Reproducing Gámiz, Mammen, Martínez-Miranda and Nielsen (2025) using the `pandemics` package

Contents

Introduction

The purpose of this vignette is to use the `pandemics` package to reproduce all the results presented in Gámiz, Mammen, Martínez-Miranda and Nielsen (2025): *Low quality exposure and point processes with a view to the first phase of a pandemic*. The `pandemics` package builds on the providing of a full dynamic system to describe and forecast the spread and the severity of a developing pandemic, based on available data.

Data have been downloaded from the official open data platform of the French government (see `www.data.gouv.fr`). The package includes two datasets:

1. `covid`. It contains counts of COVID-19 cases during the period running from 2020-03-18 to 2022-01-04. It consists of 658 observations and 5 variables: notification date, daily total number of hospitalized individuals, daily total number of in-hospital deaths, daily total number of hospital discharges, and daily total number of individuals who tested positive.
2. `covidAges`. It is similar to the first one but disaggregated by four age groups during the period from 2020-03-18 to 2022-01-04. The age categories considered are: 0-39 years, 40-59 years, 60-79 years, and 80 years and older. For each age group, the dataset provides daily counts of hospitalizations, in-hospital deaths, discharges, and positive test results by notification date. It includes 658 observations and 17 variables (notification date, daily number of infections, new hospitalizations, deaths and recoveries for each of the four age groups).

Model formulation

In order to facilitate the understanding of the terms mentioned throughout this document, we provide below a brief summary of the formulation of the full dynamic system, restricted here to only two processes, N_1 and N_2. These correspond to the daily number of infections and the daily number of new hospitalizations, respectively.

Assume that we observe n individuals in the interval $(0, T]$. Let $N_1(t)$ count the number of persons getting infected in the interval $(0, t]$, for $0 < t \leq T$. We denote by $N_1(dt)$ the number of persons infected in the interval $(t, t + dt]$. Furthermore, we write $N_1(dt, ds)$ for the number of pairs of persons where person 1 was infected in $(s, s + ds]$ and person 2 was infected by person 1 in $(t, t + dt]$, and $N_1(t, ds)$ for the number of persons infected in $(s, t]$ by a person that was infected in $(s, s + ds]$, with $s < t \leq T$. We assume that we observe the process $N_1(t)$ but not $N_1(dt, ds)$ or $N_1(t, ds)$, and define a model for N_1 based on Hawkes-type processes.

To define the transition model, we first introduce a σ-field, $\mathcal{F}_1(t)$ generated by $\{N_1(s) : s < t\}$, and

$$\lambda_1(t) = \lim_{dt \to 0} \frac{P(N_1(t + dt) - N_1(t) \geq 1 \mid \mathcal{F}_1(t))}{dt}, \quad 0 < t \leq T.$$

Then $\lambda_1(t)dt = E[N_1(dt) \mid \mathcal{F}_1(t)]$ is the probability that one person is infected in the interval $(t, t + dt]$ given $\mathcal{F}_1(t)$.

We assume that

$$\lambda_1(t)dt = \left(\int_0^{t^-} \mu_1 \left(\frac{t}{T}, t - s \right) N_1(ds) \right) dt + n\rho_1(t)dt,$$

where μ_1 and ρ_1 are some unknown functions. We have that $\mu_1(t/T, t - s)dt$ is the probability that a person infected at s infects a person in $(t, t + dt]$. This rate is assumed to have two dimensions: a one-dimensional marker (typically the notification date) and time (duration).

The estimator for the two-dimensional transition intensities in the described model, $\mu_1(t/T, t - s)$, works when individual follow-up is available. This means that the estimator of the infection rate is applicable when information is recorded on the time elapsed from when a person becomes infected until he/she causes a new infection. In what follows, we will refer to this situation as “full information”.

In case of full information, we observe the process $N_1(t, ds)$, for $s < t \leq T$. For s fixed, $N_1(t, ds)$ is a counting process with respect to a (right continuous, increasing and complete) filtration $\mathcal{F}_1(t)$, with $t \in (0, T)$. We consider the two-dimensional local linear estimator given by:

$$\hat{\mu}_1(t/T, t-s) = \frac{\int_{0 \leq v < u \leq T} C_1(s,t,v,u) K_{1,b_1}(t-u) K_{2,b_2}(t-s-(u-v)) N_1(du, dv)}{\int_{0 \leq v < u \leq T} C_1(s,t,v,u) K_{1,b_1}(t-u) K_{2,b_2}(t-s-(u-v)) N_1(dv) du},$$

with $N_1(du, dv)$ and $N_1(dv)$ as defined previously.

Since this estimator consists of a ratio of smoothed occurrences and smoothed exposure, the data are assumed to be aggregated in terms of occurrences and exposures. To reconstruct the unobserved durations appearing in the aforementioned equation, the estimator operates iteratively, following the iterative estimation schemed described in Subsection 4.2. (Eq. (4) and Eq. (5)) in Gámiz et al. (2025):

1. Use an initial guess $\hat{\mu}_1^{(0)}$ of μ_1.
2. The r-th iteration of the algorithm consists of two steps:
   - We estimate $N_1(du, dv)$ using the estimator $\hat{\mu}_1^{(r)}$ from the previous iteration and by using the observed process $N_1(t)$ as follows:

$$\hat{N}_1^{(r)}(du, dv) = \frac{\hat{\mu}_1^{(r-1)}(u/T, u-v)}{\int_0^{u^-} \hat{\mu}_1^{(r-1)}(u/T, u-w) N_1(dw)} N_1(dv) N_1(du).$$

   - The estimator of $N_1(du, dv)$ is plugged into the expression of $\hat{\mu}_1(t/T, t-s)$, providing the following update of $\hat{\mu}_1^{(r-1)}$, resulting:

$$\hat{\mu}_1^{(r)}(t/T, t-s) = \frac{\int_{0 \leq v < u \leq T} C_1(s,t,v,u) K_{1,b_1}(t-u) K_{2,b_2}(t-s-(u-v)) \hat{N}_1^{(r)}(du, dv)}{\int_{0 \leq v < u \leq T} C_1(s,t,v,u) K_{1,b_1}(t-u) K_{2,b_2}(t-s-(u-v)) N_1(dv) du}.$$

On the other hand, let $N_2(t)$ count the number of persons hospitalized in the interval $(0, t]$, for $0 < t \leq T$. We denote by $N_2(dt)$ the number of persons hospitalized in the interval $(t, t + dt]$. We write $N_2(dt, ds)$ for the number of persons hospitalized in $(t, t + dt]$ that had been infected in $(s, s + ds]$, and $N_2(t, ds)$ for the number of persons infected in $(s, t]$ that enter the hospital in the interval $(s, t]$, with $s < t \leq T$. We assume that we observe the process $N_1(t)$ and $N_2(t)$ for the whole interval $(0, T]$ but not $N_2(dt, ds)$.

Let define

$$\lambda_2(t) = \lim_{dt \to 0} \frac{P(N_2(t+dt) - N_2(t) \geq 1 \mid \mathcal{F}_1(t))}{dt}, \quad 0 < t \leq T.$$

Then $\lambda_2(t)dt = E[N_2(dt) \mid \mathcal{F}_1(t)]$ is the probability that one person is hospitalized in the interval $(t, t + dt]$ given $\mathcal{F}_1(t)$.

Assume that

$$\lambda_2(t)dt = \left(\int_0^{t^-} \mu_2\left(\frac{t}{T}, t-s\right) N_1(ds) \right) dt + n\rho_2(t)dt,$$

with μ_2 and ρ_2 are some unknown functions. In this case, $\mu_2(t/T, t-s)dt$ is the probability that a person, who was infected at s, is hospitalized in the interval $(t, t + dt]$.

