COVID-19: Comparing data to a simple compartment model

Aksel Hiorth

Oddbjørn Nødland

August 2022

Learning objectives. By completing this project, the student will:

- Learn to import data into pandas DataFrames.
- Write simple functions to generate plots, and inspect data visually.
- Constrain model parameters by comparing model output with data.
- Reflect on strengths and weaknesses of simple models for complex phenomena.

Project overview. In this introductory project we are modelling the first 250 days of the Corona virus (SARS-CoV-2) outbreak. The model we are going to use is probably the simplest one possible, and you are encouraged to keep this in mind as you reflect on what the model can, and cannot, say.

All later projects in the course MOD510 will look very much like this one. Therefore, working on this project should give you a good start.

1 Downloading COVID-19 statistics

Data on the Corona virus are readily available, e.g., good visualizations of the global situation can be found at the Johns Hopkins Coronavirus Resource Center.

For the purpose of this project, we have already downloaded country-level data for you, and stored it in a processed format in the text file `data/corona_data.dat`. Data for the Hubei province in China, where it is believed that the virus first arose [6, 3, 5], is included in the text file. The reason for doing so is that while the original data were organized by date since January 22 2020, we would like to plot the data versus the time of *the first confirmed case.* This makes it easier to apply the same model to different locations. Note that we have *not* downloaded the most recent data. If you wish to inspect the latest data, you are encouraged to checkout the Github repository Center for Systems Science and Engineering (CSSE) at Johns Hopkins University.

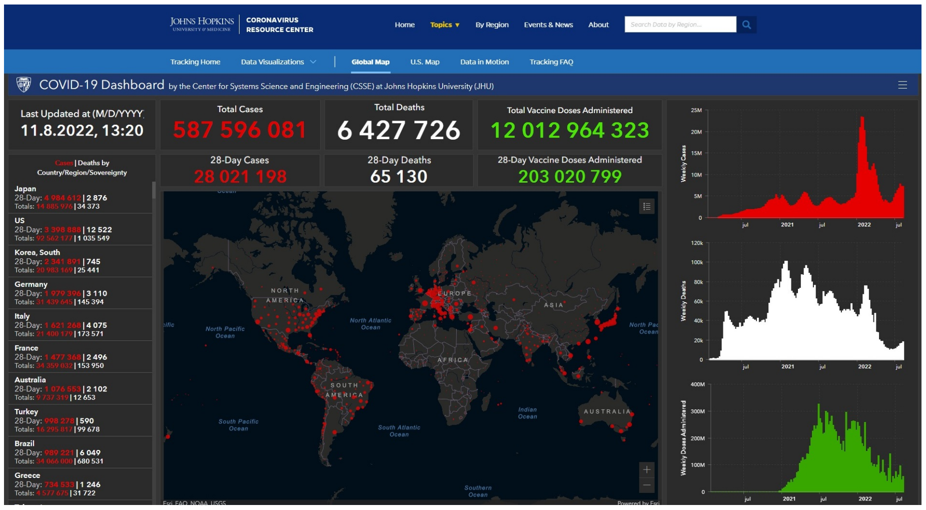

Figure 1: A screenshot of the COVID-19 Dashboard from Johns Hopkins Coronavirus Resource Center.

2 Exercise 1: Loading data into data frames

Next, we want you to get familiar with the data. The relevant text file, `corona_data.dat`, is located in a separate directory called `data`. We are going to use the pandas library to read the data from the text file into a DataFrame object. Pandas is a very flexible library, but we will only use it in a very limited way in this project.

A note regarding the use of libraries.

There are many Python libraries out there, but a good advice is to use no more libraries than necessary. For scientific computing there are a few libraries that are used by almost everyone, the most important being NumPy. Numpy is based on well-optimized C code, and it is designed to perform heavy numerical computations faster than native Python. If you see code that uses a different library to achieve something, ask yourself if the same thing could have been done with Numpy. Another important reason to prefer Numpy is that the syntax is known to almost everyone, which makes it quite easy for other people to read and understand your code by visual inspection.

