Estimating $\mathcal{R}_t$ from Covid-19 data, using SIR models

Hugh Murrell and Daniel Murrell

September 13, 2020

Abstract

Here we give an interpretation of the *effective reproduction number*, $\mathcal{R}_t$ that arises from the Susceptible-Infectious (SI) model for the spread of infectious disease. We show that a very simple smoothed frequentist estimator behaves similarly to a Bayesian posterior mean for this model and we outline a simple nowcasting scheme that corrects for onset delay. Using our fast algorithm we perform daily estimates of $\mathcal{R}_t$ on an industrial scale and we use these estimates to animate government responses to the Covid-19 outbreak. We provide an app, `https://reproduction.live` for users to view these outbreak animations.

1 Introduction

From the onset of the Covid-19 pandemic many articles have appeared in the popular press claiming that $\mathcal{R}_t$ is the metric to track when trying to ascertain the success, or lack thereof, of lockdown measures put in place by the authorities. See [4] for an excellent introduction to $\mathcal{R}_t$ and why we should be following it.

Usually, government advisors report on the current value of $\mathcal{R}_t$ and if this value is less than one the lockdown measures are deemed to be working but if it is greater than one then more stringent lockdown measures are recommended. However, methods for the computation of $\mathcal{R}_t$ are seldom outlined. The reason for this is that today's methods usually involve complicated Bayesian inference which is deemed to lie outside the skill set of the article readership, see [6] for an example of an article in the South African popular press.

An exception to the above is the work of Kevin Systrom who uses the Bayesian inference techniques of BettenCourt and Ribeiro [1] on regional USA case count data to obtain $\mathcal{R}_t$ values for each state and thus rank state-wide responses to the pandemic. System has recently re-written his code to perform for MCMC inference to obtain $\mathcal{R}_t$ rankings. See https://rt.live/ for his latest rankings. Refreshingly, Systrom has made his methods available to the public by publishing his `PyMC3` Python code with explanation as Jupyter notebooks on github [3]. Members of the public can download his notebooks and run them on their own datasets.

However, to understand Systrom's *MCMC* code, you need a background in statistics which many readers do not have. In this article we develop a simple finite difference algorithm for estimating $\mathcal{R}_t$ and we show that this simple algorithm behaves similarly to the Bayesian posterior mean of Systrom's original code.

Our initial experiments were performed on South African datasets provided at national and provincial levels in [8] and our code is made available to the public through github [10]. Results from these experiments can be viewed on the *MediaHack* dashboard at `https://mediahack.co.za/datastories/coronavirus/dashboard/`.

During the course of 2020 we will employ the methods outlined in this article on world data provided in [7] to drive animations of the pandemic evolution over time. These can be viewed at `https://reproduction.live/`.

2 The standard SI model

The Susceptible-Infectious (SI) model [2] is often used to study the spread of infectious disease by tracking the number (S) of people susceptible to the disease and the number (I) of people infectious with the disease.

Based on the model, the only way that a person can leave the susceptible group is to be infected and become immediately infectious, and the only way that a person can leave the infectious group is to recover or die. It is further assumed that those who have recovered or died from the disease are no longer susceptible.

It is also assumed that all those who have not had the disease are equally susceptible and that the probability of their contracting the disease at time $t + 1$ is proportional to the product of S and I at time t.

These assumptions lead us to a pair of difference equations for S and I, where the unit of time t is one day:

$$S_{t+1} = S_t - \beta \frac{S_t I_t}{N} \quad (1)$$

$$I_{t+1} = I_t + \beta \frac{S_t I_t}{N} - \gamma I_t \quad (2)$$

Here the parameter $\beta \geq 0$ controls the rate at which the susceptible become infected and the parameter $\gamma \geq 0$ controls the rate at which the infectious recover or die. The parameter N is the size of the **initial** susceptible population and is usually assumed constant. The number of infected persons who are no longer infectious is given by $N - (S_t + I_t)$.

