A New Method for Calculating the Effective Reproduction Number for COVID-19

A Technical Report

Patrick Grice
CloseAssociate (plc)
Wellington, New Zealand
Patrick.Grice@closeassociate.com

Stephen Grice, Ph.D.
CloseAssociate (plc)
Wellington, New Zealand
Stephen.Grice@closeassociate.com

Prof. Richard S. Laugesen
Department of Mathematics
University of Illinois at Urbana-Champaign, USA
Laugesen@illinois.edu

July 22, 2020

Abstract

A new process for estimating R_{eff} for COVID-19 is developed. It combines a deterministic SIR formula for calculating R_{eff} from positive test data and a statistical bootstrapping method for generating confidence intervals.

1 Introduction

SIR modelling is effective at predicting disease spreading through populations [9]. The dominant parameter in all SIR models is the effective reproduction number R_{eff}. This effective reproduction number is a function of time t, since it changes as the number of Susceptible individuals reduces during an epidemic, and as communities implement measures such as social distancing. Estimating R_{eff} is critical to understanding disease progression in a population, and hence is of intense interest to public health experts and policy officials.

The purpose of this technical report is to present in detail a new "SIR plus Bootstrapping" (SIR+B) process for estimating $R_{\text{eff}}(t)$. This new process has been applied to the daily case data for several countries in a separate technical report [3], and results there are compared with statistical methods such as EpiEstim.

The new SIR+B process provides public officials with a rapid and effective tool for knowing whether the reproduction number is increasing or decreasing over time, and whether it exceeds the critical value of 1 for epidemic control.

2 Overview of the method

We proceed backward through the SIR model equations to obtain a formula for the transmission coefficient multiplied by the proportion of Susceptibles in the population. From this one obtains the effective reproduction number $R_{\text{eff}}(t)$.

The formula for $R_{\text{eff}}(t)$ is then evaluated in terms of the number of Infectious-Tested cases, which we obtain from reported data on the number of confirmed, tested cases per day. Details are presented in Sections 3, 4 and 5. Sources of error and uncertainty in this SIR stage of the process include a certain amount of educated guesswork in choosing the recovery parameter γ and the testing rate c, and of course errors due to noisy data.

Model failure is a possibility too. If the SIR model is not applicable, then the method for finding the reproduction number will not give sensible output. This happens when the number of new cases per day is small, which in practice we find to mean less than about 10 per day.

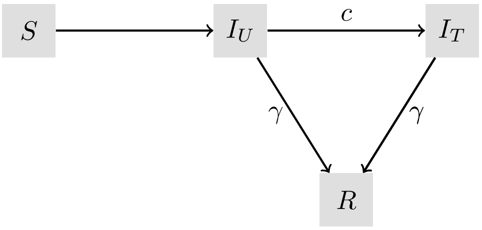

Figure 1: Progression through the SIR system: Susceptible individuals progress to Infectious-Untested, and then to either Infectious-Tested or Removed.

After the first stage of the process has calculated R_{eff}, these values are fed into the second stage, which generates a large number of daily case data sets by the wild bootstrapping method. These data sets yield confidence intervals for R_{eff}.

3 Model equations, and parameter values

Regard the population as having fixed size N, and consider the fraction of the population falling into each of the following categories:

Susceptible, Infectious-Untested, Infectious-Tested, Removed.

This SIR system is illustrated in Figure 1.

3.1 Continuous time model

The model equations are most easily stated using continuous time, that is, for a differential equations model, and so we state them that way first. The continuous time model equations are:

$$\frac{dS}{dt} = -\beta S(I_U + I_T) \tag{1}$$

$$\frac{dI_U}{dt} = \beta S(I_U + I_T) - (c + \gamma)I_U \tag{2}$$

$$\frac{dI_T}{dt} = cI_U - \gamma I_T \tag{3}$$

$$\frac{dR}{dt} = \gamma(I_U + I_T) \tag{4}$$

As a consistency check on any simulation, one should be able to compute that

$$S + I_U + I_T + R = 1$$

for all t, since the expression on the left has derivative zero.

Notes on the model:

- The model assumes random mixing among the population.
- Infectious individuals might be either symptomatic or asymptomatic.
- The transmission coefficient $\beta = \beta(t)$ is not constant. It is a function of time t.

The equations involve several parameters:

Parameter	Symbol
Transmission coefficient	$\beta(t)$
$I \to R$ transition rate	γ
Testing rate for infectious individuals	c

The values of γ and c are between 0 and 1, and should be chosen based on best available knowledge of disease behavior and testing regimes. In particular, the testing rate c represents the fraction of Infectious-Untested individuals who get tested per day.

The parameters c and γ appear explicitly in formulas (10) and (12) below for the effective reproduction number. Thus one can readily perform a sensitivity analysis to see what effect different parameter values have on the estimates of the reproduction number. Our numerical investigations show that the output is qualitatively and quantitatively robust across quite a wide range of plausible parameter values; see [3].

