Stochastic epidemic models and their link with methods from survival analysis

Hein Putter[1, 2], Jelle Goeman[1], Jacco Wallinga[1, 3]

[1]Leiden University Medical Center
[2]Mathematical Institute, Leiden University
[3]National Institute for Public Health and the Environment

2024-02-18

Abstract

Compartmental models based on ordinary differential equations (ODE's) quantifying the interactions between susceptible, infectious, and recovered individuals within a population have played an important role in infectious disease modeling. The aim of the present paper is to explain the link between stochastic epidemic models based on the susceptible-infectious-recovered (SIR) model, and methods from survival analysis. We illustrate how standard software for survival analysis in the statistical language R can be used to estimate pivotal parameters in the stochastic SIR model in the very much idealized situation where the epidemic is completely observed. Extensions incorporating interventions, age structure and heterogeneity are explored and illustrated.

1 Introduction

Mathematical modeling has played a pivotal role in understanding and managing the dynamics of infectious diseases. In particular, models based on ordinary differential equations (ODE's) quantifying the interactions between susceptible, infectious, and recovered individuals within a population, the SIR model (Kermack and McKendrick 1927) and its extensions have had enormous impact. These models allow public health professionals to predict the course of an outbreak and assess the impact of various interventions. Moreover, mathematical modeling provides a valuable tool for scenario planning, allowing for the simulation of different outbreak scenarios and the evaluation of their respective outcomes. We refer to Anderson and May (1991) and Diekmann, Heesterbeek, and Britton (2013) for more background on the usefulness and the mathematical properties of ODE models for infectious disease modeling.

Many researchers have pointed to the link with survival analysis (Becker 1989; Becker and Britton 1999; Kenah 2011, 2013, 2015; KhudaBuksh et al. 2019), however, an accessible overview with standard software is lacking. The present paper has two purposes. The first is to explain the link between infectious disease models, in particular the susceptible-infectious-recovered (SIR) model, and methods from survival analysis in a concise and self-contained way. The second is to illustrate how standard software for survival analysis in the statistical language R can be used to estimate pivotal parameters in the infectious disease model.

We concentrate on the very much idealized situation where the epidemic is completely observed, i.e., at each time point we have complete and correct information on the number of susceptible, infected and recovered individuals. The ideas in this manuscript are largely known; the purpose is to establish further connections with *standard* models used in survival analysis, like Cox's proportional hazards

model, the additive hazard model, and Poisson generalized linear models (GLM's), and to illustrate using R code how standard software for survival analysis can be used for estimating pivotal quantities in the SIR model, specifically the transmission parameter. We also show how methods from survival analysis may be used to cover extensions incorporating interventions, age structure and heterogeneity.

2 SIR model

2.1 Deterministic SIR model

The SIR model is a system of differential equations, describing the evolution over time of an infectious disease. Defining $\overline{S}(t)$, $\overline{I}(t)$ and $\overline{R}(t)$ to be the proportion of susceptibles, infectious and recovered individuals in a closed population, the system of ordinary differential equations (ODE) is defined as (Anderson and May 1991)

$$\begin{aligned} \frac{\mathrm{d}\overline{S}}{\mathrm{d}t} &= -\beta\overline{S}(t)\overline{I}(t), \\ \frac{\mathrm{d}\overline{I}}{\mathrm{d}t} &= \beta\overline{S}(t)\overline{I}(t) - \gamma\overline{I}(t), \\ \frac{\mathrm{d}\overline{R}}{\mathrm{d}t} &= \gamma\overline{I}(t). \end{aligned} \tag{1}$$

The idea behind Equation 1 is that interactions between susceptible and infectious individuals leading to a new infection occur with a rate quantified by an transmission parameter β. In case of an infection the proportion of susceptibles decreases and the proportion of infectious individuals increases by the same amount. Infectious individuals recover with a rate quantified by the recovery parameter γ. This class of models is often referred to as *compartimental models*, since it describes the interactions of units from different compartments. Apart from describing the spread of an infection through a population, compartimental models have been used in many other fields of application. Among the many examples we mention models describing the interaction of HIV and CD4+ T-cells within humans (Ho et al. 1995), predator-prey models in biology (Beddington, Free, and Lawton 1975) and economics (Murdoch, Briggs, and Nisbet 2013), and pharmacodynamics (Donnet and Samson 2013). Many extensions of this simple SIR model have been developed, with the aim of coming closer to a description of reality, but the simple SIR model has proved to be remarkably robust, and we will stick with the simple SIR model throughout.

We can numerically solve the ODE system using the {deSolve} package.

```
library(deSolve)
parameters <- c(beta = 2, gamma = 0.5) # parameter values
state <- c(S = 0.999, I = 0.001, R = 0) # starting values for the system
# Definition of the system of ODE's
SIR_ODE <- function(time, state, parameters) {
  with(as.list(c(state, parameters)), {
  # rate of change
  dS <- -beta * S * I
  dI <- beta * S * I - gamma * I
  dR <- gamma * I
  # return the rate of change
  list(c(dS, dI, dR))
  }) # end with(as.list ...
```

```
}
# Now run the system for some time, using the ode() function of {deSolve}
times <- seq(0, 12, by = 0.01)
out <- ode(y = state, times = times, func = SIR_ODE, parms = parameters)
```

Figure 1 shows the development of the epidemic over time, in the form of a stacked plot. The lowest curve shows the proportion of infected individuals. The distance between the lowest curve and the one directly above that represents the proportion of recovered individuals. The sum of these two is the proportion of individuals that have got infected over time (recovered or not). The remainder is the proportion of susceptible individuals.

```
out <- as.data.frame(out)
plot(out$time, out$I, type="l", lwd=2, xlab="Time", ylab="", ylim=c(0, 1), col="red")
lines(out$time, out$I + out$R, type="l", lwd=2)
lines(out$time, out$I + out$R + out$S, type="l", lwd=2, col="blue")
legend(12, 0.8, c("I", "I + R", "I + R + S"), lwd=2,
       col=c("red", "black", "blue"), bty="n", xjust=1)
```

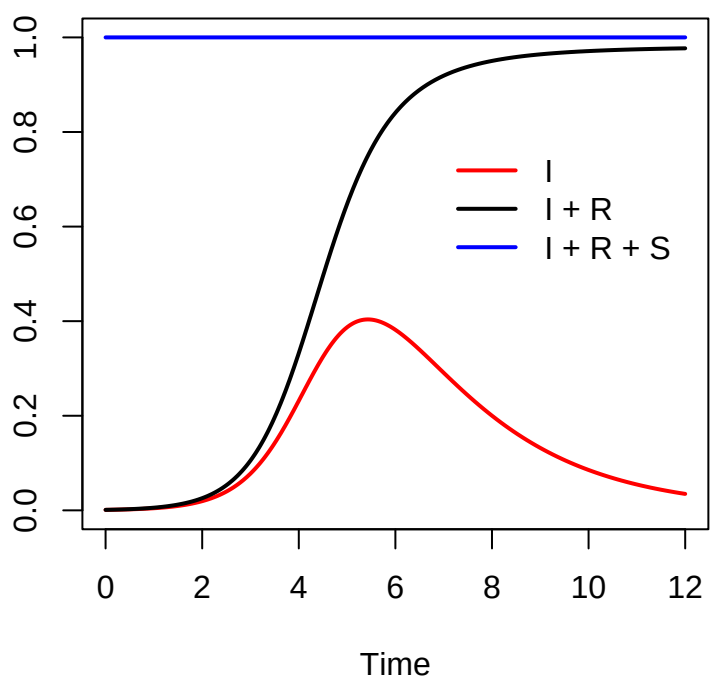


Figure 1: Stacked plot showing the proportion of susceptible, infected and recovered individuals over time in the SIR model; $\beta = 2$, $\gamma = 0.5$.

