Epidemic Timeseries - Thesis

Matthew So

October 2020

Contents

1 Relevant resources

- Meeting planner
- Overall thesis GitHub repo

- Shifts folder
- Folder for phase-plane analysis, mobility PCA, and clustering

2 Mobility PCA

2.1 Explanation of mobility PCA

Google Mobility data contains detailed daily mobility data for each country. We wanted to use this data to make a single mobility index that was applicable to all countries.

2.2 Current approach

Google Mobility tracks mobility in terms of a percent change from baseline in retail and recreation (RR), grocery and pharmacy (G), transit stations (T), workplaces (W), residential (R), and parks (P).

Mobility data for all listed metrics except P smoothed with a 7-day window (one reason was to remove weekend effects; note that performing no smoothing has similar results). Data was taken for all countries with ≥ 200 datapoints and naively accumulated. Then, the PCA algorithm from scikit-learn was fitted to this data.

The explained variance of principal component (PC) 1 is 0.894, and the explained variance of PC 2 is 0.041. The components of PC 1 were (last run): RR 0.57781228 G 0.40459858 T 0.53877504 W 0.41191798 R -0.20610189, and the components of PC 2 are: RR 0.26836183 G 0.66819787 T -0.35537018 W -0.59375593 P -0.05156992.

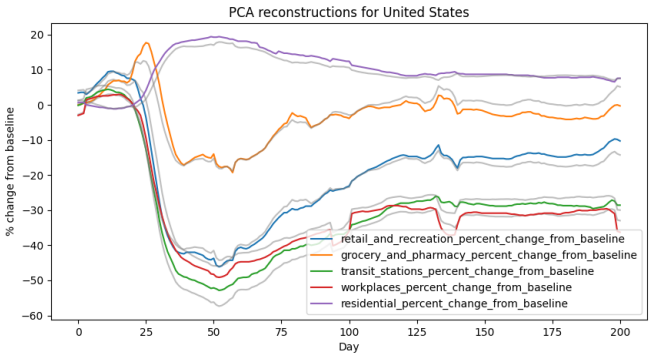

Figure 1: A PCA reconstruction (grey lines) of mobility data for United States.

2.3 Concerns

No major concerns at this moment.

Just for documentation purposes: In the past, MS had concerns about a seemingly poor reconstruction accuracy in the latter half of the pandemic. After actually plotting the inverse-transformed PCA-transformed data, I no longer believe this is an issue.

3 Clustering

3.1 Explanation of clustering

Epidemic timeseries (such as incidence, cumulative case, etc.) curves are qualitatively different. For example, the U.S. incidence timeseries is obviously different from the Canadian incidence timeseries. I would like to algorithmically cluster "similar" timeseries, and hopefully this will give some deeper insight about pandemic responses in different countries.

3.2 Current approach

I intend on trying out several different timeseries distance metrics to determine the pairwise distances between each timeseries. I've currently tried the discrete time warping-Euclidian distance, but other metrics have been reported. In terms of the actual clustering, I need to use an algorithm that can cluster points based on pairwise distance. From some of my experimentation, I thought the DBSCAN algorithm gave some good results. This algorithm seeds a new cluster if it has at least n points that are within ϵ units from it. Then, if another point is within ϵ units from a cluster point, it is added to a cluster. If two clusters share a point, they are merged into a single cluster. I define the start of the pandemic as the first point in time at which there are ≥ 1 cumulative cases in a country, although I'm not sure of how much this matters (due to the use of DTW)

3.3 Concerns

I don't understand DTW yet.

4 Phase-planes and other plots

4.1 Explanation of plots

(For myself): Phase-planes plot two variables that are connected in time against each other. They're sort of similar to the vector field plots that pop up if you search up 'phase plane' in that it shows the evolution of two functions of an underlying variable.