The estimation of the transition intensity μ_2 follows a similar approach to μ_1. Specifically, to estimate this transition we employ the iterative algorithm outlined in Subsection 4.3.

1. Use an initial guess $\hat{\mu}_2^{(0)}$ of μ_2.
2. The r-th iteration of the algorithm consists of two steps:
   - We estimate $N_2(du, dv)$ using the estimator $\hat{\mu}_2^{(r)}$ from the previous iteration and by using the observed process $N_1(t)$ and $N_2(t)$ as follows:

$$\hat{N}_2^{(r)}(du, dv) = \frac{\hat{\mu}_2^{(r-1)}(u/T, u-v)}{\int_0^{u^-} \hat{\mu}_1^{(r-1)}(u/T, u-w) N_1(dw)} N_1(dv) N_2(du).$$

   - The estimator of $N_2(du, dv)$ is plugged into the expression of $\hat{\mu}_2(t/T, t-s)$, providing the following update of $\hat{\mu}_2^{(r-1)}$, resulting:

$$\hat{\mu}_2^{(r)}(t/T, t-s) = \frac{\int_{0 \leq v < u \leq T} C_1(s,t,v,u) K_{1,b_1}(t-u) K_{2,b_2}(t-s-(u-v)) \hat{N}_2^{(r)}(du, dv)}{\int_{0 \leq v < u \leq T} C_1(s,t,v,u) K_{1,b_1}(t-u) K_{2,b_2}(t-s-(u-v)) N_1(dv) du}.$$

Principles of forecasting in a dynamic environment

Assume that we have the estimation of the dynamic infection rate, $\hat{\mu}_1(t/T, u)$, for $0 < u \leq t \leq t^*$, where t^* denotes the calendar time corresponding to the most recent estimate. We fix the forecasting horizon at time $t^* + h$, with $h > 0$. By extrapolating the dynamic infection rate $\hat{\mu}_1((t^* + s)/T, u)$, for $u \geq 0$ and $0 < s \leq h$, and assuming that the infection rate at the end of the forecasting period equals to the most recent estimate multiplied by a number $C_{t^*,h}$, we obtain:

$$\tilde{\mu}_1((t^* + s))/T, u) = \hat{\mu}_1(t^*/T, u) \times \left(1 + (C_{t^*,h} - 1)\frac{s}{h}\right),$$

for $u \geq 0$ and $0 < s \leq h$.

Forecasting can be performed by setting $C_{t^*,h} = 1$. Alternatively, information from expert knowledge can motivate the use of a value of $C_{t^*,h}$ different from one. Values below 1 reflect a reduction in the infection rate, whereas values above 1 point to an increase, compared to the dynamic infection rate evaluated in the most recent estimate $\hat{\mu}_1(t/T, t^*)$.

If, in addition, we want to assess uncertainty associated with the forecasts, we can generate bootstrap samples from the number of infections on the ith day, $N_{1,i}$. To account for the substantial overdispersion relative to the Hawkes model, the bootstrap algorithm described in Subsection 7.3 consists of the following steps:

1. Take $k > \gamma$, being γ the overdispersion factor

$$\gamma = (1/T) \sum_{i=1}^{T} \frac{(N_{1,i} - \lambda_{1,i})^2}{\lambda_{1,i}},$$

where $N_{1,i}$ is the number of infections at the ith day, $i = 1, 2, \ldots, T$, and $\lambda_{1,i}$ is the expected number of infections on the ith day, given $N_{1,0}, \ldots, N_{1,i-1}$.

2. Define $\beta = (\gamma - 1)/(k(k-1))$, and $\alpha = 1 - k\beta$.
3. Given $N_{1,0}, N_{1,1}^*, \ldots, N_{1,i-1}^*$, generate a bootstrap sample with $N_{1,\alpha,i}^* \rightarrow Pois(\alpha\lambda_{1,i}^*)$, and $N_{1,\beta,i}^* \rightarrow Pois(\beta\lambda_{1,i}^*)$, being $\lambda_{1,i}^* = N_{1,0}\hat{\mu}_1(i,i) + N_{1,1}^*\hat{\mu}_1(i, i-1) + \ldots + N_{1,i-1}\hat{\mu}_1(i,1)$ the expected number of infections at the ith day given $N_{1,0}, N_{1,1}^*, \ldots, N_{1,i-1}^*$. Define $N_{1,i}^* = N_{1,\alpha,i}^* + kN_{1,\beta,i}^*$.

Repeating step 3 above a large number of times, we are able to compute 95% prediction limits by evaluating the 2.5% and 97.5% quantiles of the bootstrap samples at each day in the forecasting horizon.

When forecasting hospitalizations, the bootstrap procedure requires a slight adjustment. In this case, the sample consists of pairs of the form $\{(N_{1,0}, N_{2,0}), \ldots, (N_{1,T}, N_{2,T})\}$, where $N_{1,i}$ denotes the number of infections on the ith day, and $N_{2,i}$ the number of new hospitalizations on the ith day, for $i = 1, 2, \ldots, T$. From these, we generate bootstrap samples that incorporate this additional step:

4. Given $N_{1,0}, N_{1,1}^*, \ldots, N_{1,i-1}^*$, generate a bootstrap sample with $N_{2,i}^* \rightarrow Pois(\lambda_{2,i}^*)$, with $\lambda_{2,i}^* = N_{1,0}\hat{\mu}_2(i,i) + N_{1,1}^*\hat{\mu}_2(i,i-1) + \ldots + N_{1,i-1}\hat{\mu}_2(i,1)$.

Figure 2: dynamic estimation of the rate of infection

To reproduce the dynamic estimation of the infection rate, we use the function `rate2Dmiss` included in the `pandemics` package. This function implements the dynamic local linear estimator of the infection (or the hospitalization) rate in the case of missing-survival-link data (Gámiz et al. 2025). The rate is assumed to have two dimensions: a one-dimensional marker (typically the notification date) and time (duration) (see Eq. (3) in Subsection 4.1.).

To estimate the infection rate, we only consider daily counts of individuals who tested positive, since we estimate the infection rate describing the transition loop from infected to infected. Firstly, we need to define an exposure vector and an occurrence vector. The exposure vector, `Ei.z1`, will contain the daily positive counts. Meanwhile, the occurrences vector, `Oi.z`, will represent the daily number of positive cases, excluding the first d days to account for a typical delay. In our case, we will consider `delay = 1`, since an infected individual may become infected one day after testing positive.

It is common to observe, in several COVID datasets, a large variation in the number of reported infections depending on the day of the week. This is particularly evident in the French COVID dataset. In the case of new infections, this variation may be due to delays in confirming cases and compiling results over the weekend. Although this variation is observed especially in the reporting of new infections, it can also be found, to a lesser extent, in the reporting of hospitalizations cases, recoveries, and hospital deaths. For this reason, in the results presented in this document, a correction is applied to these data in order to mitigate the reporting delay of cases over the weekend. This correction was proposed in Koyama et al. (2021) and, in our package, is implemented through the function `week_effect`. This function only requires the values of a single variable and returns the same variable with the correction applied to the data. In what follows, we will always apply Koyama's correction to our data.

In order to generate the dynamic local linear estimator of the infection rate, we first apply Koyama's correction to the newly infected cases. We remove the first 56 rows and the last 3 rows since there are no testing data available before May 13th, 2020, and after January 2nd, 2022. Since we want to see the dynamic local linear estimator for the entire dataset, we set the length as `M <- length(Ei.new)`, although of course, a smaller vector can also be used. After correcting for the weekend effect, we adjust the `delay` parameter, which we will set to 1.

```r
library(pandemics)
```

```r
data('covid')
Ei.new<-week_effect(covid$Posit[-c(1:56, 656:658)])
M <- length(Ei.new)
delay<-1;M<-M-delay
Oi.z<-Ei.new[-(1:delay)]; Ei.z1<-Ei.new[1:M]
```

To estimate the infection rate, we use the function `rate2Dmiss`. The function requires, as arguments, a vector of `M` grid points for the time dimension (`t.grid`), a vector of `M` grid points for the marker dimension (`z.grid`), and two vectors of length `M` with the number of people tested positive (`Ei.z1` and `Oi.z`). Regarding the grids, `z.grid` represents the times at which a person becomes infected, and `t.grid` represents the times at which the infected person transmits the infection to another person.

