Reproducing Gámiz, Mammen, Martínez-Miranda, Nielsen, Scholz and Silva-Gómez (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, Nielsen, Scholz and Silva-Gómez: *Monitoring a developing pandemic with available data*. 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

Let $N_2(t)$ count the number of persons hospitalized in the interval $(0,t]$ and $N(t)$ the total number of patients that leave hospital in $(0,t]$, for $0 < t \leq T$. Then, $N(t) = N_3(t) + N_4(t)$, where $N_3(t)$ counts the number of patients that receive medical discharge and $N_4(t)$ the number of patients that die in hospital in the interval $(0,t]$. Thus, $N(dt)$ is the number of persons that leave the hospital (due to death or recovery) in the interval $(t, t+dt]$. Furthermore, we write $N(dt, ds)$ for the number of persons who entered the hospital in $(s, s+ds]$ and leave in $(t, t+dt]$ due to any cause, with $s < t \leq T$.

Let $\mathcal{F}(t)$ denote the σ-field generated by $\{(N_2(s), N(s)) : s < t\}$ and $\lambda(t)$ the intensity function of $N(t)$, that is,

$$\lambda(t) = \lim_{dt \to 0} \frac{P(N(t+dt) - N(t) \geq 1 \mid \mathcal{F}(t))}{dt}, \; 0 < t \leq T.$$

We assume that

$$\lambda(t) = \int_0^{t^-} \mu(t, t-s) S(t,s) N_2(ds),$$

where $\mu(t, t-s)$ is the hazard function for the duration time in the hospital for an individual that entered at time s and leaves at time t, and $S(t,s)$ is the survival function of duration-time-in-hospital for an individual that enters at s. In other words, $S(t,s)$ can be seen as the probability that an individual who enters at time s, still remains at time t, with $s < t$. The hazard function, $\mu(t, t-s)$ is assumed to have two dimensions: a one-dimensional marker (typically the notification date) and time (duration).

When we have individual follow-up, we can estimate the previous hazard function with no restriction on its functional form. Ideally, we observe the process $N(t, ds) = \sum_{i=1}^{n} N_{\cdot,i}(t, ds)$, where n is the observed number of individuals in the interval $(0, T]$ and $N_{\cdot,i}(t, ds)$ takes the value 1 if the individual i, hospitalized in $(s, s+ds]$, leaves the hospital at some time in the interval $(s, t]$. We will refer to this situation as "full information". However, in our motivating application we only observe the marginal processes $N_2(t)$ and $N(t)$.

In order to estimate the hazard function, let us assume that, we observe also the marginal processes $N_3(t)$ and $N_4(t)$. We denote $\mu_3(t, t-s)$ the recovery hazard function at time t for a subject that entered the hospital at

time s, and $\mu_4(t, t-s)$ the mortality hazard function at time t for a subject that entered the hospital at time s.

Let us denote $N(t) = N_3(t)+N_4(t)$ the total number of patients leaving the hospital at time t, and $\mu = \mu_3+\mu_4$, the hazard function for duration time in hospital, regardless the patient leaves the hospital due to death or clinical discharge. Let $Y(t, s)$ the total number of individuals that enter the hospital at time s and still remain hospitalized on the day t, with $0 < s < t \leq T$.

With all that, the local-linear estimator of μ can be written as

$$\hat{\mu}(t, t-s) = \frac{\int_{0\leq v<u\leq T} D(s,t,v,u)K_{1,b_1}(t-u)K_{2,b_2}(t-s-(u-v))N(du,dv)}{\int_{0\leq v<u\leq T} D(s,t,v,u)K_{1,b_1}(t-u)K_{2,b_2}(t-s-(u-v))Y(u,v)dvdu},$$

for $0 \leq v < u \leq T$.

In the case that full-information is not available, we observe only the marginal processes $N_2(t)$ and $N(t)$. Since the 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 processes, the estimator operates iteratively, following the iterative estimation schemed described in Subsection 4.3. in Gámiz et al. (2025b):

1. Use an initial guess $\hat{\mu}^{(0)}$ of μ.
2. The r-th iteration of the algorithm consists of two steps:
   - Construct a two-dimensional process $\hat{N}^{(r)}(t, ds)$ that approximates $N(t, ds)$ and a two dimensional process $\hat{Y}^{(r)}(t, s)$ that approximates $Y(t, s)$. This is done by using the estimator $\hat{\mu}^{(r)}$ from the previous iteration and by using the observed processes $N_2(t)$ and $N(t)$ as follows:

     $$\hat{N}^{(r)}(du, dv) = \frac{\hat{S}^{(r-1)}(u, u-v)\hat{\mu}^{(r-1)}(u, u-v)N_2(dv)}{\int_0^{u^-} \hat{S}^{(r-1)}(u, w)\hat{\mu}^{(r-1)}(u, u-w)N_2(dw)}N(du),$$

     and

     $$\hat{Y}^{(r)}(u, v)dvdu = \frac{\hat{S}^{(r-1)}(u, v)N_2(dv)}{\int_0^{u^-} \hat{S}^{(r-1)}(u, w)N_2(dw)}Y(u)du,$$

     where $\hat{S}^{(r-1)}(t, s) = \prod_{s<u\leq t}\{1 - \hat{\mu}^{(r-1)}(u, u-s)du\}$ is the estimated probability that a subject hospitalized at time s still remains in hospital at time t, and $Y(u) = N_2(u) - N(u)$ is the number of individuals remaining in hospital at time u.
   - The estimator of $N(du, dv)$ and $Y(u, v)$ is plugged into the expression of $\hat{\mu}(t, t-s)$, providing the following update of $\hat{\mu}^{(r-1)}$:

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

Finally, two types of outcome are considered. We estimate the transition rate from-hospitalized-to-recovery, $\mu_3(t, t-s)$, and the transition rate from-hospitalized-to-death, $\mu_4(t, t-s)$, respectively, as follows:

Let $\hat{\mu}$ be the solution of the above iterative procedure based of $\hat{N}$ and $\hat{Y}$ obtained at the last iteration. For $j = 3, 4$, let define

$$\hat{N}_j(du, dv) = \frac{\hat{S}(u, u-v)\hat{\mu}(u, u-v)N_2(dv)}{\int_0^{u^-} \hat{S}(u, u-v)\hat{\mu}(u, u-v)N_2(dw)}N_j(du).$$

Then, for $j = 3, 4$,

$$\hat{\mu}_j(t, t-s) = \frac{\int_{0\le v<u\le T} C(s,t,v,u)K_{1,b_1}(t-u)K_{2,b_2}(t-s-(u-v))\hat{N}_j(du,dv)}{\int_{0\le v<u\le T} C(s,t,v,u)K_{1,b_1}(t-u)K_{2,b_2}(t-s-(u-v))\hat{Y}(u,v)dvdu}.$$

General considerations when monitoring and forecasting in a dynamic environment

Modelling the dynamics of a pandemic

Our dynamic model is composed of four states: the number of infected individuals, the number of hospitalized patients, the number of recovered individuals, and the number of in-hospital deaths. The system is based on two types of transitions: a standard follow-up survival model, and another type in which the number of individuals involved is affected by dynamic definitions and potential underestimation. The latter includes the transitions from infected to infected and from infected to hospitalized, which depend on a partially known exposed population. (specifically, we are only referring to individuals who have tested positive, without making any assumptions about the total number of infections). Within the follow-up survival transition type, we distinguish two transitions: from hospitalized to recovered, and from hospitalized to deceased.

In addition, we include a fifth transition, from infected to deceased, which refers to the total number of deaths in the population due to the pandemic, for which additional information is required. In order to relate in-hospital and out-of-hospital deaths, we can define a ratio of the number of deaths outside the hospital to the number of deaths inside the hospital on a given day $t > 0$,

$$F(t) = \frac{N_{\text{out}(t)}}{N_{\text{in}(t)}},$$

being $N_{\text{out}(t)}$ y $N_{\text{in}(t)}$ the number of deaths outside and inside the hospital, respectively, for $t > 0$.

Thus, we can estimate the previous ratio using the ratio of two non-parametric regression estimator. On the numerator, the smoothed regression on time of the number of deaths outside the hospital, and on the denominator, the smoothed regression on time of the number of deaths inside hospital.

Principles of forecasting in a dynamic environment

Regarding the infection process, in Gámiz et al. (2025), introduced an infection indicator representing where the future, $(T, T+h]$ for $h > 0$, will be different or equal to the immediate past. Assume that we have the estimation of the dynamic infection rate, $\hat{\mu}_1(t/T, u)$, for $0 < u \le t \le t^*$, 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 $u \ge 0$ and $0 < s \le h$, and assuming that the infection rate at the end of the forecasting period equals to the most recent estimate multiplied by a number $C_{1,h}$, we obtain:

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

for $u \ge 0$ and $0 < s \le h$.

A suitable value of $C_{1,h}$ suggested makes our method capable of forecasting the number of new infections. From those, we can predict the number of hospitalized to thus forecasting of number of deaths in hospital as well as the number of recoveries.

There is also another indicator, $C_{2,h}$ to predict the total number of deaths, both inside and outside the hospital, based on forecasted number of infected provided by the previous infection indicator and the proposed dynamic model. For any t in the interval $(0, T]$, we consider the estimator, $\hat{F}(t)$ of the ratio $F(t)$. If we fix the forecast horizon at time $T + h$, then we need to extrapolate the ratio $F(T + s)$, for $0 < s \le h$.

We assume that the ratio F at the end of the forecasting period is the more recent estimate multiplied by $C_{2,h}$, and it varies linearly in between

$$\tilde{F}(T+s) = \hat{F}(T) \times \left(1 + (C_{2,h} - 1)\frac{s}{h}\right),$$

for $0 < s \leq h$.

Forecasting can be performed by setting $C_{1,h} = C_{2,h} = 1$. Alternatively, information from expert knowledge can motivate the use of a value of $C_{1,h}$ and $C_{2,h}$ different from one.

On the other hand, if we want to assess uncertainty in our forecasts, we propose a novel bootstrap method to quantify uncertainty by constructing prediction bands for fixed C_1 and C_2 values. Asssume we observe $N_{1,0}, N_{1,1}, \ldots, N_{1,T}$, where $N_{1,i}$ is the number of infections at the ith day ($i = 1, \ldots, T$). The bootstrap algorithm consists of the following steps: (1) from a given initial number of infections, we predict new infections in the upcoming days; (2) from the generated infections we predict new daily hospitalizations; (3) from the generated hospitalizations we predict new deaths in hospital and recoveries. For further details see Subsection 5.3. in Gámiz et al. (2025b).

(1) Predicting infections:

1.1. 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 $N_{1,0}, \ldots, N_{1,i-1}$.