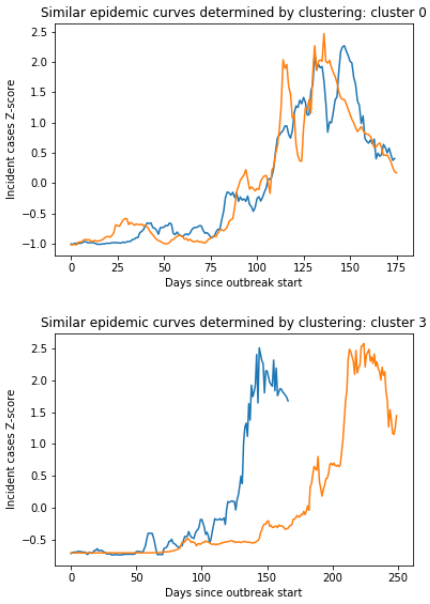

Figure 2: Similar incidence timeseries determined by clustering.

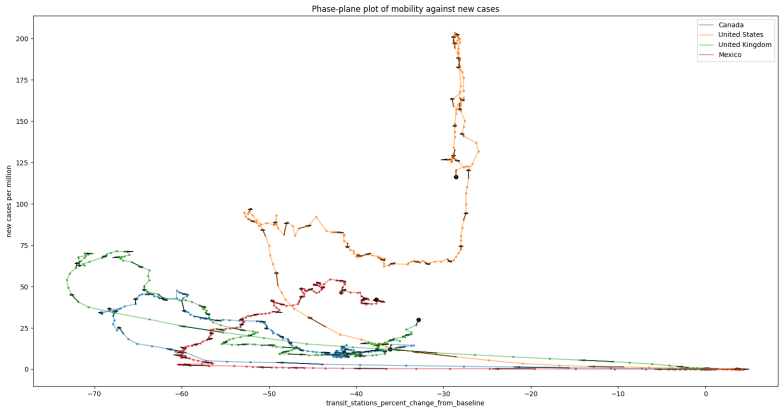

Figure 3: A phase plane of transit mobility against new cases for selected countries

4.2 Current approach

I scatter-plot the two variables under consideration, then connect each dot with lines and add arrows showing the direction of the motion. I also make heavier

dots at the end of each phase plane. The plots should be cleaned up a bit (for example, it may be worthwhile to smooth incidence curves).

4.3 Concerns

JD is colorblind, so MS should use a colorblind palette.

5 Convolution lag estimation

5.1 Explanation

An older idea looking at the feasibility of determining time lag distributions by determining convolution kernels that could possibly be used to transform one timeseries into another. For example, we could find the kernel that transforms the recorded symptomatic timeseries into the recovered timeseries. I know that you said that this is not a very promising research direction, but it was too tempting to not try this after shifts (as we KNOW the time lag distribution from E-¿I and can directly evaluate this convolution).

5.2 Current approach

At each step, I convolve the true incidence curve with a filter to attempt to get the new symptomatic cases (using scipy.optimize.minimize to minimize the mean squared error between true incidence curve and convolved symptomatic curve, with only valid convolutions used). I also use two constraints: filter[0] = 0, and I try to penalize non-smoothness by penalizing the size of the first differences of the filter. I could also force the kernel to correspond to some parametric distribution (gamma, lognormal, etc.) as I did before. I did not evaluate this idea yet, but it is simple to accomplish.

Since this method should be usable on any kind of time lag, I tried to use it on real data with the incidence/recovery curves. To reduce the fluctuations, I smoothed with a filter of 7 days.

5.3 Concerns

- How the distribution looks depends a lot on the non-smoothness penalty.
- If the original curves are not smoothed, the goodness of fit goes down drastically.