In addition, the function `rate2Dmiss` requires, as an argument, a pair of bandwidths that appear in expression (5) of the two-dimensional local linear estimator. These two bandwidths can be specified manually or estimated using the cross-validation method described in Gámiz et al. (2013) through the argument `bs.grid`. For the latter, a two-dimensional grid of possible bandwidth values is proposed. Typically, the first dimension, corresponding to time, requires more smoothing in this application, while the second dimension, the one related to delay, needs to be handled more precisely.

Among all possible combinations of bandwidths in the grid, cross-validation should ideally select the most appropriate pair. However, it often happens that the selected bandwidths for estimating either the infection or hospitalization rate fall at the boundaries of the grid. This suggests that the cross-validation procedure has failed to identify an optimal bandwidth and that a different range of values should be considered. In fact, in the following example, after the function is executed with the cross-validation method (`cv = TRUE`), it returns the two lowest values in the grid as the selected bandwidths.

```r
t.grid<-z.grid<-1:M
nb1<-10;nb2<-5
bs.grid<-expand.grid(seq(30, 2*M,length=nb2),seq(5,M,length=nb1))
RInf<-rate2Dmiss(t.grid,z.grid,Oi.z,Ei.z1,bs.grid=bs.grid,cv=TRUE)
RInf$bcv
```

```
## [1] 30  5
```

When the cross-validation method selects one of the lower bounds of the grid, it suggests that the optimal bandwidth for estimating the infection rate lies below the current grid range. Even after testing with a different grid and adjusting parameters, the method still returns the two lowest values in the grid.

```r
nb1<-10;nb2<-5
bs.grid<-expand.grid(seq(20, 30,length=nb2),seq(5,M,length=nb1))
RInf<-rate2Dmiss(t.grid,z.grid,Oi.z,Ei.z1,bs.grid=bs.grid,cv=TRUE)
RInf$bcv
```

```
## [1] 20  5
```

The only solution in these cases is to iteratively adjust the grid until a suitable pair of bandwidths is obtained. This adjustment should be done carefully, avoiding large jumps in the values defined within the grid. In fact, instead of modifying the first dimension, it may be more appropriate to adjust the delay dimension next.

```r
nb1<-10;nb2<-5
bs.grid<-expand.grid(seq(20, 30,length=nb2),seq(3,M,length=nb1))
RInf<-rate2Dmiss(t.grid,z.grid,Oi.z,Ei.z1,bs.grid=bs.grid,cv=TRUE)
RInf$bcv
```

```
## [1] 20  3
```

If, after several attempts, the cross-validation procedure still fails or does not yield satisfactory results, the infection rate can be manually adjusted. This means specifying two bandwidths that are appropriate for the analysis goals. For example, one might choose the pair $(b_1, b_2) = (15, 10)$, although other combinations such as $(b_1, b_2) = (10, 5)$ or even $(b_1, b_2) = (5, 5)$ may also be considered. Note that when a pair of bandwidths is provided directly, it is not necessary to specify a grid of values. Instead, the function `rate2Dmiss` receives a transposed vector, `bs`, with the two bandwidths.

As additional arguments, the function `rate2Dmiss` also allows including a value `epsilon` representing the tolerance in the iterative algorithm described in Section 2 of this document, and an integer value `max.ite` specifying the maximum number of iterations in the iterative algorithm. By default, these values are set as `epsilon = 1e-4` and `max.ite = 50`.

```r
t.grid<-z.grid<-1:M
bs<-t(c(15,10))
RInf<-rate2Dmiss(t.grid,z.grid,Oi.z,Ei.z1,bs.grid=bs,cv=FALSE)
hi.zt<-RInf$hi.zt # The estimated infection rate
```

We present the infection rate estimation in the form of a matrix, where each row corresponds to the grid points for the marker dimension, and each column represents the time elapsed until an individual infects another, i.e., the duration. In addition, the output of the `rate2Dmiss` function includes a two-dimensional vector with the bandwidth used to compute the estimator (estimated by cross-validation if `cv=TRUE`), a numeric value indicating the tolerance achieved by the algorithm, and the number of iterations performed.

Once the rate of infection is estimated, we choose several notification dates. In this example, we choose May 31st, June 30th, July 31st, September 30th, December 31st and September 30th, 2021 as reference dates, although these can be modified as needed.

```r
ddates<-covid$Date[-c(1:56)]
z1<-c(19,49,80,141,233,506)
nz<-length(z1)
zdates<-ddates[z1]
t.min<-min(M-z1+1)
ti<-1:t.min;n0<-length(ti)
```

A matrix `alpha.I` is created in order to plot the dynamic rate of infection. In this matrix, each row corresponds to the grid points for the marker dimension (`z.grid`), and each column represents the grid points for the time dimension (`t.grid`). In other words, this matrix reflects the probability that a person infected at `z` infects a person in `t`. As a final point, we produce the different estimates of the infection rate, each shown in a different color for several selected notification dates.

```r
alphas.I<-matrix(NA,M,M)
for(j in 1:M) alphas.I[j,j:M]<-hi.zt[j,1:(M-j+1)]
alphas<-alphas.I[z1,-(1:18)]
M2a<-ncol(alphas)
yy<-c(0,max(alphas,na.rm=TRUE)+0.02)

plot(1:M2a,alphas[nz,],type='l',ylab='',xlab='Date of notification',
     main= 'Dynamic rate of infection',ylim=yy,lwd=2,lty=1,xaxt='n')
axis(1,at=z1-18,labels=c(zdates))
for (i in 2:nz) lines(1:M2a,alphas[i-1,],lwd=3,col=i,lty=i)
legend('topright',c('Starting on date:',as.character(zdates)),
       lty=c(NA,2:nz,1),lwd=c(NA,rep(3,nz-1),2),col=c(NA,2:nz,1),bty='n',cex=0.8)
```