2.2 Stochastic SIR model

The deterministic SIR model can be thought as a "law of large numbers" limit of a stochastic system consisting of a large number of individuals, each of which has a rate of transitioning between states. The epidemic starts at time $t = 0$ with $I(0)$, $S(0)$ and $R(0)$ individuals that are infected, susceptible and recovered, respectively, for a total of $n = S(0) + I(0) + R(0)$ individuals. Because the system is closed, we in fact have $n = S(t) + I(t) + R(t)$ for all t. We again use bars for the proportions $\overline{S}(t) = S(t)/n$, $\overline{I}(t) = I(t)/n$ and $\overline{R}(t) = R(t)/n$ of susceptibles, infecteds and recovered individuals, respectively.

Becker and Britton (1999) discuss maximum likelihood estimation under complete observation. Let $N(t) = S(0) - S(t)$ be the number of individuals infected in $(0, t]$. This is a counting process, as well as $R(t)$, which is the number of individuals that recovered in $(0, t]$, if $R(0) = 0$. If $\mathcal{H}_t$ is the σ-algebra generated by the history $\{S(u), I(u); 0 \leq u < t\}$, then the epidemic can be expressed in terms of the rates of the counting processes $N(t)$ and $R(t)$, by

$$
\begin{aligned}
P(\mathrm{d}N(t) = 1, \mathrm{d}R(t) = 0 \,|\, \mathcal{H}_t) &= \beta S(t)\overline{I}(t)\mathrm{d}t + o(\mathrm{d}t), \\
P(\mathrm{d}N(t) = 0, \mathrm{d}R(t) = 1 \,|\, \mathcal{H}_t) &= \gamma I(t)\mathrm{d}t + o(\mathrm{d}t), \\
P(\mathrm{d}N(t) = 0, \mathrm{d}R(t) = 0 \,|\, \mathcal{H}_t) &= 1 - \{\beta S(t)\overline{I}(t) - \gamma I(t)\}\mathrm{d}t + o(\mathrm{d}t).
\end{aligned}
$$

At some finite (random) time τ all infectious individuals will have recovered and the epidemic is over. Increasing β leads to higher infection rates, while lower γ leads to a longer time being infected and thus more opportunity to infect others. The ratio β/γ is known as the *basic reproduction number* R_0, the average number of infections caused by one typical infectious individual in a completely susceptible environment, which in our example equals 4.

Code can be written to generate an epidemic outbreak, as a function of β, γ and the initial numbers of susceptible, infected and recovered (at $t = 0$).

```
#
# Function to generate data from an SIR model
#
gen_SIR <- function(beta, gamma, S0, I0, R0) {
  # Start with S0 susceptible, I0 infected and R0 recovered, at time t=0
  T <- 0
  S <- S0; I <- I0; R <- R0
  n <- S0 + I0 + R0
  dfr <- matrix(0, 2 * n, 5) # time, S, I, R, ev (1 for infection, 0 for recovery)
  dfr[1, ] <- c(T, S, I, R, 1)
  i <- 1
  while (I > 0) { # run until no more infecteds left
    i <- i + 1
    # currently I infected, S susceptibles, determin rates
    rate_inf <- beta * I * S / n
    rate_rem <- gamma * I
    rate_tot <- rate_inf + rate_rem
    # time point of new event
    Tev <- rexp(1, rate_tot)
    # determine type of event
    ev <- sample(0:1, size = 1, prob = c(rate_rem, rate_inf))
    T <- T + Tev
    if (ev==1) { # new infection
```

```r
      S <- S - 1
      I <- I + 1
    } else { # removal
      I <- I - 1
      R <- R + 1
    }
    dfr[i, ] <- c(T, S, I, R, ev)
  }
  dfr <- as.data.frame(dfr)
  names(dfr) <- c("T", "S", "I", "R", "ev")
  return(dfr)
}

n <- 1000
# Parameters
beta <- 2
gamma <- 0.5
# Generate
set.seed(2023)
dfr <- gen_SIR(beta, gamma, I0 = 1, S0 = n - 1, R0 = 0)
dfr <- subset(dfr, T > 0 | I > 0) # remove rows where I=0, except for T=0
dfr2023 <- dfr # save for later
head(dfr, n = 12)
```

```
          T   S  I R ev
1  0.0000000 999  1 0  1
2  0.3226596 998  2 0  1
3  1.0337027 997  3 0  1
4  1.1401370 996  4 0  1
5  1.1500565 995  5 0  1
6  1.1706431 994  6 0  1
7  1.2411637 994  5 1  0
8  1.2754007 993  6 1  1
9  1.3084457 992  7 1  1
10 1.3536666 991  8 1  1
11 1.3796063 990  9 1  1
12 1.3997104 989 10 1  1
```

```r
tail(dfr)
```

```
           T  S I   R ev
1945 15.13639 25 5 970  0
1946 15.28356 25 4 971  0
1947 15.98904 25 3 972  0
1948 16.07295 25 2 973  0
1949 16.71328 25 1 974  0
1950 17.54286 25 0 975  0
```

The data shows one realization of the schochastic process, with the same parameters as before, a population size of 1000, and a single infectious individual at $t = 0$, with (in column `T`) the time points at which events (either infections or recoveries) occur (column `ev` denotes whether it is an infection

(ev=1) or a recovery (ev=0)). The columns S, I and R denote the number of susceptibles, infected and recovered in the population (total size = 1000). Note that 25 subjects have remained uninfected. Figure 2 shows the randomly generated epidemic.

```
plot(dfr$T, dfr$I, type="s", lwd=2, xlab="Time", ylab="",
     ylim=c(0, n), col="red")
lines(dfr$T, dfr$I + dfr$R, type="s", lwd=2)
lines(dfr$T, dfr$I + dfr$R + dfr$S, type="s", lwd=2, col="blue")
legend(5, 0.8, c("I", "I + R", "I + R + S"), lwd=2,
       col=c("red", "black", "blue"), bty="n", xjust=1)
```