The susceptible time series S_t is a monotonically decreasing sequence starting at $S_0 = N$ just before the outbreak of the disease whilst the infectious time series I_t starts at $I_0 = 0$ just before the outbreak but then climbs and falls depending on interventions but eventually dies out to zero when the disease has run its course.

With this model the trajectories of S_t and I_t are pre-determined at the outset by the parameters, β, γ and N. See Figure 1 for an example trajectory.

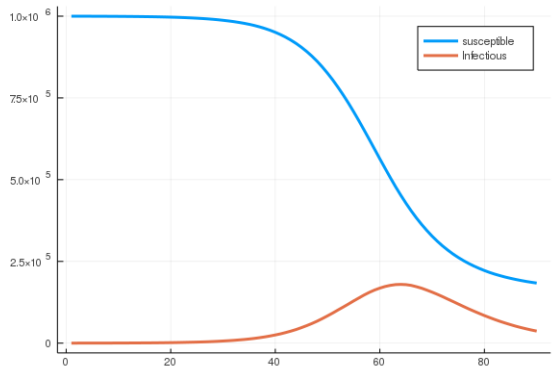

Figure 1: Susceptible and Infectious time series for user defined parameters, $\gamma = \frac{1}{7}$, $N = 10^6$ and $\beta = 2\gamma$.

3 Introducing $\mathcal{R}_t$

To gain some *control* over an epidemic, authorities can enforce quarantine or social distancing measures which may affect some of the model parameters. The parameter γ cannot be manipulated through such measures as it is a property of the disease itself. γ is the reciprocal of the length of the infectious period of the disease which in the case of Covid-19 is estimated to be about one week, so $\gamma = \frac{1}{7}$.

The other two parameters, β and N, can be manipulated via interventions and it is a common practice to define a quantity called the *effective* reproduction number, $\mathcal{R}_t$ as follows:

$$\mathcal{R}_t = \frac{\beta}{\gamma}\frac{S_t}{N} \tag{3}$$

and then recast the discrete SI model as:

$$S_{t+1} = S_t - \gamma \mathcal{R}_t I_t \tag{4}$$
$$I_{t+1} = I_t + \gamma I_t(\mathcal{R}_t - 1) \tag{5}$$

Note that $\mathcal{R}_0 = \frac{\beta}{\gamma}$ which is called the **basic** reproduction number and tells us how many persons an infected person will infect during their infectious period at the start of an epidemic and before any interventions can be mounted. In the case depicted in figure 1, the basic reproduction number is $\mathcal{R}_0 = \frac{\beta}{\gamma}\frac{S_0}{N} = 2$.

Note that if no interventions are put in place during the course of the epidemic then $\mathcal{R}_t$ will decrease monotonically as S_t decreases. The goal of interventions is to decrease $\mathcal{R}_t$ faster than the natural decline induced by the diminishing pool of susceptible persons. In particular it is desirable to force $\mathcal{R}_t < 1$ so that the pool of infectious is smaller when a person leaves the pool than when he enters it.

To see if an intervention is successful or not we must have some way of estimating the current value of $\mathcal{R}_t$ from daily new-case counts, C_t, which is how most governments currently release Covid-19 data.

4 Estimating $\mathcal{R}_t$ from case count data

Almost all modern attempts at estimating $\mathcal{R}_t$ implement Bayes's rule to estimate current $\mathcal{R}_t$ from previous values of $\mathcal{R}_t$, see for example [1] for the theory and [3] for a recent implementation. Here we will use a frequentist approach and with the use of a simple smoothing filter we will show how to estimate $\mathcal{R}_t$ directly from the data.