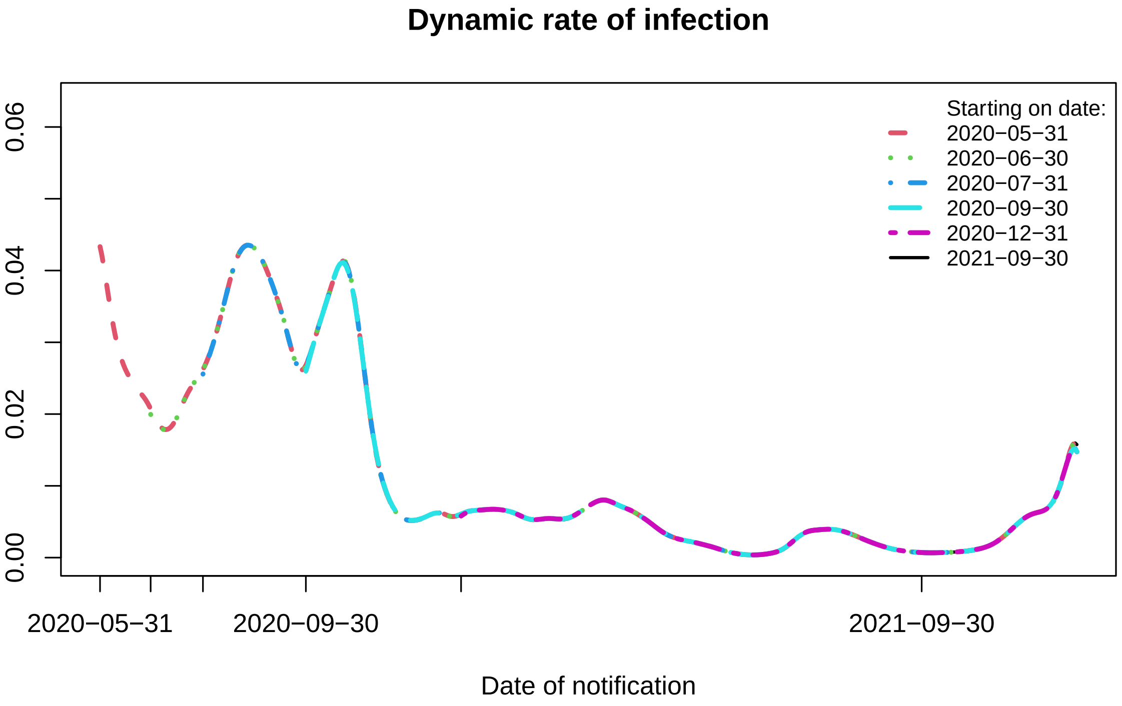


Figure 3: dynamic estimation of the rate of hospitalization

As before, to display the figure of the dynamic estimation of the hospitalization rate, we use the function `rate2Dmiss`. In this case, and unlike the infection rate, we use daily data on new hospital admissions (`Oi.z`) as occurrences together with the daily counts of individuals who tested positive (`Ei.z1`) as exposure, since we are describing the transition from infection to hospitalization.

In order to compute the daily number of new hospitalizations, we rely on the recorded data. However, in the specific case of the French dataset, we only have the total number of hospitalized individuals per day, but not the daily new hospitalizations, which are necessary to estimate the dynamic rate of hospitalization. Specifically, the number of new hospitalizations on day `i+1` can be computed as `newHi[i+1] = Hi[i+1] - newHi[i] - Di[i] - Ri[i]`, where `newHi` is the number of newly hospitalized patients, `Hi` is the total number of hospitalized individuals, `Di` is the total number of deaths, and `Ri` is the total number of recoveries.

One important remark about this calculation is that, when computing the new hospitalizations, it must be done using all available observations in the dataset. In other words, the full history of processes involving the number of hospitalized patients, deaths, and recoveries cannot be lost. Once the new hospitalizations are computed, a specific period can be selected for further analysis.

```r
Hi<-covid$Hospi ; Hi<-Hi[-1]
Ri<-diff(covid$Recov)
Di<-diff(covid$Death)
M2<-length(Di)
## New hospitalizations are Hi-Ri-Di
newHi<-Hi[-1]-(Hi[-M2]-Ri[-M2]-Di[-M2])
newHi<-c(Hi[1],newHi)
newHi[newHi<0]<-0; # possible inconsistency in the data
newHi<-as.integer(newHi)
```

We remove the first 56 rows and the last 3 rows from the counts of infected individuals. Since the newly hospitalized cases computed earlier come from differenced vectors, we remove only the first 55 and the last 3 observations from these. The reason for also removing observations from the new hospitalizations is that, in the dataset we are working with, there are no infection records prior to May 13, 2020. If infection data were available together with the other variables in our dataset, the calculation would be considerably simplified, as we would not need to select specific time periods of interest. Lastly, we apply Koyama's correction to both new infections and new hospitalizations.

```
Ei.z1<-week_effect(covid$Posit[-c(1:56, 656:658)])
newHi <- week_effect(newHi[-c(1:55, 655:657)])
Oi.z<-newHi
```

The function `rate2Dmiss` requires a vector of `M` grid points for the time dimension (`t.grid`), a vector of `M` grid points for the marker dimension (`z.grid`), a vector of length `M` with the number of people tested positive (`Ei.z1`), and a vector of length `M` with the number of new hospitalizations notified each day in `a.grid` (`Oi.z`). In this case, the grid `z.grid` represent the times at which a person becomes infected, and `t.grid` represents the times at which the person is can be hospitalized.

For the estimation of the hospitalization rate, we directly propose a pair of bandwidths. However, as previously discussed, they can also be estimated using cross-validation along with appropriate grid adjustment.

```
M<-length(Ei.z1)
t.grid<-z.grid<-1:M
bs<-t(c(15,10))
RHosp<-rate2Dmiss(t.grid,z.grid,Oi.z,Ei.z1,bs.grid=bs,cv=FALSE)
hi.zt<-RHosp$hi.zt # The estimated hospitalization rate
```

The hospitalization rate estimation is presented in the form of a matrix. Each row corresponds to the grid points for the marker dimension, that is, the times of infection, and each column represents the duration between when a person becomes infected and when they are admitted to the hospital.

To plot the hospitalization rate, we have chosen the same days used to plot the infection rate. We select May 31st, June 30th, July 31st, September 30th, December 31st, and September 30th, 2021 as reference dates.

```
ddates<-covid$Date[-c(1:56)]
z1<-c(19,49,80,141,233,506)
nz<-length(z1)
zdates<-ddates[z1]
t.min<-min(M-z1+1)
ti<-1:t.min;n0<-length(ti)
```

The matrix `alpha.I` created for the case of hospitalizations has a slightly different meaning than the one created for the infection rate. In this matrix, each row corresponds to the time at which a person becomes infected (grid points for the marker dimension), and each column represents the time at which that infected person is admitted to the hospital (grid points for the time dimension). Finally, we generate various hospitalization rate estimates, each displayed in a distinct color for several chosen notification dates.

```
alphas.I<-matrix(NA,M,M) # upper-triangular matrix
for(j in 1:M) alphas.I[j,j:M]<-hi.zt[j,1:(M-j+1)]
alphas<-alphas.I[z1,-(1:18)]
M2a<-ncol(alphas)
yy<-c(0,max(alphas,na.rm=TRUE)+0.01)

plot(1:M2a,alphas[nz,],type='l',ylab='',xlab='Date of notification',
     main= 'Dynamic rate of hospitalization',ylim=yy,lwd=2,xaxt='n')
axis(1,at=z1-18,labels=c(zdates))
for (i in 2:nz) lines(1:M2a,alphas[i-1,],lwd=3,col=i,lty=i)
legend('topright',c('Starting on date:',as.character(zdates)),
       lty=c(NA,2:nz,1),lwd=c(NA,rep(3,nz-1),2),col=c(NA,2:nz,1),bty='n',cex=0.8)
```