3.2 Discrete time model

In the real world, new cases are reported daily, and so it makes sense to use the following discrete time analogue of the model equations, where n represents the n-th day and the time step is 1 day:

$$S(n+1) - S(n) = -\beta(n)S(n)(I_U(n) + I_T(n)) \tag{5}$$
$$I_U(n+1) - I_U(n) = \beta(n)S(n)(I_U(n) + I_T(n)) - (c+\gamma)I_U(n) \tag{6}$$
$$I_T(n+1) - I_T(n) = cI_U(n) - \gamma I_T(n) \tag{7}$$
$$R(n+1) - R(n) = \gamma(I_U(n) + I_T(n)) \tag{8}$$

As a consistency check on any simulation, one should be able to compute that

$$S + I_U + I_T + R = 1$$

for all n, since the expression on the left is independent of n (as one seems by summing the left sides of (5)–(8) and observing that the right sides sum to 0).

3.3 Effective reproduction number

The effective reproduction number is defined by

$$\boxed{R_{\text{eff}}(t) = \frac{\beta(t)S(t)}{\gamma}} \qquad \text{or} \qquad \boxed{R_{\text{eff}}(n) = \frac{\beta(n)S(n)}{\gamma}},$$

depending whether the continuous time or discrete time model is used.

4 Estimating R_{eff}

Simple SIR model

For illustrative purposes, we begin with a simple SIR differential system

$$S' = -\beta SI,$$
$$I' = \beta SI - \gamma I,$$
$$R' = \gamma I,$$

which has just a single class of Infectious individuals. The second equation yields $\beta S = (I' + \gamma I)/I$, and so the effective reproduction number for this simple system is

$$R_{\text{eff}} = \frac{\beta S}{\gamma} = \frac{1}{\gamma}\frac{I' + \gamma I}{I}.$$

The right side of this formula involves only the proportion of Infectious individuals, which one can hope to observe experimentally.

In practice, not all Infectious individuals are observable, which is why this report follows James et al. [5] in dividing that group into Infectious-Untested and Infectious-Tested categories. We proceed to analyze that model, seeking a formula for βS, and hence for the reproduction number.

4.1 Continuous time SIR system with Infectious-Untested and Infectious-Tested

Now we analyze the differential equation model (1)–(4). The model equation (3) can be rearranged to express I_U in terms of I_T, as

$$cI_U = I_T' + \gamma I_T. \tag{9}$$

Substituting this equation into (2) yields

$$(I_T' + \gamma I_T)' = \beta S(I_T' + (c+\gamma)I_T) - (c+\gamma)(I_T' + \gamma I_T),$$

which can be rearranged to obtain the effective reproduction number:

$$\boxed{R_{\text{eff}}(t) = \frac{\beta(t)S(t)}{\gamma} = \frac{1}{\gamma}\frac{I_T''(t) + (c+2\gamma)I_T'(t) + \gamma(c+\gamma)I_T(t)}{I_T'(t) + (c+\gamma)I_T(t)}}. \tag{10}$$

4.2 Discrete time SIR system with Infectious-Untested and Infectious-Tested

The difference equation model (5)–(8) is analyzed the same way. The model equation (7) can be rearranged to express I_U in terms of I_T, as

$$cI_U(n) = I_T(n+1) - I_T(n) + \gamma I_T(n). \tag{11}$$

Substituting this equation into the left and right sides of (6) yields

$$\begin{aligned}&\big(I_T(n+2) - I_T(n+1) + \gamma I_T(n+1)\big) - \big(I_T(n+1) - I_T(n) + \gamma I_T(n)\big)\\ &\quad = \beta(n)S(n)\big(I_T(n+1) - I_T(n) + (c+\gamma)I_T(n)\big) - (c+\gamma)\big(I_T(n+1) - I_T(n) + \gamma I_T(n)\big).\end{aligned}$$

Rearranging the last equation enables us to evaluate the effective reproduction number $R_{\text{eff}}(n) = \beta(n)S(n)/\gamma$, finding

$$\boxed{R_{\text{eff}}(n) = \frac{1}{\gamma}\frac{\big(I_T(n+2) - 2I_T(n+1) + I_T(n)\big) + (c+2\gamma)(I_T(n+1) - I_T(n)) + \gamma(c+\gamma)I_T(n)}{I_T(n+1) - I_T(n) + (c+\gamma)I_T(n)}}. \tag{12}$$

This formula depends explicitly on the testing rate c and "recovery or death" parameter γ, as well as on the Infectious-Tested data $I_T(n)$.

To make use of formula (12) in practice, one needs values for $I_T(n)$ on some sequence of days. The initial day is not necessarily when the disease began spreading, and need not be when the first case was diagnosed. Section 5 explains how we estimate values of I_T from the testing data.

Note. The numerator and denominator of (12) should be positive. If they turn out to be negative, then either the real-world data we are relying on is too noisy, or else the real-world epidemic is not following an SIR model. For example, the denominator can be rewritten as $I_T(n+1) - (1-c-\gamma)I_T(n)$. Thus if the number of real-world Infectious-Tested cases decreases in a single day by more than fraction $c+\gamma$ of its value, one knows the denominator is negative and the reported data for the epidemic is not following the SIR model.

