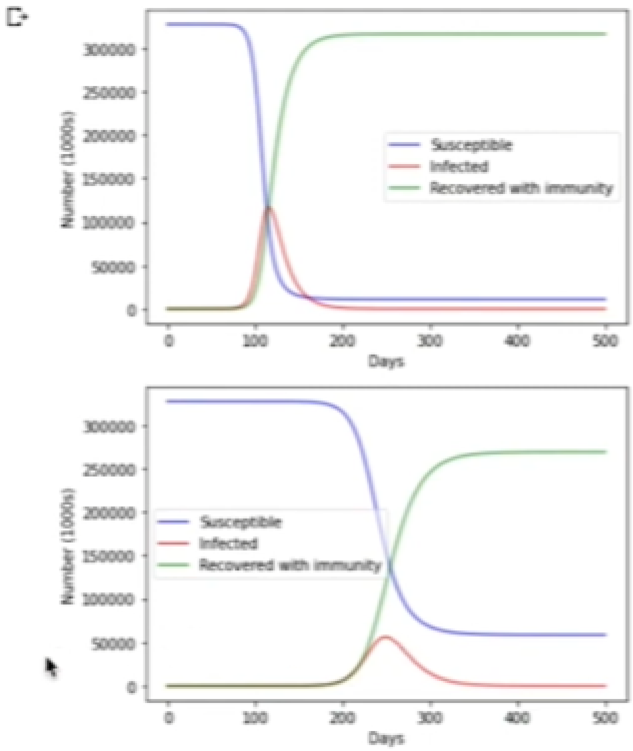
This model assumes that there is a certain population of people who can catch the disease (S), a population of people who are infected (I), and a population of people who are recovered and have immunity (R).

This model relies on solving a set of differential equations, which I'm not going to get into the details of, but in short:
The number of people who can catch the disease decreases based on their contact with those infected.
The number of people infected increases and decreases based on the number of people they have contact with and how long it takes to recover.

The number of people who are immune increases based on the time it takes to recover when infected.
This is a better model, but it also makes some assumptions and omits some important details.
Namely, it doesn't account for death. The total population does not change, just the ratio of the number of susceptible, infected, and immune people.

```
population = 327200000 # Population of the United States

initial_infected, initial_recovered = 1, 0 # Initial conditions for infected and recovered people

initial_everyone_else = population - initial_infected - initial_recovered # Initial conditions for everyone else.

initial_conditions = initial_everyone_else, initial_infected, initial_recovered

n_days = 500 # Days over which to integrate
time = np.linspace(0, n_days, n_days)

contact_rate = 0.25 # Contact Rate - We don't know this for coronavirus, so use it as a relative term for comparison.
recovery_rate = 1/14 # Recovery Rate -

# The SIR model, integreated over 500 days.

def SIR(initial_conditions, t, population, contact_rate, recovery_rate):
  S, I, R = initial_conditions
  dS = -contact_rate*S*I/population
  dI = contact_rate*S*I/population - recovery_rate*I
  dR = recovery_rate*I
  return dS, dI, dR

result = odeint(SIR, initial_conditions, time, args=(population, contact_rate, recovery_rate))
```

(Limitations:
We don't have consistent data, where you can infect other people and we don't know how long immunity lasts.)



(Recovery rate is 1 over the number days to become immune to disease.)

Conclusion:

For **Model 1** here, Everyone is infected and Everyone recovers. But the problem here is, everyone gets it really fast. That however puts a strain on the healthcare.
This might lead to more people may die due to unavailability of beds and resources, increased fatality rate.

For **Model 2** being implemented**,** What if the social distancing is implemented and the contact rate between people is decreased, so disease spread more slowly. So implementing it in the second graph.

*Fewer people end up getting the disease. Maximum infected people w.r.t time is much lower. This is flattening the curve.

-However, The model may not predict the exact absolute analysis, but it will work depending upon the integrated factors.

- We see the strong rapid peak in the epidemic, followed by decaying oscillations to the final endemic state.

-Eventually, these oscillations disappear, and the system settles in to the stable endemic equilibrium.

-The most characteristic feature of this system is the existence of two very different time scales -- the short time scale of a single epidemic, and the long generational time scale on which susceptible are replenished.

- Once the endemic state is established, any small perturbations which occur will be damped oscillations, with frequency and damping rate.

References:

WHO Data on COVID-19: https://www.who.int/emergencies/disea...

Logistic growth in Epidemiology: https://arxiv.org/pdf/2003.05447.pdf

SIR Models in Epidemiology: https://en.wikipedia.org/wiki/Compart...

Python Tutorial Used to Create Colab Example: https://scipython.com/book/chapter-8-...

Mathematical Explanation of SIR Model: http://www2.me.rochester.edu/courses/...

Recent Paper with more complex modeling on how preventative efforts impact SIR models - https://www.imperial.ac.uk/media/impe...