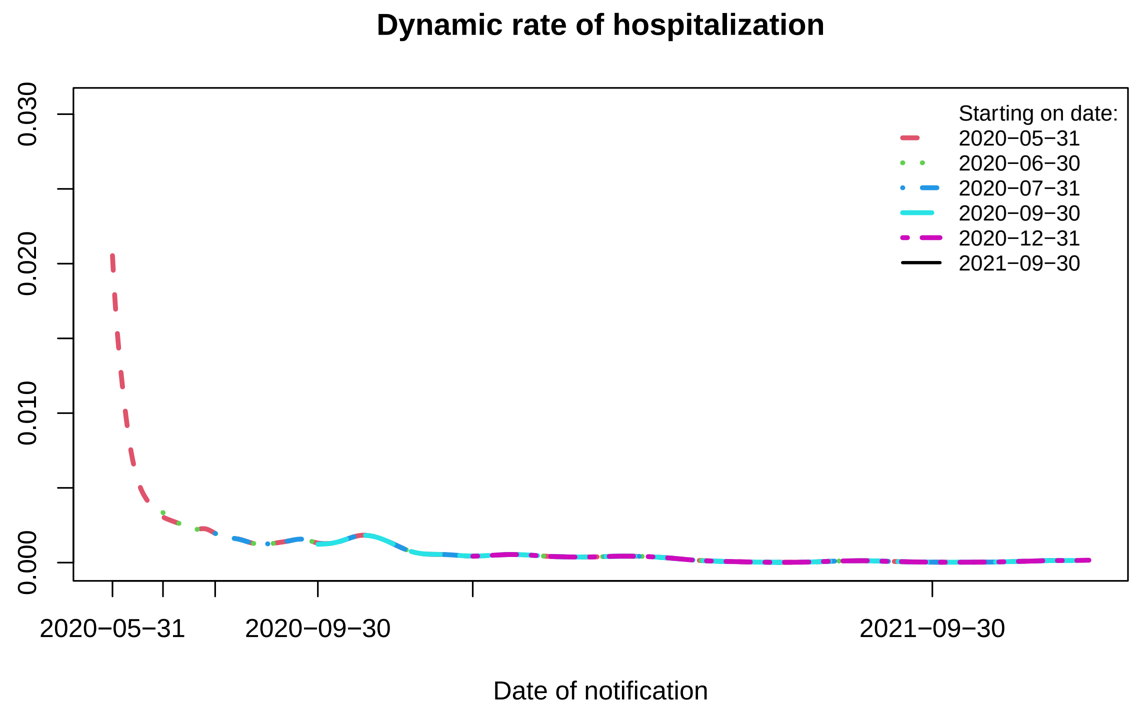


Figure 5: forecasts of new positives in October 2020

To replicate the forecasting exercise for new positive cases, we use the `forecasting` function. This approach relies on the estimation methods presented in Section 6 and the forecasting principles outlined in Section 5 in Gámiz et al. (2025).

The `forecasting` function allows generating forecasts for a given forecasting horizon. These forecasts are determined through the infection indicator $C_{t^*,h}$, introduced in Section 3 of this document. This indicator will be specified through the argument `Cval`. As previously mentioned, `Cval = 1` is often not an appropriate choice for forecasting the immediate future, as will be illustrated in the figures below. However, we show the forecasts computed with it in order to illustrate the potential of this methodology. Therefore, it is advisable to specify a value greater or smaller than 1 in order to adjust the forecasts as closely as possible to reality. An appropriate choice of `Cval` is strongly related to the reported reproduction number. In the case of forecasting the daily number of infections, the infection indicator `Cval` measures how many new infections are caused by one infected individual within a given time period. For further details and its relationship with the well-known reproduction number, see Subsection 7.1 of Gámiz et al. (2025).

In addition to the indicator, `Cval`, and the forecasting period, `period`, the `forecasting` function requires as input the observed data of the process. In this case, since we aim to present a forecasting exercise for the infection-to-infection loop, we only require the observed data on the daily number of new infections, namely `Pz`. Furthermore, associated with this process, we also need the estimation of the infection rate, `RoInf`, computed via the function `rate2Dmiss` as described earlier.

With regard to the assessment of uncertainty, the `forecasting` function optionally allows implementing the bootstrap algorithm. This option can be enabled using the boolean argument `boot`, which is set to `FALSE` by default. When uncertainty estimation is requested, it is carried out internally within the `forecasting` function through several auxiliary functions. Broadly speaking, the function first generates bootstrap samples according to the steps described in Section 3 of this document. The number of bootstrap samples must be specified via the argument `B`, and it is recommended to use a sufficiently large number of samples. By default, the number of samples generated is 500, and they are produced with a seed value of 1, which can be optionally modified using the seed argument. Subsequently, for each sample, forecasts are computed with the chosen values of Cval and period.

These bootstrap forecasts rely on a bandwidth matrix, band.matrix, containing two bandwidths used to estimate the infection rates for the bootstrap samples. It is perfectly valid to use the same bandwidths that were employed to estimate the infection rate for the observed dataset. However, it is also possible to modify these bandwidths if desired. That said, the matrix band.matrix must always be included as an argument in the forecasting function when boot = TRUE; otherwise, an error message will be displayed when running the function, of the form: "argument 'band.matrix' is required when 'boot = TRUE'." Once the samples and their corresponding forecasts are obtained, the 2.5% and 97.5% quantiles are calculated for each day in the forecasting horizon.

As its final output, the forecasting function returns a matrix composed of the observed data (Pz.obs), the fitted and predicted values (Pz.pred), and, when boot = TRUE, the lower and upper bounds of the prediction bands (Pz.PI.lwr and Pz.PI.upr). As a final remark, note that the fitted values in the output have the same length as the input vector, while the predicted values have the length of the forecasting horizon, although both are presented together in the column Pz.pred.

For this forecasting exercise, we propose to forecast the number of new positive cases in October 2020 using the data from France, covering the period from May 13th to September 30th, since there are no testing data available before May 13th, 2020. We begin by taking the French dataset and applying Koyama's correction to this variable. We then define the exposure and occurrences. In the case of forecasting infections, both represent the daily number of new positive cases.

```r
Pi <- week_effect(covid$Posit[-c(1:56, 656:658)])
Ms<- 141
Ei.z<-Pi[1:Ms]
delay<-1;Msd<-Ms-delay
Oi.z<-Ei.z[-(1:delay)]
Ei.z1<-Ei.z[1:Msd];
```

Next, we estimate the dynamic rate of infection using the rate2Dmiss function on data up to September 30th. In this case, we directly provide a pair of bandwidths, without the need to perform cross-validation, although cross-validation can also be used for the estimation.

```r
t.grid<-z.grid<-1:Msd
bs <- t(c(5,10))
RInf<-rate2Dmiss(t.grid,z.grid,Oi.z=Oi.z,Ei.z1=Ei.z1,bs.grid=bs,cv=FALSE)
RoInf <- RInf$hi.zt
```

Once the estimated infection rate has been computed, we proceed to the forecasting stage using the forecasting function. To reproduce Figure 5 in Gámiz et al. (2025), we need to call the function twice: once with the infection indicator set to 1, and once with the optimal value of the indicator. For the first case, it is not necessary to compute prediction bands using the bootstrap algorithm, since we only require the predicted number of daily new positives under the assumption that the future will behave in the same way as the immediate past. Therefore, we only store the estimated number of new positives up to September 30th (fitted values), the predicted values for October, and the observed values up to September 30th in order to plot them.

To estimate the optimal value of $C_{t^*,h}$ for forecasting infections in October, we construct a grid of possible values of the indicator and identify the one that yields the smallest mean squared error with respect to the actual values in the forecasting period. The grid we have designed ranges from 0.01 to 4, taking 100 possible values of the indicator within this interval. Note that this procedure is, of course, infeasible in practice, since future data are never available. In this case, the optimal infection indicator takes the value 1.863939. Both the code required to reproduce this value and the functions needed to obtain the value of the infection indicator are available in the GitHub repository.

