Project 01 - MT3002

Eric Rostedt

July 2020

Introduction

This brief report is based on the first project in the course: MT3002 The Mathematics and Statistics of Infectious Disease Outbreaks given by Stockholm University summer 2020. We shall look at some basic aspects in estimating magnitude of outbreaks, simulate an outbreak over time, fit parameters to an outbreak and also try to deal with delayed reporting. All programming will be done in Python (version 3.7.2), more specifically using jupyter-lab.

Dependencies:

- scipy (spceifically scipy.optimize and odeint from scipy.integrate)
- numpy
- matplotlib.pyplot
- pandas
- math

Exercise 1

In this first exercise we shall solve the final size equation numerically to see how the number of people that gets infected depends on the basic reproduction number R_0.

a)

Firstly we shall assume that the entire population are initially susceptible to get infected (no immunes). We wish to solve for the final fraction infected τ by from the final size equation:

$$1 - \tau - \exp(-R_0\tau) = 0$$

We solve this using scipy.optimize's numerical solver *fsolve* and get the following result for $R_0 \in [0,\ 5]$ with 100 equidistant points.

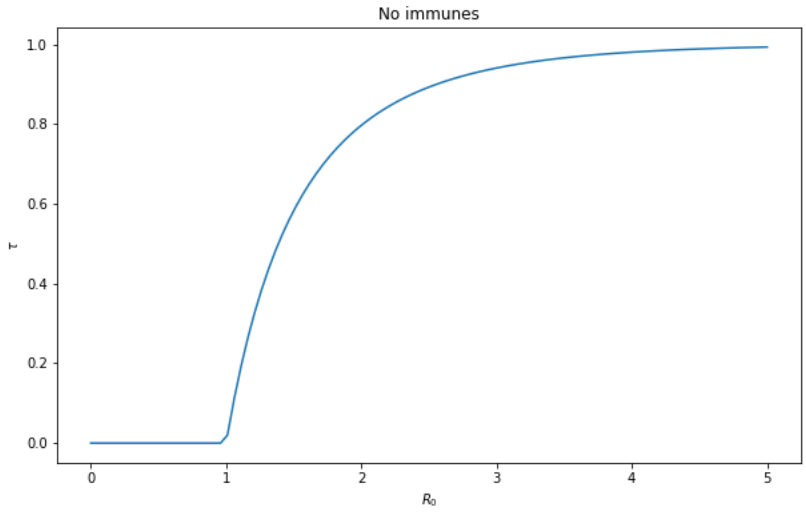

Figure 1: Fraction of people τ that gets infected as a function of the basic reproduction number R_0.

We note that when $R_0 < 1$ we get the solution $\tau = 0$. Thereafter τ is monotonically increasing and tends (when tending to positive infinity) to 1.

b)

Now we shall instead assume that some fraction r of the population is immune to the infection. We then have to instead solve the updated final size equation:

$$1 - \tau - \exp(-R_0(1 - r)\tau) = 0$$

We let $r = 0.3$ and solve this new equation in the same way as in 1a) and acquire:

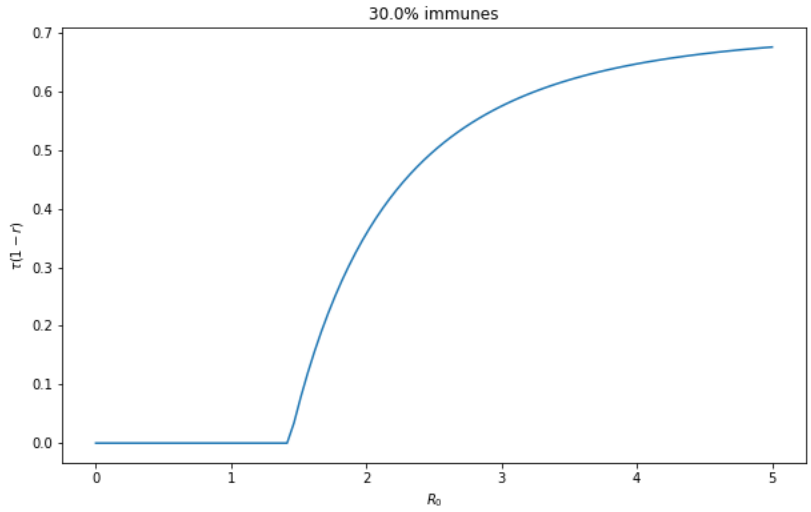

Figure 2: Fraction of people that gets infected as a function of the basic reproduction number R_0 and with 30% being immune to the infection.