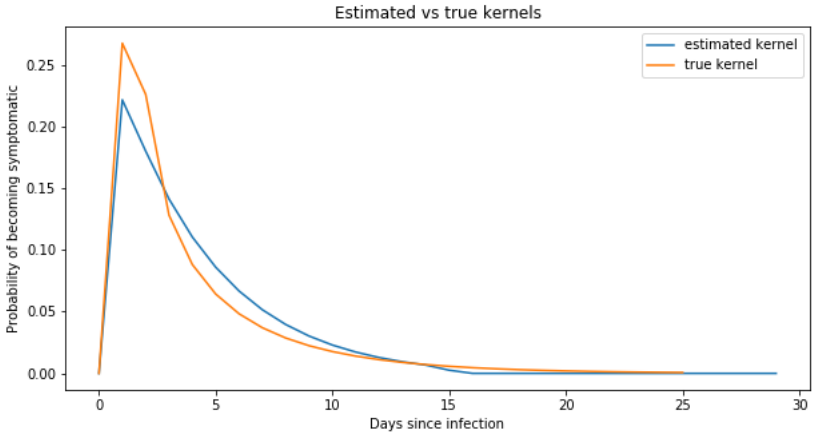

Figure 4: Estimated vs true kernel for E to I transition

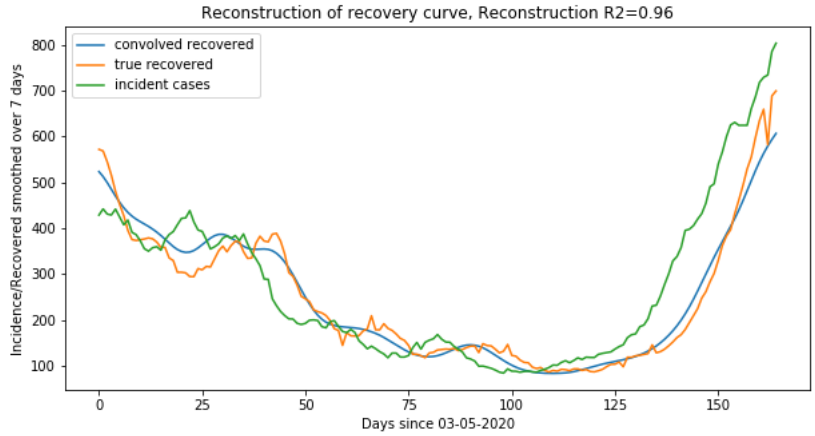

Figure 5: Reconstructed recovery curves for Ontario

6 Shifts

6.1 Explanation of shifts

An exploration of JD's idea found here. The idea is to investigate using the methods of Cori and Wallinga to estimate R(t). We will use deconvolution (currently Richardson-Lucy as used in the Gostic paper) to provide an estimate of the true incidence curve from the number of new symptomatic infections.

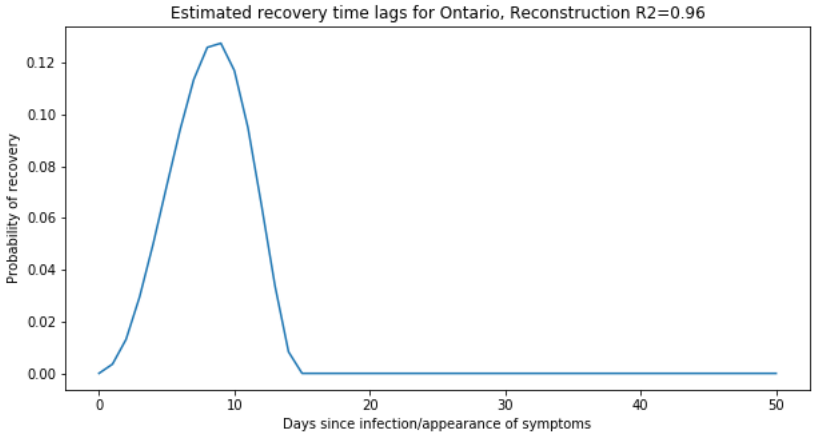

Figure 6: Estimated recovery distribution for Ontario

6.2 Current approach

I use code from the Gostic paper to perform Richardson-Lucy deconvolution in R. I use the EpiEstim library to generate R(t) estimates using the Cori and Wallinga-Tenuis methods.