Of course, some tasks are best performed by other libraries. As you will see in this introductory project, pandas lets you read data from files using very few lines of code. When doing more advanced tasks involving the file system and installed programs on your computer, the pathlib library is very powerful, as is the os module. For plotting, matplotlib is where most Python programmers begin their journey.

Part 1. Execute the following lines of code

```
df = pd.read_csv("../data/corona_data.dat", sep="\t")
print(df)  # In Jupyter notebooks, you get nice output with just: df
```

- What does the code do? Compare the output with what you see if you open the file `corona_data.dat` in a text editor.
- What happens if you remove `sep="\t"`?

Part 2. So far, we have read COVID-19 data from every country, however we are not usually only interested in a subset of the data. Run the following code:

```
is_specific_country = (df["LOCATION"] == "Afghanistan")
df = df[is_specific_country]
print(df)
```

- Explain what the code does.
- What kind of an object is `df`? (Find out how you can get Python to tell it to you if you did not know it already).

Part 3.

- Write a Python function that takes as input 1) the full DateFrame with data and 2) a specific place country or province. The function should only return the data for that location.

Alternatively, you may try to specify the path to the datafile as input argument instead of assuming you already have a DataFrame.

- Can you think of any disadvantages of doing this?

A note regarding copying code.

To copy and paste code can often help you get quickly started on a programming problem. We all do it. However, if you really want to mature as a programmer, you should try to limit how often you do it. First, if you are not careful about "borrowing" code from others, it could have legal repercussions. Second, copying code might prevent you from achieving a good understanding of the problem you are working on: especially when using code developed by others, you should strive to *understand* the individual steps of the code, as far as it is practical to do so (usually, you do not have to worry about internal implementation details of the core libraries you are using). If you are able to get a good understanding of what a piece of code does, you will later be able extend it in new directions, and/or to simplify it by removing unnecessary steps.

Even when you write all the code yourself, you may find yourself copying and pasting the same (or very similar) code many times over. This is usually a clear sign that you should think about making a function. By moving repeated code blocks into a function, you can reduce code duplication. If you ever need to make a change to a function (from experience, this is very likely), all places making use of the function will automatically be updated. In other words, the code becomes more maintainable than if you had to change the code manually in many places at once.

3 Exercise 2: Visualizing data

To make figures, we suggest that you use Matplotlib.

Plotting with pandas?

Note that pandas have some built-in functionality to make plots. This functionality is built on top of matplotlib, and it can sometimes be useful for quickly creating a figure without having to type so much. However, as you spend more time fine-tuning plots, you might conclude that it is better to work directly with matplotlib.

Part 1.

- Use the function you made in the previous exercise to inspect the data for Norway. Specifically, look at the columns `ELAPSED_TIME_SINCE_OUTBREAK` and `CONFIRMED`.
- What kind of an object is each of the columns? (assuming all of the data are stored in a DataFrame)

- Make a 2D scatter plot showing the number of confirmed cases of COVID-19 in Norway versus time.

Part 2.

- Create a new function which takes as one input argument the name of a country or province, and makes a scatter plot of the number of confirmed COVID-19 cases versus time for that country or province.
- When writing this second function, try to re-use the one you made previously.

Tip: For increased flexibility in making plots, it may be a good idea to return either a figure or Axes object from the function. If you do that, you can continue to customize the figure outside the function.

Separation of concerns.

It can be challenging to design good functions. A rule-of-thumb is that functions should not have too many responsibilities; ideally, any one function should concentrate on doing one thing very well, and not too many things at once. As an example, when working with data it is almost always a good idea to separate the pre-processing of raw data from code that works with formatted data. Similarly, it is usually not a good idea to mix code that does mathematical model calculations with code that makes plots.

Taken to the extreme, some advocates of functional programming recommend always using *pure functions*, that is, functions which always return the same output given the same input, and which have no side-effects. In practice, this will most likely not be feasible to achieve, but it is worth thinking about. If you are able to create a small library of reusable pure functions, life can become so much easier!

Part 3.

- Apply the function you just made to create a scatter plot for a different country than Norway. Include in this new figure a suitable title as well as axis labels; especially labels are important in order for others to understand what you are plotting.

To add the title and labels, you have essentially three options:

1. Use the input name of the country / province in the title and hard-code the rest inside the function.
2. Let the user pass in a title and/or axis labels as additional function arguments.