The fraction that gets infected are now $\tau(1-r) = 0.7\tau$. From the plot we can see that $0.7\tau = 0$ for $R_0 < 1/(1-r) = 10/7 \approx 1.43$. Thereafter 0.7τ is monotonically increasing and tends (when tending to positive infinity) to 0.7.

Exercise 2

In this second exercise we shall construct and solve a simple SEIR model.

a)

We shall begin by constructing the SEIR model for non-varying parameters. We shall assume that we don't have to care for births and deaths; and recovered individuals are permanently immune. Now let's define our parameters as:

β : The effective contact rate between individuals.

$1/\rho$: The mean latency period.

$1/\gamma$: The mean infectious period.

Then the SEIR model is constructed as the following system of differential equations:

$$\frac{dS}{dt} = -\beta IS \tag{1}$$

$$\frac{dE}{dt} = \beta IS - \rho E \tag{2}$$

$$\frac{dI}{dt} = \rho E - \gamma I \tag{3}$$

$$\frac{dR}{dt} = \gamma I \tag{4}$$

We note that this system is non-linear, see (1), so our best bet of solving the system is, most likely, by numerical routines.

b)

Now that we have our system of equations we can introduce a numerical integration scheme to solve the system. There are a large family of numerical integration schemes but we shall for now resort to one of the most simple: The Euler method. The Euler method calculates a finite difference gradient with step size Δt at some time t and follows the gradient Δt far. Let t_0 be the left bound and t_n be the right bound of the integral. We shall discretize the interval $[t_0,\ t_n]$ into n + 1 equidistant points with the time update as: $t_{k+1} = t_k + \Delta t$. Now let $\frac{dy}{dt} = f(t, y)$, then the Euler integration scheme (in one direction) has the following form:

$$y_{k+1} = y_k + \Delta t f(t_k, y_n)$$

We now simply need to supply initial conditions and iterate since the derivatives for the update is precisely the equations in the equation system. Eg:

$$S_{k+1} = S_k + \Delta t(-\beta I_k S_k), \quad S_0 = N - 1$$

Set $(N,\ \beta,\ \rho,\ \gamma) = (100,\ 0.004,\ 1/5,\ 1/7)$ and now we solve the system using Euler integration with initial condition $[N - 1, 0, 1, 0]$ and plot the susceptible and infected as functions of time.

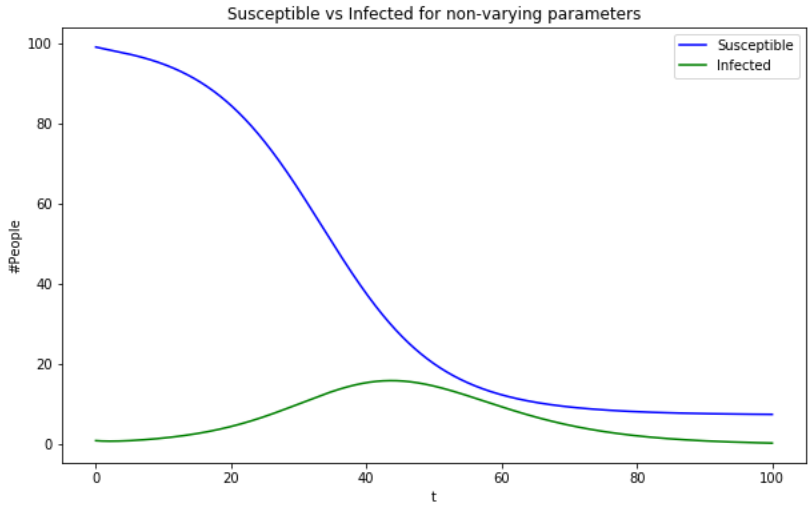

Figure 3: SEIR simulation for parameters: $(N,\ \beta,\ \rho,\ \gamma)$ $= (100,\ 0.004,\ 1/5,\ 1/7)$

We can see from the plot that most people actually gets infected with most people being sick during the interval $[20,\ 70]$.

c)