An *instantaneous* estimate for yesterday's $\mathcal{R}_t$ value can be obtained by comparing the size of yesterday's infectious pool with the size of today's infectious pool by rewriting equation 5 as

$$\mathcal{R}_t = 1 + \frac{1}{\gamma}\left(\frac{I_{t+1} - I_t}{I_t}\right) \tag{6}$$

As we can estimate the current size of the infectious pool, $I(t)$, by summing *new case counts*, C_t, over the preceding infectious period:

$$I_t = \sum_{j=t-\frac{1}{\gamma}+1}^{t} C_t \tag{7}$$

Equations 6 and 7 provide us with an algorithm for estimating $\mathcal{R}_t$. However are three drawbacks to this algorithm.

1) Case reporting is very erratic due to things like testing backlogs and data collection. Indeed, new case counts often exhibit a 7-day cycle due to under-reporting over week-ends. To overcome erratic reporting we apply a 7-day smoothing filter to the time series and assume that the real world process is not as stochastic as the actual reporting.

2) There is a natural delay, when the patient is infectious but un-diagnosed, between the infectious *onset* of the disease and the appearance of symptoms and case *reporting*. In the literature, this problem is solved by convolving the new cases time series with an empirical estimate of reporting delays. This procedure shifts the time series a few days earlier and causes case counts to shrink at the end of the time series. This shrinkage is repaired using a a technique called *nowcasting*. We attempt to implement a simple version of onset correction on our Covid-19 data by convolving a Weibull distribution with our case time series

augmented by a short exponential (growth/decay) forecast. This is achieved by implementing simple linear regression in the log domain.

3) The difference scheme proposed in equation 6 is unstable when the epidemic has been vanquished and the pool of infectious persons is near zero. At this stage a sudden appearance of one or two new cases will send the estimate for $\mathcal{R}_t$ sky high. The correct way to deal with this problem is to implement Bayes' rule for updating $\mathcal{R}_t$ that requires the calculation of *priors* from preceding values of $\mathcal{R}_t$ and then selects the *most likely* $\mathcal{R}_t$ trajectory. We implement a fast version of Bayes' rule in a further attempt to *smooth* our estimates.

4.1 Pseudocode

Given a new-case count time series, $\mathcal{C}_t$, for a region an $\mathcal{R}_t$ time series can be computed as follows:

input : $\mathcal{C}_t$ daily reported new-case counts
truncate $\mathcal{C}_t$ by removing leading zeros;
compute SC_t as cumulative sum of $\mathcal{C}_t$;
impute missing values on SC_t using linear interpolation;
reconstruct $\mathcal{C}_t$ by using first differences on SC_t;
apply a 7-day *moving-average* to the $\mathcal{C}_t$ series;
apply linear regression in the log domain to *forecast* new $\mathcal{C}_t$ data;
apply *onset estimation* to the $\mathcal{C}_t$ series;
compute the infectious pool time series according to equation 7;
compute the $\mathcal{R}_t$ time series according to equation 6;
moderate the $\mathcal{R}_t$ time series using Bayes' rule;
output: $\mathcal{R}_t$ effective reproduction rate estimates

Algorithm 1: pseudocode for constructing $\mathcal{R}_t$ from reported case counts. No attempt is made to estimate the under reported asymptomatic cases. We assume that the proportion of under reporting is constant over short time periods.

It remains to give further details on *onset estimation* and *credible intervals*.

4.2 Onset correction

Onset correction translates positive case counts to the dates where they likely occurred. If we have the onset delay distribution, we can distribute

case counts back in time according to that distribution. To obtain an onset distribution we need case data that gives reporting time and estimated contraction time for each case. In South Africa we are unable to obtain such data and so to proceed we rely on work done using USA data for the `rt.live` dashboard where an empirical distribution for onset delays was obtained from US case data, see figure 2.

