JBF $R_e(t)$ for NL

Jan van Rongen

2021-07-25

Explanation

JBF emulates the RIVM data quite well, Can it do better? And can we clean up the messy code and make a proper error estimate?

Load RIVM Data

```
rivm_ts <- make_cases(repro_rivm[, c(1,3)])
FROM= as.Date("2020-08-01"); TO= as.Date("2021-06-25")
rivm_ts<- rivm_ts[FROM <= rivm_ts$date & rivm_ts$date < TO,]
cases_all<- make_all_cases()
```

Try other methods

As an example we use the Golay filter.

```
for (m in 8+(1:6))  {
  JBF_Rt <- Rt_JBF(cases_all, SMOOTH_DATA=m, SMOOTH_Method="Golay")
  JBF_ts <- make_cases( JBF_Rt[, 1:2])
  JBF_ts<- JBF_ts[FROM <= JBF_ts$date & JBF_ts$date < TO,]
  cat("\nResemblance: ", m, rev_rmse(JBF_ts$cases, rivm_ts$cases), "%")
}
```

```
##
## Attaching package: 'signal'

## The following object is masked from 'package:dplyr':
##
##     filter

## The following objects are masked from 'package:stats':
##
##     filter, poly

##
## Resemblance:  9 93.64 %
## Resemblance:  10 94.77 %
```

```
## Resemblance:  11 95.47 %
## Resemblance:  12 95.64 %
## Resemblance:  13 95.33 %
## Resemblance:  14 94.61 %
```

We have tried various other filters, but the Gaussian remains the closest with 98.51% and a span of 11, followed by the moving average centered with a span of 7 days and a score of 95.6%. Third and very close is the Golay filter with a span again of 11 days and a score of 95.5%.

improving the error estimates

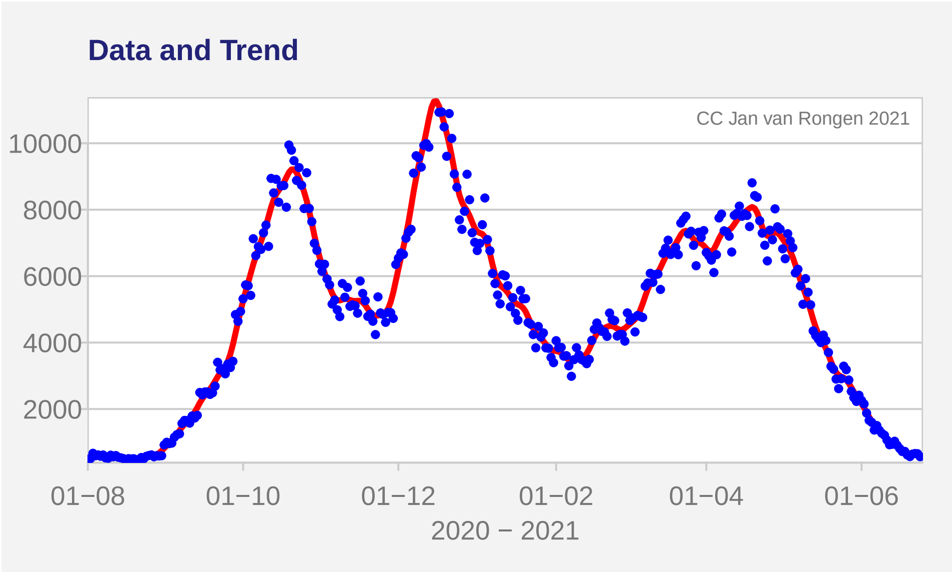
The initial code of JBF was just a messy hack. It worked, Okay? But we can do better.