Assuming that the effective contact rate is completely constant during an infection outbreak is usually a very strong condition. Instead the effective contact rate is usually a function of time $\beta(t)$ since societies (hopefully) introduces safety measure to reduce the spread. We shall now assume that the effective contact rate function $\beta(t)$ is constant during the earlier stages, then linearly changes towards another value in the intermediate stage and finally constant once again but at the other value throughout the rest of the outbreak. We know that the function has the following appearance:

$$\beta(t) = \begin{cases} \beta_0, & \text{if } t \leq t_1 - w \\ \beta_1, & \text{if } t_1 + w < t \end{cases}$$

But now wish to find the expression for the value of $\beta(t)$ for $t \in [t_1 - w,\ t_1 + w]$. Since we know that the function should change linearly we simply need to find the function $y = kx + m$ which is β_0 at $t_1 - w$ and β_1 at $t_1 + w$. We could simply insert the conditions and solve a linear equation system, but we can solve the problem even faster. First we shall shift the linear equation by $t_1 - w$ to the left so that we get the y intercept at $m = \beta_0$ Then we know that the curve goes from β_0 to β_1 when going from $t_1 - w$ to $t_1 + w$ so we can find the slope as: $\frac{\beta_1 - \beta_0}{t_1 + w - (t_1 - w)} = \frac{\beta_1 - \beta_0}{2w}$. Which grants us the equation $\beta(t) = \beta_0 + (t - t_1 + w)\frac{\beta_1 - \beta_0}{2w},\ t \in [t_1 - w,\ t_1 + w]$. So the final result for the effective contact rate function is:

$$\beta(t) = \begin{cases} \beta_0, & \text{if } t \leq t_1 - w \\ \beta_0 + (t - t_1 + w)\frac{\beta_1 - \beta_0}{2w}, & \text{if } t_1 - w < t \leq t_1 + w \\ \beta_1, & \text{if } t_1 + w < t \end{cases} \tag{5}$$

d)

We have now a SEIR model but with the time varying effective contact rate function in 2c). We shall now solve the new system, but not by using Euler integration this time but instead opt to use scipy.integrate's odeint function which is a wrapped solver for the lsoda solver in FORTRAN. When solving for parameters: $(N,\ \beta_0,\ \beta_1,\ t_1,\ w,\ \rho,\ \gamma) = (100,\ 0.004,\ 0.0012,\ 30,\ 5,\ 1/5,\ 1/7)$ and initial condition $[N - 1, 0, 1, 0]$ we get the following.

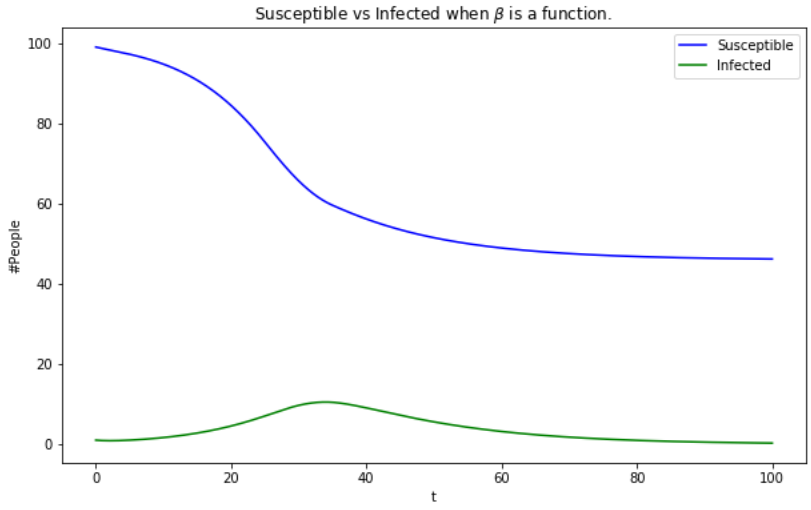

Figure 4: SEIR simulation for parameters: $(N, \beta_0, \beta_1, t_1, w, \rho, \gamma)$ = (100, 0.004, 0.0012, 30, 5, 1/5, 1/7)

As one can clearly see, with this model we get a significantly better result with a lot fewer infected, compared to 3. This of course does not mean that this model is more correct for a real outbreak, but if this kind of outbreak would occur then it would be preferable to the situation in 3.