5 Computing R_{eff} deterministically from daily case numbers

Fully real-world data is not available for the number of Infectious-Tested individuals, since those people are not being tested every day to determine exactly when they recover. Estimated values for I_T are obtained instead by smoothing out the daily new-case counts using the Discrete Cosine Transform (see below) to get the smoothed daily case count $j(n)$. The estimated I_T value on day $n+1$ is now computed from its value on day n by the update formula

$$I_T(n+1) = I_T(n) + \frac{1}{N}j(n) - \gamma I_T(n), \qquad n = 1, 2, 3, \ldots, \tag{13}$$

with the initial value $I_T(1) = i(1)/N$ coming simply from the number of cases on that first day. The update formula is based on model equation (7), with the newly Infectious-Tested term $cI_U(n)$ in the model equation being replaced in the update formula by the smoothed new-case fraction $j(n)/N$, which we obtained from real-world data. Recoveries and deaths are estimated by subtracting the fraction γ of the cases, in keeping with the model equation (7). The resulting function $NI_T(n)$ gives the estimated number of Infectious-Tested individuals on day n.

One then calculates R_{eff} using equation (12).

Discrete Cosine Transform

Real world COVID-19 test data is noisy, and the noise comes from a variety of sources. The methods described in this paper use difference equations, which tend to amplify noise. Therefore a choice of smoothing procedure is critical to extracting the disease signal from the data. Several smoothing approaches were considered. We have chosen the Discrete Cosine Transform (DCT) method because its global nature handles residuals over the entire interval, and it permits filtering of high frequency components that are not epidemiologically justified.

The DCT can be affected by missing data and extreme outliers. To treat days where no cases were reported, we replace these data points with fitted values from a piecewise linear model (described in Section 6). Extreme outliers are identified with Tukey's intervals. First we take the lower quartile and upper quartile of the residuals, denoted LQ and UQ, and we take the observations for which the the residuals lie outside of the interval $[5LQ, 5UQ]$. We also replace these observations with the fitted values.

Then we concatenate the data with its even reflection, and repeat this data series ten times to get an extended data series x_0,...,x_n. (Reflecting and repeating the data seems to improve stability.) Apply the DCT to the natural log of this data series, that is, to $\ln x_0, \ldots, \ln x_n$, getting a transformed series $y_0, \ldots, y_n$. Filter the transformed data in the frequency domain by letting $z_k = y_k \times 10^{-3k/n}$. Apply the inverse DCT to obtain a data series $z_0, \ldots, z_n$ in the time domain. Remember this series is twenty times as long as the original data interval, and so the DCT-smoothed version of the original data consists of the first $1/20$-th of the series $z_0, \ldots, z_n$.

The filtering step reduces higher frequency noise in the disease signal, hence improving stability of the numerical derivatives in formula (12) for the effective reproduction number.

6 Confidence intervals — quantifying the stability of the deterministic method

Confidence intervals will be generated by fitting a piecewise linear curve to the log of the daily case data, and then randomly resampling the residuals following the wild-bootstrap method. This synthetic data is then smoothed using the Discrete Cosine Transform (DCT) [8], and a synthetic I_T is calculated using equation (13). We can then apply equation (12) to the synthetic I_T to calculate confidence intervals for R_{eff}. Details of the process are given below.

6.1 Fitting a linear model to the daily case data

To generate synthetic daily case data, first fit a piecewise linear polynomial.

- First order polynomials are used to fit the log of the daily case data. Higher order polynomials overfit the data when the daily new cases levels out, and at the endpoints.
- The simplest approach would be to equally space the knots, but this overfits the data when the daily cases approximately plateaus. A method to select non-equally spaced knots has been implemented (see below).
- The polynomial coefficients are calculated by fitting the curve to the *logarithm* of the daily case data, in order to reduce scale dependence. However the knots are selected by fitting the model to the raw daily case data. When the daily case data is low fewer knots are required to prevent over fitting as the disease signal is harder to detect.
- We aim to avoid fluctuations in the model. A "fluctuation" is defined as three consecutive intervals where the first derivative oscillates around 0 and the knots are 15 days or fewer apart.
- We want to take an interval of the data from the time the disease has been established in the population to the present. Compute the lower quartile of the daily case data (excluding the days where fewer than 10 cases were reported) and take the first day where the number of daily cases exceeds the (adjusted) lower quartile. Take a interval of the data from this day to the present.

We introduce a smoothing factor $s \geq 0$ to solve for the optimal knots. The smoothing parameter is a scalar which balances the closeness of fit with smoothness of fit. For a curve fitted with $s = 0$, a knot will be placed at every data point and the model will fit the data exactly. For a curve fitted with sufficiently large $s > 0$, knots will be placed at the endpoints and the data will be fitted with a single polynomial. The smoothing procedure is described by Dierckx [2], and we follow the implementation in scipy function `scipy.interpolate.splrep` [8].