We call the forecasting function using the optimal value of the infection indicator. We specify the option `boot = TRUE` in order to compute the prediction bands. We perform a total of `B = 500` bootstrap samples and store the forecasts for October along with the lower and upper bounds of the prediction bands, `Pz.PI.lwr` and `Pz.PI.upr`, respectively.

```
Cval<-1
period<-31  # Forecasts up to 31st October, 31 days
Pz<-Pi[1:Ms]

fore<-forecasting(Cval=1,period=period,Pz=Pz,RoInf=RoInf)
Pz.fitted<-fore$Pz.pred[1:Ms]
Pz.predC1<-fore$Pz.pred[(Ms+1):(Ms+period)]
Pz.obs<-fore$Pz.obs

Cval<-1.86
band.matrix <- matrix(c(5,10),nrow =1,ncol=2)
fore_opt<-forecasting(Cval=Cval,period=period,Pz=Pz,RoInf=RoInf,boot=TRUE,B=500,band.matrix=band.matrix
Pz.predCopt<-fore_opt$Pz.pred[(Ms+1):(Ms+period)]
Pz.PI.lwr<-fore_opt$Pz.PI.lwr[(Ms+1):(Ms+period)]
Pz.PI.upr<-fore_opt$Pz.PI.upr[(Ms+1):(Ms+period)]
```

We display two types of points: black dots represent observed data up to September, while white dots correspond to the number of infections recorded in October. We also display three types of lines: the black solid line shows the fitted values for the number of infections, while the black and red dotted-dashed line represents the forecasts obtained using `Cval=1` and `Cval=1.86`, respectively. We also plot the prediction bands based on the forecasts obtained with the optimal infection indicator in the bootstrap samples.

```
Sys.setlocale("LC_TIME", "C");
```

```
## [1] "C"
```

```
ddates<-as.Date(1:(Ms+period),origin='12/05/2020',format="%d/%m/%y")
ddates<-format(ddates,format="%D")

t1<-1:(Ms+period);y1<-c(rep(NA,Ms),Pz.PI.lwr);y2<-c(rep(NA,Ms),Pz.PI.upr);m2<-length(t1)
x21<-t1;x22<-t1[m2:1];y21<-y1;y22<-y2[m2:1]

plot(1:(Ms+period),Pz.obs,lty=2,col=1,ylab='',ylim=c(0,80000),
     main='Forecasts of new positives in October 2020 (including uncertainty)',
     xlab='Date of notification',type='n',xaxt='n',pch=20)
oat<-c(1,17,32,47,62,78,93,109,124,139,154,170)
olab<-ddates[oat]
axis(1,at=oat,labels=olab)

points(1:(Ms+period),Pz.obs,pch=20,col='black')
lines(1:Ms,Pz.fitted,lty=1,col=1,lwd=2)
polygon(c(x21,x22,x21[1]),c(y21,y22,y21[1]),col='mistyrose',border=F)
lines((Ms+1):(Ms+period),Pz.predC1,lty=4,lwd=2,col=1)
lines((Ms+1):(Ms+period),Pz.predCopt,lty=2,lwd=2,col=2)

Pz.obsOct<- Pi[(Ms+1):(Ms+period)]
points((Ms+1):(Ms+period),Pz.obsOct,pch=21)

legend('topleft',c('Daily number of new positives until 30th September (adjusted for weekday effect)',
                   'Daily number of new positives in October (adjusted for weekday effect)',
  'Estimated number of new positives until 30th September',
  paste0('Forecasts with C[t*,h]=',Cval,' (optimal to predict new positives in the period 1-31 October)
  paste0('Forecasts with C[t*,h]=',1),
  paste0('95% Prediction bands using C[t*,h]=',Cval)),
  col=c(1,1,1,2,1,'mistyrose'),lty=c(NA,NA,1,2,4,1),
  lwd=c(NA,NA,2,2,2,2),pch=c(20,21,NA,NA,NA,NA),bty='n',cex=0.8)
```

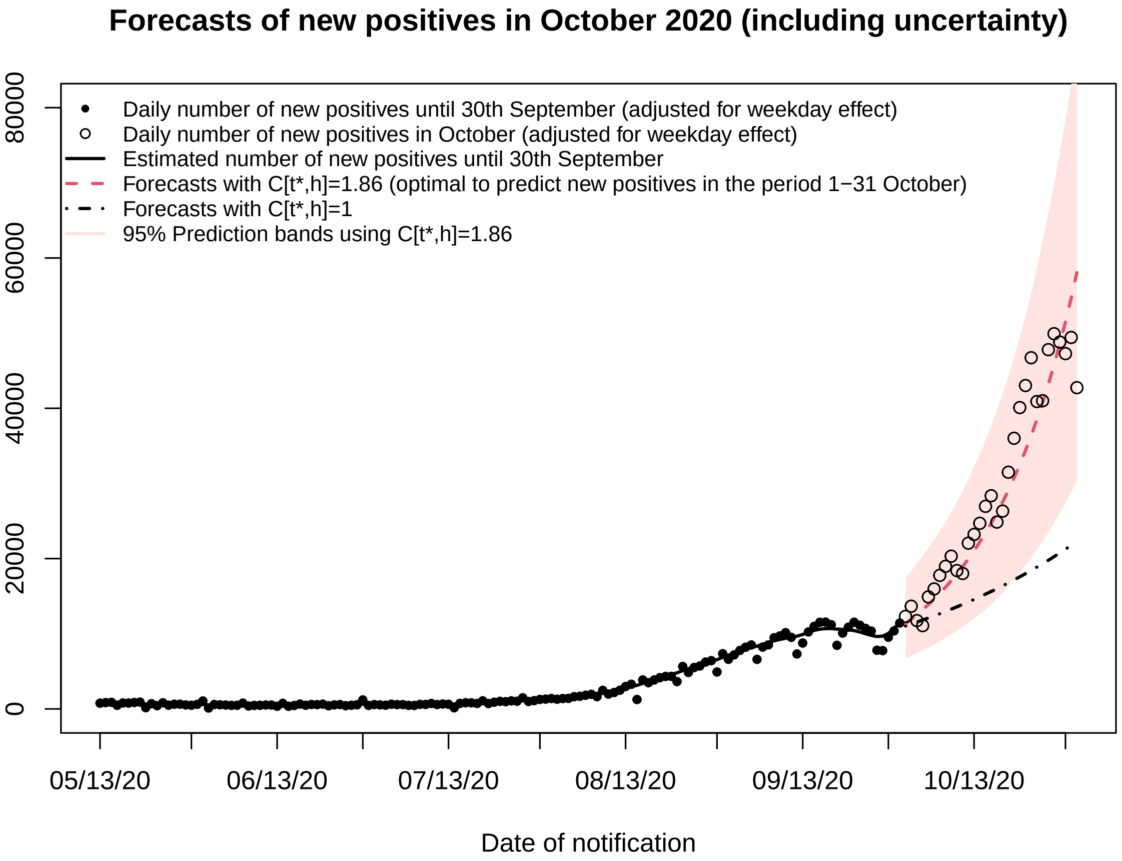


Figure 6: forecasts of new hospitalizations in October 2020

We now present a final forecasting exercise. In this case, the focus is on the number of new hospitalizations over the same period as before, from May 13 to September 30, with the goal of forecasting the daily counts for the month of October. As in the previous scenario, we use the `forecasting` function to calculate forecasts and incorporate uncertainty through the bootstrap scheme. Following the latter, the steps to be carried out for new hospitalizations are very similar to those for new infections. The only difference is the addition of a fourth step to the bootstrap method described in detail in Subsection 7.3 of Gámiz et al. (2025).

We begin by estimating the hospitalization rate. For this, what we need are the observed data on the daily number of infections and the daily number of new hospitalizations. The latter has already been calculated in the estimation of the dynamic rate of hospitalization and stored in the variable `newHi`. To this end, we select the estimation and forecasting periods from our data and specify the bandwidths used for their estimation. As in the previous case, cross-validation can be carried out for the hospitalization rate, but we directly provide a pair of bandwidths.

```
Ms<- 141
Ei.z1 <- Pi[1:Ms]
Oi.z <- newHi[1:Ms]
```

```r
t.grid<-z.grid<-1:Ms
bs <- t(c(5,10))
RHosp<-rate2Dmiss(t.grid,z.grid,Oi.z=Oi.z,Ei.z1=Ei.z1,bs.grid=bs,cv=FALSE)
RoHosp <- RHosp$hi.zt
```

We next estimate the optimal value of $C_{t^*,h}$ for forecasting hospitalizations in October. By again applying a grid of possible indicator values and using the mean squared error as the measure of error, we arrive at the estimation that, for October 2020, a value of $C_{t^*,h} = 3.07303$ is obtained. Similar to the infection indicator, the necessary code to reproduce and obtain the value of this indicator can be found in the GitHub repository. As before, this value can only be obtained if the data for the forecasting period in which this methodology is to be applied are available. Otherwise, expert information must be incorporated through the dynamic indicator to improve the forecasts and make them as realistic as possible. See Subsection 7.1 of Gámiz et al. (2025) for further details and its relationship with the well-known reproduction number.