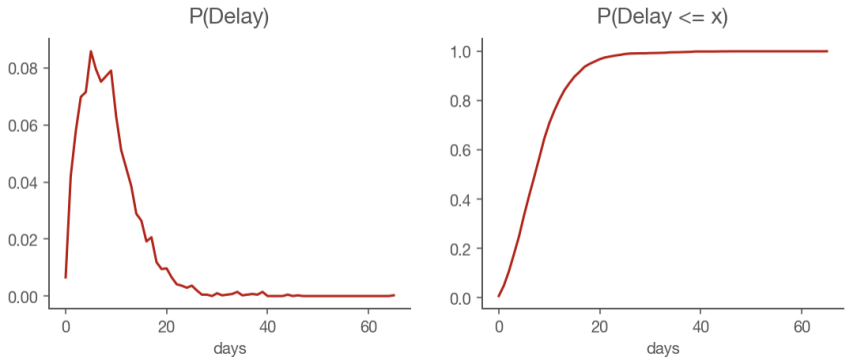

Figure 2: Empirical onset delay distribution from US case data, see [3] for details.

Although we don't have actual numbers for this distribution we can get some idea of its *shape* and *scale* by studying figure 2 and then try and find a known statistical distribution to do the job for us. The distribution we select is the Weibull distribution with shape 2 and scale 5, see figure 3

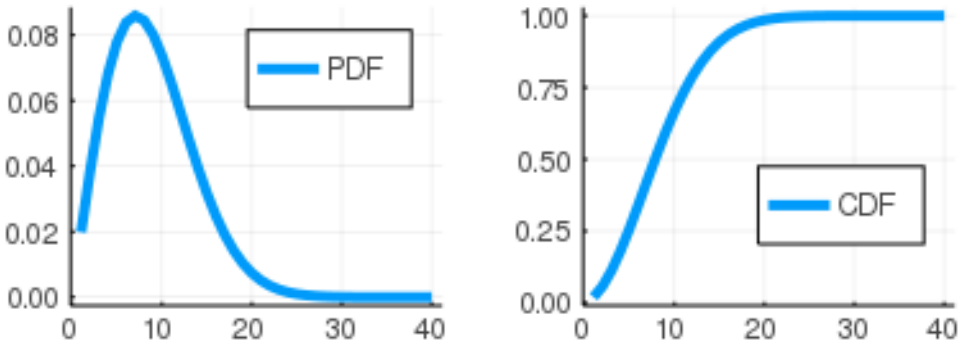

Figure 3: Weibull(2,5) distribution for simulating onset delay.

Now that we have an onset delay distribution to work with we can perform the onset correction. As expained by Kevin Systrom in [3] we accomplish this, by reversing the new-case time series, convolving it with the delay distribution and then reversing the series again to obtain the onset curve. However this algorithm results in *right censoring* of the onset curve due to zero padding at the boundaries of the signal.

4.3 Nowcasting

To avoid the right censoring we implement simple *nowcasting* by first augmenting our time series into the future by repeatedly using regression over the last s days of the time series to forecast a new data point for the next day.

For a simple *linear* regression scheme, we have a two parameter model:

$$c(t) = \theta_0 + \theta_1 t = \boldsymbol{\theta}^T \boldsymbol{t}$$

with

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} \text{ and } \boldsymbol{t} = \begin{bmatrix} 1 \\ t \end{bmatrix} \tag{8}$$

With this model the normal equations for the parameters are usually given in the form

$$\boldsymbol{\theta} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

where in our case the data for the regression comes from the last s days of the case count time series, vis:

$$\boldsymbol{X} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & s \end{bmatrix} \text{ and } \boldsymbol{Y} = \begin{bmatrix} y_1 & = & c_{n-s+1} \\ y_2 & = & c_{n-s+2} \\ y_3 & = & c_{n-s+3} \\ \cdot & = & \cdot \\ \cdot & = & \cdot \\ \cdot & = & \cdot \\ y_s & = & c_n \end{bmatrix}$$

and thus

$$\boldsymbol{X}^T\boldsymbol{X} = \begin{bmatrix} s & \frac{s(s+1)}{2} \\ \frac{s(s+1)}{2} & \frac{s(s+1)(2s+1)}{6} \end{bmatrix}$$

and according to *Wolfram Alpha*

$$\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1} = \frac{2}{(s-1)s}\begin{bmatrix} 2s+1 & -3 \\ -3 & \frac{6}{s+1} \end{bmatrix}$$