The goal is to fit a piece-wise linear model to the log of the daily case data. We start by choosing knot locations (the days on which the linear model will have corners) by fitting the model to the raw daily case data (not the log of the data, for reasons explained below). To choose the knots, we take $s = 100 \times 2^n$ where n is the minimum integer such that:

- there exists a linear model fitted to the raw data with smoothing parameter s that has at most one knot for every 15 days of data,
- the linear model refitted to the log of the daily case data with the same knots contains no fluctuations.

This algorithm, which defines knots on the raw daily case data, in practice prevents over fitting where the daily case numbers are small.

For some populations the fit of the linear model can be poor if the trend of the daily case data drops too low between the knots, because the knot placement is dependent on the scale of the data. To handle this problem, take the current knots

k and takes intervals of the data between each of the knots. For each interval of data, set $l_0 = 1$, $u_0 = 1000$, $i := 0$ and proceed as follows.

1. Begin the i-th iteration by setting $s_i = (l_i + u_i)/2$ and fit the linear model to the log of the interval of the daily case data with the smoothing parameter s_i. Let k_i be the knots of the fitted model. (Note that we are using the log of the daily case data in these steps)
2. Form the global knots by concatenating the previous knots k and the new knots k_i to get a combined list of knots k'_i. Fit the linear model to the log of the total daily case data.
3. If the fitted linear model is feasible, set $u_{i+1} = s_i$, $l_{i+1} = l_i$. Otherwise, set $u_{i+1} = u_i$, $l_{i+1} = s_i$
4. If $u_i - l_i < 10^{-6}$, stop iterating and take the new knots k' to be the knots found by fitting the linear model to the daily case data with the smoothing parameter u_i. Otherwise, increment i and return to (1).

In some cases, this method is not good at fitting the data near the right endpoint. If the final knot is more than 28 days from the endpoint, we refit the linear model 14 times with an extra knot from 21 to 7 days from the right endpoint. Of these 14 linear fits, we choose the fit with the least residual sum of squares.

6.2 Uncertainty in the daily case data

Random errors are introduced when the SIR+B method is applied to noisy real world data.

The data we have used comes from three sources: the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University [11], the European Centre for Disease Prevention and Control (ECDC; an agency of the European Union) [13], and USAFacts, a "not-for-profit, nonpartisan civic initiative providing the most comprehensive and understandable government data" [14].

There are differences between these data sets which we cannot explain.

Compare, for example, the French daily new cases as reported by the CSSE and the ECDC (Figure 2). An extreme outlier appears in the CSSE data around 12 April, with over 25,000 new cases reported in just one day. The CSSE justifies its methodologies on github (`https://github.com/CSSEGISandData/COVID-19/issues/2459`) for example the decision to include probable cases as well as confirmed cases — "France has made the explicit decision to not test these cases and they will not be reflected in the official case count. We believe this masks the true extent of the infectious spread in the country and should be captured by those interested in understanding the case burden across time and space." It is unknown whether the methods of the CSSE correctly adjust for this so called masking of probable cases.

The CSSE French data set [11] has other problems too. On 19 April, the cumulative number of reported cases actually drops, which should of course be impossible. Other drops occur on 22 April, 29 April and 14 May. Further, the number of daily new cases seems unusually volatile in certain periods, such as late April.

Most countries and regions suffer from such problems with the data, although rarely to such an extent as with France. Another example of real-world data anomalies was demonstrated by the recent decision by the WHO to reassign 189 infections on the Ruby Princess cruise ship to the Australian state of New South Wales (NSW). Passengers disembarked in Sydney on 19 March 2020, but the cases were not attributed until 03 July 2020, resulting in an anomalous spike that day in the NSW time series.

These issues with the data and others, suggest that a certain amount of smoothing of the data is desirable, thus justifying the use of the DCT filtering in the previous section. Even with this smoothing, a certain variability will remain in the data on top of the underlying disease signal, and for that reason we apply bootstrapping.

6.3 Bootstrapping method

The wild bootstrap method is used to estimate 50% and 95% confidence intervals for R_{eff}. The term bootstrapping is used for any technique involving random re-sampling with replacement. Davidson et al. [1] state, "The wild bootstrap is based on an idea suggested by Wu (1986), and has been explored in detail by Hardle (1989, 1990) and Mammen (1992). The effectiveness of the wild bootstrap, particularly for studentized coefficients, was demonstrated by Mammen (1993)." Bootstrapping methods have been used in epidemiological models that account for the delay between case onset date and notification date [12].

The advantage of bootstrapping methods is that they can be used when the observed process and the nature of the measurement errors is not fully understood. We chose the wild bootstrap method because the number of daily new cases can be unusually volatile in certain periods (for example in the CSSE France data set), and hence the assumption that the measurement errors are normally distributed is not appropriate.