Like the Gostic paper, I've used a vanilla SEIR model (We need to be able to replicate the results first).

$$\frac{dS}{dt} = -\beta(t)\frac{S}{N}I \tag{1}$$
$$\frac{dE}{dt} = \beta(t)\frac{S}{N}I - \gamma E \tag{2}$$
$$\frac{dI}{dt} = \gamma E - \mu I \tag{3}$$
$$\frac{dR}{dt} = \mu I \tag{4}$$

Therefore, these are equations representing the instantaneous reproductive number:

$$R_0 = \beta(t)/\mu \tag{5}$$
$$R(t) = \beta(t)/\mu \cdot S/N \tag{6}$$

This is how (I think) you can obtain the generation interval distribution analytically. I may be totally wrong, please help here. It is an addition of exponential distributions of $E \to I$ and $I \to R$, exploiting the fact that both the means and variances add up, as well as that two exponential distributions

multiplied by each other make a Gamma distribution. It also may not be super necessary now.

G: generation interval distribution
k: shape parameter of Gamma distribution
θ: scale parameter of Gamma distribution

$$Mean(E \to I) = 1/\gamma \tag{7}$$
$$Mean(I \to R) = 1/\mu \tag{8}$$
$$Var(E \to I) = 1/\gamma^2 \tag{9}$$
$$Var(I \to R) = 1/\mu^2 \tag{10}$$
$$Mean(G) = 1/\gamma + 1/\mu = k\theta \tag{11}$$
$$Var(G) = 1/\gamma^2 + 1/\mu^2 = k\theta^2 \tag{12}$$
$$k = Mean(\mathrm{G})^2/Var(\mathrm{G}) \tag{13}$$
$$\theta = Var(\mathrm{G})/Mean(\mathrm{G}) \tag{14}$$

This equation represents the cohort reproductive number, which will be estimated with the Wallinga method.

$$R_{case}(t) = R(t) * G \tag{15}$$

6.3 Progress

From preliminary testing, the deconvolution method fits the true incidence quite well. However, further testing is required to see if it might work well in practice. It also works reasonably well with a changing R(t).

The Cori method to the deconvoluted incidence also seems to work well with the static β and slowly changing R(t). The Wallinga-Tenuis method seems to fit the slowly changing cohort R(t), but is incredibly slow.

6.4 Concerns

- The Cori estimates are no longer wrong, but the Wallinga-Tenuis ones are still slightly wrong.
- I don't understand most of the math here. How was the R(t) calculated for the SEIR/other models? How does the RL deconvolution method work? How do the Cori and Wallinga estimation methods work?