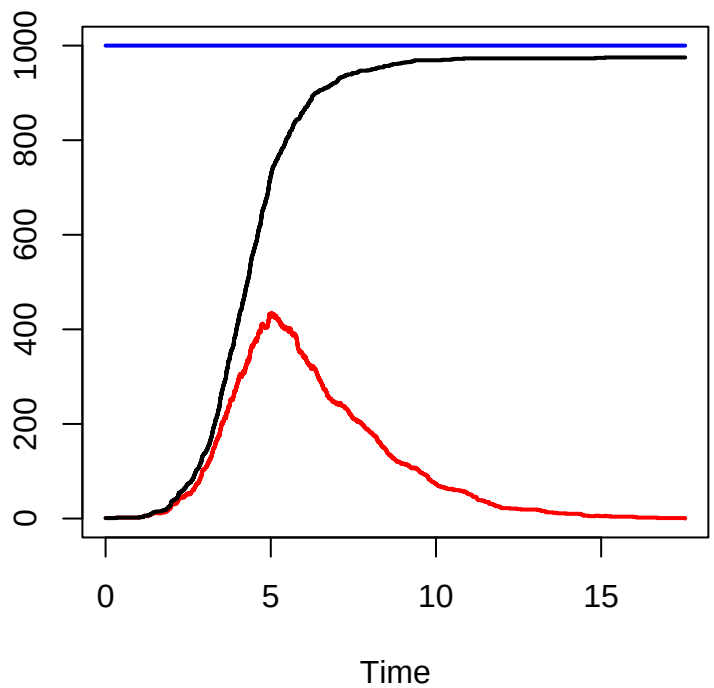
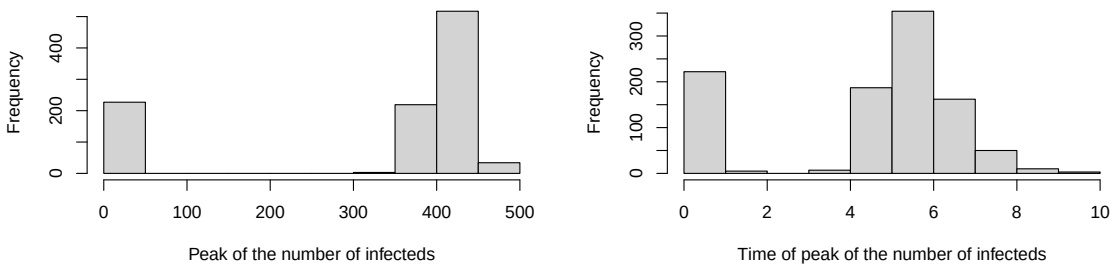


Figure 2: Stacked plot showing the number of susceptible, infected and recovered individuals over time from a stochastic SIR model.

This looks very much like the deterministic system, but with, on average, a slight delay in the time of peak prevalence. Note, however, that the system is now stochastic. With another seed we would obtain a different epidemic. Occasionally the epidemic might just not start off, because by chance the early infected individuals recover before they managed to infect new individuals. Figure 3 shows a histogram of (a) the peaks and (b) the time at which they occurred in 1000 simulated epidemics, with the same parameters as before, a population size of 1000 and one initial infected individual.

```r
M <- 1000; maxs <- times <- rep(NA, M)
for (seed in 1:M) {
  set.seed(seed)
  dfr <- gen_SIR(beta, gamma, I0=1, S0=999, R0=0)
  maxI <- which.max(dfr$I)
  maxs[seed] <- dfr$I[maxI]
  times[seed] <- dfr$T[maxI]
}
hist(maxs, xlab="Peak of the number of infecteds", main="")
hist(times, xlab="Time of peak of the number of infecteds", main="")
```



(a) Peak of the number of infecteds

(b) Time of peak of the number of infecteds

Figure 3: Results of 1000 simulated SIR epidemics

We see that in about 8% of the simulated epidemics, only a very limited number of individuals (here at most five, including the index case) got infected. This coincides with mathematical theory saying that with one initial infectious individual the probability of the epidemic not spreading to a large part of the population equals $1/R_0$ (Diekmann, Heesterbeek, and Britton 2013).

Becker and Britton (1999) derived the likelihood under complete observation, as

$$\ell(\beta,\gamma) = \int_0^\tau \Big[\log\{\beta S(t)\overline{I}(t)\}\mathrm{d}N(t) - \beta S(t)\overline{I}(t)\mathrm{d}t + \log\{\gamma I(t)\}\mathrm{d}R(t) - \gamma I(t)\mathrm{d}t\Big].$$

The first thing to notice is that the log-likelihood can be written as a sum of terms only depending on β and only on γ, respectively, implying that β and γ can be maximized separately. The maximum likelihood estimator (MLE) of β can then be analytically derived by taking the derivative of the part involving β, with respect to the parameter β, leading to (ignoring a β^{-1} term)

$$U(\beta) = \int_0^\tau \left\{\mathrm{d}N(t) - \beta S(t)\overline{I}(t)\mathrm{d}t\right\}, \tag{2}$$

and setting this to zero. This yields

$$\widehat{\beta}_{\mathrm{MLE}} = N(\tau)/\Big\{\int_0^\tau S(t)\overline{I}(t)\mathrm{d}t\Big\} \tag{3}$$

as the maximum likelihood estimator of β. Similarly (assuming $R(0) = 0$), we arrive at

$$\widehat{\gamma}_{\mathrm{MLE}} = R(\tau)/\Big\{\int_0^\tau I(t)\mathrm{d}t\Big\}.$$

These estimates can be readily calculated from the current data.

```
dfr <- dfr2023 # reuse the first data with seed 2023
n <- dfr$S[1] + dfr$I[1] + dfr$R[1]
Ntau <- sum(dfr$ev[-1]) # total number of new infections
Ndfr <- nrow(dfr)
dfr$length_int <- c(dfr$T[-1] - dfr$T[-Ndfr], 0)
intSI <- sum(dfr$S * dfr$I * dfr$length_int)
betahat_MLE <- Ntau / (intSI / n)
betahat_MLE
```

```
[1] 1.987948
```

```
intI <- sum(dfr$I * dfr$length_int)
gammahat_MLE <- max(dfr$R) / intI
gammahat_MLE
```

```
[1] 0.508712
```

Estimates of the variances of $\widehat{\beta}_{\text{MLE}}$ and $\widehat{\gamma}_{\text{MLE}}$ are also provided in Becker and Britton (1999). From now on we concentrate on the transmission parameter β. We see from Becker and Britton (1999) that $\text{se}(\widehat{\beta}_{\text{MLE}}) = \widehat{\beta}_{\text{MLE}}/\sqrt{N(\tau)}$. This gives the following value for the standard error of $\widehat{\beta}_{\text{MLE}}$:

```
sebetahat_MLE <- betahat_MLE / sqrt(Ntau)
sebetahat_MLE
```

```
[1] 0.06369797
```

3 The link with survival analysis

Counting processes and their associated rates play a pivotal role in survival analysis so it is not surprising that methods from survival analysis can be used to estimate transmission parameters in SIR models. From now on, we restrict to estimation of β and ignore γ. The rates of the counting processes defined above are all aggregated over all susceptible and infected individuals. Restricting our attention to $N(t)$, which is counting the total number of new infections occurring in $(0, t]$, the link with survival analysis becomes clearer if we consider each of the individual counting processes $N_i(t)$ leading to $N(t) = \sum_{i=1}^{n} N_i(t)$. Thus, $N_i(t)$ counts the number of new infections *of individual* i occurring in $(0, t]$, and this will be 1 or 0, depending on whether individual i has been infected or not before time t. Define $S_i(t)$, $I_i(t)$, and $R_i(t)$ to be random indicator variables taking the value 1 if at time t individual i is susceptible, infected or recovered, respectively, and 0 otherwise. The total number of susceptible, infected and recovered individuals is then $S(t) = \sum_{i=1}^{n} S_i(t)$, $I(t) = \sum_{i=1}^{n} I_i(t)$, and $R(t) = \sum_{i=1}^{n} R_i(t)$, respectively. Individual i is only at risk of becoming infected while being susceptible. The usual notation in survival to indicate whether or not individual i is at risk at time t is $Y_i(t)$. Since the event of interest here is becoming infected, being at risk is the same as being susceptible, for which we have already defined the notation $S_i(t)$, so we have $Y_i(t) = S_i(t)$. In the remainder of this paper we will be using $Y_i(t)$ when recalling general theory from survival analysis, and $S_i(t)$ when applying this to the SIR model. The rate of $N_i(t)$ is given by $\lambda_i(t) = \beta\overline{I}(t)$, while being susceptible, which can then be written as $\lambda_i(t) = \beta S_i(t)\overline{I}(t)$. In this way, adding over all individuals, we see that the rate of $N(t)$ equals $\sum_{i=1}^{n} \lambda_i(t) = \sum_{i=1}^{n} \beta S_i(t)\overline{I}(t)$, leading to a total rate of $\beta S(t)\overline{I}(t)$, the same as in Equation 2. Also, each *infected* individual has a constant rate γ of recovering.