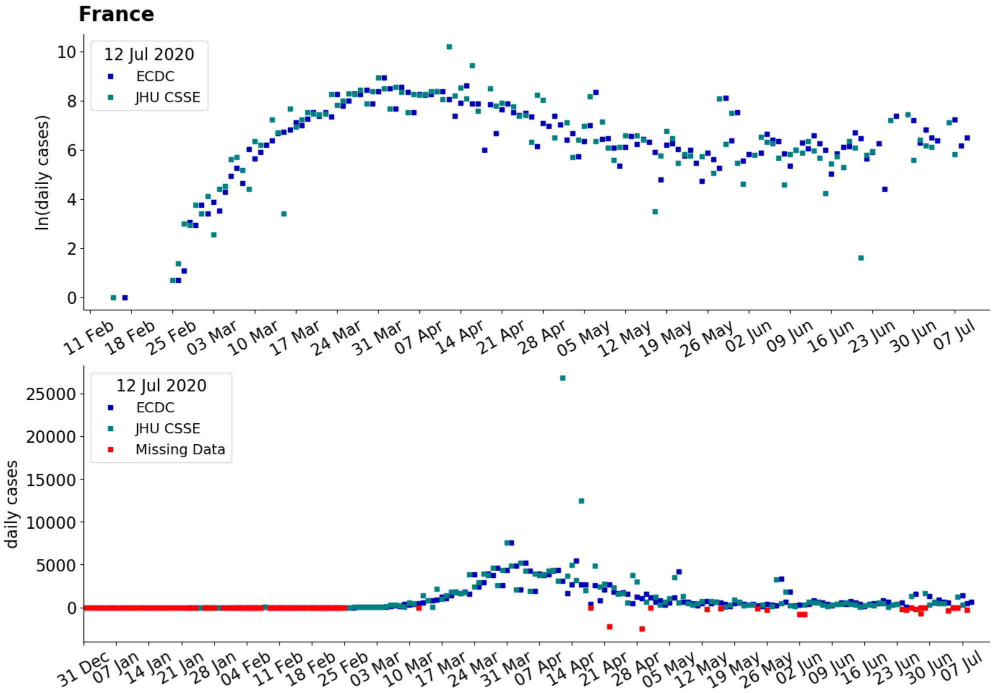

Figure 2: Top: The natural log of daily new case data for France as reported by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University and the European Centre for Disease Prevention and Control (ECDC). Bottom: The France daily new case data on a linear scale.

First we describe wild bootstrapping in general, and then explain how it is used in this paper. To generate bootstrapped observations, first consider a linear model $Y = X\beta + \epsilon$ and fit it by the method of least squares as $y = X\beta + u$, where X is the design matrix, β is the least squares predictor, Y is the vector of true values, y is the vector of observations, ϵ is the (unobservable) error vector and u is the vector of residuals. Then one randomly re-samples the errors, as follows. The bootstrapping is "wild" because the error of the i-th observation is modelled with an independent random variable F_i. These variables are chosen [6] such that the expectations satisfy $E(F_i) = 0$, $E(F_i^2) = u_i^2$ and $E(F_i^3) = u_i^3$ where u_i is the i-th residual.

Mammen provides three different choices for the random variable F_i. The one we choose to work with is $F_i = u_i U_i$ where

$$U_i = (\delta_1 + V_{i,1}/\sqrt{2})(\delta_2 + V_{i,2}/\sqrt{2}) - \delta_1\delta_2,$$

with $\delta_1 = (3/4 + \sqrt{17}/12)^{1/2}$ and $\delta_2 = (3/4 - \sqrt{17}/12)^{1/2}$ and where $V_{i,1}$ and $V_{i,2}$ are independent normal random variables with mean 0 and variance 1.

We want to apply this wild bootstrapping method to the log of the daily case data. Unfortunately, some of the residuals are not well defined, because on some days no new cases were reported (meaning we would need to take the log of zero), or the cumulative number of reported cases goes down rather than up, which is impossible. Mammen's F_i is not well defined for these exceptional observations in the data, and so we need to create somehow residual values for those days. Another issue is that having a small residual u_i on a certain day does not mean we have more certainty in the underlying data for that day (since the data is somewhat noisy). Thus it would seem unjustified to always force a small variance by using the value u_i in the bootstrapped F_i.

To address these issues and handle all data points in a consistent fashion, a modified residual u_i^* is defined below. First randomly choose a residual $u_{j(i)}$ from the 5 closest observations for which the residual is well defined (inclusive of the i-th observation), with each residual having equal probability of being chosen. If the i-th residual is not defined, choose from the nearest two residuals on the left and two on the the right that are well defined. Several exceptional situations must be handled, for example if the residual of a neighboring data point is also not defined. Such points can be dealt with by hand as necessary. Second, normalise the chosen residuals by letting

$$u_i^* = \sigma(u_{j(i)} - \widetilde{u})$$

where $\widetilde{u}$ is the mean of the values $u_{j(1)}, \ldots, u_{j(n)}$ and σ is the standard deviation of the well defined residuals u_i divided by the standard deviation of the $u_{j(i)}$. This normalization ensures that

$$\frac{1}{n}\sum_{i=1}^{n} u_i^* = 0, \qquad \frac{1}{n}\sum_{i=1}^{n}(u_i^*)^2 = \frac{1}{m}\sum_{i}(u_i - \overline{u})^2,$$

where $\overline{u}$ is the mean of the residual values u_i that are well defined, with m being the number of such residuals. We then define $F_i^* = u_i^* U_i$ and generate a bootstrapped observation at the i-th data point by randomly re-sampling ϵ_i^* from F_i^* and computing $y_i^* = X\beta + \epsilon_i^*$.