1.2. Define $\beta = (\gamma - 1)/(k(k-1))$, and $\alpha = 1 - k\beta$.

1.3. Given $N^*_{1,0}, N^*_{1,1}, \ldots, N^*_{1,i-1}$, generate a bootstrap sample with $N^*_{1,\alpha,i} \to Pois(\alpha\lambda^*_{1,i})$, and $N^*_{1,\beta,i} \to Pois(\beta\lambda^*_{1,i})$, being $\lambda^*_{1,i} = N^*_{1,0}\hat{\mu}_1(i/T, i) + N^*_{1,1}\hat{\mu}_1(i/T, i-1) + \ldots + N_{1,i-1}\hat{\mu}_1(i/T, 1)$ the expected number of infections at the ith day given $N_{1,0}, N^*_{1,1}, \ldots, N^*_{1,i-1}$. Define $N^*_{1,i} = N^*_{1,\alpha,i} + kN^*_{1,\beta,i}$.

(2) Predicting hospitalizations:

2.1. Given $N^*_{1,0}, N^*_{1,1}, \ldots, N^*_{1,i-1}$, generate a bootstrap sample with $N^*_{2,i} \to Pois(\lambda^*_{2,i})$, with $\lambda^*_{2,i} = N^*_{1,1}\hat{\mu}_2(i/T, i) + N^*_{1,2}\hat{\mu}_2(i/T, i-1) + \ldots + N_{1,i}\hat{\mu}_2(i/T, 1)$.

(3) Predicting outcome of hospital:

3.1 Given $N^*_{2,0}, N^*_{2,1}, \ldots, N^*_{2,i}$, generate $N^*_{3,i}$ and $N^*_{4,i}$ as follows:

i. Define the expected number of patients who recover on day i after having spent a total of d days in hospital as $\lambda^*_{3,i}(d) = N^*_{2,i-d+1}S(i/T, i-d+1)\hat{\mu}_3(i/T, d)$, where

$$S(i/T, i-d+1) = \prod_{j=1}^{d-1}[1 - \hat{\mu}((i-j)/T, j)], \quad d = 1, \ldots, i,$$

with $\hat{\mu} = \hat{\mu}_3 + \hat{\mu}_4$, and assuming $S(i/T, i) = 1$. Generate $N^*_{3,i,d} \longrightarrow Pois(\lambda^*_{3,i}(d))$.
Set $N^*_{3,i} = \sum_{d=1}^{i} N^*_{3,i,d}$.

ii. Define the expected number of patients who die in hospital after a stay of d days as $\lambda^*_{4,i}(d) = N^*_{2,i-d+1}S(i/T, i-d+1)\hat{\mu}_4(i/T, d)$. Generate $N^*_{4,i,d} \longrightarrow Pois(\lambda^*_{4,i}(d))$.
Set $N^*_{4,i} = \sum_{d=1}^{i} N^*_{4,i,d}$.

Repeating the resembling scheme 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.

Figure 2: hazard rate of time spent in hospital for individuals entering the hospital at different dates

In order to reproduce the hazard rate estimation of the time spent in hospital for those individuals who enter the hospital at different dates, we can use the function `hazard2Dmiss`. This function implements the local linear estimator of the marker-dependent hazard in the case of missing-survival-link data (Gámiz et al., 2025b). Hazard is assumed having two dimensions: a one-dimensional marker and time.

This function is based on the local linear estimator of the marker-dependent hazard introduced by Nielsen (1998), also implemented in the `pandemics` package through the `hazard2D` function. This marker-dependent hazard local linear estimator assumes that full information is available in the form of occurrences and exposures. However, in an emerging or developing pandemic, this estimator is likely to be infeasible with the data available. Therefore, we propose the algorithm described in Subsection 4.3. in Gámiz et al. (2025b) to attempt to reconstruct the hidden processes and make use of the `hazard2dmiss` function.

To estimate the hazard rate, the function requires several 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`.
- A vector of length `M` with the daily number of hospitalizations each day, `Ei.z`.
- A vector of length `M` with the daily number of deaths each day, `Oi1.z`.
- A vector of length `M` with the daily number of recoveries each day, `Oi2.z`.

In addition, the function `hazard2Dmiss` requires a pair of bandwidths of the two-dimensional local linear estimator (see Eq. 13 in Gámiz et al. (2025b)). These two bandwidths can be specified manually or estimated using the cross-validation method described in Gámiz et al. (2013). 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. Thus, to select the bandwidth estimates through cross-validation, it is enough to provide a grid of bandwidths using the argument `bs.grid` and set `cv=TRUE`, since by default it is set to `FALSE`.

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 the hazard 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. For a brief discussion of this issue in practice, see *Reproducing Gámiz, Mammen, Martínez-Miranda and Nielsen (2025) using the* `pandemics` *package*. Regarding this last point, we note that, depending on the number of data points, the selection of bandwidths can take quite a long time, even several hours. For this reason, in this document, all estimations are performed by directly providing a pair of bandwidths.

Finally, the function `hazard2Dmiss` also allows including, as arguments, a value `epsilon` representing the tolerance in the iterative algorithm, and an integer value `max.ite` specifying the maximum number of iterations. By default, these values are set as `epsilon = 1e-4` and `max.ite = 50`.

As output, three hazard estimates are produced: one for deaths, one for recoveries, and another considering hospital discharge regardless of the cause (deaths plus recoveries). Each hazard estimate takes the form of a matrix, where each row corresponds to the grid points for the marker dimension, and each column represents the duration from the time a patient is admitted to the hospital until discharge, either due to recovery or death. In addition, the output also 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.

To estimate the marker-dependent hazard when the final event is either death or recovery, we consider hospitalized individuals as the exposure, and the occurrences as either deaths or recoveries, depending on which hazard function we aim to estimate. It is common to observe, in several COVID datasets, a large variation in the number of reported cases depending on the day of the week. This is particularly evident in

the French COVID dataset. 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 order to correct the reporting delay in confirming cases. In what follows, we will always apply Koyama's correction to our data.

```
library(pandemics)
```

```
data('covid')
Oi1.z<-diff(covid$Death)   # deaths
Oi2.z<-diff(covid$Recov)   # recoveries
M<-length(Oi1.z)
Ei.z<-covid$Hospi[-(M+1)]   # exposure

Oi1.z<-week_effect(Oi1.z)
Oi2.z<-week_effect(Oi2.z)
Ei.z<-week_effect(Ei.z)
```

When using the function `hazard2Dmiss`, we recommend always starting with the full dataset from the very first day and, once the hazard has been estimated, trimming the estimation to the desired period. The reason for this is that it is not possible to analyze a period of the pandemic without accounting for its history; in other words, we need to know what has happened previously in order to understand its future development. To reproduce this specific example, we will keep the entire dataset.

Once the data period has been defined and Koyama's correction applied, the key step in estimating the hazard is the selection of the two bandwidths required to estimate the hazard function. In this case, given the sample size, we directly specify two bandwidth values for the hazard estimation. However, for a first initial approximation of the pair of optimal bandwidths, we recommend using the cross-validation method at least once. After generating the grid points for both the time dimension and the marker dimension, we can call the `hazard2Dmiss` function and store the two hazard estimates of interest for this example: that corresponding to deaths and that corresponding to recoveries.

```
t.grid<-z.grid<-1:M
bs <- t(c(150,150))
res.h<-hazard2Dmiss(t.grid,z.grid,Oi1.z,Oi2.z,Ei.z,bs.grid=bs,cv=FALSE)
hi1.zt<-res.h$hi1.zt ## 2D-hazard of deaths
hi2.zt<-res.h$hi2.zt ## 2D-hazard of recoveries
```

Once the hazard function estimates for recoveries and/or deaths has been saved, we can plot the hazard of death and/or recovery on several selected dates. In this example, we choose April 30th, June 30th, September 30th, December 31th, March 31th, 2021 and June 30th, 2021 as reference dates, although these can be modified as needed, along with the number of days considered from hospital admission. For this particular case, we set the time window to 35 days after hospital admission.

```
## Plot of hazard of deaths on several dates
ddates<-covid$Date
z1<-c(44,105,197,289,379,470)
zdates<-ddates[z1] ; nz<-length(z1)
t.min<-35
ti<-1:t.min ; n0<-length(ti)
```

We present the estimated hazard rate from hospital admission to death (left panel) and from admission to recovery (right panel). Hazard estimations corresponding to several selected reporting dates are shown in different colours.

```r
par(mfrow = c(1, 2))

yy<-range(hi2.zt[z1,1:n0],na.rm=TRUE)
yy[2]<-yy[2]+.02
plot(ti,hi2.zt[z1[nz],1:n0], lwd=2,main='Recoveries',type='l',
    xlab='Time from admission (days)',ylab='Hazard',ylim=yy)
for(i in 2:nz) lines(ti,hi2.zt[z1[i-1],1:n0],lwd=3,col=i,lty=i)
legend('topright',c('Date of admision',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)

yy<-range(hi1.zt[z1,1:n0],na.rm=TRUE)
yy[1]<-yy[1]-.0003;yy[2]<-yy[2]+.005
plot(ti,hi1.zt[z1[nz],1:n0], lwd=2,main='Deaths',type='l',
    xlab='Time from admission (days)',ylab='Hazard',ylim=yy)
for(i in 2:nz) lines(ti,hi1.zt[z1[i-1],1:n0],lwd=3,col=i,lty=i)
legend('topright',c('Date of admision',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)
```