This perspective is called *individual-based* or *agent-based* modeling, see KhudaBuksh et al. (2019). With the purpose of estimating β, we can exploit the link to survival analysis, again under complete observation, by translating the original data, documenting the number of S, I, R individuals over time, into a long format data set where each susceptible subject occurs individually, and where at the same time the number of infected individuals is kept track of, using start-stop notation, also called Andersen-Gill (Andersen and Gill 1982) or counting process notation. The following code establishes this, starting from the original `dfr` data.

```
library(survival)
library(tidyverse)
SIR2surv <- function(SIRdata)
{
  n <- SIRdata$S[1] + SIRdata$I[1] + SIRdata$R[1] # first extract the total size
  wh <- which(SIRdata$ev == 1) # select the infection events
  ninf <- length(wh) # number of observed infections in the time window
  tinf <- SIRdata$T[wh]
  d <- data.frame(id = 1:ninf, time = tinf, status = 1)
  d$w <- 1 # give weight 1 to observed infections
  # First is not really an observed event, so remove
  d <- d[-1, ]
  # Add the rest of the population to the data with number of never infecteds
  d <- rbind(d, data.frame(id=ninf+1, time=max(SIRdata$T), status=0, w=n-ninf))
  # Prepare long data format
  tt <- SIRdata$T
  dlong <- survSplit(Surv(time, status) ~ ., data=d, cut=tt[-1])
  # Add proportion of infecteds as time-dependent covariate
  dlong$pinf <- SIRdata$I[match(dlong$tstart, SIRdata$T)] / n
  dlong$logpinf <- log(dlong$pinf)
  dlong$fuptime <- dlong$time - dlong$tstart # length of follow-up interval
  dlong$logfuptime <- log(dlong$fuptime)
  dlong <- subset(dlong, w>0)
  return(as_tibble(dlong))
}
dfr <- dfr2023
dlong <- SIR2surv(dfr)
```

The head and tail of this expanded data set looks like this. The column `fuptime` has the length of the time interval between `tstart` and `time`, and its logarithm, `logfuptime`, will prove to be useful for Poisson regression later.

```
head(dlong, 12)
```

```
# A tibble: 12 x 9
      id     w tstart  time status  pinf logpinf fuptime logfuptime
   <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>
 1     2     1  0     0.323      1 0.001   -6.91 0.323       -1.13
 2     3     1  0     0.323      0 0.001   -6.91 0.323       -1.13
 3     3     1  0.323 1.03       1 0.002   -6.21 0.711       -0.341
 4     4     1  0     0.323      0 0.001   -6.91 0.323       -1.13
 5     4     1  0.323 1.03       0 0.002   -6.21 0.711       -0.341
```

```
 6     4     1  1.03  1.14       1 0.003   -5.81 0.106       -2.24
 7     5     1  0     0.323      0 0.001   -6.91 0.323       -1.13
 8     5     1  0.323 1.03       0 0.002   -6.21 0.711       -0.341
 9     5     1  1.03  1.14       0 0.003   -5.81 0.106       -2.24
10     5     1  1.14  1.15       1 0.004   -5.52 0.00992     -4.61
11     6     1  0     0.323      0 0.001   -6.91 0.323       -1.13
12     6     1  0.323 1.03       0 0.002   -6.21 0.711       -0.341
```

```
tail(dlong, 12)
```

```
# A tibble: 12 x 9
      id     w tstart  time status  pinf logpinf fuptime logfuptime
   <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>
 1   976    25   14.4  14.4      0 0.008   -4.83  0.0275     -3.59
 2   976    25   14.4  14.5      0 0.007   -4.96  0.0413     -3.19
 3   976    25   14.5  14.6      0 0.006   -5.12  0.151      -1.89
 4   976    25   14.6  14.8      0 0.005   -5.30  0.195      -1.64
 5   976    25   14.8  14.9      0 0.006   -5.12  0.123      -2.10
 6   976    25   14.9  15.1      0 0.005   -5.30  0.163      -1.81
 7   976    25   15.1  15.1      0 0.006   -5.12  0.0243     -3.72
 8   976    25   15.1  15.3      0 0.005   -5.30  0.147      -1.92
 9   976    25   15.3  16.0      0 0.004   -5.52  0.705      -0.349
10   976    25   16.0  16.1      0 0.003   -5.81  0.0839     -2.48
11   976    25   16.1  16.7      0 0.002   -6.21  0.640      -0.446
12   976    25   16.7  17.5      0 0.001   -6.91  0.830      -0.187
```

A more concise version of this long format data can be obtained by combining all the rows with the same time interval (`tstart`, `time`), and adding all the weights, separately for `status`=0 and `status`=1.

```
dlong_short <- as_tibble(dlong) %>%
  group_by(tstart, time, status) %>%
  summarize(pinf = min(pinf), fuptime = min(fuptime),
            w = sum(w)) %>%
  mutate(logpinf = log(pinf), logfuptime = log(fuptime)) %>%
  ungroup()
```

```
`summarise()` has grouped output by 'tstart', 'time'. You can override using
the `.groups` argument.
```

```
head(dlong_short, n=12)
```

```
# A tibble: 12 x 8
   tstart  time status  pinf fuptime     w logpinf logfuptime
    <dbl> <dbl>  <dbl> <dbl>   <dbl> <dbl>   <dbl>      <dbl>
 1  0     0.323      0 0.001 0.323     998   -6.91     -1.13
 2  0     0.323      1 0.001 0.323       1   -6.91     -1.13
 3  0.323 1.03       0 0.002 0.711     997   -6.21     -0.341
 4  0.323 1.03       1 0.002 0.711       1   -6.21     -0.341
 5  1.03  1.14       0 0.003 0.106     996   -5.81     -2.24
 6  1.03  1.14       1 0.003 0.106       1   -5.81     -2.24
```

```
 7  1.14  1.15       0 0.004 0.00992   995   -5.52     -4.61
 8  1.14  1.15       1 0.004 0.00992     1   -5.52     -4.61
 9  1.15  1.17       0 0.005 0.0206    994   -5.30     -3.88
10  1.15  1.17       1 0.005 0.0206      1   -5.30     -3.88
11  1.17  1.24       0 0.006 0.0705    994   -5.12     -2.65
12  1.24  1.28       0 0.005 0.0342    993   -5.30     -3.37
```

```
tail(dlong_short, n=12)
```

```
# A tibble: 12 x 8
   tstart  time status  pinf fuptime     w logpinf logfuptime
    <dbl> <dbl>  <dbl> <dbl>   <dbl> <dbl>   <dbl>      <dbl>
 1   14.5  14.6      0 0.006  0.151     27   -5.12     -1.89
 2   14.6  14.8      0 0.005  0.195     26   -5.30     -1.64
 3   14.6  14.8      1 0.005  0.195      1   -5.30     -1.64
 4   14.8  14.9      0 0.006  0.123     26   -5.12     -2.10
 5   14.9  15.1      0 0.005  0.163     25   -5.30     -1.81
 6   14.9  15.1      1 0.005  0.163      1   -5.30     -1.81
 7   15.1  15.1      0 0.006  0.0243    25   -5.12     -3.72
 8   15.1  15.3      0 0.005  0.147     25   -5.30     -1.92
 9   15.3  16.0      0 0.004  0.705     25   -5.52     -0.349
10   16.0  16.1      0 0.003  0.0839    25   -5.81     -2.48
11   16.1  16.7      0 0.002  0.640     25   -6.21     -0.446
12   16.7  17.5      0 0.001  0.830     25   -6.91     -0.187
```

```
dim(dlong)
```

```
[1] 693611      9
```

```
dim(dlong_short)
```

```
[1] 2923    8
```

