Forecasting epidemic trajectories: Time Series Growth Curves package tsgc

Michael Ashby
Downing College
Cambridge University

Andrew Harvey
Faculty of Economics
Cambridge University

Paul Kattuman*
Judge Business School
Cambridge University

Craig Thamotheram
TAC Index

Abstract

This paper documents the Time Series Growth Curves (**tsgc**) package for R, which is designed for forecasting epidemics, including the detection of new waves and turning points. The package implements time series growth curve methods founded on a dynamic Gompertz model and can be estimated using techniques based on state space models and the Kalman filter. The model is suitable for predicting future values of any variable which, when cumulated, is subject to some unknown saturation level. In the context of epidemics, the model can adjust to changes in social behavior and policy. It is also relevant for many other domains, such as the diffusion of new products. The **tsgc** package is demonstrated using data on COVID-19 confirmed cases.

Keywords: Covid-19, Gompertz growth curve, Kalman filter, reproduction number, state space model, stochastic trend, turning points.

1. Forecasting epidemic trajectories

Outbreaks of infectious diseases with epidemic potential require real-time responses by public health authorities. Accurate real-time forecasting of the trajectory of the epidemic over the near future is of great value in this regard.

The R package, **tsgc**, is intended for use in monitoring and forecasting the progress of an epidemic, including the detection of *new waves* and *turning points*.[1] It develops and implements time series growth curve methods first reported in Harvey and Kattuman (2020) (hereinafter referred to as HK). HK develop a class of time series models for predicting future values of a variable which, when cumulated, is subject to an unknown saturation level. In a single wave of an epidemic, as more and more people get infected, the pool of susceptible individuals dwindles. This results in the decline of new infections, and the cumulative number of infections approaches its saturation level. The model can take account of deviations relative to

*Correspondence to: p.kattuman@jbs.cam.ac.uk

[1] The package is available from `https://github.com/Craig-PT/tsgc`.

this canonical trajectory due to changes in social behavior and policy. Models in this family are relevant for many other disciplines, such as marketing (when estimating the demand for new products). While attention here is focused on the spread of epidemics and the applications used for illustration relate to coronavirus, this package is designed with a view to wider applicability.

Given the number of different modeling approaches for epidemics, there are many notable packages that can be used for monitoring epidemics. For the most part, these seek to model explicitly the mechanism by which the disease spreads through the population. For example, **EpiModel** (Jenness, Goodreau, and Morris 2018), **EpiEstim** (Cori, Ferguson, Fraser, and Cauchemez 2013), **epinowcast** (Abbott and Monticone 2021) to name a few, can be categorized as belonging to the class of 'mechanistic' models in the language of philosophy of science, in that they require structural knowledge of the disease spread mechanism in order to obtain predictions.

In contrast, the empirical approach implemented in **tsgc** falls into the class of models described as 'phenomenological'. Although it is motivated by the archetypal pattern in the dynamics of disease spread, it does not rely on structural assumptions derived from epidemiological theory. There are advantages to not requiring assumptions about values of parameters relating to, inter alia, disease infectiousness, disease severity, or contact patterns, which are difficult to pin down with sufficient precision, especially in real-time during an epidemic. Our approach makes minimal assumptions and merely requires past observations of the epidemic variable of interest, to which we apply time-series methods to provide predictions over short future time horizons. The model can be estimated quickly and straightforwardly, and subjected to standard diagnostic tests. A statistical model of this type is a useful complement to mechanistic models that attempt to describe the epidemic in terms of underlying processes.

Section 2 sets out the state space formulation of the dynamic Gompertz growth curve and the way nowcasts and forecasts are obtained from predictive recursions. It is then shown how these numbers translate into estimates of the instantaneous reproduction number R_t. Section 3 explains how multiple waves can be accommodated by reinitializing the series at the start of new waves. The start of a new wave is not obvious in real time but a rule for triggering reinitialization that works well in practice is presented. Section 4 describes the functionality of **tsgc**. Section 5 sets out a full working example of the use of the package to forecast COVID infection in Gauteng province in South Africa. Section 6 concludes.

2. Theory

2.1. Gompertz curve

Our model is based on the sigmoidal growth curve pattern that characterizes epidemics. We start by assuming that the cumulative number of cases follows a Gompertz curve, which is a parsimonious model for the canonical sigmoid shape of cumulative case numbers in a one-wave epidemic. Over the course of a wave, the number of new infected cases increases up to a peak before declining to zero as the pool of susceptible individuals declines. Specifically, if the cumulative number of cases at time t, $\mu(t)$, follows a Gompertz curve, we can write

$$\mu(t) = \bar{\mu} \exp\{\gamma_0 e^{\gamma t}\},$$

where $\bar{\mu}$ is the unknown saturation level for the cumulative number of cases, $\gamma_0 < 0$ is a parameter related to $\mu(0)$ and $\gamma < 0$ is the growth rate parameter. Defining $\dot{\mu}(t) \equiv d\mu(t)/dt$ and $g(t) \equiv \dot{\mu}(t)/\mu(t)$, it is straightforward to show that

$$\ln g(t) = \delta + \gamma t, \tag{1}$$

where $\delta = \ln \gamma_0 \gamma$.

The observational model needs to be specified in discrete, rather than continuous, time. This is straightforward. Let Y_t be the observed cumulative number of cases on day t and $y_t = Y_t - Y_{t-1}$ be the number of daily new cases.[2] We can then define the growth rate of Y_t as $g_t = y_t/Y_{t-1}$ and replace $\ln g(t)$ with $\ln y_t - \ln Y_{t-1}$.

2.2. Dynamic Gompertz model

The deterministic trend implied by (1) is too inflexible for practical time-series modeling of an epidemic. Replacing it with a stochastic trend allows the model to adapt to changes in dynamics during the course of the epidemic. We call this stochastic-trend counterpart of (1) the dynamic Gompertz model. It is a local linear trend model specified as

$$\ln g_t = \delta_t + \varepsilon_t,\ \varepsilon_t \sim NID(0, \sigma_\varepsilon^2),\ t = 2, ..., T, \tag{2}$$

where $\ln g_t = \ln y_t - \ln Y_{t-1}$ and

$$\begin{aligned} \delta_t &= \delta_{t-1} + \gamma_{t-1}, & (3)\\ \gamma_t &= \gamma_{t-1} + \zeta_t,\ \zeta_t \sim NID(0, \sigma_\zeta^2), & (4) \end{aligned}$$

where the disturbances ε_t and ζ_t are mutually independent, and $NID(0, \sigma^2)$ denotes normally and independently distributed with mean zero and variance σ^2. Note that the larger the signal-to-noise ratio, $q_\zeta = \sigma_\zeta^2/\sigma_\varepsilon^2$, the faster the estimate of the slope parameter, γ_t, which can be interpreted as the growth rate of the growth rate of cumulative cases, changes in response to new observations. Conversely, a lower signal-to-noise ratio induces more smoothness to the estimates. When $\sigma_\zeta^2 = 0$, the trend is deterministic as in (1).

2.3. State space form and estimation

It is convenient to write the dynamic Gompertz model in general state space form:

$$\begin{aligned} \ln g_t &= Z\alpha_t + \varepsilon_t & \varepsilon_t &\sim NID(0, \sigma_\varepsilon^2)\\ \alpha_{t+1} &= T\alpha_t + R\eta_t & \eta_t &\sim NID(0, Q) \end{aligned}$$

with

$$\alpha_t = (\delta_t, \gamma_t)',\ Z = (1, 0),\ \eta_t = (0, \zeta_t)',\ T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix},\ R = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix},\ Q = \begin{pmatrix} 0 & 0 \\ 0 & \sigma_\zeta^2 \end{pmatrix}.$$

This model can be estimated using techniques based on the Kalman filter once a prior is specified. The prior is

[2] Of course, while y_t denotes daily new cases here, it could equally denote weekly sales of a new product, etc. None of the analysis here is dependent on the data frequency.

$$(\delta_1, \gamma_1)' \sim N(a_1, P_1),$$

where a_1 is a 2×1 vector of prior means and P_1 a 2×2 prior variance matrix. We use a diffuse prior due to the absence of prior information about the epidemic when the model is first estimated: i.e., we set $a_1 = (0, 0)'$, $P_1 = \kappa I$, and let $\kappa \to \infty$. Model estimation, including implementation of the diffuse prior, is carried out using the **KFAS** package (Helske 2017).

The Kalman filter outputs estimates of the state vector $(\delta_t, \gamma_t)'$. The estimates at time t conditional on information up to and including time t are denoted $(\hat{\delta}_{t|t}, \hat{\gamma}_{t|t})'$ and given by the contemporaneous filter; the predictive filter estimates the state at time $t+1$ from the same information set, outputting $(\hat{\delta}_{t+1|t}, \hat{\gamma}_{t+1|t})'$.