To make forecasts using the `forecasting` function, in the case of forecasting the number of new hospitalizations, the function requires as arguments the value `Cval`, the `period` argument, the observed data on the daily number of new infections and new hospitalizations, `Pz` and `newHz`, and the estimated matrices for the rates of infection and hospitalization, `RoInf` and `RoHosp`, respectively. To establish the prediction intervals on the forecasts with the indicator value, `boot=TRUE` must be selected, along with the number of bootstrap samples to be computed and the bandwidths matrix. As a note on the estimation of the bootstrap samples, the argument `band.matrix` allows the inclusion not only of the pair of bandwidths for estimating the dynamic rate of infection, but also the pair of bandwidths for estimating the hospitalization rate. It is sufficient to specify the pair of bandwidths for infections in the first row and the pair of bandwidths for hospitalizations in the second row.

As before, the `forecasting` function returns a matrix composed of the observed data (`Pz.obs` and `newHz.obs`), the fitted and predicted values (`Pz.pred` and `newHz.pred`), and, when `boot = TRUE`, the lower and upper bounds of the prediction bands for both new infections and new hospitalizations.

```r
Cval<-1
period<-31
Pz<-Pi[1:Ms]
newHz<-newHi[1:Ms]

fore<-forecasting(Cval=Cval,period=period,Pz=Pz,newHz=newHz,
                  RoInf=RoInf,RoHosp=RoHosp)
newHz.fitted<-fore$newHz.pred[1:Ms]
newHz.predC1<-fore$newHz.pred[(Ms+1):(Ms+period)]
newHz.obs<-fore$newHz.obs

Cval<-3.07
band.matrix <- matrix(c(5,5,10,10),nrow=2,ncol=2)
fore_opt<-forecasting(Cval=Cval,period=period,Pz=Pz,newHz=newHz,
                  RoInf=RoInf,RoHosp=RoHosp,boot=TRUE,B=500,band.matrix=band.matrix)
newHz.predCopt<-fore_opt$newHz.pred[(Ms+1):(Ms+period)]
newHz.PI.lwr<-fore_opt$newHz.PI.lwr[(Ms+1):(Ms+period)]
newHz.PI.upr<-fore_opt$newHz.PI.upr[(Ms+1):(Ms+period)]
```

Lastly, we display two types of points: black dots represent observed data up to September, while white dots correspond to the number of hospitalizations recorded in October. We also display three types of lines: the black solid line shows the fitted values for the number of hospitalizations, while the black and red dotted-dashed line represents the forecast obtained using `Cval=1` and `Cval=3.07`, respectively. We also plot the prediction bands for the forecasts.

```r
Sys.setlocale("LC_TIME", "C");
```

```
## [1] "C"
```

```r
ddates<-as.Date(1:(Ms+period),origin='12/05/2020',format="%d/%m/%y")
ddates<-format(ddates,format="%D")

t1<-1:(Ms+period);y1<-c(rep(NA,Ms),newHz.PI.lwr);y2<-c(rep(NA,Ms),newHz.PI.upr);m2<-length(t1)
x21<-t1;x22<-t1[m2:1];y21<-y1;y22<-y2[m2:1]

plot(1:(Ms+period),newHz.obs,lty=2,col=1,ylab='',ylim=c(0,4500),
     main='Forecasts of new hospitalizations in October 2020
     (including uncertainty)',
     xlab='Date of notification',type='n',xaxt='n',pch=20)
oat<-c(1,17,32,47,62,78,93,109,124,139,154,170)
olab<-ddates[oat]
axis(1,at=oat,labels=olab)

points(1:(Ms+period),newHz.obs,pch=20,col='black')
lines(1:Ms,newHz.fitted,lty=1,col=1,lwd=2)
polygon(c(x21,x22,x21[1]),c(y21,y22,y21[1]),col='mistyrose',border=F)
lines((Ms+1):(Ms+period),newHz.predC1,lty=4,lwd=2,col=1)
lines((Ms+1):(Ms+period),newHz.predCopt,lty=2,lwd=2,col=2)

newHz.obsOct<- newHi[(Ms+1):(Ms+period)]
points((Ms+1):(Ms+period),newHz.obsOct,pch=21)

legend('topleft',c('Daily number of new hospitalized until 30th September (adjusted for weekday effect)
                   'Daily number of new hospitalized in October (adjusted for weekday effect)',
  'Estimated number of new hospitalizations until 30th September',
  paste0('Forecasts with C[t*,h]=',Cval,' (optimal to predict new hospitalizations in the period 1-31 Oc
  paste0('Forecasts with C[t*,h]=',1),
  paste0('95% Prediction bands using C[t*,h]=',Cval)),
  col=c(1,1,1,2,1,'mistyrose'),lty=c(NA,NA,1,2,4,1),
  lwd=c(NA,NA,2,2,2,2),pch=c(20,21,NA,NA,NA,NA),bty='n',cex=0.8)
```