In the remainder of this section we will review a number of standard models in survival analysis, show how they apply to the SIR model, and illustrate how standard software for survival analysis can be used, in combination with the long format data, to fit these models.

3.1 Additive hazards models

The additive hazards model (Aalen 1980, 1989) specifies the rate of new infections as a sum of (typically time-dependent) linear combinations of the covariates, which themselves may also be time-dependent:

$$\lambda_i(t) = Y_i(t)\{\beta_0(t) + \beta_1(t)X_{i1}(t) + \ldots + \beta_p(t)X_{ip}(t)\}.$$

The first term within brackets, $\beta_0(t)$, is an intercept, comparable to the baseline hazard in the Cox model, $X_{i1}(t), \ldots, X_{ip}(t)$ is a set of possibly time-dependent covariates, and $\beta_1(t), \ldots, \beta_p(t)$ are the regression coefficients. Typically these are taken to be time-dependent, but we will also consider the case later where the $\beta(t)$'s are constant over time. Aalen's ordinary least squares (OLS) estimates focus on the cumulative regression functions $B_j(t) = \int_0^t \beta_j(s)\mathrm{d}s$.

Defining vectors $\mathbf{N}(t) = (N_1(t), \ldots, N_n(t))^\top$, $\mathbf{B}(t) = (B_0(t), B_1(t), \ldots, B_p(t))^\top$, and the matrix $\mathbf{X}(t)$, with ith row $(Y_i(t), Y_i(t)X_{i1}(t), \ldots, Y_i(t)X_{ip}(t))$, then Aalen, Borgan, and Gjessing (2008) derive

$$\mathrm{d}\widehat{\mathbf{B}}(t) = \left(\mathbf{X}(t)^\top \mathbf{X}(t)\right)^{-1} \mathbf{X}(t)^\top \mathrm{d}\mathbf{N}(t) \tag{4}$$

as estimate of the increment of $\mathbf{B}(t)$, provided $\mathbf{X}(t)$ has full rank. In the absence of the intercept term $\beta_0(t)$ in the model, the constant elements in the first column of $\mathbf{X}(t)$ are removed.

Applying this model to the SIR setting, since the rate of each susceptible individual equals $\beta\overline{I}(t)$, we can view this as a term β/n being added to the hazard for each infectious individual. So we would have no intercept, $p = 1$, and $X_{i1}(t) = X_i(t) = \overline{I}(t)$ for each individual, and $\mathbf{X}(t)$ would be a $N \times 1$ vector with ith element $Y_i(t)X_i(t) = S_i(t)\overline{I}(t)$. Using Equation 4 leads to

$$\widehat{\beta}(t) = \left\{\sum_{i=1}^{n} (S_i(t)\overline{I}(t))^2\right\}^{-1} \sum_{i=1}^{n} S_i(t)\overline{I}(t)\mathrm{d}N_i(t) = \frac{\overline{I}(t)\mathrm{d}N(t)}{S(t)\{\overline{I}(t)\}^2} = \frac{\mathrm{d}N(t)}{S(t)\overline{I}(t)}. \tag{5}$$

Here we have used that $S_i^2(t) = S_i(t)$ and $S_i(t)\mathrm{d}N_i(t) = \mathrm{d}N_i(t)$ (because an event can only happen when someone is at risk). This $\widehat{\beta}(t)$ can be estimated in the {timereg} package in R, by fitting an additive hazards model with $\overline{I}(t)$ (in column `pinf`) as covariate, *excluding* an intercept (hence `pinf - 1` in the formula below). This idea has been used before in Wolkewitz et al. (2002). The plot in Figure 4 shows an estimate of $\widehat{\mathbf{B}}(t)$ over time.

```r
library(timereg)
# Fit additive hazards model without intercept
ahfit1 <- aalen(Surv(tstart, time, status) ~ pinf - 1,
                data = dlong, weights = dlong$w)
summary(ahfit1)
```

```
Additive Aalen Model

Test for nonparametric terms

Test for non-significant effects
     Supremum-test of significance p-value H_0: B(t)=0
pinf                          3.55               0.003

Test for time invariant effects
           Kolmogorov-Smirnov test p-value H_0:constant effect
pinf                          6.99                       0.721
             Cramer von Mises test p-value H_0:constant effect
pinf                           271                        0.72



  Call:
aalen(formula = Surv(tstart, time, status) ~ pinf - 1, data = dlong,
    weights = dlong$w)
```

```r
par(mfrow=c(1, 1))
plot(ahfit1, main = "")
```

```
lines(c(0, 15), c(0, 30), lty = 3, col = "blue")
```

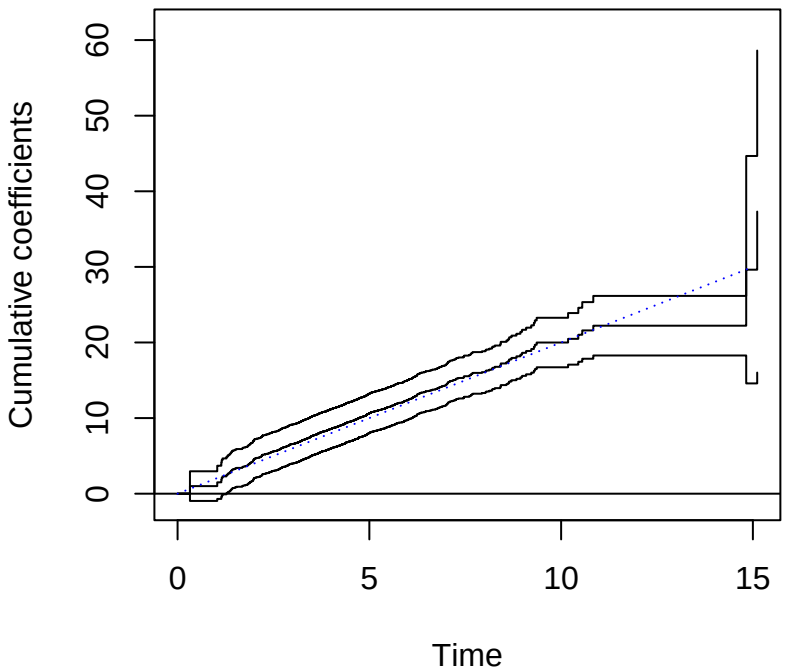


Figure 4: Estimate of $B(t)$ over time; the dotted line represents the true $B(t)$.