Uncertainties after a kernel approximation

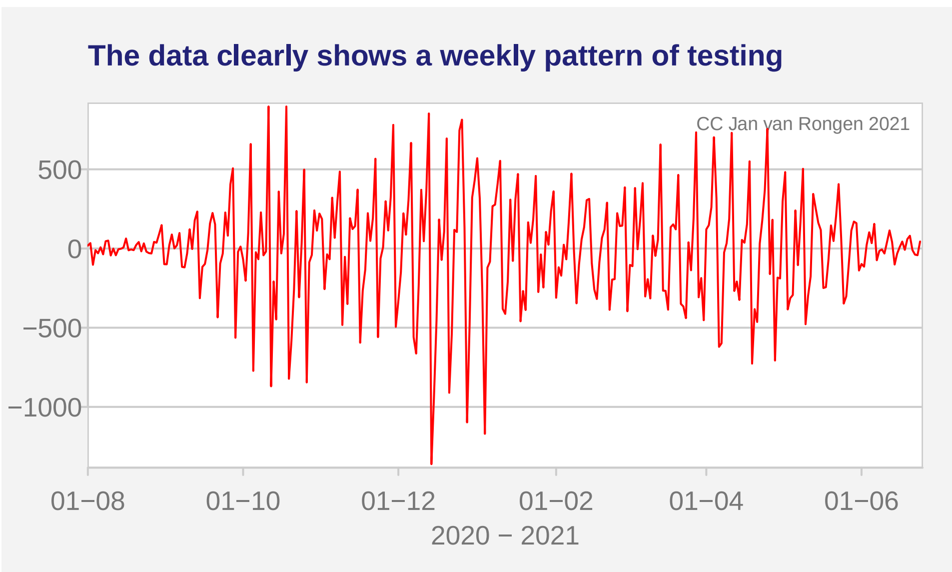
However we smooth, we calculate a trend in a time series. We fit the best possible trend (in our class of models), but certain can we be about that trend?

With t the date variable, we start with the model that $Y(t) = T(t) + E(t)$ wher T is the trens, E is the error (noise) and Y are the daily observations. This is a familiar refression model and we will immediately discover that $E(t)$ is not so randomly distributed.

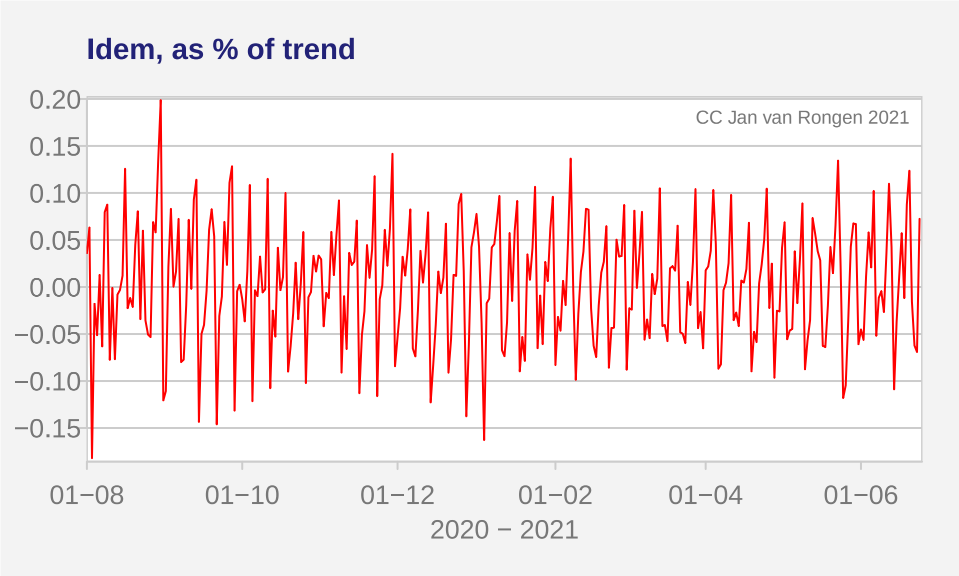
```
TO= as.Date("2021-06-25")
# change to mid july to see how absurd the Dutch situation was end june

# data
cases_all<- make_all_cases()
trend_all<- smoothed(cases_all, 11)
cases_ts<- cases_all[FROM <= cases_all$date & cases_all$date < TO, ]
trend_ts<- trend_all[FROM <= trend_all$date & trend_all$date < TO, ]

pretty_date(trend_ts, lwd=3, start=FROM, end=TO, main="Data and Trend")
pretty_date(add=T, type="p", CEX=0.9, cases_ts, kleur=2)
```




```
a<- trend_ts; a$cases<- a$cases- cases_ts$cases
pretty_date(a, start=FROM, end=TO, main="The data clearly shows a weekly pattern of test
```




```
a$cases<- ifelse(trend_ts$cases>0, a$cases/trend_ts$cases, 0)
pretty_date(a, start=FROM, end=TO, main="Idem, as % of trend")
```




```r
mean(abs(a$cases))
```

```
## [1] 0.05157477
```

Although this isn't very scientific, we can conclude that locally the trend might be deviated by 5%, so with a std of 2.5%. So when we want to know the uncertainties, we might sample from N(trend(t), 0.025*trend(t)) in a simulation.