It may be useful to review past movements of the state vector $(\delta_t, \gamma_t)'$. This can be done using the smoothed estimates $(\hat{\delta}_t, \hat{\gamma}_t)'$, which denotes the estimates of the state vector at time t based on all T observations in the series.

Estimation of the unknown variance parameters (σ^2_ε and σ^2_ζ) is by maximum likelihood (ML) and is carried out using **KFAS** following the procedure described in Helske (2017). We retain the option of either estimating the signal-to-noise ratio q_ζ, or of fixing it at a plausible value. In practice, for coronavirus applications, we set the value of q_ζ based on experience and judgment, reducing the number of parameters to be estimated by one. Tests for normality and residual serial correlation are based on the standardized innovations, that is one-step ahead prediction errors, $v_t = \ln g_t - \delta_{t|t-1}$, $t = 3, ..., T$.

Daily effects, which are generally quite pronounced in the coronavirus data, can be included in the model as described in the Appendix.

2.4. Forecasts and peak prediction

Forecasts of future observations are obtained from the predictive recursions

$$\begin{aligned} \widehat{g}_{T+\ell|T} &= \exp(\hat{\delta}_{T|T} + \hat{\gamma}_{T|T}\ell), \;\; \ell = 1, 2, .. \\ \widehat{\mu}_{T+\ell|T} &= \widehat{\mu}_{T+\ell-1|T}(1 + \widehat{g}_{T+\ell|T}) \end{aligned}$$

so that

$$\widehat{y}_{T+\ell|T} = \widehat{g}_{T+\ell|T}\widehat{\mu}_{T+\ell-1|T} = Y_T \exp \hat{\delta}_{T+\ell|T} \prod_{j=1}^{\ell-1}(1 + \exp \hat{\delta}_{T+j|T}) \tag{5}$$

and $\widehat{Y}_{T+\ell|T} = \widehat{\mu}_{T+\ell|T}$; the initial value is $\widehat{\mu}_{T|T} = Y_T$.

We construct forecast intervals for y_t based on the prediction intervals for δ_t. The conditional distribution of future values of $\hat{\delta}_t$ is Gaussian. We replace $\hat{\delta}_{T+j|T}$ in (5) with the upper bound of a prediction interval for $\hat{\delta}_{T+j|T}$ to compute the upper bound of our forecast interval for $y_{T+\ell}$ and likewise for the lower bound.[3] In effect, the forecast intervals are based on inference on the log cumulative growth rate, δ_t.

[3] These are not proper prediction or confidence intervals for $y_{T+\ell}$. The one-step-ahead predictive distribution of $\hat{y}_{T+\ell|T}$ (for $\ell = 1$) is lognormal. This is not the case more than one step ahead due to the presence of the cumulative total in equation (2).

The filtered growth rate $\hat{g}_{y,t|t}$ of new cases y_t, can be extracted from the continuous-time incidence curve: $\mu'(t) = g(t)\mu(t)$, where $\mu(t)$ is the growth curve and $g(t)$ is its growth rate. Taking logarithms and differentiating

$$\hat{g}_{y,t|t} = \hat{g}_{t|t} + \hat{\gamma}_{t|t}, \tag{6}$$

where $\hat{g}_{t|t} = \exp \hat{\delta}_{t|t}$. The sampling variability of $\hat{g}_{t|t}$ is dominated by that of $\hat{\gamma}_{t|t}$ (see Harvey and Kattuman 2021). Therefore when constructing confidence intervals for $\hat{g}_{t|t}$ we treat $\hat{g}_{y,t}$ as if it has a normal distribution centered on $\hat{g}_{y,t|t}$ with variance $\mathrm{Var}(\hat{\gamma}_{t|t})$.

Even when the nowcast $\hat{g}_{y,T|T}$ is positive and daily cases are growing, there will be a saturation level for the cumulative total, Y_t, so long as $\hat{\gamma}_{T|T}$ is negative. The nowcasts of y_t peak when $\hat{g}_{y,t|t} = 0$, which requires $\hat{\gamma}_{t|t}$ to be sufficiently negative to outweigh $\hat{g}_{t|t}$, which is, of course, always positive. This can be seen from the expression for the growth rate of daily cases:

$$\hat{g}_{y,T|T} = \exp \hat{\delta}_{T|T} + \hat{\gamma}_{T|T} = \hat{g}_{T|T} + \hat{\gamma}_{T|T}. \tag{7}$$

When $\hat{\gamma}_{T|T}$ is negative, there is a flattening of the curve and a signaling of an upcoming peak in the trend of y_t. As shown in [HK, p10], the peak in the trend is predicted to be ℓ_T days ahead where[4]

$$\ell_T = \frac{\ln(-\hat{\gamma}_{T|T}) - \hat{\delta}_{T|T}}{\hat{\gamma}_{T|T}} = \frac{\ln(-\hat{\gamma}_{T|T}/\hat{g}_{T|T})}{\hat{\gamma}_{T|T}}, \quad -\hat{g}_{T|T} < \hat{\gamma}_{T|T} < 0.$$

The generation of forecasts is demonstrated in Section 4.

2.5. Reproduction Number R_t

The path of the epidemic is best tracked by nowcasts and forecasts of $g_{y,t}$, the growth rate of y_t, which are constructed by HK from the filtered estimates in the state space model, (2), (3) and (4). Wallinga and Lipsitch (2007) describe how the estimates of $g_{y,t}$ can be translated into estimates of the instantaneous reproduction number R_t. Harvey and Kattuman (2021) propose

$$\widetilde{R}_{t,\tau} = 1 + \tau g_{y,t|t} \quad \text{or} \quad \widetilde{R}^e_{\tau,t} = \exp(\tau g_{y,t|t}), \tag{8}$$

where τ is the generation interval – the typical number of days between an infected person becoming infected and them transmitting the disease to someone else. We construct credible intervals for $\widetilde{R}_{t,\tau}$ and $\widetilde{R}^e_{\tau,t}$ by substituting the upper and lower bounds of the confidence intervals for $g_{y,t}$ into (8) to get the upper and lower bounds of the credible intervals. See Harvey, Kattuman, and Thamotheram (2021) for an application.

The estimates of R_t can be used for tracking and forecasting the epidemic. The nowcasts of y_t peak when $\hat{g}_{y,t|t} = 0$, corresponding to $\widetilde{R}_{t,\tau} = \widetilde{R}^e_{\tau,t} = 1$. Based on (7), predictions of $g_{y,t}$ are given by

$$\hat{g}_{y,T+\ell|T} = \exp \hat{\delta}_{T+\ell|T} + \hat{\gamma}_{T+\ell|T} = \exp(\hat{\delta}_{T|T} + \hat{\gamma}_{T|T}\ell) + \hat{\gamma}_{T|T}, \qquad \ell = 1, 2, . \tag{9}$$

We can then obtain predictions of R_t, as in (8). If $\hat{\gamma}_{T|T}$ is zero, the estimated growth of y_t is exponential and it is helpful to characterize it by the doubling time, $\ln 2/\hat{g}_{y,T|T} = 0.693 \exp(-\hat{\delta}_{T|T})$.

[4] Note the change in sign of γ as compared with HK.

When $\exp \hat{\delta}_{T|T} + \hat{\gamma}_{T|T} > 0$, the nowcast $\hat{g}_{y,T|T}$ is positive and the estimate of R_t given by (8) is greater than one. So long as $\hat{\gamma}_{T|T}$ is negative, then as $T \to \infty$, $\widetilde{R}^{e}_{\tau,T+\ell|T} \to \exp(\tau\hat{\gamma}_{T|T}) < 1$, and a saturation level for Y appears on the horizon.

We now turn to case where γ_t potentially turns positive in a typically short-lived phase, as a new wave emerges.

3. Reinitialization

The coronavirus pandemic was characterized by multiple waves punctuated by plateaus. At the beginning of a new wave the growth rate of daily cases, $g_{y,t}$, turns positive. The initial surge may be explosive to the point where the growth is *super exponential.* In this case, γ_t, the growth rate of g_t (which is the growth rate of cumulative cases) can also turn positive, with no peak in prospect for y_t. Such a phase can be expected to be transient, with γ_t dropping back to zero (exponential growth in infection, accompanied by an upcoming peak in y_t), and then falling below zero (sub-exponential growth in infection).