When the model is fitted with the least squares, the residual vector is automatically orthogonal to the columns of the design matrix:

$$X^T u = X^T(y - X\beta) = X^T y - X^T X(X^T X)^{-1} X^T y = 0,$$

where we used the standard formula (12) for β. Note the vector of modified residuals u^* need not be orthogonal to the column space.

The Bootstrap method is summarised as

1. Fit the linear model using the procedure outlined in Section 6.1.
2. Compute the residuals. At the i-th data point (inclusive of the days where less than 1 case was reported), randomly choose $u_{j(i)}$ from the nearest neighbours, compute u_i^* and define $F_i^* = u_i^* U_i$ for $i = 1, ..., n$ (see above).
3. Generate bootstrapped observations by randomly re-sampling the error ϵ_i from F_i^* to arrive at $y_i^* = X_{ij}\beta_j + \epsilon_i$.
4. Compute R_{eff} for the bootstrapped data set.
5. Repeat steps 2 – 4 a large number of times (e.g. 1000 times). Smooth the raw and bootstrapped data on the linear scale 3 times with the 3-point Newton-Cotes rule, the 7-point Newton-Cotes rule and the 3-point Newton Cotes rule respectively.
6. Compute R_{eff} for the raw data and bootstrapped data. Construct a 50% and 95% confidence interval by taking the 5th, 25th, 75th and 95th percentiles of the bootstrapped R_{eff}'s for each day.

The wild bootstrap produces artificial daily case data which compares well to the raw data. For example, Figure 3 shows the raw daily case data for France (data source ECDC) together with daily case data generated by bootstrapping.

7 Results

In a separate technical report Grice et al. [3] have used the SIR+B process to calculate R_{eff} for several countries. In that technical report the authors compared the SIR+B results with the results of other statistical approaches. We refer the reader to the technical report to assess the value of the SIR+B process when applied to real world data.

8 Conclusion

A SIR+B process for estimating the effective reproduction number from daily case data has been developed. The process has the following desirable attributes:

- it is fast, taking about 0.1 seconds per country analysed, when coded in Python on a standard desktop machine,
- it is based directly on the epidemiological SIR model, giving an analytic formula for the effective reproduction number, with confidence intervals subsequently generated by wild bootstrapping,
- it involves many fewer assumptions than other methods such as EpiEstim,
- the deterministic calculation of R_{eff} depends on only two parameters, γ and c,
- it is relatively insensitive to those parameters (see [3]).

Thus the paper demonstrates the potential of the SIR+B process to inform public health policy in the immediate future.