With this linear regression setup we estimate our next day's case counts as:

$$\begin{aligned} c_{n+1} &= y_{s+1} \\ &= \boldsymbol{\theta}^T \begin{bmatrix} 1 \\ s+1 \end{bmatrix} \\ &= \left(\frac{2}{(s-1)s}\begin{bmatrix} 2s+1 & -3 \\ -3 & \frac{6}{s+1} \end{bmatrix}\begin{bmatrix} \sum y_i \\ \sum iy_i \end{bmatrix}\right)^T \begin{bmatrix} 1 \\ s+1 \end{bmatrix} \\ &= \frac{2}{(s-1)s}\left[3\sum iy_i - (s+2)\sum y_i\right] \end{aligned} \tag{9}$$

In our code we transform the data into the *log* domain before using 9 to estimate $\log(c_{n+1})$ and then we *exponentiate* to transform back. We perform this estimation repeatedly to generate realistic *padding* for the signal before applying the Weibull convolution.

5 Credible Intervals

To obtain *confidence* in our estimates we must invoke *Bayes* and consider a range of possible $\mathcal{R}_t$ values at any one time. To this end we maintain a day by day probability distribution of $\mathcal{R}_t$ values, call them $\mathcal{R}_t^s$, and for each one estimate the next day infectious population using equation 5 as

$$\mathcal{I}_{t+1}^s \quad = \quad I_t(1 + \gamma(\mathcal{R}_t^s - 1)) \tag{10}$$

After seeing the next day's data, $k = I_{t+1}$, we calculate the *likelihood* of each of our estimates by computing the probability that a *Poisson* distribution with parameter $\lambda = \mathcal{I}_{t+1}^s$ would deliver the observed data:

$$\mathcal{L}_{t+1}^s \quad = \quad \mathcal{P}(I_{t+1}|\mathcal{R}_t^s) = \text{Pois}(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{11}$$

We then use *Bayes' rule* and some *Gaussian smoothing* to tabulate the probability distribution $\mathcal{R}_t^s$ having seen the next day's data.

$$\mathcal{P}(\mathcal{R}_t^s|I_{t+1}) \quad \propto \quad \mathcal{P}(I_{t+1}|\mathcal{R}_t^s)\mathcal{P}(\mathcal{R}_t^s) \tag{12}$$

and we select the most likely $\mathcal{R}_t$ as $\mathcal{R}_t^{s^*}$ where

$$s^* \quad = \quad \arg\max_s(\mathcal{P}(\mathcal{R}_t^s|I_{t+1})) \tag{13}$$

and as a bonus we obtain a so called *90% credible interval* for our most likely estimate by finding the region of the distribution, $\mathcal{R}_t^s$, that supports 90% of its density, see [5] for a full description.

We note here that using Bayes in this way is computationally expensive, of the order $\mathcal{O}(n * d^3)$ where n is the length of our time series and d is the size of our distribution grid. For our *full Bayesian* inference we make use of FFT techniques to perform the *Gaussian smoothing* involved which allows us to compute $\mathcal{R}_t$ estimates for thousand of time series each day to be presented on demand to the user via our `https://reproduction.live` app, see later.

6 Implementation on South African data

We obtain the latest case counts from a github repository maintained by the Data Science for Social Impact research group of the University of Pretoria, see [8].

The results of moving average smoothing, onset correction and nowcasting are depicted in figure 4.