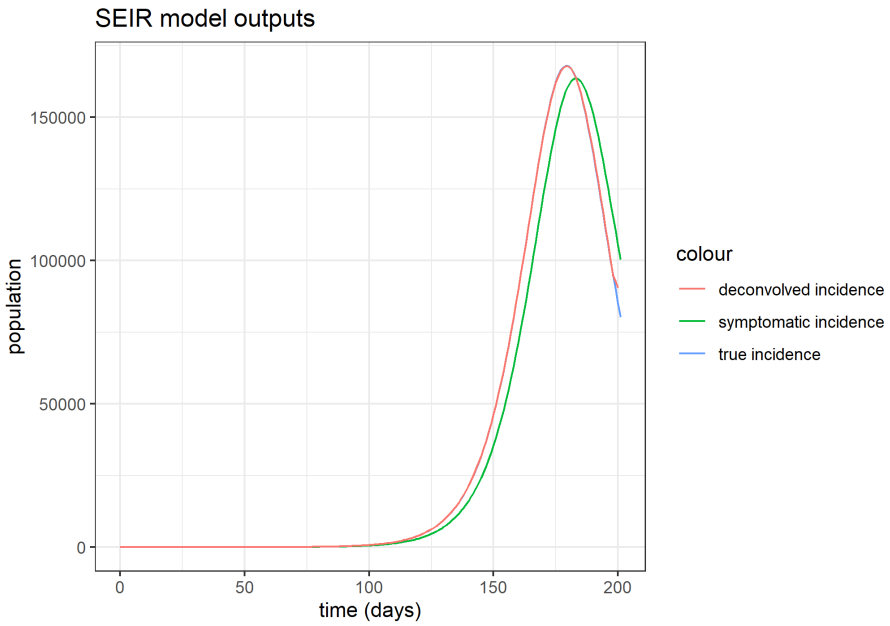

Figure 7: Deconvolved incidence, true incidence, and symptom onset curves

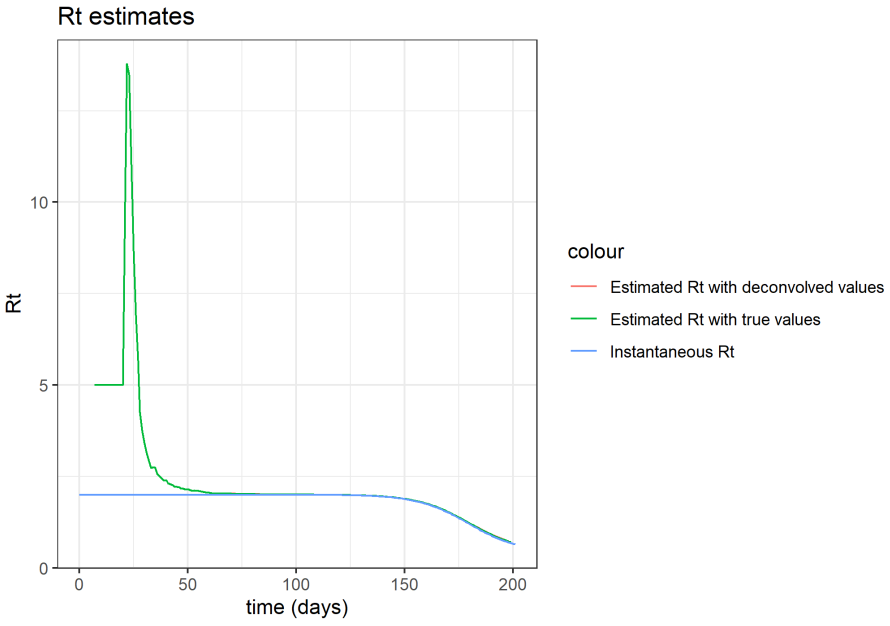

Figure 8: Instantaneous R(t) estimated with the method of Cori.

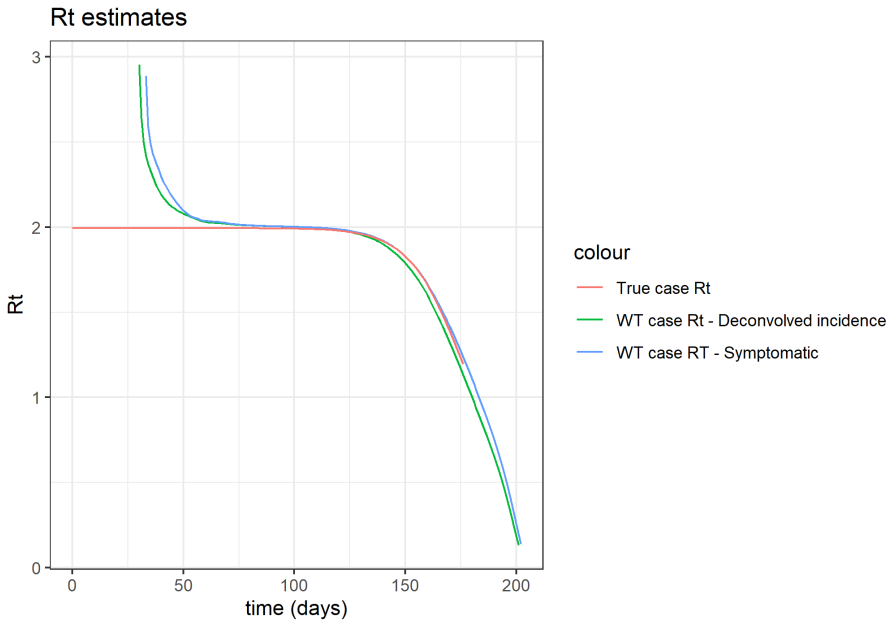

Figure 9: Cohort R(t) estimated with the method of Wallinga and Tenuis.