From the point-of-view of forecasting an epidemic, a peak must be in prospect even if it can only be expected some way into the future. There is thus a need for a solution to the problem of the estimated $\hat{\gamma}_{t|t}$ rising to positive values as it adapts to the upward surge in y_t, and remaining positive for any protracted period. This upward shift in $\hat{\gamma}_{t|t}$ can be averted by *reinitializing* the $\ln g_t$ series at the start of a new wave. This involves setting the cumulative total of cases Y_t back to zero at, or around, the start of a new wave and setting γ_t to zero so as to impose exponential growth. From the point-of-view of the relationship $g_{y_t} = g_t + \gamma_t$, the re-initialization effectively shifts "surplus" γ_t emanating from super-exponential growth, into δ_t and therefore into g_t (since $g_t = \exp \delta_t$). Note that, on the date of the re-initialization, $g_{y,t} = g_t$, since $\gamma_t = 0$, and both $g_{y,t}$ and g_t will be high because a new wave is taking off.

3.1. Reinitalizing the data series

Let $t = r$ denote the re-initialization date and let r_0 denote the date at which the cumulative series is set to 0. Then:

$$\begin{aligned}
\ln g_t &= \ln y_t - \ln Y_{t-1} && t = 1, \ldots, r \\
\ln g^r_t &= \ln y_t - \ln Y^r_{t-1} && t = r+1, \ldots, T \qquad (10)\\
Y^r_t &= Y^r_{t-1} + y_t && t = r, \ldots, T \qquad (11)
\end{aligned}$$

where Y^r_t denotes the cumulative cases after re-initialization. We set $Y^r_{r-1} = 0$, so that the growth rate of cumulative cases is available from $t = r+1$ onwards. Note that $Y^r_t = Y_t - Y_{r_0}$. The gap between the two series becomes apparent by writing

$$\ln g^r_t = \ln g_t + \ln \frac{Y_{t-1}}{Y^r_{t-1}} = \ln g_t + \ln \frac{Y_{t-1}}{Y_{t-1} - Y_{r_0}} \qquad t = r+1, \ldots, T \tag{12}$$

In the next section, where we illustrate the working of the program, it can be seen that in contrast to the original $\ln g_t$ series, which continues to increase, the reinitialized $\ln g_t$ series begins to decrease from the reinitialization date. The reinitialization enforces the canonical Gompertz curve with the log of growth rate of cumulative cases sloping down.

3.2. Reinitalizing the model

We reinitialize the model by specifying the appropriate prior distribution for the initial states: $\alpha_1^r \sim N(a_1^r, P_1^r)$ with $a_1^r = (a_{\delta,1}^r, a_{\gamma,1}^r)'$ and P_1^r defined as follows. Let $\mathcal{F}_t = \ln g_t, \ln g_{t-1}, \ldots, \ln g_1$ and define $a_t = E(\alpha_t|\mathcal{F}_{t-1})$, $a_{t|t} = E(\alpha_t|\mathcal{F}_t)$, $P_t = Var(\alpha_t|\mathcal{F}_{t-1})$, $P_{t|t} = Var(\alpha_t|\mathcal{F}_t)$. Then,

$$\begin{aligned} a_{\delta,1}^r &= a_{\delta,r+1} + \ln(Y_r/y_r) \\ a_{\gamma,1}^r &= 0 \\ P_1^r &= P_{r+1}, \end{aligned}$$

where $a_{\delta,r+1}$ and P_{r+1} are obtained from the non-reinitialized model estimated over $t = 1, \ldots, r$ via the usual Kalman filter recursions. Adding $\ln(Y_r/y_r)$ to $a_{\delta,r+1}$ corrects for the shift down in the log cumulative cases caused by reinitializing the cumulative case series. Setting $a_{\gamma,1}^r = 0$ ensures the model starts off with exponential, rather than super-exponential, growth.

We reinitialize the model through the priors in this way rather than simply re-estimating the model from scratch for two reasons. First, it allows us to impose a proper (rather than diffuse) prior centered on zero for γ, so that the starting point is exponential growth. Second, it enables us to make use of data from before the reinitialization date. One needs a reasonable sample size for the estimated model and forecasts to be reliable, but if a new wave is taking off, forecasts need to be generated quickly. This was particularly true with the emergence of the Omicron variant, which caused an explosive increase in infection over a short period of time.

We do not re-estimate the σ_ε^2 or σ_ζ^2 parameter in the reinitialized model. Rather, we use the values estimated in the original model over $t = 1, \ldots, r$. The one-step-ahead prediction error at $t = r$ is the same in both the initialized and reinitialized models, but after $t = r$, the prediction errors diverge.

The reinitialization procedure is very similar in the case where we have seasonal terms. If we let $\alpha_{s,t}$ be the vector of seasonal states and maintain an analogous notation to that above, the prior mean of the seasonal components in the reinitialized model is

$$a_{s,1}^r = a_{s,r+1}.$$

The prior variance of α_1^r remains P_{r+1} where P_{r+1} is appropriately re-defined to include the seasonal term, as described in the Appendix.

4. Functionality of tsgc

The two main classes in **tsgc** are `SSModelDynamicGompertz` and `SSModelDynGompertzReinit`. These implement the models described in (2)-(4), with and without reinitialization, respectively. They both inherit from a common base class `SSModelBase`, which acts as a wrapper around **KFAS** to set up the state space model and define consistent update and estimation methods for it. The unknown parameters are estimated with the `estimate` method in both classes. This estimation returns an object of the `FilterResults` class, which is a wrapper around the **KFAS** `KFS` class with a date index and additional methods for prediction attached.

The `SSModelDynamicGompertz` needs only a cumulative series `Y` as an input. In our application, this is the cumulative number of new coronavirus cases. There is an option to specify the signal-to-noise ratio q_ζ, rather than estimate it, and an option to specify the model to have a seasonal component, using the `sea.type` option. The period of a seasonal component is specified through the `sea.period` option.

The `SSModelDynGompertzReinit` class allows the model to be estimated for a new wave without losing information from prior waves. It will accept the reintialization date specified by `reinit.date` or a `FilterResults` object from which it can extract the initial values. If the user wishes to reinitialize the model without using prior information (i.e. treat the new wave as an entirely separate epidemic), a reinitialization date can be specified through the `renit.date` option and `use.presample.info` can be set to `FALSE`.

The `FilterResults` class contains prediction methods which can be applied to estimated dynamic Gompertz curve models (both reinitialized and non-reinitialized). `get_growth_y` will return filtered or smoothed estimates of the growth rate of new cases (g_t), while `get_gy_ci` will return the same with confidence intervals. Forecasts of the incidence variable (new cases, y_t) can be obtained with the `predict_level` call, and forecasts of all the states can be obtained with the `predict_all` call.

Several functions are available to generate plots of smoothed and filtered estimates and forecasts. `plot_forecast` will plot actual and realised values of $\ln(g_t)$. `plot_gy` and `plot_gy_ci` can be used to plot the smoothed or filtered growth rate, its components, and confidence intervals, respectively. Forecasts of the incidence variable (y_t) and forecast intervals can be plotted using `plot_new_cases`, while `plot_holdout` adds plots of prediction intervals and of realized outcomes over a holdout period to help evaluate forecast accuracy. Finally, the `reinitialise_dataframe` function can be used to reinitialise a dataframe at a given `reinit.date`.

More details on how to use the methods and functions described are presented in the following section.

5. Illustration of the tsgc package

In this section we provide a full working example of the **tsgc** package in R which implements the modeling framework for time series growth curves-based epidemic forecasting.

tsgc comes with two example data sets relating to COVID-19: one for Gauteng province in South Africa (sourced from https://sacoronavirus.co.za/, South Africa's official coronavirus online news and information portal) and another for England (sourced from the official UK government dashboard for data and insights on coronavirus, https://coronavirus.data.gov.uk/). In the example that follows, we use the data on confirmed cases in Gauteng. The data series is in cumulative form and is loaded as an `xts` object with a date index, as follows.

```
data(gauteng, package = "tsgc")
```

New COVID-19 cases reported for Gauteng province and their centered 7-day moving average presented in Figure 1 show a sequence of four waves over the period between 10 March 2020 and 5 January 2022.