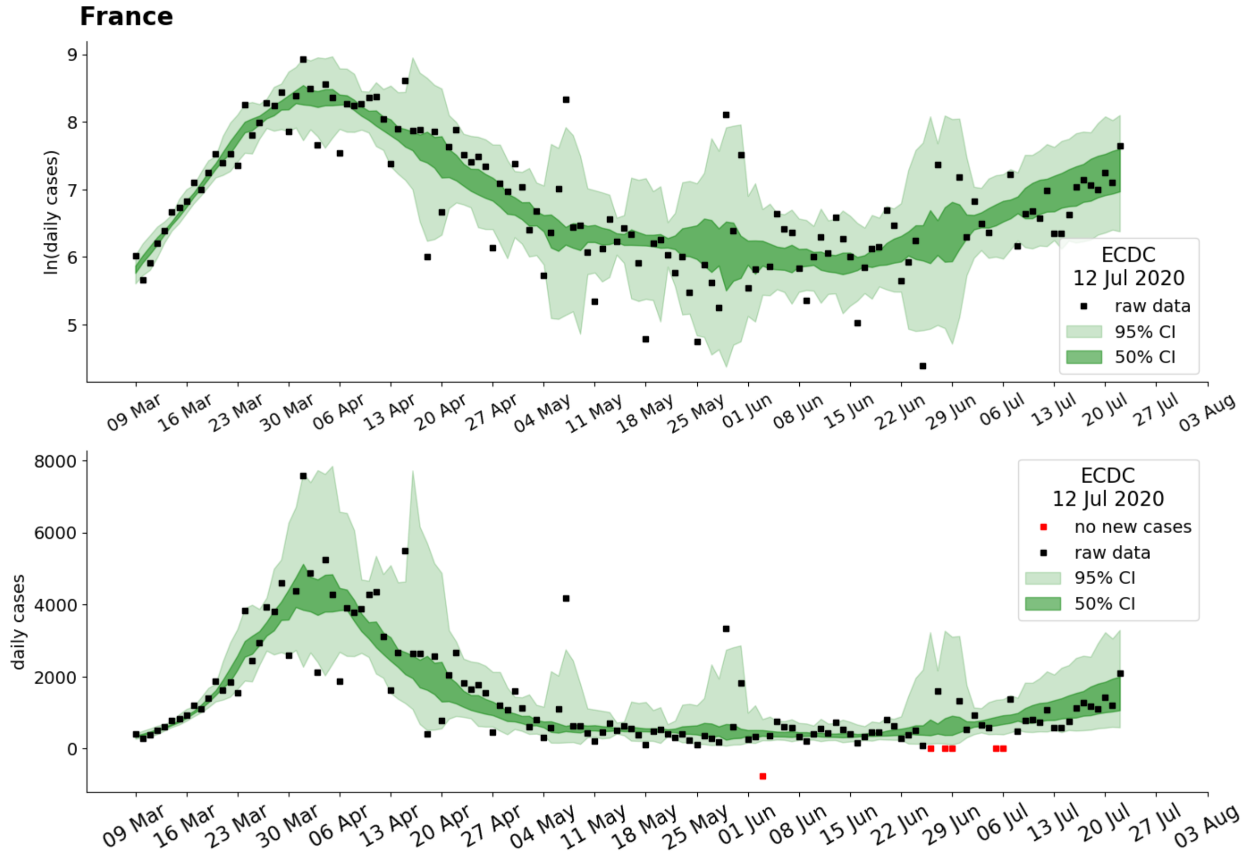

Figure 3: Top: ECDC natural log of daily new cases (France) versus the confidence intervals for the synthetic daily case data. Bottom: The top plot on the displaying the raw data on the linear scale.

A Using Synthetic data to reconstruct deterministic R_{eff} values

In this appendix, we test the deterministic part of the SIR+B process on a numerically exact solution of the SIR system that features rapid changes in $R_{\text{eff}}(t)$. Consider the SIR system (5)–(8) over a full year ($1 \leq n \leq 365$), with parameter values $\gamma = 1/15, c = 1/15$, and initial conditions

$$S(1) = 1 - 10/(5 \times 10^6), \quad I_U(1) = 10/(5 \times 10^6), \quad I_T(1) = 0, \quad R(1) = 0, \quad D(1) = 0.$$

That is, the epidemic begins with 10 infected people (untested) in a population of 5 million.

The transmission coefficient $\beta(n)$ that is applied to the system is chosen to exhibit rapid changes between two levels, as shown in Figure 4, corresponding to normal societal conditions ($\beta/\gamma = 2.5$) and severe lockdown conditions ($\beta/\gamma = 0.5$). These values correspond to the "R_0" values that a fully susceptible population ($S = 1$) would experience, under these conditions.

Solving the discrete time SIR system numerically yields the plots of $I_U(n)$ and $I_T(n)$ shown on the right of Figure 4. The effective reproduction number $R_{\text{eff}} = \beta S/\gamma$ is plotted on the left of Figure 5. It drops off somewhat, because $S(n)$ decreases with time.

This oscillating R_{eff} is detected by the method, on the right of Figure 5. Notice the method is not reliable until about day 30. Before that time, the number of daily new tested cases is very small (less than 10), and our rounding of the underlying solution to give integer-valued case numbers is enough to throw off the method. After day 30 the number of cases gets larger, and the method starts giving sensible estimates for the reproduction number.

This synthetic result gives reason to believe that the method can reproduce R_{eff} for real world data also. It is important to note, though, that the daily case numbers in this example are derived from an exact computed solution to an SIR system, so that even after rounding, the approximate I_T values being fed into formula (12) are high on signal and low

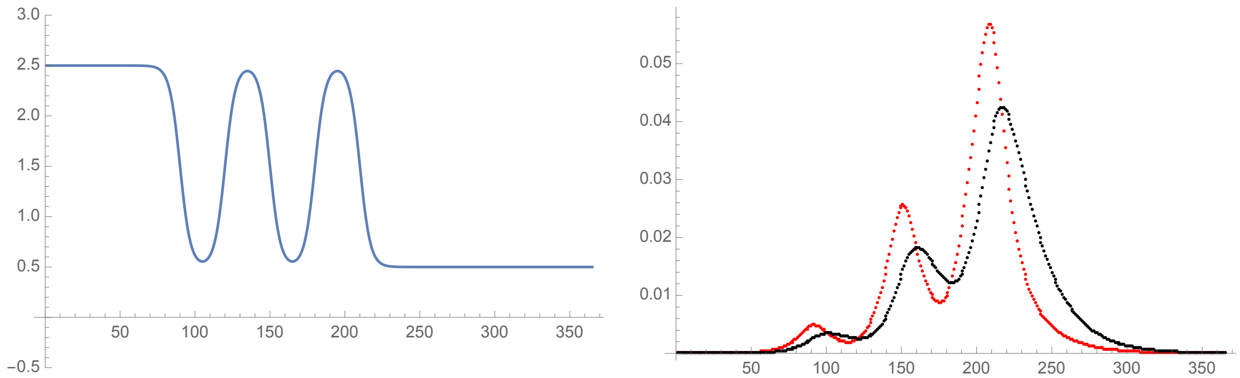

Figure 4: Left: plot of the applied reproduction number $\beta(n)/\gamma$ in the test of the method, in Section A. Right: The Infectious-Untested I_U (red, higher peak) and Infectious-Tested I_T (black, lower peak) curves computed for that discrete time SIR system.