It is reassuring that the estimate of $B(t)$ indeed seems to follow a straight line, indicating constant $\beta(t)$. The hypothesis of $\beta(t)$ being constant is actually added in the "tests for time invariant effects", of which both the Kolmogorov - Smirnov and the Cramer - von Mises tests show no significant departures from time invariant effects. For background on these tests we refer to Martinussen and Scheike (2006). The slope is approximately equal to $\beta = 2$, as we would hope, see the dotted line in Figure 4.

Inspired by least squares theory, a least squares estimator would be given by a variant of the displayed equation between (2.6) and (2.7) of Lin and Ying (1994), the difference being the absence of the baseline hazard term, given by

$$U(\beta) = \sum_{i=1}^{n} \int_0^{\tau} X_i(t) \left\{ \mathrm{d}N_i(t) - Y_i(t)\beta X_i(t)\mathrm{d}t \right\},$$

where we recall that the time-dependent covariate is given by $X_i(t) = \overline{I}(t)$, and $Y_i(t)$ is the at risk indicator, which in our case was earlier denoted by $S_i(t)$. We will first elaborate on the general theory, then replace $X_i(t)$ by $\overline{I}(t)$ and $Y_i(t)$ by $S_i(t)$ to apply it to the SIR case. Setting $U(\beta)$ to zero and

solving for β yields

$$\widehat{\beta}_{\text{OLS}} = \frac{\sum_{i=1}^{n} \int_0^\tau X_i(t)\mathrm{d}N_i(t)}{\sum_{i=1}^{n} \int_0^\tau Y_i(t)X_i^2(t)\mathrm{d}t}.$$

Following Lin and Ying (1994), define

$$A = n^{-1} \sum_{i=1}^{n} \int_0^\tau Y_i(t)X_i^2(t)\mathrm{d}t, \quad B = n^{-1} \sum_{i=1}^{n} \int_0^\tau X_i^2(t)\mathrm{d}N_i(t).$$

Then the variance of $\widehat{\beta}_{\text{OLS}}$ can be consistently estimated by $B/(NA^2)$, which equals

$$\frac{\sum_{i=1}^{n} \int_0^\tau X_i^2(t)\mathrm{d}N_i(t)}{\left(\sum_{i=1}^{n} \int_0^\tau Y_i(t)X_i^2(t)\mathrm{d}t\right)^2}.$$

We go back to the formulas for $\widehat{\beta}_{\text{OLS}}$ and its variance, and now replace $X_i(t)$ by $\overline{I}(t)$ and $Y_i(t)$ by $S_i(t)$, leading to

$$\widehat{\beta}_{\text{OLS}} = \frac{\int_0^\tau \overline{I}(t)\mathrm{d}N(t)}{\int_0^\tau \{\overline{I}(t)\}^2 S(t)\mathrm{d}t}. \tag{6}$$

Note first that $\widehat{\beta}_{\text{OLS}}$ is of a similar form as the estimate $\widehat{\beta}(t) = \frac{\overline{I}(t)\mathrm{d}N(t)}{\{overline I(t)\}^2 S(t)}$ of Equation 5, but with both the numerator and denominator being integrated over time. Note also that $\widehat{\beta}_{\text{OLS}}$ is similar to $\widehat{\beta}_{\text{MLE}}$ in Equation 3, except that $\widehat{\beta}_{\text{OLS}}$ has an extra weighting term $\overline{I}(t)$ in both numerator and denominator.

The estimate of its variance is given by

$$\frac{\int_0^\tau \{\overline{I}(t)\}^2 \mathrm{d}N(t)}{\left(\int_0^\tau \{\overline{I}(t)\}^2 S(t)\mathrm{d}t\right)^2}.$$

Here is the result when implementing these formulas in our data, given the following estimate:

```
dfr$Ibar <- dfr$I / n
num <- sum(dfr$Ibar * dfr$ev)
denom <- sum((dfr$Ibar)^2 * dfr$S * dfr$length_int)
betahat_OLS <- num / denom
betahat_OLS
```

```
[1] Inf
```

and its estimated variance:

```
A <- denom
B <- sum( (dfr$Ibar)^2 * dfr$ev )
varbetahat_OLS <- B / A^2
varbetahat_OLS
```

```
[1] Inf
```

```
sebetahat_OLS <- sqrt(varbetahat_OLS)
```

The standard error equals Inf, which is about ∞% larger than the one obtained by Becker and Britton (1999), which was 0.064. Summarizing, the estimate and its 95% confidence interval are given by

```
res_beta <- data.frame(betahat = betahat_OLS, SE = sebetahat_OLS,
                       lower = betahat_OLS - qnorm(0.975) * sebetahat_OLS,
                       upper = betahat_OLS + qnorm(0.975) * sebetahat_OLS)
res_beta
```

```
  betahat  SE lower upper
1     Inf Inf   NaN   Inf
```

The {timereg} package can in principle also estimate constant $\beta(t)$, but somehow not without an intercept, so as far as I have been able to see the above procedure can not be implemented in {timereg}.

3.2 Multiplicative models

3.2.1 Cox models

Recall that the rate of $N_i(t)$ equals $\beta\overline{I}(t)$ while being at risk. The Cox model assumes that the rate of the event equals $Y_i(t)h_0(t)\exp(\beta_1 X_{i1}(t) + \ldots + \beta_p X_{ip}(t)) = Y_i(t)h_0(t)\exp(\beta^\top X_i(t))$. Here $Y_i(t)$ again is the at risk indicator, $h_0(t)$ is an unspecified baseline hazard, $X_{i1}(t), \ldots, X_{ip}(t)$ are possibly time-dependent covariates, and $\beta_1, \ldots, \beta_p$ are regression coefficients to be estimated. In the absence of ties, the vector of regression coefficients is estimated by maximizing the *partial likelihood*, given by

$$\prod_{i=1}^{n}\prod_{t\geq 0}\left\{\frac{Y_i(t)\exp(\beta^\top X_i(t))}{\sum_{j=1}^{n} Y_j(t)\exp(\beta^\top X_j(t))}\right\}^{\mathrm{d}N_i(t)}.$$

This partial likelihood is maximized with respect to β to obtain estimates $\widehat{\beta}$ of β. For given β, in the absence of ties the estimate of the baseline hazard increment is given by Breslow's estimate

$$\widehat{h}_0(t) = \frac{\mathrm{d}N(t)}{\sum_{i=1}^{n} Y_i(t)\exp(\beta^\top X_i(t))}. \tag{7}$$

If we take $p = 1$, $X_{i1}(t) = \log(\overline{I}(t))$, and fix $\beta_1 = 1$, we get a hazard rate of $h_0(t)\exp(\log(\overline{I}(t))) = h_0(t)\overline{I}(t)$. Using Equation 7 we get

$$\widehat{h}_0(t) = \frac{\mathrm{d}N(t)}{\sum_{i=1}^{n} S_i(t)\exp(\log(\overline{I}(t)))} = \frac{\mathrm{d}N(t)}{S(t)\overline{I}(t)},$$

which can be seen to equal the hazard increments $\widehat{\beta}(t)$ from Equation 5 obtained from the additive hazards model!