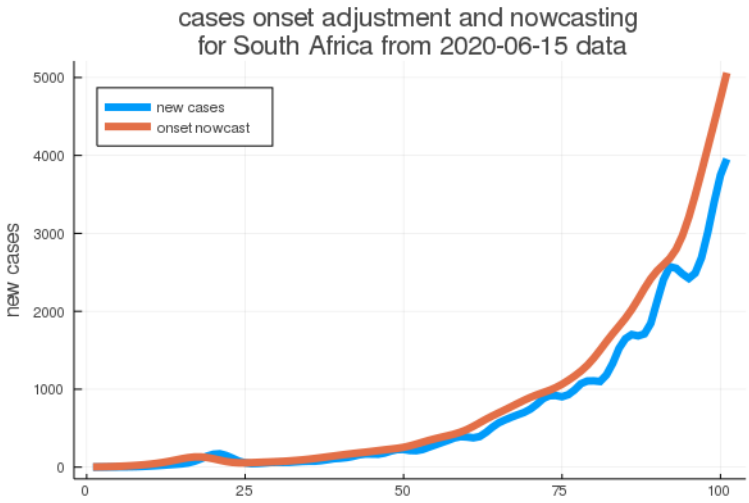

Figure 4: South African case counts after onset and nowcasting correction.

From the corrected new-case counts we compute the Infectious counts, I_t, by summing new cases over the infectious period and then we estimate $\mathcal{R}_t$. The result is shown in figure 5

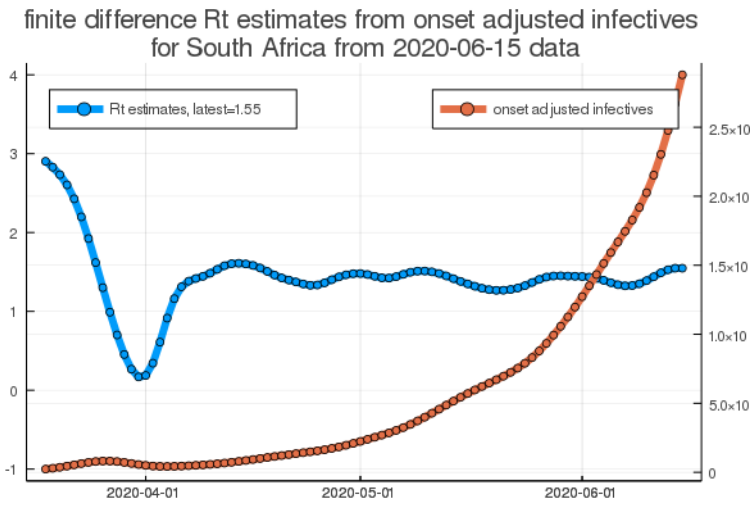

Figure 5: Simple finite difference estimates of $\mathcal{R}_t$ from infectious estimates after onset nowcasting correction of South African case counts.

If we want the associated *credible intervals* then we must implement the full Bayesian procedure as outlined above and we note that once death counts have risen sufficiently we can repeat the whole procedure using death counts instead of case counts to obtain $\mathcal{R}_t$ estimates. The only change we must make is to the parameters of our Weibull distribution to cater for the longer time delay between onset and death (which we estimate at about 21 days) Both estimates with credible intervals are performed on South African data and results are shown in figure 6.

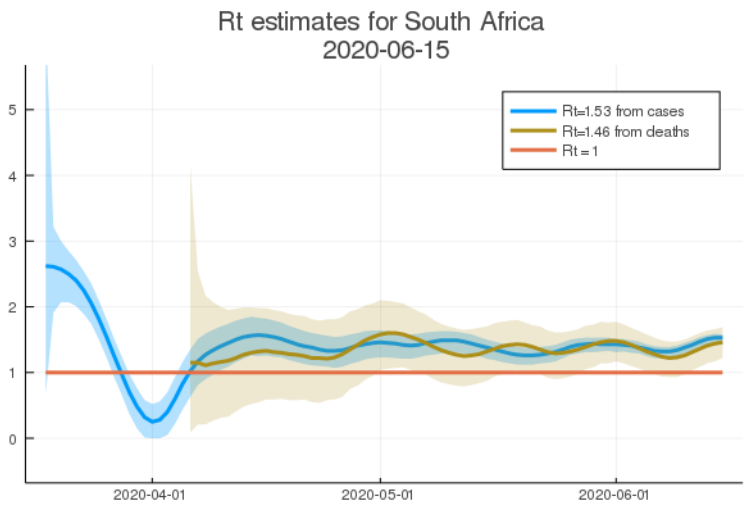

Figure 6: Tracking $\mathcal{R}_t$ using both reported cases and reported deaths.