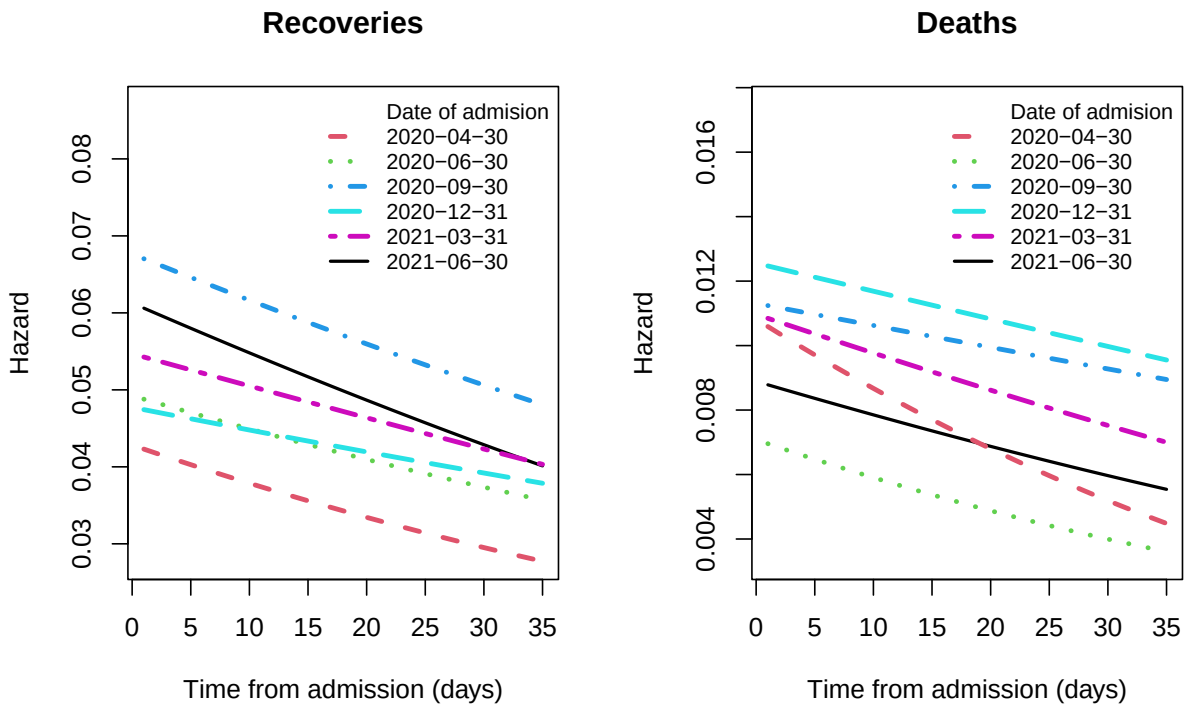


Figure 3: hazard rate of time since admission until death for individuals entering the hospital at different dates

With the `covidAges` dataset, similar plots to the previous two can be created. It is enough to define the exposure and the variable that will take the place of occurrences, in this case by age group. We generate the estimated hazard rate of time from admission until death for individuals admitted to the hospital on different dates, grouped by age. However, a similar code will provide the hazard rate of time since admission until recovery. It is only necessary to take into account the matrix of the estimated hazard corresponding to the process of interest.

As before, we start by defining the exposure and the occurrences for each age group. These are the number

of daily hospitalizations and the number of hospital deaths. Note that, as in the previous case, since the number of deaths is cumulative, differentiating the occurrence vector results in the loss of one day. Therefore, we use all hospitalization data except for the last day.

Once the exposure and occurrences are defined, we estimate the hazard using the function `hazard2Dmiss`, providing the grid points for the time and marker dimensions (`t.grid` and `z.grid`). As before, since this is a very large dataset, we do not estimate the bandwidths through cross-validation; instead, we directly use the same bandwidths as those used for the `covid` dataset.

After estimating the hazard, we plot it for different calendar times. As before, we choose April 30th, June 30th, September 30th, December 31st, March 31st, 2021, and June 30th, 2021 as reference dates, and we set the time window to 35 days after hospital admission.

```
data('covidAges')

## Ages (0, 40)
Oi1.z<-week_effect(diff(covidAges$Death_0_39))   # deaths
Oi2.z<-week_effect(diff(covidAges$Recov_0_39))   # recoveries
M<-length(Oi1.z)
Ei.z<-week_effect(covidAges$Hospi_0_39[-(M+1)])   # exposure for survival analysis

## Hazard estimate with a fixed bandwidth
t.grid<-z.grid<-1:M
bs<-t(c(150,150))
res.h_0_39<-hazard2Dmiss(t.grid,z.grid,Oi1.z,Oi2.z,Ei.z,
                         bs.grid=bs,cv=FALSE)
hi1.zt_0_39<-res.h_0_39$hi1.zt ## 2D-hazard of deaths


## Ages (40, 60)
Oi1.z<-week_effect(diff(covidAges$Death_40_59))   # deaths
Oi2.z<-week_effect(diff(covidAges$Recov_40_59))   # recoveries
M<-length(Oi1.z)
Ei.z<-week_effect(covidAges$Hospi_40_59[-(M+1)])    # exposure for survival analysis

## Hazard estimate with a fixed bandwidth
t.grid<-z.grid<-1:M
bs<-t(c(150,150))
res.h_40_59<-hazard2Dmiss(t.grid,z.grid,Oi1.z,Oi2.z,Ei.z,
                          bs.grid=bs,cv=FALSE)
hi1.zt_40_59<-res.h_40_59$hi1.zt ## 2D-hazard of deaths


## Ages (60, 80)
Oi1.z<-week_effect(diff(covidAges$Death_60_79))  # deaths
Oi2.z<-week_effect(diff(covidAges$Recov_60_79))   # recoveries
M<-length(Oi1.z)
Ei.z<-week_effect(covidAges$Hospi_60_79[-(M+1)])    # exposure for survival analysis

## Hazard estimate with a fixed bandwidth
t.grid<-z.grid<-1:M
bs<-t(c(150,150))
res.h_60_79<-hazard2Dmiss(t.grid,z.grid,Oi1.z,Oi2.z,Ei.z,
                          bs.grid=bs,cv=FALSE)
hi1.zt_60_79<-res.h_60_79$hi1.zt ## 2D-hazard of deaths
```

```r
## Ages (80, --)
Oi1.z<-week_effect(diff(covidAges$Death_80_plus))   # deaths
Oi2.z<-week_effect(diff(covidAges$Recov_80_plus))   # recoveries
M<-length(Oi1.z)
Ei.z<-week_effect(covidAges$Hospi_80_plus[-(M+1)])    # exposure for survival analysis

## Hazard estimate with a fixed bandwidth
t.grid<-z.grid<-1:M
bs<-t(c(150,150))
res.h_80_plus<-hazard2Dmiss(t.grid,z.grid,Oi1.z,Oi2.z,Ei.z,
                            bs.grid=bs,cv=FALSE)
hi1.zt_80_plus<-res.h_80_plus$hi1.zt ## 2D-hazard of deaths


## Plot of hazard of deaths on several dates
ddates<-covidAges$Date
z1<-c(44,105,197,289,379,470)
zdates<-ddates[z1] ; nz<-length(z1)
t.min<-35
ti<-1:t.min ; n0<-length(ti)

par(mfrow = c(2, 2))  # Set 2x2 plot layout

yy <- c(-0.001,0.02)
plot(ti,hi1.zt_0_39[z1[nz],1:n0], main='Deaths ages (--,40)',type='l',
     xlab='Time from admission (days)',ylab='Hazard',ylim=yy,lwd=2)
for(i in 2:nz) lines(ti,hi1.zt_0_39[z1[i-1],1:n0],lwd=3,col=i,lty=i)
legend('topright',c('Date of admision', 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')

plot(ti,hi1.zt_40_59[z1[nz],1:n0], main='Deaths ages [40,60)',type='l',
     xlab='Time from admission (days)',ylab='Hazard',ylim=yy,lwd=2)
for(i in 2:nz) lines(ti,hi1.zt_40_59[z1[i-1],1:n0],lwd=3,col=i,lty=i)

plot(ti,hi1.zt_60_79[z1[nz],1:n0], main='Deaths ages [60,80)',type='l',
     xlab='Time from admission (days)',ylab='Hazard',ylim=yy,lwd=2)
for(i in 2:nz) lines(ti,hi1.zt_60_79[z1[i-1],1:n0],lwd=3,col=i,lty=i)

plot(ti,hi1.zt_80_plus[z1[nz],1:n0], main='Deaths ages [80,--)',type='l',
     xlab='Time from admission (days)',ylab='Hazard',ylim=yy,lwd=2)
for(i in 2:nz) lines(ti,hi1.zt_80_plus[z1[i-1],1:n0],lwd=3,col=i,lty=i)
```

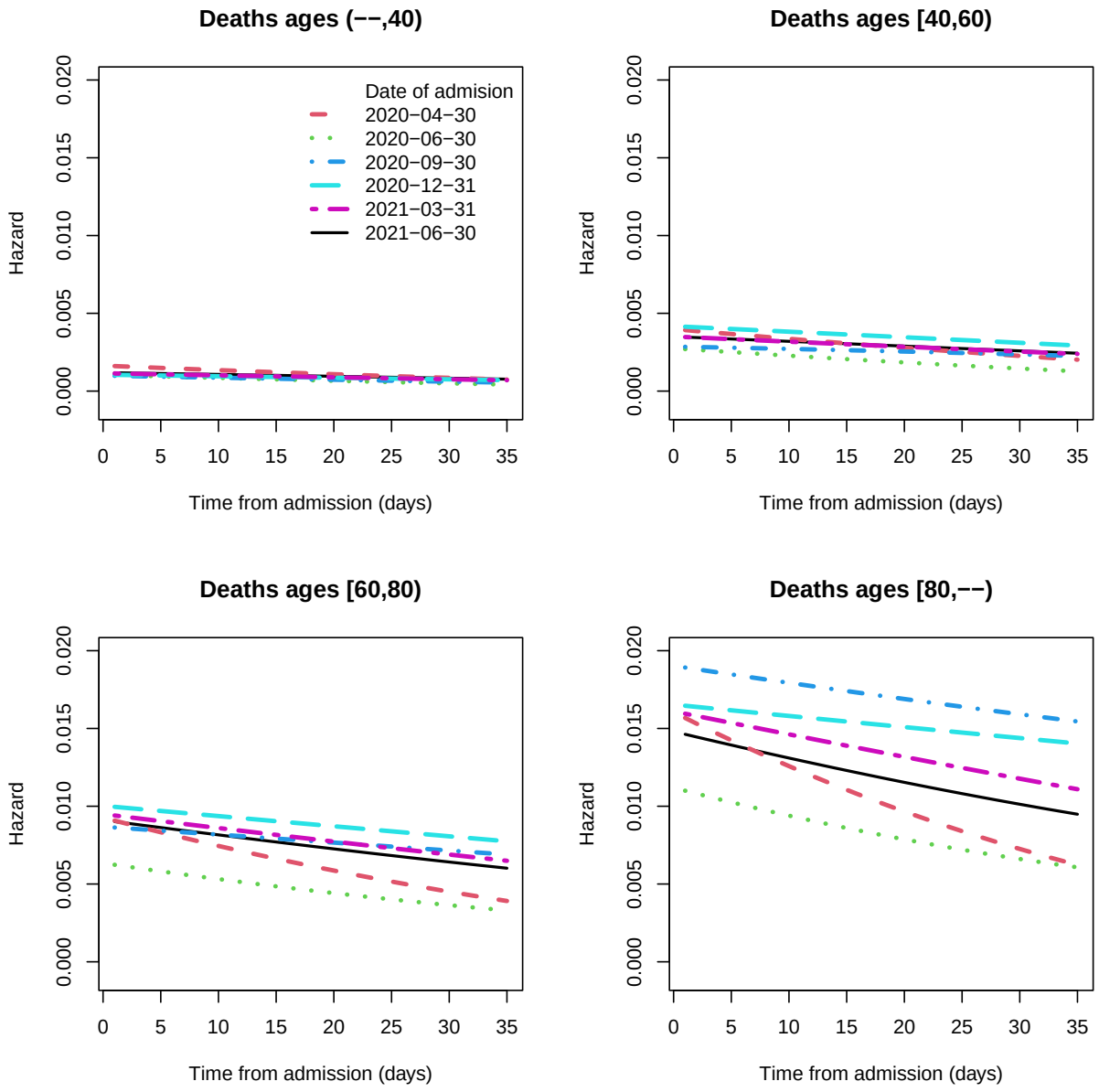


Figure 4: median of the time spent in hospital by date of admission

The hazard rate estimator described previously provide sufficient information to estimate additional indicators. In this case, we propose calculating the median time from hospital admission to exit, either due to recovery or death, for a given patient depending on the calendar time.