If we fit a Cox model with $\log(\overline{I}(t))$ (`logpinf`) as an *offset* (meaning we are not estimating the associated regression coefficient, but setting it to one), then the baseline hazard rate of the Cox model becomes $h_0(t)\exp(\log(\overline{I}(t))) = h_0(t)\overline{I}(t)$, the estimated baseline rate $\widehat{h}_0(t)$ should equal $\widehat{\beta}(t)$, and the cumulative baseline hazard $\widehat{H}_0(t) = \sum_{t_i \leq t}\widehat{h}_0(t_i)$ should resemble βt, in other words a straight line with intercept 0 and slope β. Fitting the Cox regression seems to recover the constant rate, although care is needed to extract the baseline hazard due to the offset term. When adding the estimated cumulative hazard of Aalen's additive hazard model to the resulting estimate, we indeed see that the Cox and Aalen based baseline hazards are exactly the same.

```r
dlong <- dlong_short
c0 <- coxph(Surv(tstart, time, status) ~ offset(logpinf), data=dlong, weights = w)
bh0 <- basehaz(c0, centered=FALSE)
par(mfrow=c(1, 1))
bh0 <- rbind(data.frame(hazard=0, time=0), bh0) # add time 0
bh0 <- bh0[, c(2, 1)] # change column order of time and hazard
bh0 <- bh0[!duplicated(bh0$hazard), ]
# Divide by exponent of (weighted) mean of offset variable
scaling_factor <- exp(weighted.mean(dlong$logpinf, dlong$w))
bh0$hazard <- bh0$hazard / scaling_factor

bh0 <- cbind(bh0, ahfit1$cum[, 2]) # add cumulative hazard of Aalen model
names(bh0)[2:3] <- c("hazard_Cox", "hazard_Aalen")
head(bh0)
```

```
       time hazard_Cox hazard_Aalen
1 0.0000000   0.000000     0.000000
2 0.3226596   1.001001     1.001001
3 1.0337027   1.502003     1.502003
4 1.1401370   1.836339     1.836339
5 1.1500565   2.087343     2.087343
6 1.1706431   2.288348     2.288348
```

```r
tail(bh0)
```

```
         time hazard_Cox hazard_Aalen
1873 10.18134   20.48141     20.48141
1879 10.44775   21.01904     21.01904
1883 10.55053   21.59376     21.59376
1888 10.83856   22.22032     22.22032
1942 14.82600   29.62773     29.62773
1944 15.11207   37.32004     37.32004
```

3.2.2 Poisson regression

The Cox model uses an unspecified baseline hazard and does not use the assumption that the infection rate is constant. We could attempt to estimate the infection rate, assuming that it is constant, based on the well established epidemiologic occurrence over exposure (O/E), see Clayton and Hills (1993). The variance of the log rate is the inverse of the number of events, by which we could construct a 95% confidence interval of the estimated rate.

```r
OE <- dlong %>%
  summarise(O = sum(status * w),
            E = sum(pinf * fuptime * w)) %>%
  mutate(rate = O / E,
         lograte = log(rate),
         SElograte = 1 / sqrt(O),
         lower = exp(lograte - qnorm(0.975) * SElograte),
         upper = exp(lograte + qnorm(0.975) * SElograte))
OE
```

```
# A tibble: 1 x 7
      O     E  rate lograte SElograte lower upper
  <dbl> <dbl> <dbl>   <dbl>     <dbl> <dbl> <dbl>
1   974  490.  1.99   0.687    0.0320  1.87  2.12
```

This quite nicely returns the true rate 2. It is also identical to the value of 1.9879483 for β obtained by the direct formula of Becker and Britton (1999).

Alternatively, Poisson regression can be used. The expected number of infections in a short interval of length Δt around t equals $\beta \overline{I}(t)\Delta t$, so with a log link this equals $\exp(\log \beta + \log \overline{I}(t) + \log \Delta t)$. This means that fitting a GLM Poisson model with log link, and with both $\log \overline{I}(t)$ and $\log \Delta t$ as offset terms, we obtain an estimate of $\log \beta$. Taking the exponent of the estimate of $\log \beta$ then provides an estimate of β.

```r
poisreg <- glm(status ~ offset(logpinf) + offset(logfuptime),
               data=dlong, weights=w, family="poisson")
summary(poisreg)
```

```
Call:
glm(formula = status ~ offset(logpinf) + offset(logfuptime),
    family = "poisson", data = dlong, weights = w)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.68710    0.03204   21.44   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 13835  on 2922  degrees of freedom
Residual deviance: 13835  on 2922  degrees of freedom
AIC: 15785

Number of Fisher Scoring iterations: 9
```

```r
cipoisreg <- confint(poisreg)
cipoisreg
```

```
    2.5 %    97.5 %
0.6236365 0.7492528
```

```r
tmp <- c(poisreg$coef, cipoisreg)
exp(tmp)
```

```
(Intercept)       2.5 %      97.5 %
   1.987948    1.865700    2.115419
```

This again gives exactly the same result for $\widehat{\beta}$ as Becker and Britton (1999). The 95% confidence intervals given by the occurrence/exposure formulas and Poisson regression differ slightly, but this is due to different ways of constructing these confidence intervals (Wald versus likelihood-based). The

standard errors from the Poisson regression and the occurrence / exposure formulas *are* exactly the same.

The Poisson regression model also gives the opportunity to estimate $\beta(t)$ as a smooth function of time using splines. We use the natural splines (cubic splines that are linear beyond the outermost knots), as implemented in the `Ns()` function in the {Epi} package, but other splines can also be used.

```
library(Epi)
tinf <- sort(dfr$T[dfr$ev == 1])
t.kn <- summary(tinf)[-3] # excluding median
poisfit2 <- glm(status ~ Ns(time, knots=t.kn) + offset(logfuptime) + offset(logpinf),
               family="poisson", data=dlong,
               weights = w)
summary(poisfit2)
```

```
Call:
glm(formula = status ~ Ns(time, knots = t.kn) + offset(logfuptime) +
    offset(logpinf), family = "poisson", data = dlong, weights = w)

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)               1.1825     0.4458   2.653  0.00799 **
Ns(time, knots = t.kn)1  -0.5073     0.4279  -1.185  0.23587
Ns(time, knots = t.kn)2  -0.2778     0.3373  -0.824  0.41022
Ns(time, knots = t.kn)3  -1.1291     0.9909  -1.140  0.25448
Ns(time, knots = t.kn)4  -0.5775     0.7076  -0.816  0.41445
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 13835  on 2922  degrees of freedom
Residual deviance: 13833  on 2918  degrees of freedom
AIC: 15791

Number of Fisher Scoring iterations: 9
```

Note that none of the effects of the spline coefficients is significant. This is consistent with a constant infection rate. Otherwise the output is not very informative. It is more informative to make a plot of the fitted time-dependent infection rate. The function `matshade()` from the Epi package makes a nice plot, shown in Figure 5.

```
tt <- seq(0, 12, by=0.05)
nd <- data.frame(time=tt, logfuptime=0, logpinf=0)
lambda <- ci.pred(poisfit2, nd) # the rates from the spline model
par(mfrow=c(1, 1))
matshade(tt, lambda, plot=TRUE, col="blue", lwd=2,
         xlab="Time", ylab="Infection rate",
         xlim=c(0, 12), ylim=c(0, 8))
```