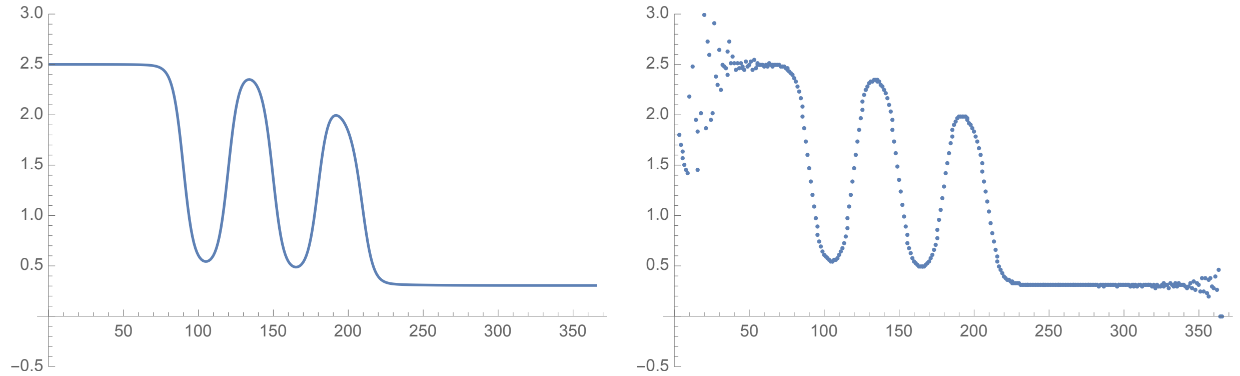

Figure 5: Left: plot of the exact $R_{\text{eff}}(n)$, which equals the applied reproduction number $\beta(n)/\gamma$ multiplied by the computed Susceptible fraction $S(n)$. Right: the effective reproduction number $R_{\text{eff}}(n)$ computed numerically from (12), by using the rounded, exact new-case value $i(n+1) = \text{Round}\,(NcI_U(n))$ for day $n+1$ in order to get approximate values for I_T as in Section 5. The computed values on the right correctly pick out the oscillations in R_{eff}, showing that the method has adequate resolution even in the face of repeated changes in the applied reproduction number (i.e., changes in lockdown conditions).

on noise. Real data for real populations will not be as favorable, as is seen when using the method to estimate $R_{\text{eff}}(t)$ using real-world data [3].

References

[1] A. C. Davison and D. V. Hinkley. Bootstrap Methods and their Application. Cambridge University Press, Cambridge, 1997.

[2] P. Dierckx. Curve and surface fitting with splines. Oxford University Press (June 15, 1995)

[3] P. Grice, S. Grice and R. S. Laugesen. Calculating the effective reproduction number for COVID-19 using a new process for various countries Technical report. `https://www.covid19-blog.closeassociate.com/`

[4] J. D. Hunter. Matplotlib: A 2D Graphics Environment. Comput. Science & Engin. CiSE 9 (2007), no. 3, 90–95.

[5] A. James, S. C. Hendy, M. J. Plank and N. Steyn. Suppression and mitigation strategies for control of COVID-19 in New Zealand. (26 March 2020).
`https://cpb-ap-se2.wpmucdn.com/blogs.auckland.ac.nz/dist/d/75/files/2017/01/Supression-and-Mitigation-Strategies-New-Zealand-TPM-1.pdf`

[6] E. Mammen. Bootstrap and wild bootstrap for high dimensional linear models. Ann. Statist. 21 (1993), no. 1, 255–285.

[7] W. McKinney. Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference (2010), 51–56. http://conference.scipy.org/proceedings/scipy2010/mckinney.html

[8] SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, 51–56. https://doi.org/10.1038/s41592-019-0686-2

[9] J. D. Murray. Mathematical Biology. I and II. An Introduction. Third edition. Interdisciplinary Applied Mathematics, 17 and 18. Springer–Verlag, New York, 2002 and 2003.

[10] S. van der Walt, S. C. Colbert and G. Varoquaux. The NumPy Array: A structure for efficient numerical computation.

[11] COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University https://github.com/CSSEGISandData/COVID-19

[12] Epiforecasts methods.Rmd https://github.com/epiforecasts/covid/blob/master/methods.Rmd

[13] European Centre for Disease Prevention and Control https://www.ecdc.europa.eu/en/publications-data/download-todays-data-geographic-distribution-covid-19-cases-worldwide

[14] USAFacts https://usafacts.org/visualizations/coronavirus-covid-19-spread-map/