3. Do not change the function at all, instead modify the figure object after returning it from the function.

Discuss advantages and drawbacks of each of the three mentioned approaches.

Part 4.

- Make a plot showing the number of confirmed cases since the time of the first outbreak in the Hubei province in China.
- Compare it to the data for Norway, either in the same figure, or in two separate figures side-by-side.

Figure 2 shows a possible solution.

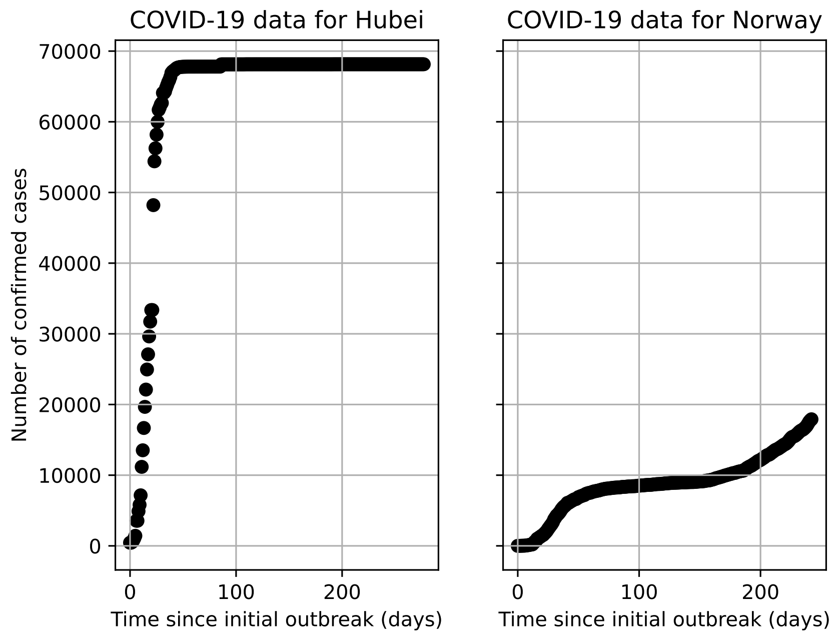

Figure 2: COVID-19 data for Hubei and Norway.

4 Modelling COVID-19: The SI-model

Compartment models [2] are widely used to study how an epidemic disease might spread in a population. In these models, the total population is partitioned into compartments based on a set of possible "disease states", such as "Susceptible", "Infected", "Recovered", and "Dead". Differential equations are set up to describe how individuals "flow" from one compartment to another. The equations can be either deterministic or stochastic. While stochastic models are more realistic, they are also more challenging to use and interpret. In this project, we will only consider deterministic models.

We start by studying the SI-model, which consists of only two compartments:

1. S - Susceptible: people at risk of infection.
2. I - Infected.

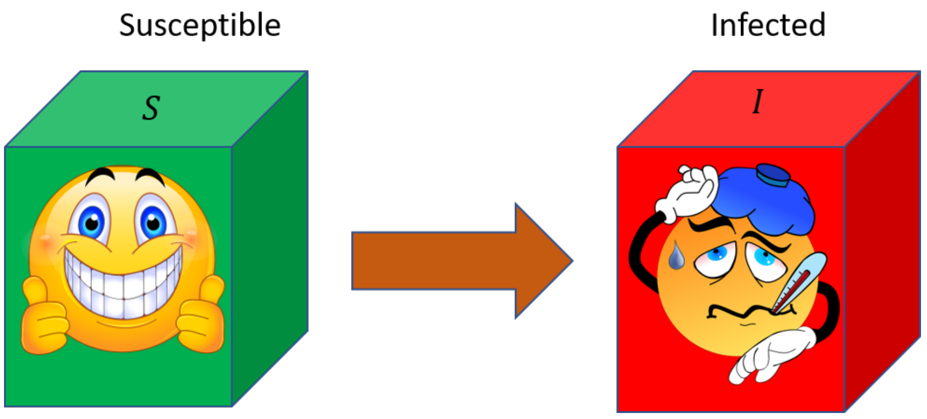

Figure 3: The SI-model: All individuals are either "Susceptible" or "Infected". Once you become sick, there is no recovery.