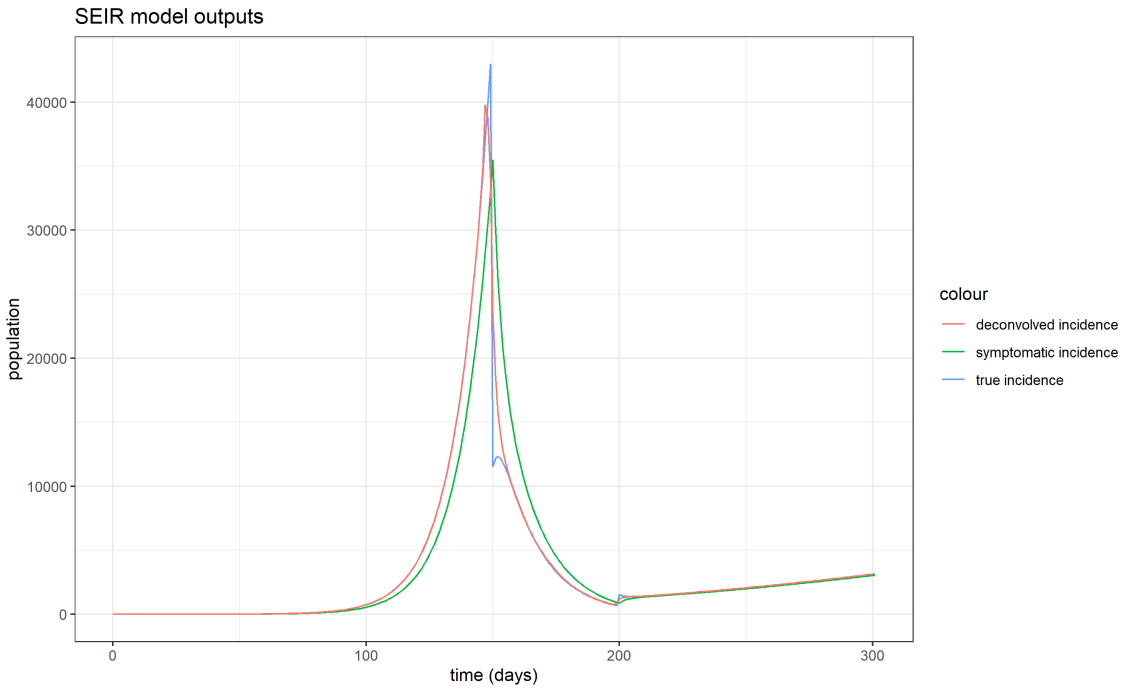

Figure 10: Deconvolution of incidence works reasonably well even with changing R(t).