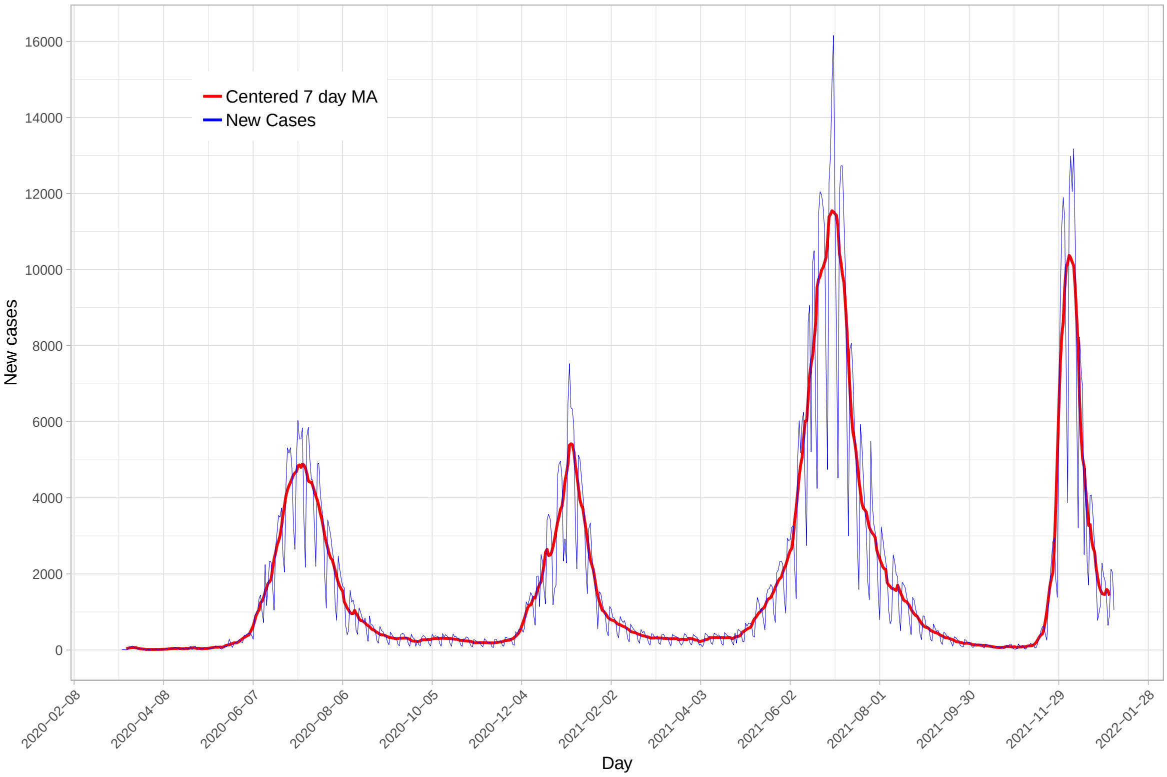

Figure 1: New Cases and their centered 7-day moving average for Gauteng province in South Africa between 10 March 2020 and 5 January 2022.

5.1. Setting up the forecasting exercise

We begin by specifying a number of options for the forecasting exercise, as defined below.

- `Y` is the data, in the form of time series of cumulative confirmed cases. In this example the object holding this series is called `gauteng`.
- `estimation.date.start` is the date of the first observation in the sample to be used for estimating the model. By default, it is the first date in the `xts` object `Y`.
- `estimation.date.end` is the date of the last observation in the sample to be used for estimating the model. By default, it is the last date in the `xts` object `Y`.
- `n.forecasts` is the number of days or periods for which forecasts are to be made. E.g., if `n.forecasts = 14`, forecasts will be generated for up to 14 days following `estimation.date.end`.
- `q` is the signal-to-noise ratio, which controls the smoothness of the estimated trend. A lower value will lead to more smoothness. By default, we use `q = 0.005`, which in

our experience ensures a good balance between the smoothness of the trend and the speed with which changes in estimates respond to new observations. Alongside, `q` can be estimated and compared with the default value.

- `confidence.level` sets the coverage of the confidence intervals for $\ln(g_t)$ which is then used to generate the prediction intervals for forecasts. Here, we use 0.68, corresponding to the probability that the forecast lies within one standard deviation of the point forecast.
- `plt.length` sets a truncation date to enhance the clarity of plots, e.g. showing only the last 30 days of the estimation sample. The date range for plotting can be set as `plt.length` days upto `estimation.date.end`.

In this example the data is the cumulative confirmed cases time series for Gauteng. The start and end dates (`estimation.date.start` and `estimation.date.end`) that define the sample used for estimation are chosen as appropriate for the exercise. We begin with the sample period set from 1 February to 19 April 2021. This marks the beginning of the third wave in Gauteng as can be seen in Figure 1. The options are specified as below.

```
file.path <- here()
res.dir <- here::here(file.path, 'results')
date.format <- "%Y-%m-%d"
Y <- gauteng
estimation.date.start <- as.Date("2021-02-01")
estimation.date.end <- as.Date("2021-04-19")
n.forecasts <- 14
q <- 0.005
confidence.level <- 0.68
plt.length <- 30
```

5.2. Estimation

We begin by selecting the data series (`Y`) for the defined sample period.

```
idx.est <-
    (zoo::index(Y) >= estimation.date.start) &
    (zoo::index(Y) <= estimation.date.end)
y <- Y[idx.est]
```

We then estimate the model using a diffuse prior distribution for the initial state vector. The signal-to-noise ratio can be left as a free parameter to be estimated, as in the code below.

```
model_q <- SSModelDynamicGompertz$new(Y = y)
res_q <- model_q$estimate()
```

In the rest of this example we estimate the model setting the signal-to-noise ratio at 0.005. As mentioned, in our experience this value strikes a useful balance between the smoothness of the estimate of the slope parameter γ, and the speed with which it adapts to new observations.

```
model <- SSModelDynamicGompertz$new(Y = y, q = q)
res <- model$estimate()
```

5.3. Results

We can now plot the forecast of $\ln g_t$ – the log of the growth rate of Y, the cumulative cases – which is the transformation of the data series that is taken to the model, and we can compare these forecasts to the actual $\ln g_t$ series. We do this by passing the output (`res`) of the estimation step along with an evaluation sample to a plotting function. We specify the evaluation sample by converting the cumulative cases series to the log of the growth rate of the cumulative cases.

First, we create the evaluation sample.

```
y.eval <- Y %>%
subset(index(.) > tail(res$index,1)) %>%
tsgc::df2ldl()
```

`tsgc::plot_forecast` then creates and plots forecasts of $\ln(g_t)$.

```
tsgc::plot_forecast(
 res<-res,
 y.eval <- y.eval, n.ahead <- n.forecasts,
 plt.start.date <-  tail(res$index, 1) - plt.length
)
```

From these results we can recover the forecasts of new cases from 20 April 2021, with their prediction intervals.

```
tsgc::plot_new_cases(
    res <- res, Y <- Y,
    n.ahead <- n.forecasts,
    confidence.level <- confidence.level,
    date_format <- "%Y-%m-%d",
    plt.start.date <- tail(res$index, 1) - plt.length
  )
```

To assess accuracy, we plot these forecasts against the actual new cases, that have been held back from the estimation sample, using the `plot_holdout` function. The model forecasts are compared with the the first differences of `Y.eval`, the cumulative series for the forecast window.

```
tsgc::plot_holdout(res, Y<-Y,
                   Y.eval <- Y[(tail(res$index,1)+0:n.forecasts)],
                   confidence.level <- 0.68,
                   date_format <- "%Y-%m-%d")
```

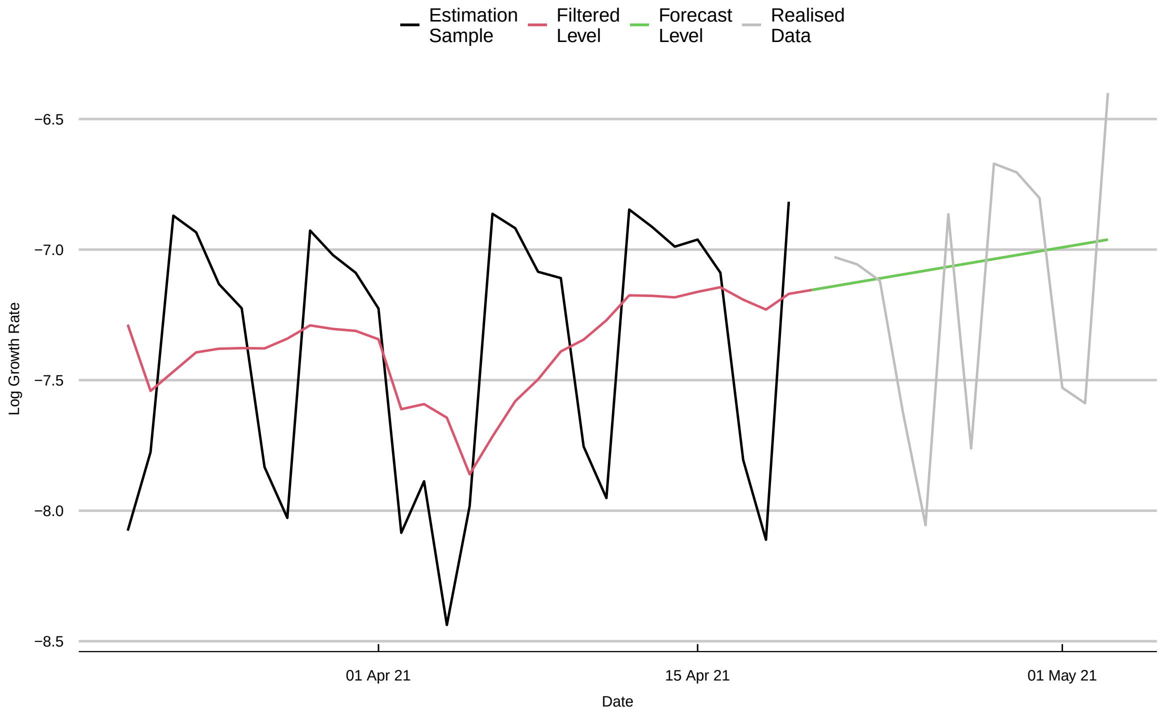

Figure 2: Fourteen-day forecast of $\ln(g_t)$ from 20 April 2021 for Gauteng province in South Africa.