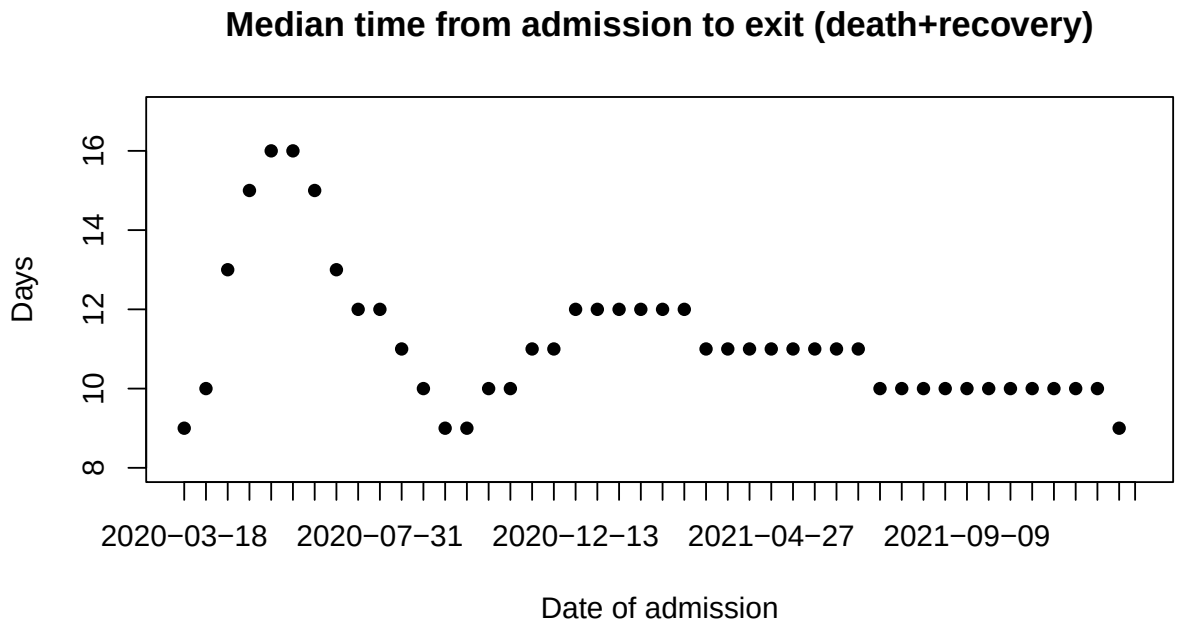
The median hospital stay can be obtained using the `medtime` function from the `pandemics` package. This function requires two input arguments: a matrix (`M` × `M`) containing the estimated hazard of death plus recovery (as provided by the `hazard2Dmiss` function), and a vector of indices between 1 and `M` specifying the admission days at which the median is to be evaluated. In the present example, given the selected time period, we consider admission days spaced every 15 units. This choice, however, can be adapted to the time frame in order to enhance visualization. If no index vector is provided, the function defaults to a sequence from 1 to `M-1` in steps of 2. The output of the `medtime` function is a vector containing the estimated median times for each specified admission day.

We again consider the number of hospitalized individuals as the exposure, and the number of recoveries and deaths (adjusted for the weekday effect) as the occurrences. We compute the estimated hazard using the `hazard2Dmiss` function, retaining only the matrix with the estimated hazard of death plus recovery evaluated over the grid of time points.

```
hi.zt<-res.h$hi.zt
```

Once the hazard for the two possible hospital discharge outcomes has been estimated, we call the `medtime` function to estimate the median time from admission to discharge for any cause.

```
z1<-c(seq(1,M-1,by=15),M)
nz<-length(z1)
res<-medtime(hi.zt,z1)

plot(z1,res,ylab='Days',xaxt = "n",type='p',pch=16,
  xlim=range(z1), ylim=c(8,17),xlab='Date of admission',
  main='Median time from admission to exit (death+recovery)')
axis(1,at=z1,labels=ddates[z1],cex=1)
```



It is also possible to perform a similar analysis using the `covidAges` dataset. Below, we reproduce a plot showing the median hospital stay for each of the four age groups: 0-39, 40-59, 60-79, and 80 years and older. The only difference compared to the previous plot lies in the data used; with regard to the code, the procedure is the same.

```
plot(z1,res,ylab='Days',xaxt = "n",type='p',pch=16,
  xlim=range(z1), ylim=c(4,21),xlab='Date of admission',
  main='Median time by age groups')
axis(1,at=z1,labels=ddates[z1],cex=1)

hi.zt_0_39<-res.h_0_39$hi.zt
res_0_39<-medtime(hi.zt_0_39,z1)
text(z1, res_0_39, labels = 1, col = 5)
```

```
hi.zt_40_59<-res.h_40_59$hi.zt
res_40_59<-medtime(hi.zt_40_59,z1)
text(z1, res_40_59, labels = 2, col = 4)

hi.zt_60_79<-res.h_60_79$hi.zt
res_60_79<-medtime(hi.zt_60_79,z1)
text(z1, res_60_79, labels = 3, col = 3)

hi.zt_80_plus<-res.h_80_plus$hi.zt
res_80_plus<-medtime(hi.zt_80_plus,z1)
text(z1, res_80_plus, labels = 4, col = 2)

lg <- legend("topright",
       legend = c("(--,40)", "[40,60)", "[60,80)", "[80,--)", "All ages"),
       bty = "n", cex = 0.6)
text(lg$rect$left, lg$text$y[1:4], labels = c(1,2,3,4), col = 5:2, adj = 1)
points(lg$rect$left, lg$text$y[5], pch = 16, col = "black")
```

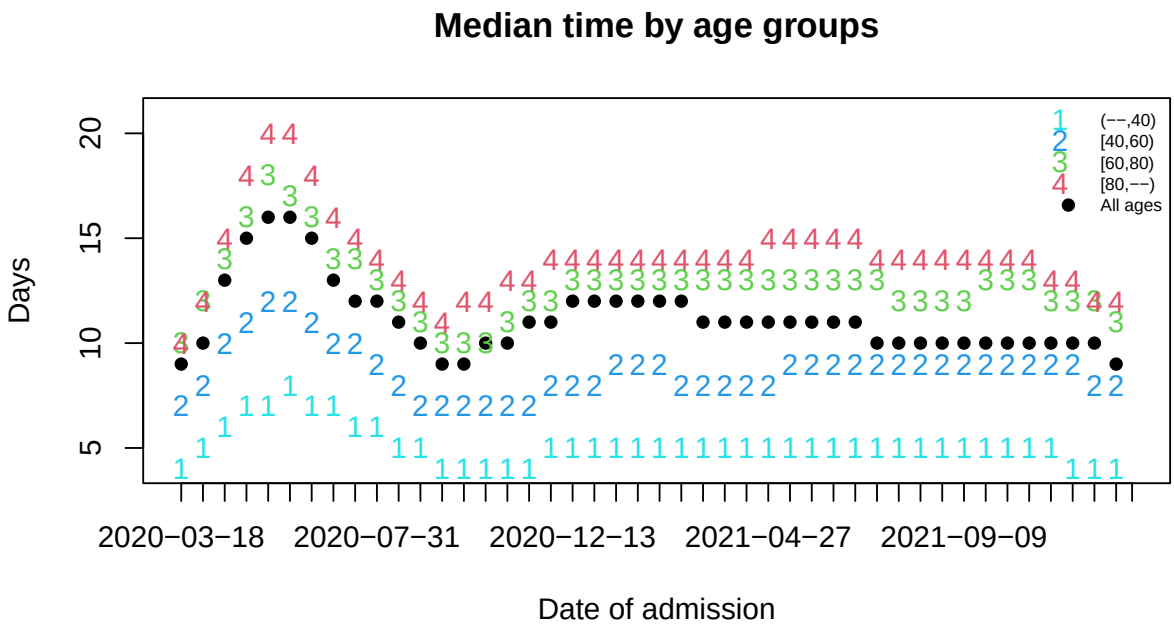


Figure 5: probability of outcome by cause specific

From the two-dimensional estimated hazard of deaths and recoveries, we can compute the probability that a person, who has been in hospital for a number of days, leaves the hospital alive or death, depending on the date of admission. For that, the key function is `poutcome`, that can estimate the probability of survival or death.

To use this function, we need as input the estimated matrices of the hazard for deaths and recoveries (`hi1.zt` and `hi2.zt`). As mentioned previously, these estimates are obtained as output from the `hazard2Dmiss` function. Additionally, the function optionally accepts a vector of indices indicating the admission days at which to evaluate the probabilities. If it is missing, the function defaults to a sequence from `1` to `M-1` in steps

of 2. The output of the `poutcome` function consists of two matrices containing the computed probabilities of leaving or dying in the hospital: `alive.zt` and `death.zt`, respectively.