Exercise 3

Being able to model what will likely happen granted some parameters are all fine and dandy, however is quite useless unless one actually has appropriate parameters for modeling the outbreak accurately. In this exercise we shall look at how one can estimate parameters for using data on how many people actually gets infected.

a)

In this first part we shall do a simple parameter fitting using reported cases of Covid-19 in Stockholm during February-April 2020. The population $N \approx 2.37455 \cdot 10^6$, let's assume that time zero ($t = 0$) is February 17:th and that the number of infected on that day is 1 i.e $I(0) = 1$. We shall fit the SEIR model from task 2d), i.e a model which follows the equations (1) $\longrightarrow$ (4) with time-varying $\beta(t)$ in accordance to (5). We are also granted that $\rho = 1/5$ and now wish to fit the parameters $\boldsymbol{\theta} = (\beta_0,\ \beta_1,\ t_1,\ w,\ \gamma)^\top$. We shall use a simple least square method defined as:

$$\text{argmin}_{\boldsymbol{\theta}}\ L(\boldsymbol{\theta}) = \sqrt{\sum_{t=0}^{T} \left(I(t;\boldsymbol{\theta}) - I_{observed}(t)\right)^2}$$

This function is then minimized using scipy's minimization routine. With an initial guess of parameters $\boldsymbol{\theta} = (7 \cdot 10^{-8},\ 8 \cdot 10^{-8},\ 30,\ 5,\ 1/7)$ we get the following fit:

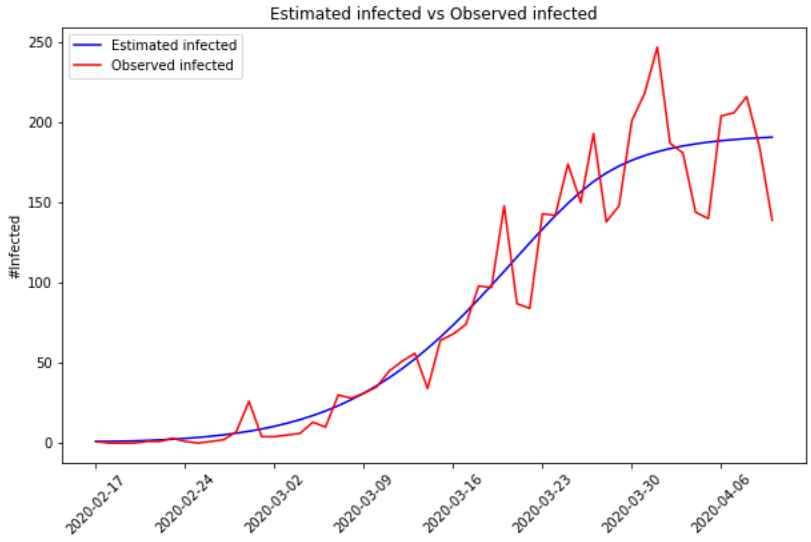

Figure 5: Observed infections vs estimated infections where parameters for the fitted model are: $\boldsymbol{\theta} = (3.31 \cdot 10^{-7},\ 7.45 \cdot 10^{-12},\ 5.74,\ 33.69,\ 1.15 \cdot 10^{-6})$

We note that we can actually acquire a quite good fit by this method. The fitted model fails to capture the larger fluctuations but appears to capture the overall trend very well.

b)

We have now constructed and solved a rather simple SEIR model and shall compare it to a more sophisticated one, namely the one from Folkhälsomyndigheten (Folkhälsomyndigheten, 2020). We will summarize some parts of their model which are not present in our more simplified model:

N : Our model had baked the 1/N into the β whilst Folkhälsomyndigheten hadn't, therfore their model has β/N instead of just β.

b_t : Time varying infectivity rate $b_t = b(t, t_b, \theta, \delta, \epsilon) = \theta(\delta + \frac{1-\delta}{1+\exp(-\epsilon(t-t_b))})$ where the infectivity rate is assumed to vary between θ and $\delta\theta$ and the midpoint is t_b. ϵ is the speed of change in infectivity. We modeled using a time varying effective contact rate which was linear and simpler.

I_r : The model from Folkhälsomyndigheten split up the infected variable into two, one for reported infections and one for unreported infections.

q_o : Scaling parameter $q_0 \in [0,\ 1]$ for infectiousness among unreported cases compared to infectiousness among reported cases. If an infection is unreported it is probably due to lack of or mild symptoms and therefore probably not as infectious as the cases where the infection is reported.