Figure 4 shows that the forecasts were accurate over the first seven days, with a mean absolute percentage error (MAPE) of 13.9%. Note that reported cases were unusually low on 27 April due to the fact that it is Freedom Day and a public holiday in South Africa. That day aside, over the six days from 28 April the MAPE was 12.8%. Over the full 14 days of the forecasts, the MAPE was 27%.

As discussed in Section 2.5, the reproduction numbers R_t and their 68% credible intervals can be calculated. The plot in Figure 5 reveals that R_t remains above one through the period, indicating that the (third) wave in Gauteng had launched by this time.

```
r.t <- tail(exp(res$get_gy_ci()*gen_int),7)
```

A CSV file (named `y-forecast`) is written to the directory specified. The forecast options specified earlier are retained.

```
tsgc::write_results(
 res<-res,
 res.dir <- res.dir,
 Y<-Y,
 n.ahead <- n.forecasts,
```

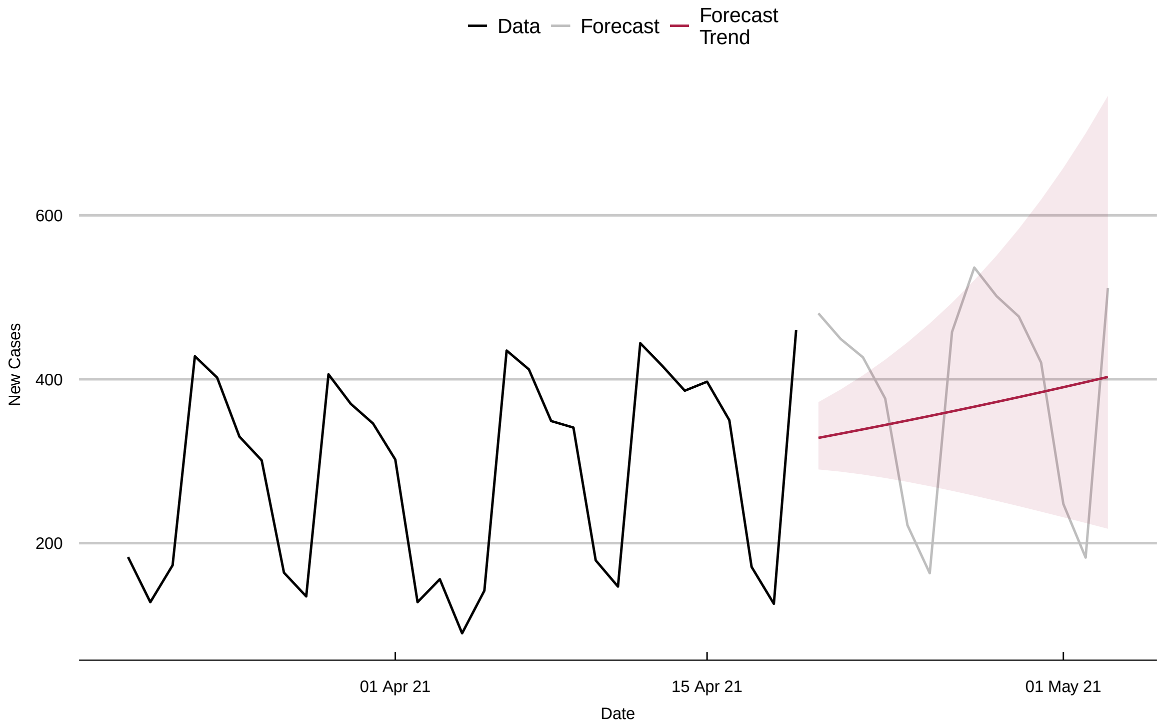

Figure 3: Fourteen-day forecast of new cases from 20 April 2021 for Gauteng province in South Africa.

```
 confidence.level <-  confidence.level
)
```

5.4. Reinitialization

In all countries the coronavirus pandemic was characterized by a series of recurring waves due to a combination of biological, behavioral, and environmental reasons. In an epidemic, the onset of a new wave is signalled when the slope parameter γ, which measures of the growth rate of the growth rate of new cases, rises above zero for a sustained period. Such a super-exponential phase of the epidemic in which the growth rate of new cases is itself increasing over time is typically short.

This section illustrates the reinitialization procedure which allows us to apply the model to the new wave as it begins, without jettisoning information from the wave that has just ended. We extend the estimation window to 25 June 2021, by which date the third wave is well on course with its peak within sight (see Figure 1). All other options remain the same.

```
estimation.date.end <- as.Date("2021-06-25")
```

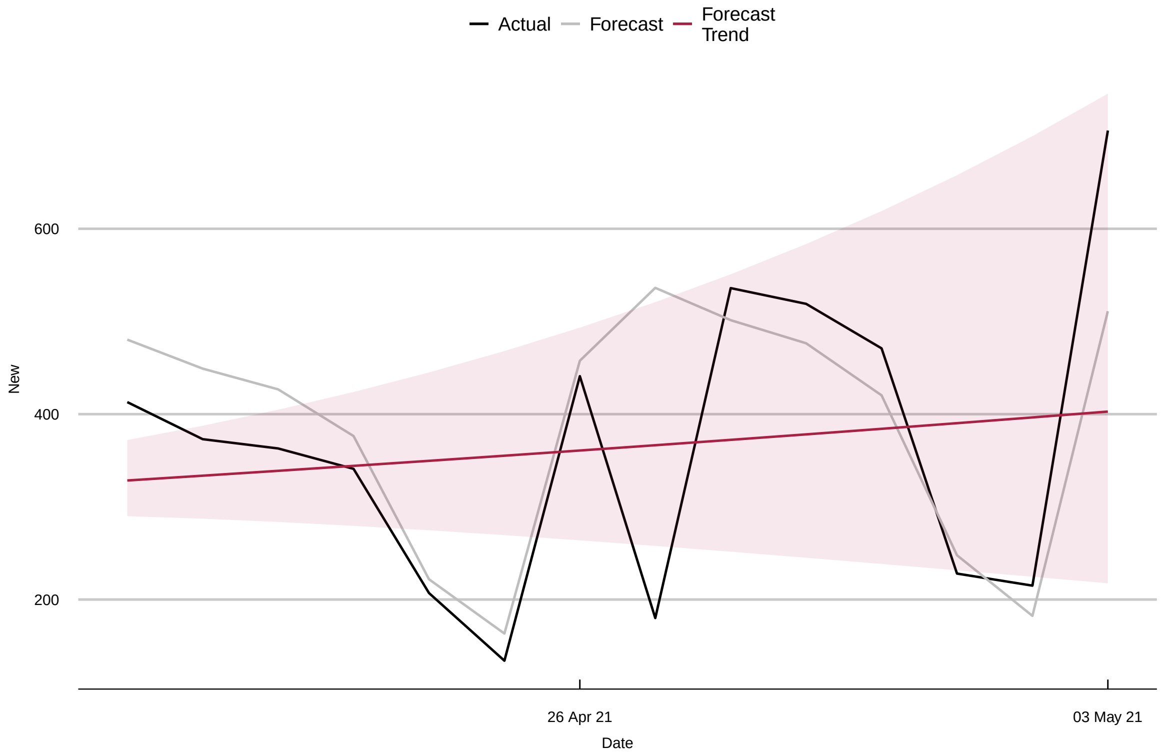

Figure 4: Accuracy of the fourteen-day forecast of new cases from 20 April 2021, for Gauteng province in South Africa.

Triggering reinitialization It is not obvious, a priori, when precisely to reinitialize the model. Based on experiments a reasonable option is to reinitialize when the estimate of the slope parameter, γ_t, breaches a threshold of two standard errors above zero, and at that point backdate reinitialization to when the estimate of γ_t first turned positive. In applying the above heuristic there is a choice between the filtered slope estimate and the smoothed slope estimate. Experiments suggests that the greater noisiness of the filtered estimate of γ_t often triggers reinitialization too early. The smoothed estimate is more reliable.