Once the hazards for deaths and recoveries have been estimated, we can, for example, set the admission dates to April 30th, June 30th, September 30th, and December 31st, 2020, as well as March 30th and June 30th, 2021. Along with these dates, we also specify the number of days elapsed since hospital admission for which we want to observe the progression of the probabilities. In this case, as before, we set 35 days. Once all of this is selected, we can use the `poutcome` function, storing the matrices of the computed probabilities of leaving or dying in the hospital. Finally, we present the estimated probabilities of being discharged alive (left panel) and of dying in the hospital (right panel).

```
z1<-c(44,105,197,289,379,470)
zdates<-ddates[z1] ; nz<-length(z1)
t.min<-35
ti<-1:t.min ; n0<-length(ti)

res<-poutcome(hi1.zt,hi2.zt,z1)
alive.zt<-res$alive.zt
death.zt <- res$death.zt

par(mfrow = c(1, 2))

plot(ti,alive.zt[1:n0,nz], lwd=2,main='Probability to get out alive',type='l',
    xlab='Time from admission (days)',ylab="Probability",ylim=c(0,1))
for(i in 2:nz) lines(ti,alive.zt[1:n0,i-1],lwd=3,col=i,lty=i)
legend('bottomleft',c('Date of admision',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)

plot(ti,death.zt[1:n0,nz], lwd=2,main='Probability of dying in the hospital',type='l',
    xlab='Time from admission (days)',ylab="Probability",ylim=c(0,1))
for(i in 2:nz) lines(ti,death.zt[1:n0,i-1],lwd=3,col=i,lty=i)
legend('topleft',c('Date of admision',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)
```

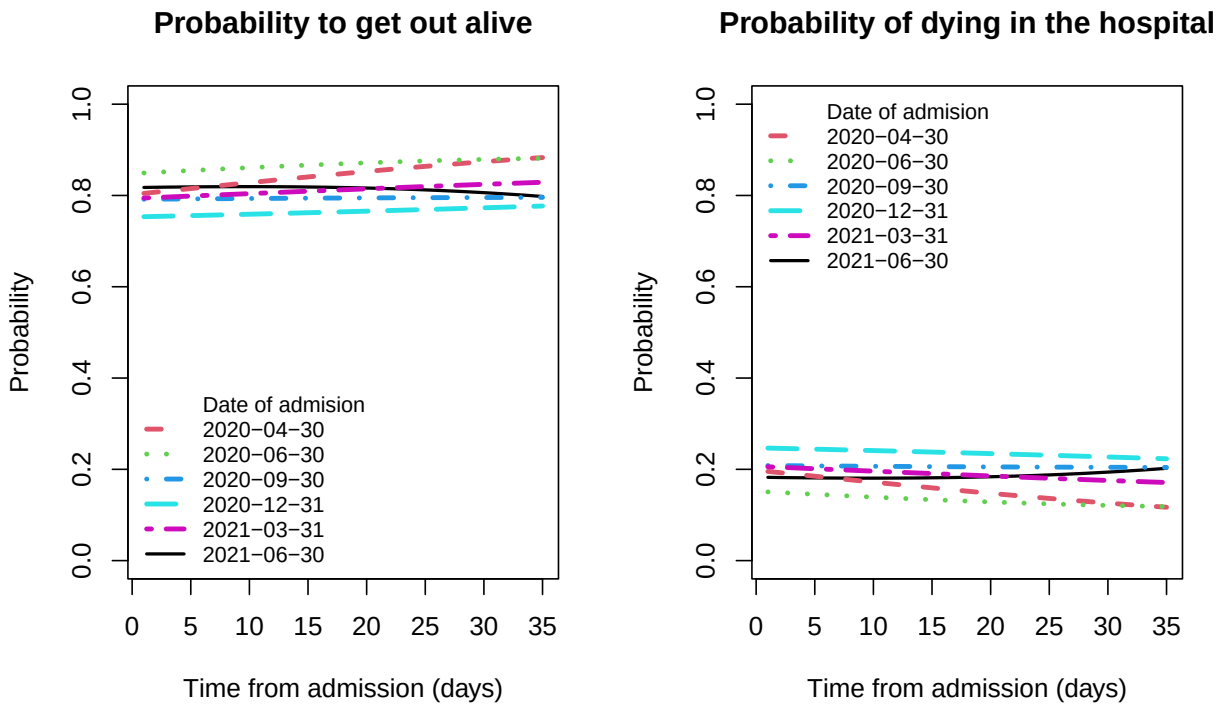


Figure 6: probability of leaving the hospital due to recovery by age group

Using the `covidAges` dataset, similar plots to the previous ones can be generated. We calculate in this section the probability of leaving the hospital due to recovery, grouped by age. It is only necessary to consider the hazard estimates for each age group and apply the `poutcome` function, as discussed in the previous section

```
z1<-c(44,105,197,289,379,470)
zdates<-ddates[z1] ; nz<-length(z1)
t.min<-35
ti<-1:t.min ; n0<-length(ti)

hi1.zt_0_39<-res.h_0_39$hi1.zt # 2D-hazard of deaths
hi2.zt_0_39<-res.h_0_39$hi2.zt # 2D-hazard of recoveries
res_0_39<-poutcome(hi1.zt_0_39,hi2.zt_0_39,z1)
alive.zt_0_39<-res_0_39$alive.zt

hi1.zt_40_59<-res.h_40_59$hi1.zt # 2D-hazard of deaths
hi2.zt_40_59<-res.h_40_59$hi2.zt # 2D-hazard of recoveries
res_40_59<-poutcome(hi1.zt_40_59,hi2.zt_40_59,z1)
alive.zt_40_59<-res_40_59$alive.zt

hi1.zt_60_79<-res.h_60_79$hi1.zt # 2D-hazard of deaths
hi2.zt_60_79<-res.h_60_79$hi2.zt # 2D-hazard of recoveries
res_60_79<-poutcome(hi1.zt_60_79,hi2.zt_60_79,z1)
alive.zt_60_79<-res_60_79$alive.zt

hi1.zt_80_plus<-res.h_80_plus$hi1.zt # 2D-hazard of deaths
hi2.zt_80_plus<-res.h_80_plus$hi2.zt # 2D-hazard of recoveries
res_80_plus<-poutcome(hi1.zt_80_plus,hi2.zt_80_plus,z1)
```

```r
alive.zt_80_plus<-res_80_plus$alive.zt

par(mfrow = c(2, 2))  # Set 2x2 plot layout

plot(ti,alive.zt_0_39[1:n0,nz],ylim=c(0,1),lwd=2,type='l',
     main='Probability to get out alive, ages (--,40)',
     ylab='Probability',xlab='Time from admission (days)')
for(i in 2:nz) lines(ti,alive.zt_0_39[1:n0,i-1],lwd=3,col=i,lty=i)
legend('bottomleft',legend=zdates,lty=c(2:nz,1),
       lwd=c(rep(3,nz-1),2),col=c(2:nz,1),bty='n',cex=0.8)

plot(ti,alive.zt_40_59[1:n0,nz],ylim=c(0,1),lwd=2,type='l',
     main='Probability to get out alive, ages [40,60)',
     ylab='Probability',xlab='Time from admission (days)')
for(i in 2:nz) lines(ti,alive.zt_40_59[1:n0,i-1],lwd=3,col=i,lty=i)

plot(ti,alive.zt_60_79[1:n0,nz],ylim=c(0,1),lwd=2,type='l',
     main='Probability to get out alive, ages [60,80)',
     ylab='Probability',xlab='Time from admission (days)')
for(i in 2:nz) lines(ti,alive.zt_60_79[1:n0,i-1],lwd=3,col=i,lty=i)

plot(ti,alive.zt_80_plus[1:n0,nz],ylim=c(0,1),lwd=2,type='l',
     main='Probability to get out alive, ages [80,-)',
     ylab='Probability',xlab='Time from admission (days)')
for(i in 2:nz) lines(ti,alive.zt_80_plus[1:n0,i-1],lwd=3,col=i,lty=i)
```