Let N denote the total population size. For each time t, let $S(t)$ denote the number of susceptible people, and $I(t)$ the number of infected people. To develop a model, we need to calculate the rate of flow between the two compartments in figure 3. We start by making some observations:

- During each time interval Δt, a certain number of individuals will come into contact with each other.
- We only care about susceptible (healthy) - infected encounters, because that is the only scenario in which the number of infected people can increase.
- Whenever a healthy person meets an infected person, there is a certain probability that the healthy person becomes infected.

We shall take our imagined population to be *well mixed*, meaning that pairs of individuals interact with equal probability. Let $\mathcal{C}(N)$ denote the rate at which *any* individual in the population contacts *any* another individual, i.e., the average number of contacts made per unit time. Then, we can estimate the change in the healthy population from time t to $t + \Delta t$ as:

$$S(t+\Delta t) - S(t) = -\mathcal{C}(N) \cdot \Delta t \cdot p \cdot q \cdot S(t)\,. \tag{1}$$

where p denotes the conditional probability that a given contact is between a susceptible and infected individual, and q is the probability that such an encounter leads to disease transmission. Because of the well-mixed condition, we can assume that $p = I(t)/N$. Thus, the remaining challenge is to estimate $\mathcal{C}(N)$ and q. In principle, both of these parameters may vary in time, but for now we shall regard them as constant. By merging the two factors into a single transmission rate, β, we get

$$S(t+\Delta t) - S(t) = -\beta \cdot \Delta t \cdot \frac{S(t)I(t)}{N}\,, \tag{2}$$

Finally, by dividing by Δt and letting $\Delta t \to 0$, we obtain the following ordinary differential equation (ODE):

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t} = -\beta \cdot \frac{S(t)I(t)}{N}\,. \tag{3}$$

Similarly, the evolution of the sick population is given by:

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = +\beta \cdot \frac{S(t)I(t)}{N}\,. \tag{4}$$

This last equation can also be derived at once from the relation $N = S(t) + I(t)$.

How to interpret β?.

By saying that β is constant, we have several very strong assumptions:

- People make the same number of contacts regardless of the population size.
- People make the same number of contacts independent of time.
- The probability of becoming infected never changes.

In reality, β is time-dependent, as it implicitly accounts for a lot of biomedical, physical, and sociological factors. For example, in the beginning of an outbreak, β is likely to be large, because people might not yet understand the severity of the situation, or they may be in denial. As people start to realize the danger and fight back against the disease, β will most likely decrease. However, as we have seen with COVID-19, new strains of the virus may appear and cause transmission rates to go up again. Another complication is that people perceive the threat from the virus very differently, and at least in in some countries, this seems to be influenced by political factors.

5 Exercise 3: [OPTIONAL] Analytical solution

Clearly, if there are no infected individuals at time zero, the above equations predict that nothing will happen later either. We shall therefore assume that the initial number of sick is close to one; typically $I_0 = I(0) = 1$.

Part 1.

- Show that the analytical solution to the SI-model, equations (3) and (4), is

$$S(t) = \frac{(S_0 + I_0)\frac{S_0}{I_0}\exp(-\beta t)}{1 + \frac{S_0}{I_0}\exp(-\beta t)}\,, \tag{5}$$

$$I(t) = \frac{S_0 + I_0}{1 + \frac{S_0}{I_0}\exp(-\beta t)}\,, \tag{6}$$

where $S_0 = S(0)$, and thus $S_0 + I_0 = N$.

Hint: A good way to start is to insert $I(t) = N - S(t)$ into equation (3).

6 Exercise 4: Is the model any good?

"All models are wrong, but some are useful" is a famous quote attributed to G. E. Box [1]. The only way we can investigate if a model is useful is to compare it with data. Clearly, the SI-model presented above is very simple - it contains a *single* parameter, β. Can the model still be useful? That is, can we use it to learn something about the spread of a virus, and can this insight lead to, e.g., preventive measures?

Part 1.

- Make a Python function that finds the solution to both equation (5) and equation (6) as a function of time.
- The function should not contain any hard-coded parameters.