Figure 6 shows that for the third wave in Gauteng, the smoothed slope estimate exceeded twice its standard error on 1 May 2021, having risen above zero on 21 April 2021.

The date for reinitialization is set accordingly.

```
reinit.dates <- "2021-04-21"
```

Estimating the reinitialized model `SSModelDynGompertzReinit` takes the same arguments as `SSModelDynGompertz`, with the addition of the `reinit.dates` argument.

```
model <- SSModelDynGompertzReinit$new(
  Y <- y, q <- q,
  reinit.date <- as.Date(reinit.dates,
```

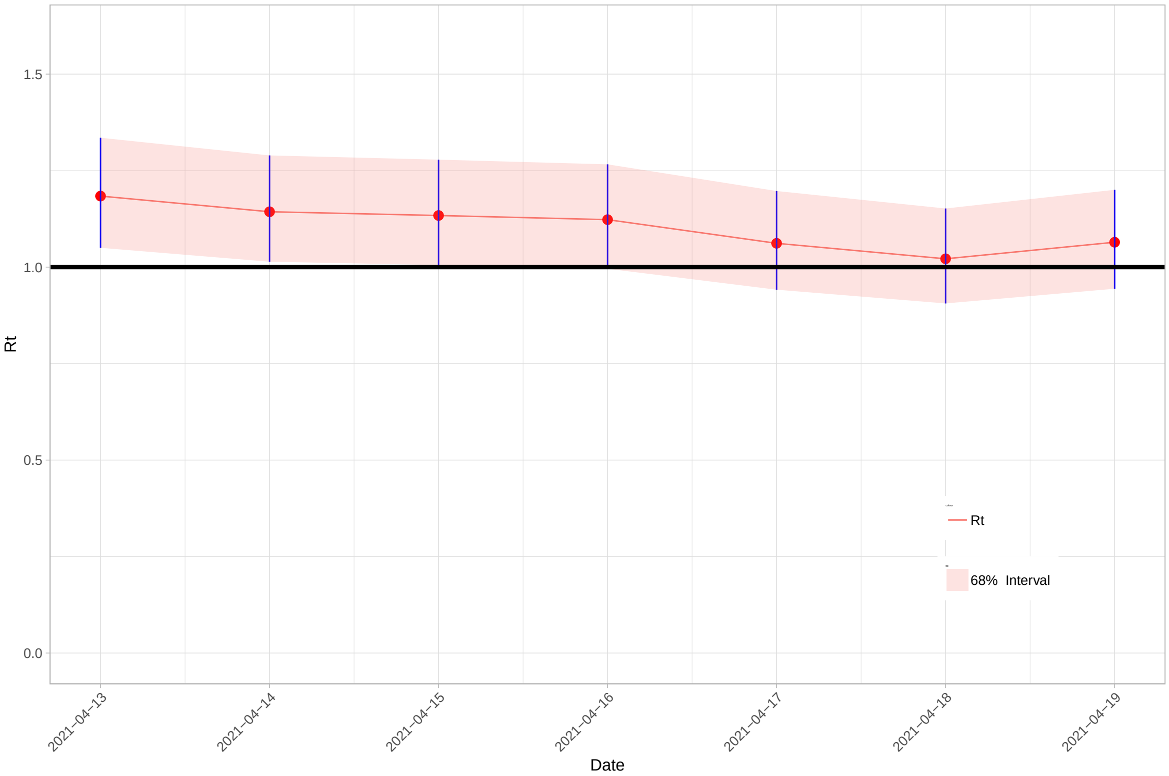

Figure 5: Reproduction numbers for the 7-day period to 19 April 2021, for Gauteng province in South Africa.

```
  format <- date.format)
)
res.reinit <- model$estimate()
```

We generate the reinitialized data frame by setting cumulative cases to 0 at the appropriate point, as discussed in Section 3.1 and extract the evaluation sample from the reinitialized data frame as below.

```
y.eval.reinit <- Y %>%
  reinitialise_dataframe(., reinit.dates) %>%
  df2ldl() %>%
  subset(index(.) > tail(res.reinit$index,1))
```

Estimating the model with the reinitialized series, the actual and forecast $\ln(g_t)$ can be plotted as in Figure 7, which is analogous to Figure 2 for the non-reinitialized series.

```
tsgc::plot_forecast(
    res <- res.reinit,
    y.eval <- y.eval.reinit,
    n.ahead <- n.forecasts,
```

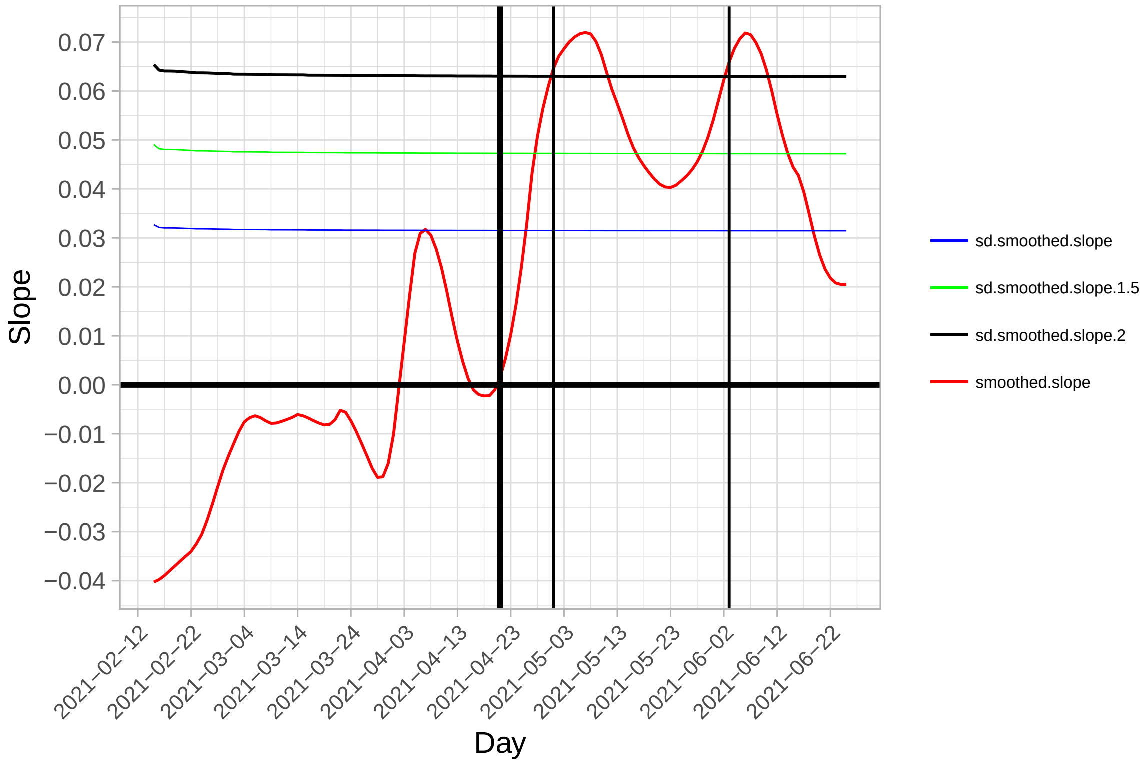

Figure 6: Trigger for reinitialization: when filtered slope $\hat{\gamma}_{t|t}$ exceeds twice its standard error $\hat{\sigma}_{\gamma,t|t}$ above zero, reinitialization is triggered to the date when $\hat{\gamma}_{t|t}$ crossed zero.

```
    plt.start.date <-  tail(res.reinit$index, 1) - plt.length
)
```

The plot of the forecasts of new cases, and that of these forecasts against the actual number of new cases can be produced as before. See Figures 8 and 9. The trend has begun to turn down in model forecasts with with the reinitialized series.

```
tsgc::plot_new_cases(
    res.reinit, Y <- Y.reinit,
    n.ahead <- n.forecasts,
    confidence.level <- confidence.level,
    date_format <- "%Y-%m-%d",
    plt.start.date <- tail(res.reinit$index, 1) - plt.length
  )


tsgc::plot_holdout(
    res <- res.reinit,
    Y <- Y.reinit[index(y)],
    Y.eval <- Y[(tail(res.reinit$index,1)+0:n.forecasts)],
```