I_u : Is the estimated unreported cases.

p_r : The probability that a unreported case gets reported.

Model comparison

Our model:

$$\frac{dS}{dt} = -\beta(t)IS$$
$$\frac{dE}{dt} = \beta(t)IS - \rho E$$
$$\frac{dI}{dt} = \rho E - \gamma I$$
$$\frac{dR}{dt} = \gamma I$$
$$\beta(t) = \begin{cases} \beta_0, & \text{if } t \leq t_1 - w \\ \beta_0 + (t - t_1 + w)\frac{\beta_1 - \beta_0}{2w}, & \text{if } t_1 - w < t \leq t_1 + w \\ \beta_1, & \text{if } t_1 + w < t \end{cases}$$

Model of Folkhälsomyndigheten:

$$\frac{dS}{dt} = -S\frac{b_t I_r}{N} - Sq_0\frac{b_t I_u}{N}$$
$$\frac{dE}{dt} = S\frac{b_t I_r}{N} + Sq_0\frac{b_t I_u}{N} - \rho E$$
$$\frac{dI_u}{dt} = p_0\rho E - \gamma I_u$$
$$\frac{dI_r}{dt} = (1 - p_0)\rho E - \gamma I_r$$
$$\frac{dR}{dt} = \gamma(I_u + I_r)$$
$$b_t = \theta\left(\delta + \frac{1-\delta}{1+\exp(-\epsilon(t-t_b))}\right)$$

Exercise 4

In this last exercise we shall try to deal with delayed reporting of deaths due to Covid-19 in Sweden. We are granted a reporting triangle beginning second of April 2020 and ending 29:th of June 2020.

a)

In this first part we shall simply calculate the total number of deaths during the period. By summing over all cells in the data frame we get that the total number of perople that died during the period is 4 816.

b)

Now we wish to plot how many people that died on each day during the period. By summing throughout the rows in the data frame we get the deaths for each day. We wish to also add a 7 day moving average to the plot. A (central) moving average process of order 7 is defined as:

$$m(t) = \frac{1}{7}\sum_{i=-3}^{3} y_{t-i}$$

We will note that for a $MA(7)$ we will need $m(-3),\ m(-2), m(-1), m(T+1), m(T+2)$ and $m(T+3)$ which we do not know and will be set to zero, therefore the moving average will be inaccurate near the edges of the data. One could use padding techniques to improve the results but we opt to leave that out for now. We plot the deaths per day together with the 7 day moving average:

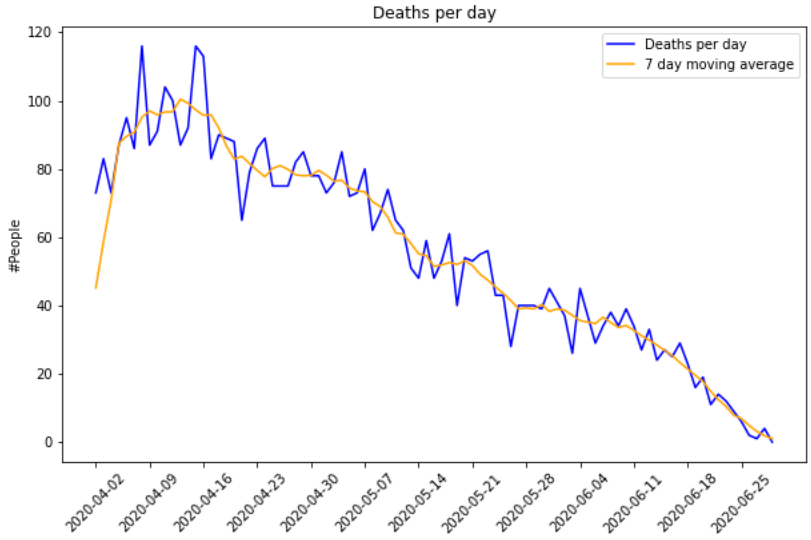

Figure 6: Deaths per day and 7-day moving average.