To be able to do this, the function need to take as input the initial condition as well as the desired report times at which to calculate the solutions. For example, one possible function signature could be:

```python
def calc_SI_model(report_times, S0, I0, beta):
    """
    :param report_times: A list or array of times at which to compute solutions.
    :param S0: The initial number of susceptible people: S0=S(0).
    :param I0: The initial number of infected people: I0=I(0).
    :param beta: The disease transmission rate.
    :return: A tuple that holds S(t) and I(t).
    """
    pass  # Type your code here...
```

The string surrounded by triple quotes which immediately follows the function definition is an example of a docstring. It is regarded as good practice to document your (non-obvious) Python functions with docstrings. This way, users of your code may easily learn about how the function is supposed to work by typing `help(name_of_function)`.

Assuming we finished implementing the above function, it can be invoked like this:

```
St, It = calc_SI_model(report_times, S0, I0, beta)
# Continue to use St and It in the code below...
```

Part 2.

- Make an example figure where you plot the analytical solution to the SI-model together with the actual number of confirmed cases observed for a given country (in the same figure).
- Try to re-use the functions you made earlier when you do this (to do so, you may find that you have to change something).

Part 3. Let us first consider the Hubei province in China where the city of Wuhan is located.

- Assume for simplicity that $I_0 = 1$. Adjust S_0 and β to investigate whether it is possible to match the reported number of confirmed cases in Hubei with the SI model.

Part 4. In the database there are also data from the Corona virus outbreak aboard the cruise ship "Diamond Princess". There were 3711 people on the ship, of which 712 were infected [4].

- Let $S_0 = 711$ and $I_0 = 1$. Run the SI-model with the same β value you found for Hubei. What do you observe? (You should be able to generate curves similar to the ones shown in figure 4).

Part 5.

- Do the same for Norway. Which value of β do you need to use in order to find a reasonable match for Norway?
- Does the model fit the data better or worse in this case than it did for Hubei?

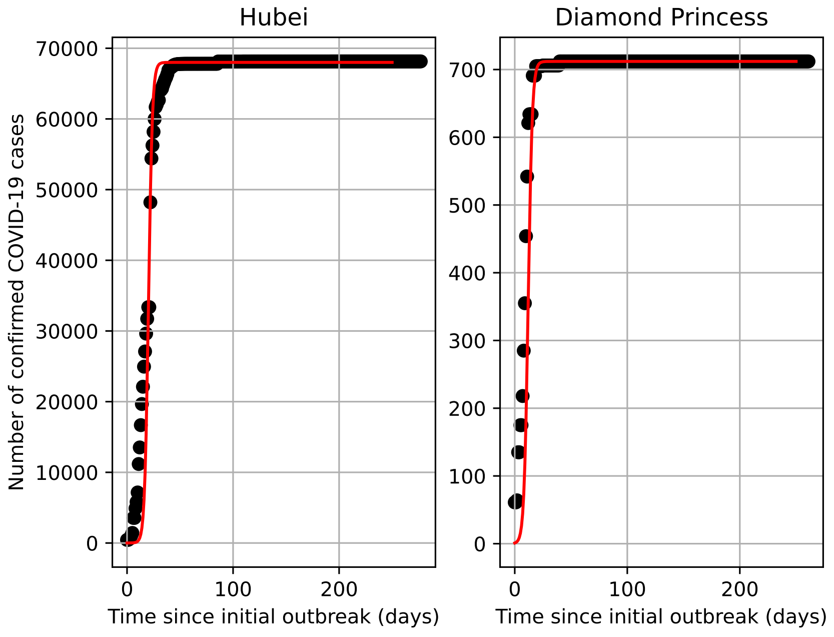

Figure 4: COVID-19 data (black circles) and model (red, solid lines) for Hubei and Diamond Princess.

7 Exercise 5: Extending the model

When doing exercise 4 you might have noticed that the SI-model works quite well in situations where the disease initially spreads quickly, and then stabilizes. However, a clear weakness with our approach is that we tuned the number S_0 to the data, essentially setting it equal to the final number of confirmed cases. In reality, S_0 should be the total population of a country, or at least the total number of people in a city where there is a large outbreak. In just Wuhan there are 11 million people, and in the entire Hubei province there are 58.5 million people. Setting S_0 to 11 or 58.5 million would not work in the model, though, because that would greatly exaggerate the number of infected people.