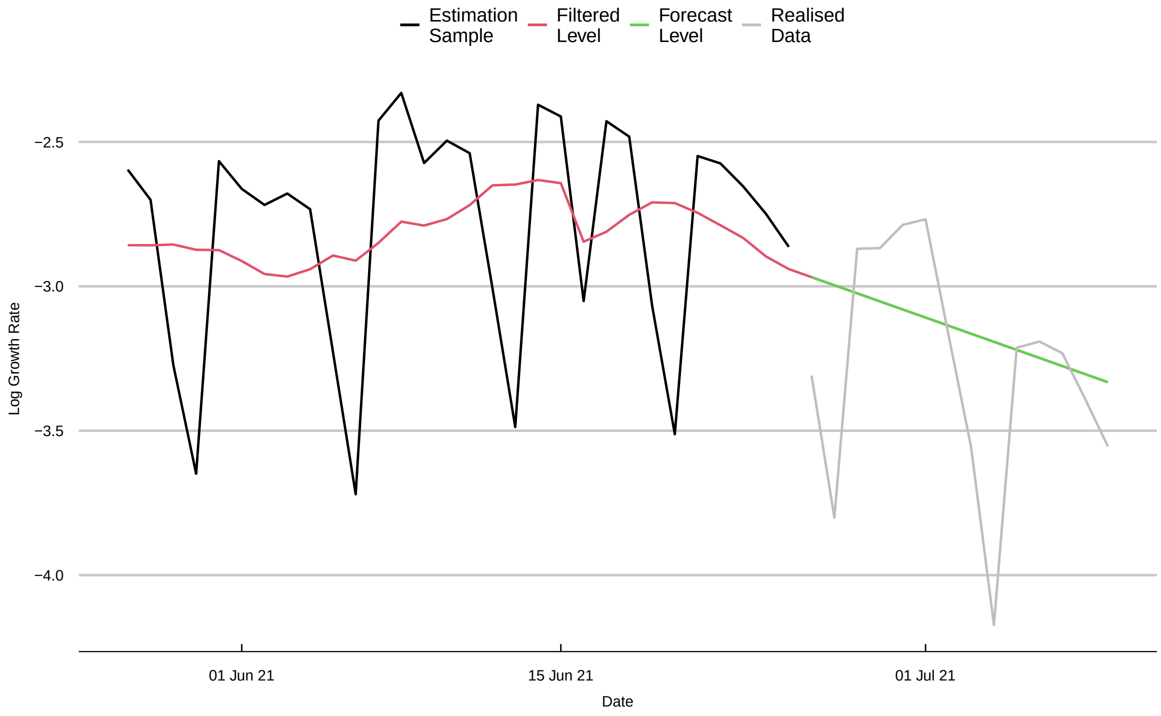

Figure 7: Forecast of $\ln(g_t)$ after reinitialization.

```
    confidence.level <- 0.68,
    date_format <- "%Y-%m-%d")
```

Comparing forecasts, without reinitialization the 14-day forecast MAPE is 41.9% (see Figure 10). With reinitialization, it falls to 20.2%. The corresponding MAPE figures for 7-day forecasts is 15.2% without reinitialization and 9.5% with reinitialization.

Just as for the standard (non-reinitialized) model, the returned estimation results contained in `res.reinit` are a `FilterResults` object, and can be written to CSV using the `write_results` method.

6. Conclusions

The **tsgc** package is based on a dynamic Gompertz curve model for the log of the growth rate of cumulative cases in an epidemic, with seasonal terms that capture day-of-the-week effects. The estimation is carried out using the **KFAS**, a package for state-space modeling in R.

The Kalman filter is used to estimate the state vector at each time point. The filter is initialized using a diffuse prior for the initial state vector. We allow the option for the signal-to-noise ratio to either be estimated or fixed at some value based on experience and judgment. Future observations are forecast using predictive recursions.

Epidemics are often characterized by multiple waves. A natural problem in this context is

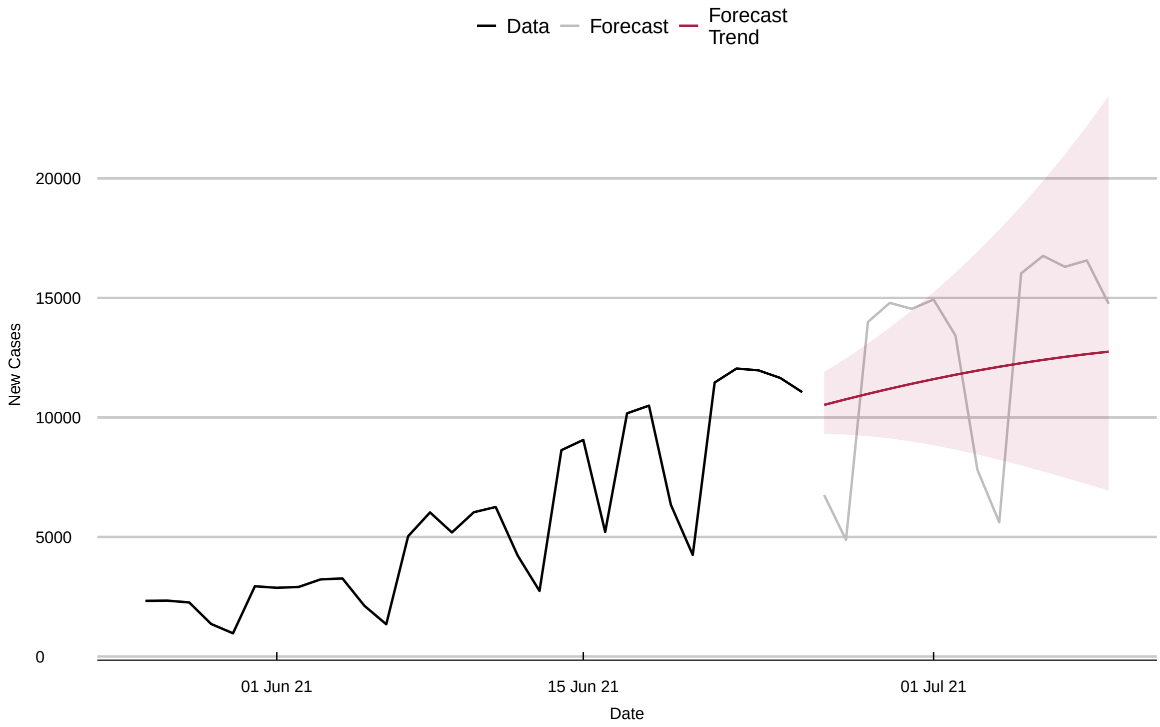

Figure 8: Forecast of new cases after reinitialization.

that there is very little data pertinent to the new wave available in its initial stages. The package employs a reinitialization method using priors in a way that allows data from before the beginning of the new wave to be used in estimation. The package also includes several functions for generating plots and forecasts.

The package is demonstrated using COVID-19 data from South Africa, but it can be used to model and forecast any time series variable where a growth curve-like trajectory is expected. Examples might include sales of a new product, innovation adoption or website traffic.

Acknowledgements

We thank The Cambridge Centre for Health Leadership & Enterprise, University of Cambridge Judge Business School, and Public Health England/UK Health Security Agency for generous support. We are indebted to Thilo Klein and Stefan Scholtes for constructive comments. Andrew Harvey's work was carried out as part of the University of Cambridge Keynes Fund project 'Forecasting and Policy in Environmental Econometrics'.

References

Abbott S, Monticone P (2021). "epiforecasts/epinowcast: Evaluation in Germany Initial