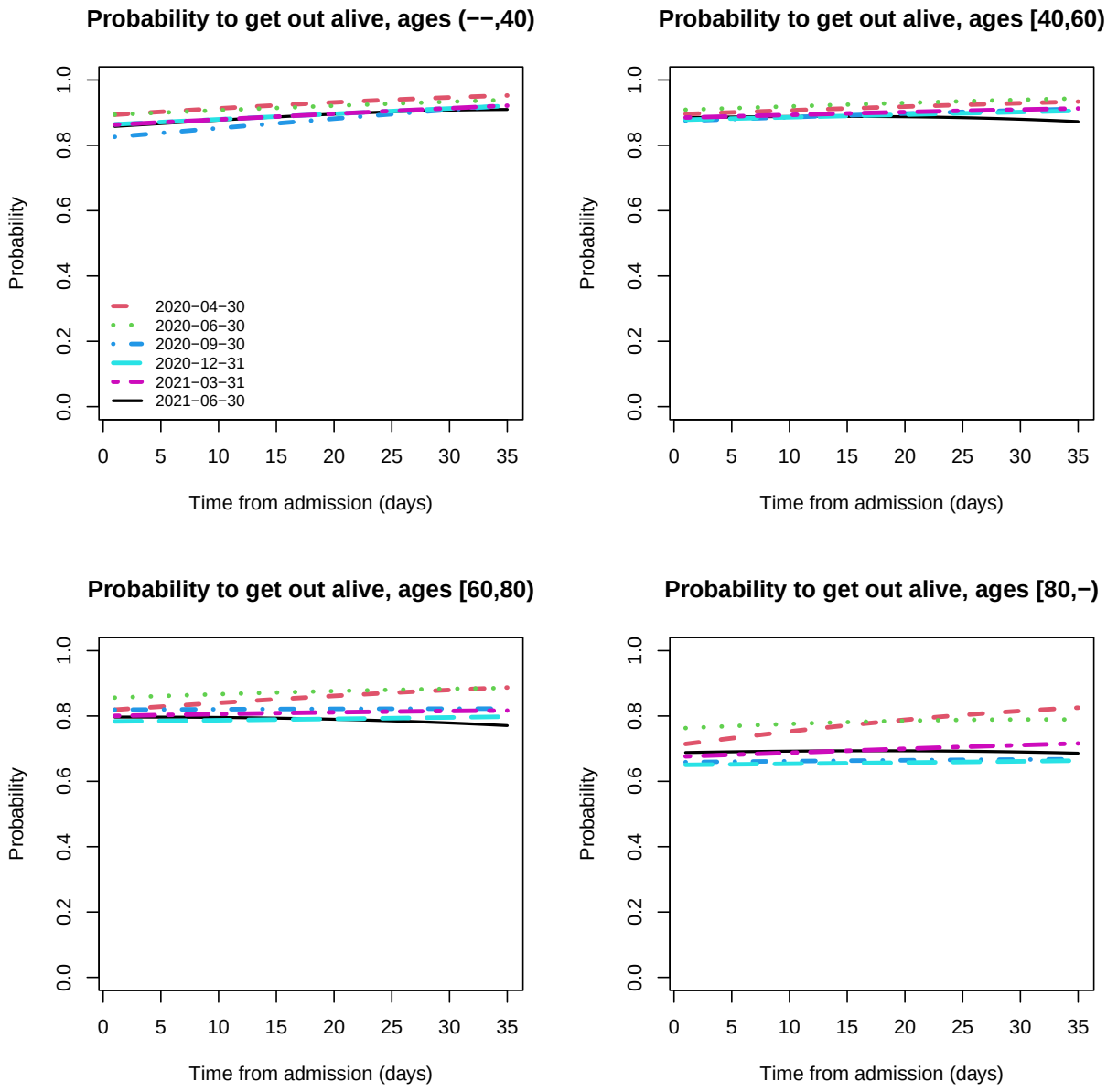


Figure 7: probability of leaving the hospital due to death by age group

As before, we can calculate the probability of leaving the hospital due to death, grouped by age. It is only necessary to consider the hazard estimates for each age group and apply the `poutcome` function. In this case, since we want to calculate the probability of leaving the hospital due to death, from the output of `poutcome`, we store the matrix corresponding to deaths.

```
z1<-c(44,105,197,289,379,470)
zdates<-ddates[z1] ; nz<-length(z1)
t.min<-35
ti<-1:t.min ; n0<-length(ti)

death.zt_0_39<-res_0_39$death.zt
death.zt_40_59<-res_40_59$death.zt
```

```r
death.zt_60_79<-res_60_79$death.zt
death.zt_80_plus<-res_80_plus$death.zt

par(mfrow = c(2, 2))  # Set 2x2 plot layout

plot(ti,death.zt_0_39[1:n0,nz],ylim=c(0,1),lwd=2,type='l',
     main='Probability of dying, ages (--,40)',
     ylab='Probability',xlab='Time from admission (days)')
for(i in 2:nz) lines(ti,death.zt_0_39[1:n0,i-1],lwd=3,col=i,lty=i)
legend('topleft',legend=zdates,lty=c(2:nz,1),
       lwd=c(rep(3,nz-1),2),col=c(2:nz,1),bty='n',cex=0.8)

plot(ti,death.zt_40_59[1:n0,nz],ylim=c(0,1),lwd=2,type='l',
     main='Probability of dying, ages [40,60)',
     ylab='Probability',xlab='Time from admission (days)')
for(i in 2:nz) lines(ti,death.zt_40_59[1:n0,i-1],lwd=3,col=i,lty=i)

plot(ti,death.zt_60_79[1:n0,nz],ylim=c(0,1),lwd=2,type='l',
     main='Probability of dying, ages [60,80)',
     ylab='Probability',xlab='Time from admission (days)')
for(i in 2:nz) lines(ti,death.zt_60_79[1:n0,i-1],lwd=3,col=i,lty=i)

plot(ti,death.zt_80_plus[1:n0,nz],ylim=c(0,1),lwd=2,type='l',
     main='Probability of dying, ages [80,-)',
     ylab='Probability',xlab='Time from admission (days)')
for(i in 2:nz) lines(ti,death.zt_80_plus[1:n0,i-1],lwd=3,col=i,lty=i)
```

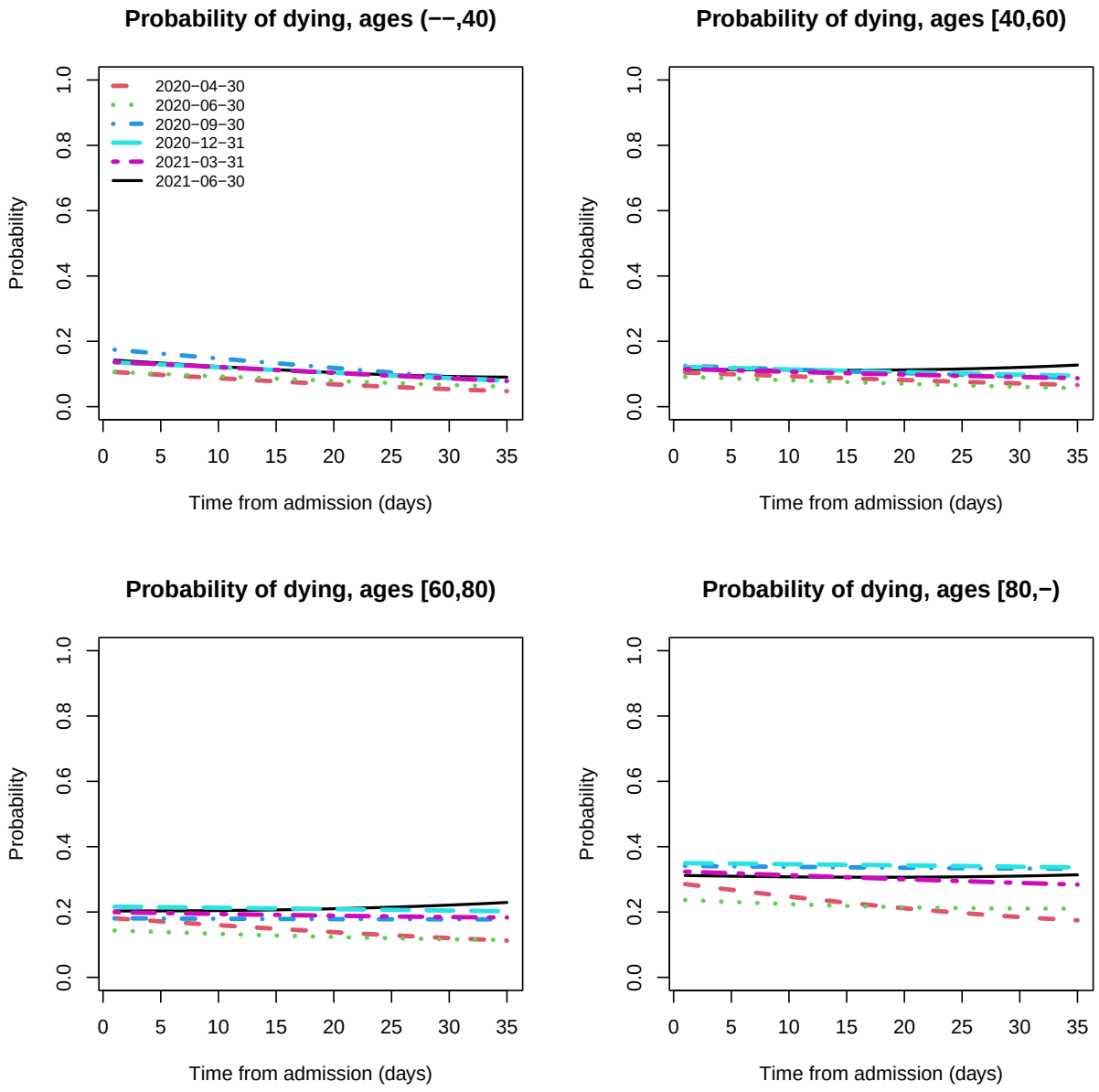


Figure 8: estimated ratio of number of deaths (inside and outside the hospital)

To estimate the ratio between deaths occurring outside and inside hospitals, it is necessary to have data on deaths outside the hospital setting. For this purpose, we use the database of medicalized nursing homes in France, which is available in our GitHub repository: https://github.com/germansilva-gomez/pandemics/

The estimation begins on April 1, which is the first date for which out-of-hospital death records are available, and ends on December 31, 2020. Although the dataset includes additional variables, only the numbers of out-of-hospital and in-hospital deaths are required to calculate the ratio.

```
library(locpol)
ehpad<-ehpad_25A21[16:290,]
```

If we look at our data, we find that there are some missing values, both for in-hospital and out-of-hospital deaths. To address this issue, we approximate these missing values using linear interpolation. Once these

values are estimated, we apply Koyama's correction to the in-hospital and out-of-hospital deaths. It is important to note that the death counts are reported cumulatively; therefore, before correcting for the weekend reporting delay, the values must first be differenced.

```
## Deaths inside hospital
Oi.in<-ehpad$deces;
M.in<-length(Oi.in)

ii<-which(is.na(Oi.in))
x.in<-1:M.in
Oi.in[ii]<-round(approx(x=x.in[-ii],y=Oi.in[-ii],xout=ii)$y)
Oi.in<-week_effect(diff(Oi.in)) ; Oi.in[Oi.in<0]<-0
M.in<-M.in-1
x.in<-1:M.in

## Deaths outside hospital
Oi.out<-ehpad$deces_ehpad
M.out<-length(Oi.out)
ii<-which(is.na(Oi.out))

x.out<-1:M.out
Oi.out[ii]<-round(approx(x=x.out[-ii],y=Oi.out[-ii],xout=ii)$y)
Oi.out<-week_effect(diff(Oi.out));Oi.out[Oi.out<0]<-0
M.out<-M.out-1
```

Finally, since our ratio is defined as the smoothed ratio between the number of out-of-hospital deaths and the smoothed number of in-hospital deaths, we apply a local linear smoother with an optimal bandwidth selected via cross-validation. We use the function `regCVBwSelC` included in the package `locpol` to implement cross-validation bandwidth selector and the function `locLinSmootherC` to compute the local estimation. Naturally, alternative smoothing methods as well as different bandwidth selection strategies could also be valid in this context.

```
x.eval<-1:M.in
b.in<-regCVBwSelC(x=x.in,y=Oi.in,deg=1,interval=c(20,M.in),kernel=EpaK)
m.in<-locLinSmootherC(x=x.in,y=Oi.in,xeval=x.eval,bw=b.in,kernel=EpaK)$beta0

x.out<-1:M.in
b.out<-regCVBwSelC(x=x.out,y=Oi.out,deg=1,interval=c(20,M.in),kernel=EpaK)
m.out<-locLinSmootherC(x=x.out,y=Oi.out,xeval=x.eval,bw=b.out,kernel=gaussK)$beta0
```

Finally, we calculate two ratios: the raw ratio, composed of the data on deaths both outside and inside the hospital, and a smooth ratio based on the smoothed counts of deaths. Once the graph is plotted, the dots represent the daily ratio of observed deaths outside the hospital to observed deaths inside the hospital. The black line shows the estimated smooth ratio.

```
ratio.s<-(m.out/m.in) # Smooth ratio
ratio<-Oi.out/Oi.in   # Raw ratio

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

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

```
ddates<-ehpad$date
ddates <- format(as.Date(ddates), "%d/%m/%y")
at<-c(1,31,62,92,123,154,184,215,245,275)
at.lab<-ddates[at]
```