As we can see the number of deaths per day have significantly shrunk from beginning of April to the end of June.

c)

In this part of exercise 4 we shall do some modifications to the reporting triangle. First we set all delays larger than 20 into the 20 delay column of the data frame. Thereafter we assume that the delay distribution at time T is stable for the interval $[T-20,\ T]$. Let 29:th of June be T then we only look at the dates from 2020-06-10 to 2020-06-29. We shall now try to estimate the cumulative distribution function of the delay distribution. We are granted the following information:

$$g(d) = P(D = d | D \leq d)$$
$$F(d) = \prod_{x=d+1}^{20} (1 - g(x))$$

The function $g(d)$ can be estimated by summing all entries in column with delay $= d$ in the data frame and divide by all entries in columns from delay $= 0$ up to delay $= d$. By seeing the reporting triangle as a matrix A where the rows are dates and columns are delays than one can write this using matrix notation:

$$g(d) = \frac{\sum_{i=0}^{T} A_{i,d}}{\sum_{i=0}^{T} \sum_{j=0}^{d} A_{i,j}}$$

We can insert this into the formula for the cumulative distribution function and we get the estimation as:

$$F(d) \approx \prod_{x=d+1}^{20} \left(1 - \frac{\sum_{i=0}^{T} A_{i,x}}{\sum_{i=0}^{T} \sum_{j=0}^{x} A_{i,j}}\right)$$

We are asked to report the value of $F(3)$ which is estimated to be 0.324.

d)

In this sub task we shall calculate the values for $F(d),\ d \in [0, 20]$. We utilize the formula from the previous sub task and summarize the result in in the following table:

Table 1: Cumulative distribution function for delayed reporting.

d	$F(d)$
0	0.028
1	0.175
2	0.262
3	0.324
4	0.408
5	0.470
6	0.594
7	0.690
8	0.769
9	0.831
10	0.882
11	0.935
12	0.969
13	0.992
14	0.997
15	0.997
16	1.00
17	1.00
18	1.00
19	1.00
20	1.00

e)

We now have a way of estimating the cumulative distribution function (CDF) of the delay distribution, with this we can do the following nowcast procedure:

$$N(t,\ \infty) = \frac{N(t,\ T)}{\hat{F}(T-t)}$$

Where $\hat{F}$ is our estimated CDF and $N(t,\ T) = \sum_{d=0}^{\min(\text{T-t},\ \text{D})} n_{t,d}$ with $n_{t,d}$ being the number of people that died at time t with delay d. We calculate $N(t,\ \infty)$ for $t \in$ [2020-06-20, 2020-06-26] and acquire:

Table 2: Estimated number of people to die on given dates due to Covid-19 from our estimates

t	$N(t,\ \infty)$
2020-06-20	21.55
2020-06-21	13.24
2020-06-22	18.21
2020-06-23	17.39
2020-06-24	15.14
2020-06-25	12.75
2020-06-26	4.90

f)

Lastly we shall compare the result from 2e) with results from Folkhälsomyndigheten (Altmejd, 2020). We are only granted a plot but by roughly eyeballing their result we could summarize their reuslt in the following table:

Table 3: Estimated number of people to die on given dates due to Covid-19 from estimates made by Folkhälsomyndigheten.

t	$N(t,\ \infty)$
2020-06-20	19
2020-06-21	11
2020-06-22	14
2020-06-23	12
2020-06-24	9
2020-06-25	5
2020-06-26	2

Obviously these numbers are not exact since the comparison is by simply eyeballing a plot, however it appears that their estimates are lower. This could most certainly be answered by the fact that they are most likely are using much more sophisticated estimation methods. But all in all one should not be too disappointed with our estimates, since after all, the methods used are rather simple.

References

All information used are taken from the course lectures and course slides. There are two exceptions which are the links prescribed in the exercise sheet, and are referenced below:

Folkhälsomyndigheten, (2020). "Estimates of the peak-day and the number of infected individuals during the covid-19 outbreak in the Stockholm region, Sweden February – April 2020". Accessed at: `https://www.folkhalsomyndigheten.se/publicerat-material/publikationsarkiv/e/estimates-of-the-peak-day-and-the-number-of-infected-individuals-during-the-covid-19-outbreak-in-the-stockholm-region-sweden-february--april-2020/` on July 29:th 2020.

Altmejd, A. (2020). "Confirmed daily Covid-19 deaths in Sweden". Image accessed at: `https://github.com/adamaltmejd/covid/blob/master/docs/archive/deaths_lag_sweden_2020-06-29.png` on July 29:th 2020.