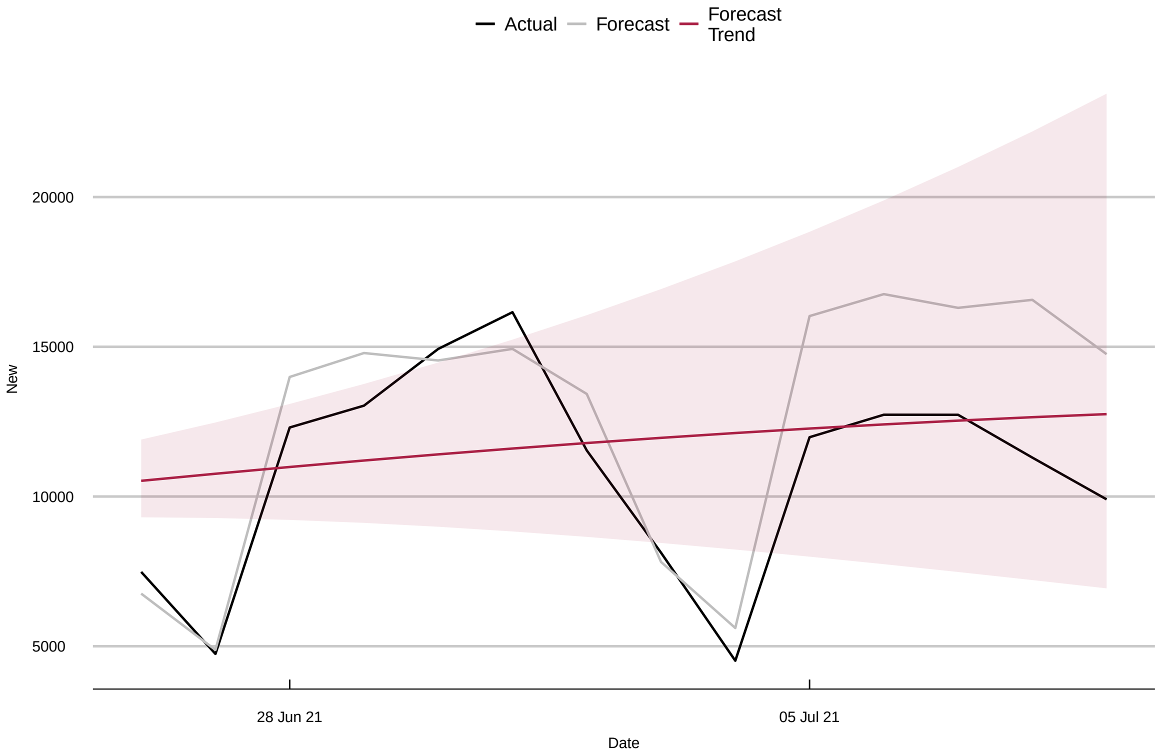

Figure 9: Forecast accuracy of the reinitialized model over the hold-out sample period: 14 days past 25 June 2021.

release.” doi:10.5281/zenodo.5637165.

Cori A, Ferguson NM, Fraser C, Cauchemez S (2013). “A New Framework and Software to Estimate Time-Varying Reproduction Numbers During Epidemics.” *American Journal of Epidemiology*, **178**(9), 1505–1512. ISSN 0002-9262. doi:10.1093/aje/kwt133.

Durbin J, Koopman SJ (2012). *Time series analysis by state space methods*. Oxford University Press.

Harvey A, Kattuman P (2020). “Time series models based on growth curves with applications to forecasting coronavirus.” *Harvard Data Science Review*. URL https://doi.org/10.1162/99608f92.828f40de.

Harvey A, Kattuman P (2021). “A farewell to R: time-series models for tracking and forecasting epidemics.” *Journal of the Royal Society Interface*, **18**. URL http://doi.org/10.1098/rsif.2021.0179.

Harvey A, Kattuman P, Thamotheram C (2021). “Tracking the mutant: forecasting and nowcasting COVID-19 in the UK in 2021.” *National Institute Economic Review*, **256**(1), 110–126. URL http://doi:10.1017/nie.2021.12.

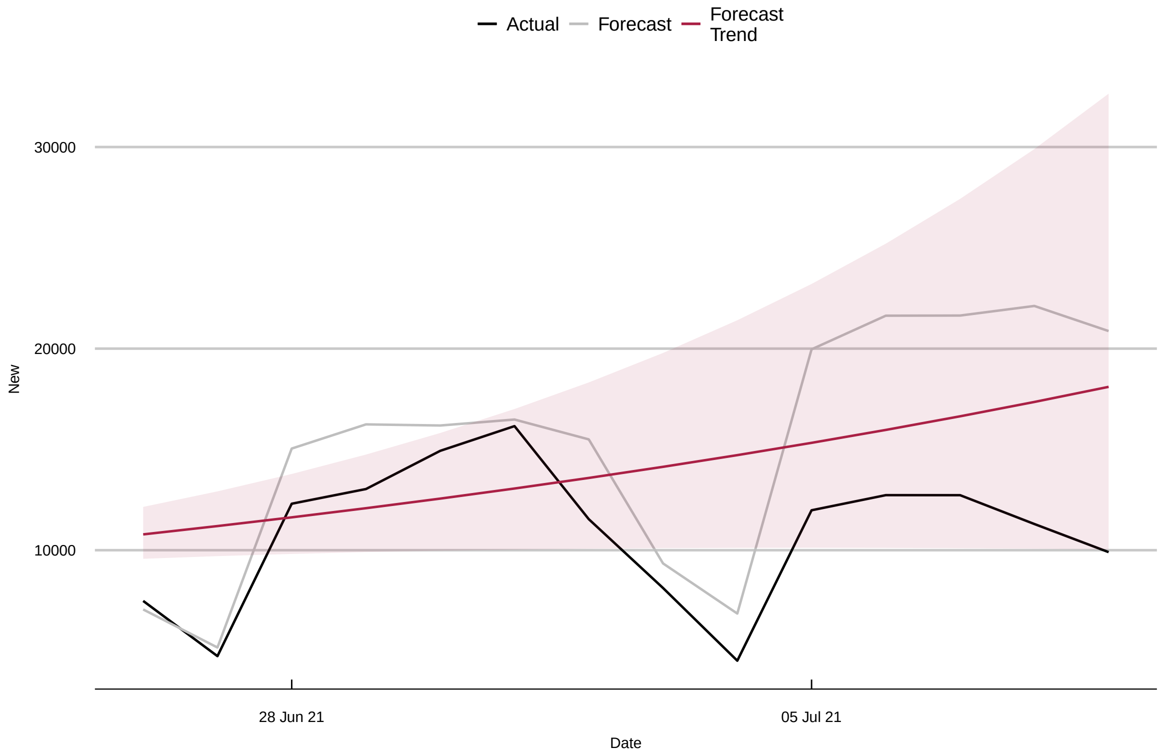

Figure 10: Forecast accuracy of the model without reinitialization over the hold-out sample period: 14 days from 25 June 2021.

Helske J (2017). "KFAS: Exponential Family State Space Models in R." *Journal of Statistical Software*, **78**(10). URL `https://doi.org/10.18637/jss.v078.i10`.

Jenness SM, Goodreau SM, Morris M (2018). "EpiModel: an R package for mathematical modeling of infectious disease over networks." *Journal of Statistical Software*, **84**(8), 1–47. URL `https://doi.org/10.18637/jss.v084.i08`.

Proietti T (2000). "Comparing seasonal components for structural time series models." *International Journal of Forecasting*, **16**(2), 247–260. URL `https://doi.org/10.1016/S0169-2070(00)00037-6`.

Wallinga J, Lipsitch M (2007). "How Generation Intervals Shape the Relationship Between Growth Rates and Reproductive Numbers." *Proceedings of the Royal Society B*, **274**, 599–604.

Appendix

Incorporating seasonal terms into the state space model

When we add a seasonal term to the model, the observation equation in the dynamic Gompertz curve (2) becomes

$$\ln g_t = \delta_t + \nu_t + \varepsilon_t, \ \varepsilon_t \sim NID(0, \sigma_\varepsilon^2), \ t = s, ..., T,$$

where ν_t is the seasonal component, δ_t remains defined by (3), and ε_t remains the iid Normal disturbance.

There are two options for specifying the evolution of the seasonal component. We can either use a trigonometric seasonal or a dummy variable seasonal. In our application, we use a trigonometric seasonal, although the two specifications are closely related. Indeed, Proietti (2000) shows that, under certain conditions, the two approaches are equivalent.

In the trigonometric seasonal approach, the seasonal pattern is captured by a set of trigonometric terms at the seasonal frequencies $\lambda_j = \frac{2\pi j}{2}$ for $j = 1, \ldots s^*$, where $s^* = \frac{s}{2}$ if s, the periodicity of the seasonal effect, is even, and $s^* = \frac{s-1}{2}$ if s is odd (Durbin and Koopman 2012). Our applications use daily data and we set $s = 7$ to capture day-of-the-week effects.

Letting $\nu_{j,t}$ be the effect of season j at time t, the seasonal terms evolve according to

$$\nu_t = \sum_{j=1}^{s^*} \nu_{j,t}, \tag{13}$$

where

$$\nu_{j,t} = \nu_{j,t-1} \cos \lambda_j + \nu^*_{j,t-1} \sin \lambda_j + \omega_{j,t} \tag{14}$$

$$\nu^*_{j,t} = -\nu_{j,t-1} \sin \lambda_j + \nu^*_{j,t-1} \cos \lambda_j + \omega^*_{j,t}, \qquad j = 1, \ldots, s^*, \tag{15}$$

and $\omega_{j,t}$ and $\omega^*_{j,t}$ are mutually independent, iid $N(0, \sigma_\omega^2)$ variables.

When reinitializing the model with seasonal terms, P_1^r is a block-diagonal matrix based on P_{r+1} which sets the covariances between $(\delta_t, \gamma_t)'$ and $(\nu_{1,t}, \nu_{2,t}, \ldots, \nu_{s^*,t})'$ to be zero. The covariance between δ_t and γ_t, as well as the covariances between $\nu_{1,t}, \nu_{2,t}, \ldots, \nu_{s^*,t}$, are permitted to be non-zero and come directly from P_{r+1}.