```r
M<-length(ratio)
plot(1:M,ratio,pch=20,col='darkgray',
     main='Ratio: deaths-outside-hospital / deaths-at-hospital',
     xaxt='n',ylab='Ratio',xlab='Date of notification', ylim=c(0,3))
lines(1:M,ratio.s,lwd=2)
axis(1,at=at,labels=at.lab)
abline(h=1,lwd=2,lty=3,col=2)
```

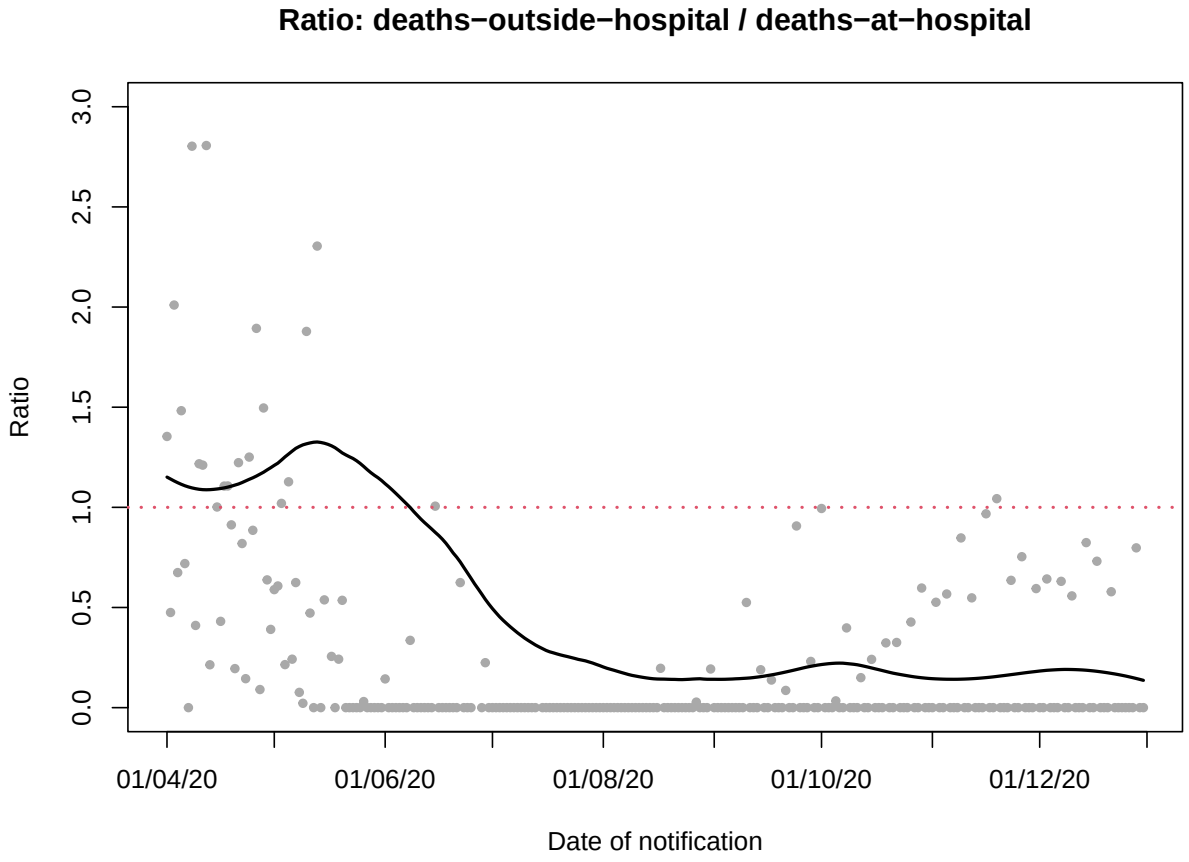


Figure 9: forecasts of total number of deaths (inside and outside the hospital) in October 2020

Finally, we propose reproducing the total number of deaths, both inside and outside hospitals. In general, we refer readers to Section 5 in Gámiz et al. (2025b) for further details in relation to forecasting. For this purpose, we use data from the `ehpad_25A21` dataset, which includes the number of deaths occurring in hospitals as well as those taking place outside them. This dataset can be found in the following GitHub repository: https://github.com/germansilva-gomez/pandemics/

To facilitate the implementation of our method for forecasting the total number of deaths, two functions are provided in the GitHub repository: `ratio_death` and `forecasting_all_deaths`. The first function implements the code used to reproduce Figure 8 of this document, offering users a simpler way to carry out the necessary steps. Specifically, it takes as input two vectors, `death.in` and `death.out`, which represent the cumulative number of deaths inside and outside hospitals, respectively. As output, the function returns the number of deaths inside and outside hospitals after applying Koyama's correction and performing linear interpolation when missing values (NAs) are present (`Oi.in` and `Oi.out`); the smoothed densities of deaths

inside and outside hospitals (`m.in` and `m.out`); the raw ratio (`ratio`); and the smoothed ratio (`ratio.s`). To use this function, the `locpol` package must be loaded, or installed if it is being used for the first time.

On the other hand, the `forecasting_all_deaths` function allows generating forecasts for a giving forecasting horizon. These forecasts are determined through the two dynamic indicators, $C_{1,h}$ and $C_{2,h}$, typically provided by experts, to determine whether the near future is expected to behave similarly or differently from the recent past. An appropriate choice of $C_{1,h}$ is strongly related to the reported reproduction number. For further details and its relationship with the well-known reproduction number, see Subsection 7.1 of Gámiz et al. (2025a). In addition, regarding the survival indicator, the data can be adjusted according to prior knowledge about the distribution of deaths inside and outside hospitals.

In addition to the two indicators, the `forecasting_all_deaths` function requires the forecasting period, `period`, and the observed data of the process. For the latter, since we aim to present a forecasting exercise for the total number of deaths, all variables of interest are required. Generating forecasts for the daily number of in-hospital deaths requires extrapolating the infection, hospitalization, and death rates. Specifically, the infection rate is estimated from the daily number of new infections; the hospitalization rate from both the daily new infections and hospitalizations; and, as noted at the beginning of this document, the hazard of death from the daily number of hospitalized patients, recoveries, and deaths. For all these reasons, in this case the `forecasting_all_deaths` function requires as arguments:

- A numeric value `Cval1` with the infection indicator.
- A numeric value `Cval2` with the survival indicator.
- An integer value `period` with the number of days to forecast.
- A vector `Pz` with the observed number of new infections each day.
- A vector `newHz` with the daily number of new hospitalizations each day.
- A vector `Hz` containing the observed total number of patients currently hospitalized each day.
- A vector `Dz` with the daily number of deaths.
- A vector `Rz` with the number of recoveries occurring within the hospital each day.
- A matrix `RoInf` with the estimated infection rate.
- A matrix `RoHosp` with the estimated hospitalization rate.
- A matrix `RoRec` containing the estimated hazard of recoveries.
- A matrix `RoDeath` containing the estimated hazard of deaths.

The rate estimates can be obtained using the `rate2Dmiss` function included in this package. For further details on the use of this function within the `pandemics` package, we refer the reader to the vignette *Reproducing Gámiz, Mammen, Martínez-Miranda and Nielsen (2025) using the* `pandemics` *package.*

With regard to the assessment of uncertainty, the `forecasting_all_deaths` function optionally allows implementing the bootstrap algorithm. This option can be enabled using the Boolean argument `boot`, which is set to `FALSE` by default, meaning that uncertainty is not computed unless specified. Broadly speaking, the function first generates bootstrap samples according to the steps described in the section *General considerations when monitoring and forecasting in a dynamic environment* 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 modified using the `seed` argument. Subsequently, for each sample, forecasts are computed with the chosen values of the optimal indicators and period. 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 output, the function produces a list containing three vectors. The first corresponds to the forecasts obtained with $C_{1,h} = C_{2,h} = 1$ (`Dz.pred_11`); the second provides the forecasts for the total number of

deaths using the optimal infection indicator and $C_{2,h} = 1$ (`Dz.pred_C1`); and the third contains the forecasts obtained with the optimal infection and survival indicators (`Dz.pred_CC`). If `boot = TRUE`, two additional vectors are included, representing the lower and upper limits of the 95% prediction bands, `Dz.in.out.PI.lwr` and `Dz.in.out.PI.upr`, respectively.

We propose forecasting the total number of deaths in October 2020 using data from France, covering the period from May 13 to September 30, 2020 (141 days), as no testing data are available prior to May 13, 2020. We begin by defining the variables of interest and computing the daily number of new hospitalizations. Specifically, the number of new hospitalizations on day `i+1` can be calculated as `newHi[i+1] = Hi[i+1] - newHi[i] - Di[i] - Ri[i]`, where `newHi` denotes the number of newly hospitalized patients, `Hi` the total number of hospitalized individuals, `Di` the total number of deaths, and `Ri` the total number of recoveries. It is important to note that both recoveries and deaths must be expressed as daily counts, not cumulative values as they are reported in the `covid` dataset.

We remove the first 56 rows and the last 3 rows from the counts of infected individuals. Since the newly hospitalized, deceased and recovered cases come from differenced vectors, we remove only the first 55 and the last 3 observations from these. Once the period has been defined, we apply Koyama's correction method to all variables in order to adjust for reporting delays caused by weekends. In the French dataset, this issue is particularly pronounced in the daily number of new infections, and to a lesser extent in the other variables.

```
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
## We apply data adjustment for variations by day
Pi <- week_effect(covid$Posit[-c(1:56, 656:658)])
newHi <- week_effect(newHi)[-c(1:55, 655:657)]
Hi <- week_effect(Hi)[-c(1:56, 656:658)]
Di <- week_effect(Di)[-c(1:55, 655:657)]
Ri <- week_effect(Ri)[-c(1:55, 655:657)]
```

For this forecasting exercise, the absence of infection data before May 13 may lead to a loss of information when estimating the hazard of death. This is because, in the model we propose, we monitor the evolution of a developing pandemic using all available data from the very beginning. Selecting only a subset of the data discards valuable information that is important for understanding the course of the pandemic.

However, to forecast deaths, as discussed in previous paragraphs, we need to extrapolate the infection, hospitalization, and death rates, all of which must have the same number of data points. Consequently, since we have no infection data prior to May 13, deaths must also be considered from that date onward. This causes the resulting rates to fail to capture the full development of the pandemic during its early days, although they later reach stability. This effect is particularly evident when plotting the fitted values for deaths, as during the initial days the fitted values are very close to zero and underestimate the observed ones.

To partially mitigate this loss of information, if desired, we propose defining May 12 as a special reference day: all new hospitalizations, deaths, and recoveries that occurred before this date are aggregated and recorded as taking place on May 12. Since no infection data are available for that day, we assign an `NA` value to infections on May 12. Note also that the number of new hospitalizations on May 12 coincides with the total number of hospitalizations reported on that date. To estimate the infection and hospitalization rates, we remove the initial `NA` element in the new infections data. After estimating the death and recovery hazards using the `hazard2Dmiss` function, we can remove the first row and the last column of each matrix so that the resulting

hazard matrices have dimension `M`. Although this adjustment does not fully recover the lost information, it helps mitigate the issue. With another dataset containing complete data from the very beginning of the pandemic, this discrepancy would likely not be noticeable.

Once the data are prepared, we can proceed to estimate the matrices of infection and hospitalization rates, as well as the hazard matrices for deaths and recoveries, obtained by evaluating the `rate2Dmiss` and `hazard2Dmiss` functions. To estimate the infection and hospitalization rates, we use the variables for new infections (`Pi`) and new daily hospitalizations (`newHi`). Note that the infection rate is returned as an `(M-1)`$\times$`(M-1)` matrix, while the hospitalization rate is returned as an `M`$\times$`M` matrix, with `M=141` in our example. Bandwidths in both the rates and hazards can be estimated using the cross-validation method described in Gámiz et al. (2013). However, in our application we directly specify a pair of bandwidths for each rate.