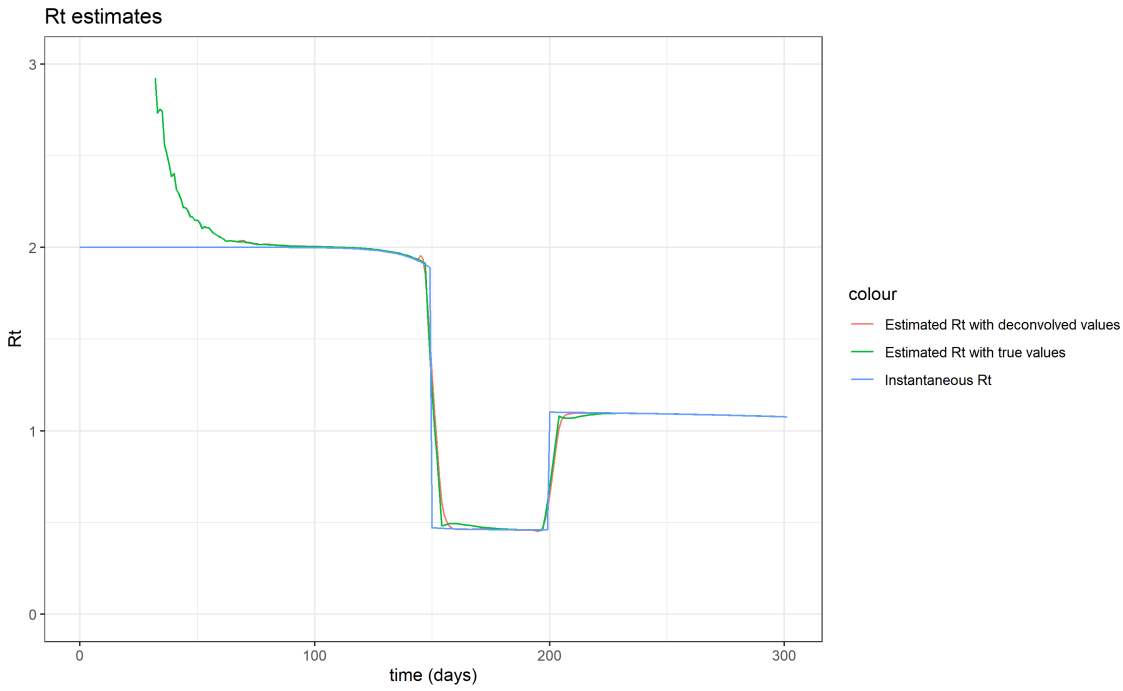

Figure 11: Cori estimates track R(t) well after the initial 50 days even with changing R(t).

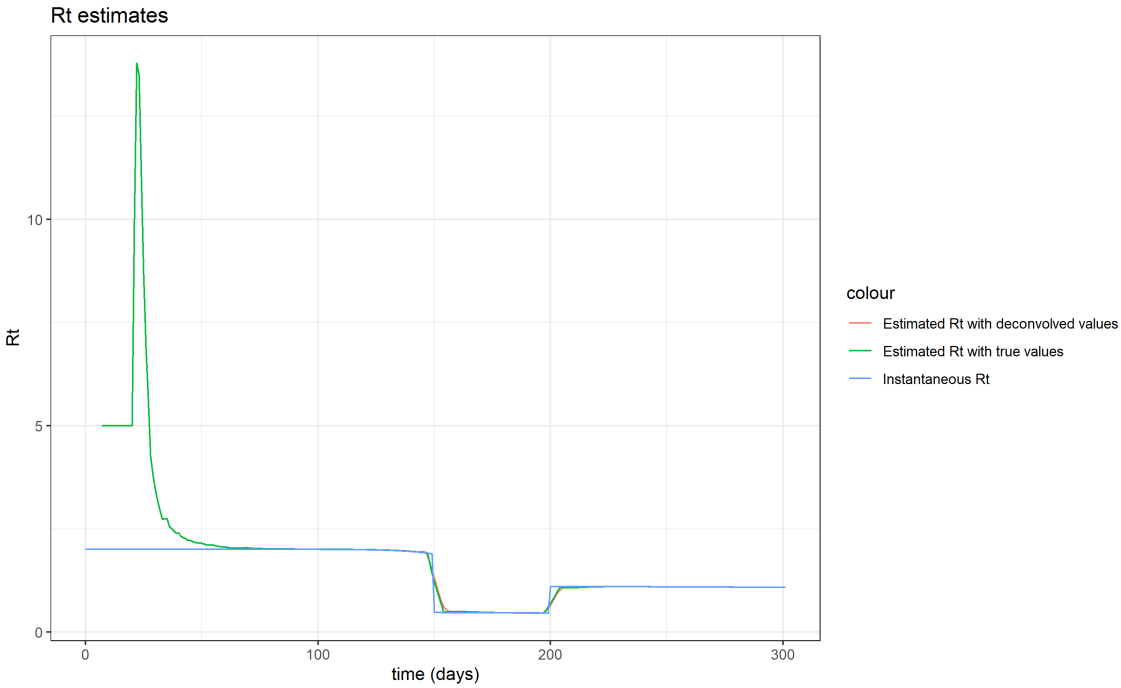

Figure 12: WT estimates of R(t) when R(t) is changing.