There are several shortcomings with the model. One obvious problem is our assumption about a constant disease transmission rate, β, which results in the model predicting that everyone who is susceptible will be infected eventually. In many countries, including China, drastic counter-measures were taken once the spread of the virus became widely known, which means that β should decrease as a function of time. To capture this behavior, and thus create a slightly more realistic model, we will now assume that β declines exponentially:

$$\beta(t) = \beta_0 e^{-\lambda t}. \tag{7}$$

Analytical solution for the extended SI-model.

It turns out that we can still use equations (5) and (6) to compute the analytical solution, provided we replace the product of of the constant β-parameter and time as follows:

$$\beta t \to \int_0^t \beta_0 e^{-\lambda t} dt = \frac{\beta_0}{\lambda}(1 - e^{-\lambda t}). \tag{8}$$

In the above expression, $\beta_0 = \beta(0)$ is the initial disease transmission rate. The magnitude of the factor λ determines how fast β decreases, i.e., a high value indicates strong disease-prevention measures.

Part 1.

- Make a function that plots the number of confirmed cases for a given country, as well as the extended analytical solution with β_0 and λ as input parameters.

Tip: Instead of creating a completely new function, you might want to modify your previous one to always use both β and λ as input arguments, but let $\lambda = 0$ by default (this way, the code stays backwards-compatible).

Part 2.

- Tune λ and β_0 to match the Hubei data. This is quite difficult, but one strategy is to first adjust β_0 to match the initial rise in the confirmed cases (For Hubei, a value for β_0 close to 2 gives a decent match). Next, adjust λ to match the data.

Tip: If/when you have time, it is worth familiarizing yourself with some of the existing Python functions to do curve-fitting, e.g., `scipy.optimize.curve_fit`. This will become very useful later in the course.

Part 3.

- Use the $\beta-$ and $\lambda-$ values you found for Hubei to predict the disease outbreak in Norway (You need to reduce S_0 to 5.4 million, which is the population of Norway). What do you observe? Do the results match your expectations?

Part 4.

- How do you need to change λ and β to improve the match for Norway? If we assume that the value of λ represents the government response, do the different values (λ_{Norway}, λ_{Hubei}) match your expectations?

Part 5.

- In light of what has happened during the entire course of the pandemic, write a paragraph or two where you reflect on the challenges with modelling disease outbreaks. Are there any important aspects that we have not touched upon in this project?

References

[1] George EP Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, 1976.

[2] William Ogilvy Kermack and Anderson G. McKendrick. A Contribution to the Mathematical Theory of Epidemics, part I. *Proceedings of the Royal Society of London. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927.

[3] Jonathan E. Pekar, Andrew Magee, Edyth Parker, Niema Moshiri, Katherine Izhikevich, Jennifer L. Havens, Karthik Gangavarapu, Lorena Mariana Malpica Serrano, Alexander Crits-Christoph, Nathaniel L. Matteson, et al. The molecular epidemiology of multiple zoonotic origins of SARS-CoV-2. *Science*, page eabp8337, 2022.

[4] Wikipedia. COVID-19 Pandemic on Diamond Princess. https://en.wikipedia.org/wiki/COVID-19_pandemic_on_Diamond_Princess, 2021. Accessed: 2022-08-13.

[5] Michael Worobey, Joshua I. Levy, Lorena Malpica Serrano, Alexander Crits-Christoph, Jonathan E. Pekar, Stephen A. Goldstein, Angela L. Rasmussen, Moritz UG Kraemer, Chris Newman, Marion PG Koopmans, et al. The Huanan Seafood Wholesale Market in Wuhan was the early epicenter of the COVID-19 pandemic. *Science*, page abp8715, 2022.

[6] Xiao Xiao, Chris Newman, Christina D. Buesching, David W. Macdonald, and Zhao-Min Zhou. Animal sales from Wuhan wet markets immediately prior to the COVID-19 pandemic. *Scientific reports*, 11(1):1–7, 2021.