```
Ms <- 141

## 1.1. Infection rate
Ei.z<-Pi[1:Ms]
delay<-1;Msd<-Ms-delay
Oi.z<-Ei.z[-(1:delay)]; Ei.z1<-Ei.z[1:Msd];

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

## 1.2. Hospitalization rate
Ei.z1<-Pi[1:Ms]
Oi.z<-newHi[1:Ms]
t.grid<-z.grid<-1:Ms
bs<-t(c(5,10))
RHosp<-rate2Dmiss(t.grid=t.grid,z.grid=z.grid,Oi.z=Oi.z,Ei.z1=Ei.z1,
                  bs.grid=bs,cv=FALSE)
RoHosp<-RHosp$hi.zt
RoHosp <- RoHosp

## 1.3. Hazards of deaths and recoveries
Oi1.z<-Di[1:Ms]
Oi2.z<-Ri[1:Ms]
Ei.z<-Hi[1:Ms]
t.grid<-z.grid<-1:Ms

bs <- t(c(150,150))
res.h<-hazard2Dmiss(t.grid,z.grid,Oi1.z,Oi2.z,Ei.z,
                    bs.grid=bs,cv=FALSE)
RoDeath<-res.h$hi1.zt
RoRec<-res.h$hi2.zt
```

When working with deaths occurring inside and outside hospitals, we start by considering the period from April 1, 2020, to September 30, 2020. The reason for choosing this starting point is that, in this dataset, there are no available observations prior to April 1 for deaths occurring outside hospitals, and we always work with the entire period available. However, when plotting the graph, we display it starting from May 13. We call the `ratio_death` function with the cumulative number of deaths inside and outside hospitals from the dataset, and we keep only the smoothed ratio as output. In fact, we are only interested in the estimated ratio on the last day, September 30, which is why we store that value.

```
ehpad<-ehpad_25A21[16:198,]
deaths_in.out <- ratio_death(ehpad$deces,ehpad$deces_ehpad)
ratio.s_deaths_in.out <- deaths_in.out$ratio.s
last.ratio <- ratio.s_deaths_in.out[length(ratio.s_deaths_in.out)]
```

As stated before, we proposed two numerical indicators. The first, labeled as the infection indicator, $C_{1,h}$, refers to whether the future will differ from or resemble the immediate past when forecasting the number of new infections over a given horizon. To estimate this value, we construct a grid of possible indicator values and identify the one that minimizes the mean squared error with respect to the actual values observed in the forecasting period. The grid we designed ranges from 0.01 to 4, with 100 candidate values within this interval. In this example, the estimated value was 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. It should be noted, however, that in practice this value cannot be computed, since future infection data are never available. Therefore, the information provided by the reported reproduction number should be incorporated.

We proposed a second indicator, $C_{2,h}$ to forecast the total number of deaths, both inside and outside the hospital, based on the forecasted number of infections provided by $C_{1,h}$ and the dynamic model. In order to estimate this survival indicator, it is necessary to have the observed number of in-hospital deaths, the number of out-of-hospital deaths, and the estimated number of in-hospital deaths with an appropiate choice of the infection indicator. Based on this, we proposed a grid of candidate values for the survival indicator, specifically 200 values ranging from 4.01 to 8. In this least-square minimization problem, both the observed total number of deaths and the forecasted number of deaths (inside and outside the hospital) were included. The forecasts of out-of-hospital deaths was computed as a projection of the predicted in-hospital deaths, multiplied by the different sequences generated with the values from the grid. Moreover, these sequences incorporated the smoothed ratio between out-of-hospital and in-hospital deaths as of September 30, estimated at 0.2158769. With all this, the estimated value of the indicator was $C_{2,h} = 6.817035$. Obviously, this value of the survival indicator may vary if the grid of possible values is modified. As in the case of the infection indicator, the code and functions required to compute and reproduce the value of the survival indicator are available in the GitHub repository.

Finally, we call the `forecasting_all_deaths` function using this last value and specify the argument `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, `Dz.in.out.PI.lwr` and `Dz.in.out.PI.upr`, respectively.

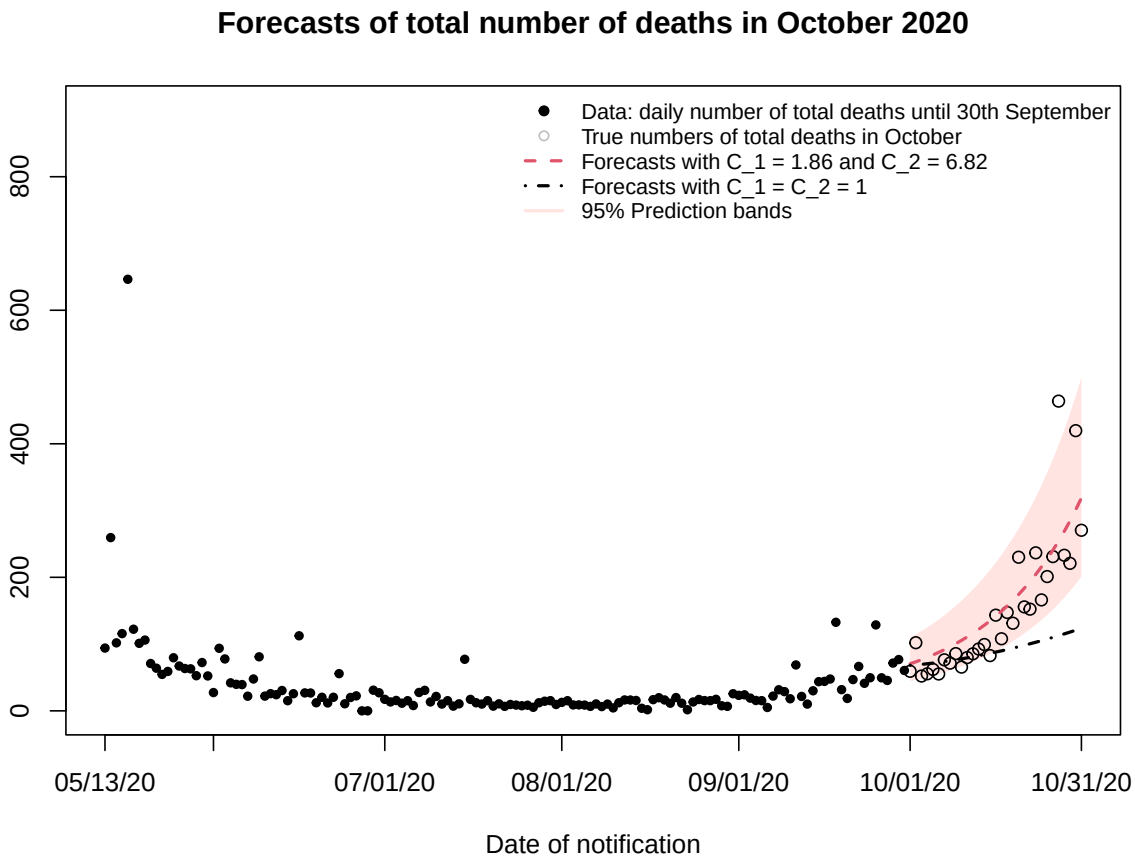
```
Cval1 <- 1.86
Cval2 <- 6.82
period <- 31
fore_deaths <- forecasting_all_deaths(Cval1=Cval1,Cval2=Cval2,period=period,RoInf=RoInf,RoHosp=RoHosp,
                                      RoRec=RoRec,RoDeath=RoDeath,Pz=Pi[1:Ms],newHz=newHi[1:Ms],
                                      Hz=Hi[1:Ms],Rz=Ri[1:Ms],Dz=Di[1:Ms],last.ratio = last.ratio,
                                      boot=TRUE,B=500,seed=1)
Dz.pred_1 <- fore_deaths$Dz.pred_11
Dz.pred_CC <- fore_deaths$Dz.pred_CC
Dz.in.out.PI.lwr <- fore_deaths$Dz.in.out.PI.lwr
Dz.in.out.PI.upr <- fore_deaths$Dz.in.out.PI.upr
```

We display two types of points: black dots represent observed data up to September, while white dots correspond to the number of deaths recorded in October. We also display two types of lines: the black dotted-dashed line represents the forecasts obtained with $C_{1,h} = C_{2,h} = 1$, and the red-dashed line reflects the forecasts using the optimal infection indicator $C_{1,h} = 1.86$ and the optimal survival indicator of $C_{2,h} = 6.82$.

```
Dz.true <- deaths_in.out$Oi.in + deaths_in.out$Oi.out

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")

plot(1:Ms,Dz.true[-c(1:41)],ylab='',xlab='Date of notification',
     main=paste('Forecasts of total number of deaths in October 2020',sep=''),
     pch=20,ylim=c(0,900),xaxt='n',xlim=c(1,Ms+period))
oat<-c(1,20,50,81,112,142,172)
olab<-ddates[oat]
axis(1,at=oat,labels=olab)

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

polygon(c(x21,x22,x21[1]),c(y21,y22,y21[1]),col='mistyrose',border=F)

ehpad<-ehpad_25A21[16:229,]
ratio.oct <- ratio_death(ehpad$deces,ehpad$deces_ehpad)
Oi.in <- ratio.oct$Oi.in
Oi.out <- ratio.oct$Oi.out
Dz.true.oct <- Oi.in + Oi.out
M1 <- 182
points((Ms+1):(Ms+period),Dz.true.oct[(M1+1):(M1+period)],col=1,pch=1)

lines((Ms+1):(Ms+period),Dz.pred_CC,col=2,lty=2,lwd=2)
lines((Ms+1):(Ms+period),Dz.pred_1,col=1,lty=4,lwd=2)

legend('topright',c('Data: daily number of total deaths until 30th September',
                    'True numbers of total deaths in October',
                    paste0('Forecasts with C_1 = ',Cval1,' and C_2 = ',Cval2),
                    paste0('Forecasts with C_1 = C_2 = 1',sep=''),
                    paste0('95% Prediction bands')),
       col=c(1,'grey',2,1,'mistyrose'),lty=c(NA,NA,2,4,1),lwd=c(NA,NA,2,2,2),
       pch=c(19,1,NA,NA,NA),bty='n',cex=0.8)
```



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. (2025a). Low quality exposure and point processes with a view to the first phase of a pandemic. *arXiv preprint arXiv:2308.09918*

Gámiz, M. L., Mammen, E., Martínez-Miranda, M. D., Nielsen, J. P., Scholz, M., and Silva-Gómez, G. E. (2025b). Monitoring a developing pandemic with available data. *arXiv preprint arXiv:2308.09919*

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.

Nielsen, J. P. (1998). Marker dependent kernel estimation from local linear estimation. *Scandinavian Actuarial Journal*, 2, 113-124.