By comparing figure 5 with figure 6 the reader can see that after onset correction a simple finite difference scheme gives a reasonable estimate of the Bayesian posterior mean. The Julia script made available in [10] enables the reader to repeat the analysis using either simple finite differences or full Bayesian inference.

7 Industrial scale $\mathcal{R}_t$ estimates

Using open data from `https://github.com/open-covid-19/data`, see [7], we compute $\mathcal{R}_t$ estimates each day at national, provincial and regional level and we present them via a `Nuxt` app to the user at `https://reproduction.live`.

New $\mathcal{R}_t$ estimates are computed each day using a Julia script and these precomputed time series are stored in an *infinite tree-like directory* of `json` files on our server so that access to $\mathcal{R}_t$ estimates of interest is fast enough to provide animations of the Covid-19 outbreak in your region.

At the time of writing we process 7183 time series nightly and this number will grow as the *open-covid-19* repository [7] finds new data sources.

8 Conclusions

The results of our simple frequentist implementation look fairly reasonable. The technique affords the layman the opportunity to analyse his own data and tweak model parameters as he sees fit. Personalised models can be used to decide on a strategy for curtailing an epidemic through quarantines, lockdowns and social distancing. Access to pre-computed $\mathcal{R}_t$ estimates with Bayesian creditable intervals is provided via a web based `Nuxt` app at `https://reproduction.live`.

9 Acknowledgements

First and foremost, the authors thank *Kevin Systrom* for making his code and methods available via github [3].

We thank *Benjamin Murrell* for enlightening us as to the intricacies of Bayesian techniques.

We thank *MediaHack* for displaying our $\mathcal{R}_t$ charts on their South African Covid-19 dashboard and making this document available for public consumption via their website, [9].

We thank *Max Price* for suggesting the use of `death-counts` instead of `case-counts` to track $\mathcal{R}_t$.

We thank *Philip Machanick* for keeping an eye on our charts and alerting us to glitches that appear from time to time.

We thank *Devin Mahon* for lively discussions concerning under reporting of case data.

References

[1] Bettencourt MA and Ribeiro RM, *Real Time Bayesian Estimation of the Epidemic Potential of Emerging Infectious Diseases*, Plos One, Volume 3, Issue 5, May 2008.

[2] *Compartmental models in epidemiology*, wikipedia

[3] Kevin Systrom, Thomas Vladek and Mike Krieger. Rt.live (2020). GitHub repository, https://github.com/rtcovidlive/covid-model

[4] Max Price, Covid-19: *Making sense of R*, GroundUp article

[5] Makowski D, *et al.*, *Describing Effects and their Uncertainty, Existence and Significance within the Bayesian Framework*. Journal of Open Source Software, 4(40), 1541, see: Credible Intervals

[6] Sarah Evans, *Lockdown and other interventions potentially saved 20 000 lives*,News 24, news24 article

[7] Oscar Wahltinez, Kevin Murphy, Michael Brenner, Matt Lee, Anthony Erlinger and Mayank Daswani. *COVID-Open-Data: curating a fine-grained, global-scale COVID-19 data repository*, https://github.com/open-covid-19/data/blob/main/README.md

[8] Data Science for Social Impact research group, University of Pretoria, github repository

[9] Media Hack Collective, *Coronavirus in South Africa*, Dashboard.

[10] Hugh Murrell, *SIR models for Covid-19 data*, hughmurrell.github.io/CoVmodel/index.html.