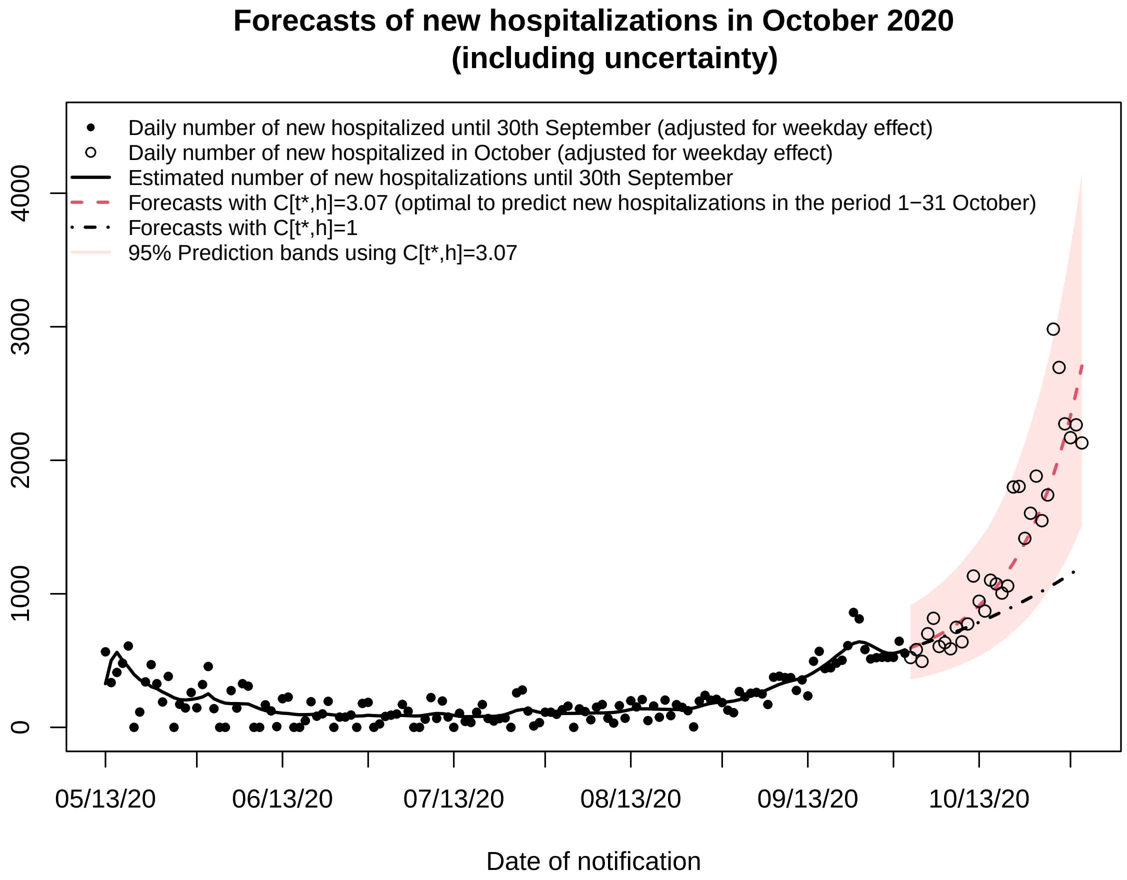


Figure 7: forecasts of new positives in October 2020 using the R number

To generate forecasts of new positives in October 2020 using the R-number, it is enough to build on the previous code lines and apply the `forecast` function with two different values of the infection indicator: one corresponding to the optimal value of 1.86, and another equal to the R-number, 1.34. This R-number value is taken as the average of the reported R-numbers calculated with data not later than 31st of October. See Subsection 7.1. and Subsection 7.4.

```
Ms<- 141
Pz <- Pi[1:Ms]

Cval<-1.86
period<-31  # Forecasts up to 31st October, 31 days

fore<-forecasting(Cval=Cval,period=period,Pz=Pz,RoInf=RoInf)
Pz.fitted<-fore$Pz.pred[1:Ms]
Pz.predCopt<-fore$Pz.pred[(Ms+1):(Ms+period)]
Pz.obs<-fore$Pz.obs

Cval<-1.34
fore_R<-forecasting(Cval=Cval,period=period,Pz=Pz,RoInf=RoInf)
Pz.predR<-fore_R$Pz.pred[(Ms+1):(Ms+period)]
```

The figure display two forecast lines: the red dashed line reflects forecasts using the optimal infection indicator

(Cval=1.86), while the blue dotted line shows forecasts derived from the R-number.

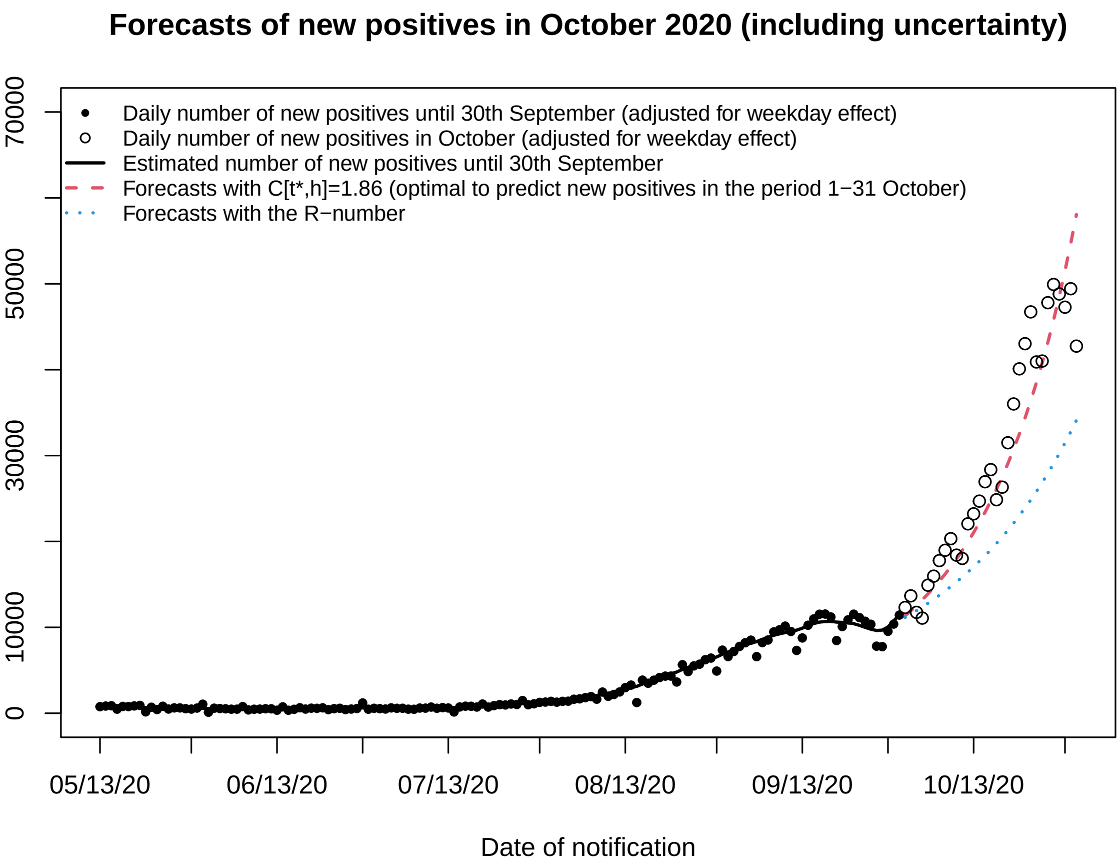
```
ddates<-as.Date(1:(Ms+period),origin='12/05/2020',format="%d/%m/%y")
ddates<-format(ddates,format="%D")

plot(1:(Ms+period),Pz.obs,lty=2,col=1,ylab='',ylim=c(0,70000),
     main='Forecasts of new positives in October 2020 (including uncertainty)',
     xlab='Date of notification',type='n',xaxt='n',pch=20)
oat<-c(1,17,32,47,62,78,93,109,124,139,154,170)
olab<-ddates[oat]
axis(1,at=oat,labels=olab)

points(1:(Ms+period),Pz.obs,pch=20,col='black')
lines(1:Ms,Pz.fitted,lty=1,col=1,lwd=2)
lines((Ms+1):(Ms+period),Pz.predCopt,lty=2,lwd=2,col=2)
lines((Ms+1):(Ms+period),Pz.predR,lty=3,lwd=2,col=4)

points((Ms+1):(Ms+period),Pz.obsOct,pch=21)

legend('topleft',c('Daily number of new positives until 30th September (adjusted for weekday effect)',
                   'Daily number of new positives in October (adjusted for weekday effect)',
  'Estimated number of new positives until 30th September',
  'Forecasts with C[t*,h]=1.86 (optimal to predict new positives in the period 1-31 October)',
  'Forecasts with the R-number'),
  col=c(1,1,1,2,4),lty=c(NA,NA,1,2,3),
  lwd=c(NA,NA,2,2,2),pch=c(20,21,NA,NA,NA),bty='n',cex=0.8)
```


Forecasts of new positives in October 2020 (including uncertainty)
● Daily number of new positives until 30th September (adjusted for weekday effect)
○ Daily number of new positives in October (adjusted for weekday effect)
Estimated number of new positives until 30th September
Forecasts with C[t*,h]=1.86 (optimal to predict new positives in the period 1−31 October)
Forecasts with the R−number
0
10000
30000
50000
70000
05/13/20
06/13/20
07/13/20
08/13/20
09/13/20
10/13/20
Date of notification

References

Gámiz, M. L., Janys, L., Martínez-Miranda, M. D. and Nielsen, J. P. (2013). Bandwidth selection in marker dependent kernel hazard estimation, *Computational Statistics & Data Analysis*, 68, 155-169.

Gámiz, M. L., Mammen, E., Martínez-Miranda, M. D., and Nielsen, J. P. (2025). Low quality exposure and point processes with a view to the first phase of a pandemic.

Koyama, S., Horie, T., Shinomoto, S. (2021). Estimating the time-varying reproduction number of COVID-19 with a state-space method. *PLoS computational biology*, 17(1), e1008679.