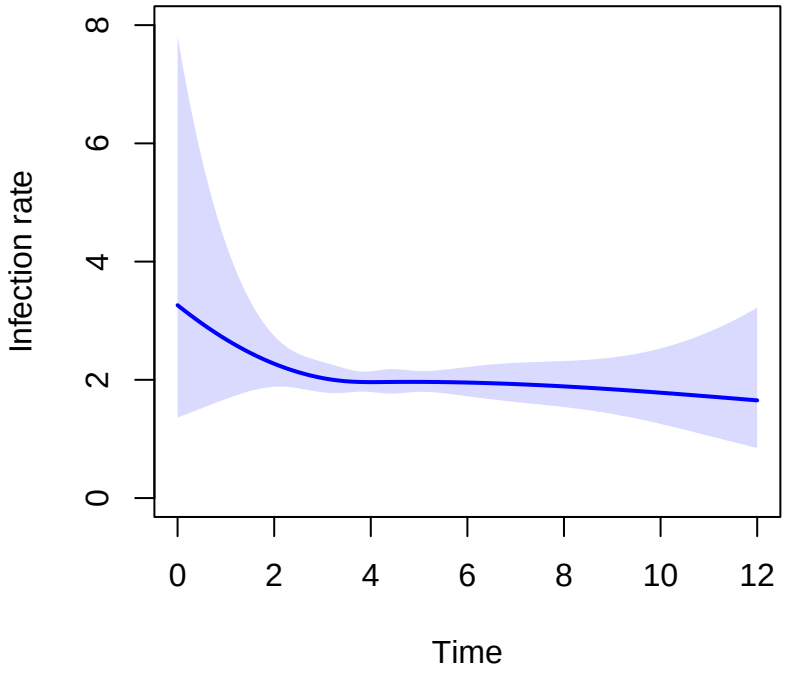


Figure 5: Smooth estimate of $\beta(t)$.

It can be seen that the curve is more or less constant, with wider confidence intervals in the beginning.

4 Modeling perspectives

4.1 Additive versus multiplicative (Aalen versus Cox)

In the previous section we saw that for estimating β, when not assuming β to be constant, the estimate of $\beta(t)$ obtained from the additive hazards model and the Cox model gave exactly the same result. The perspective of the additive hazards model is that we can view the rate $\beta\overline{I}(t)$ of a susceptible individual as additive in the number of infected individuals at time t; each infected individual *adds* β/n to the hazard. In contrast, the multiplicative hazards model views $\log(\overline{I}(t))$ as a multiplicative term, with regression coefficient fixed to one (an offset term).

Without any other covariates both perspectives are equally valuable and they give the same result when $\beta(t)$ is estimated non-parametrically as a time-dependent function, and slightly different but comparable results when estimating a time-constant β. Different contexts will call for different modeling strategies, and we argue that in many cases a hybrid form might be advantageous.

4.2 Multiplicative models

Implementations of intervention measures are most naturally expressed in a multiplicative way. It makes sense to assume that such measures will decrease the hazard of every individual by a certain percentage. In ideal settings the effect of the intervention can be estimated together with the transmission parameter, using Poisson regression.

We are now going to explore the use of Cox and Poisson regression models in a situation where intervention measures are put into place, triggered here by the proportion of infected individuals reaching a certain threshold. The function below incorporates a single intervention, which is suspended by the time the proportion of infected individuals again reaches a lower threshold. The function is the same as before, with three extra parameters, the first, `effect`, being the effect of the intervention as a rate ratio, the second and third, `pinf_start` and `pinf_stop`, determining at what stage (in terms of percentage of infected individuals) the intervention is started and stopped.

```
SIR2 <- function(beta, gamma, n, effect, pinf_start, pinf_stop) {
  # Start with one susceptible and n-1 susceptibles, at time t=0
  I <- 1; S <- n - 1; R <- 0; T <- 0
  dfr <- matrix(0, 2 * n, 5) # time, S, I, R, ev
  dfr[1, ] <- c(T, S, I, R, 1)
  i <- 1
  intervention <- 0 # intervention has not yet started
  beta0 <- beta # reserve beta0 for baseline beta
  while (I > 0) {
    if (I >= pinf_start * n & I < pinf_start * n + 1 & intervention == 0) {
      # the first time I crosses pinf_start * n
      intervention <- 1 # intervention has started
      beta <- beta0 * effect # effective beta is affected by intervention
      t1 <- T # record time of intervention start
    }
    if (I <= pinf_stop * n & I > pinf_stop * n - 1 & intervention == 1) {
      # the first time I crosses pinf_stop * n
      intervention <- 0 # intervention has stopped
      beta <- beta0 # effective beta is no longer affected by intervention
      t2 <- T # record time of intervention stop
    }
    i <- i + 1
    # currently I infected, S susceptibles
    rate_inf <- beta * I * S / n
    rate_rem <- gamma * I
    rate_tot <- rate_inf + rate_rem
    Tev <- rexp(1, rate_tot) # time point of new event
    ev <- sample(0:1, size = 1, prob = c(rate_rem, rate_inf)) # 1 = new infection
    T <- T + Tev
    if (ev==1) { # new infection
      S <- S - 1
      I <- I + 1
    } else { # removal
      I <- I - 1
      R <- R + 1
    }
```

```
    dfr[i, ] <- c(T, S, I, R, ev)
  }
  dfr <- as.data.frame(dfr)
  names(dfr) <- c("T", "S", "I", "R", "ev")
  dfr <- subset(dfr, S + I + R > 0)
  attr(dfr, "tpinf_start") <- t1
  attr(dfr, "tpinf_stop") <- t2
  return(dfr)
}
```

We now generate data according to this SIR model with intervention. The intervention is started as soon as 25% of the population is infected, and suspended as soon as that percentage has dropped to 10%. The effect of the intervention is 80%, in that the original transmission parameter β is 2, and after intervention it is $2 \cdot 0.2 = 0.4$.

```
set.seed(2023)
# Parameters
beta <- 2
gamma <- 0.5
# Total population
n <- 1000
# Effect of intervention
effect <- 0.2
# First and second intervention when 5 and 10% have been infected
pinf_start <- 0.25
pinf_stop <- 0.1
```

Figure 6 shows what the outbreak looks like. The time points where the intervention is implemented (when 25% of the population is infected) and stopped (when 10% of the population is infected) are indicated at the bottom.

```
dfr <- SIR2(beta = beta, gamma = gamma, n = n,
            effect = effect,
            pinf_start = pinf_start, pinf_stop = pinf_stop)
t1 <- attr(dfr, "tpinf_start")
t2 <- attr(dfr, "tpinf_stop")
plot(dfr$T, dfr$I, type="s", lwd=2, ylim=c(0, n), col="red",
     xlab="Time", ylab="Cumulative number of infecteds")
lines(dfr$T, dfr$I + dfr$R, type="s", lwd=2)
lines(dfr$T, dfr$I + dfr$R + dfr$S, type="s", lwd=2, col="blue")
legend(5, 0.8, c("I", "I + R", "I + R + S"), lwd=2,
       col=c("red", "black", "blue"), bty="n", xjust=1)
lines(c(t1, t1), c(0, pinf_start * n), lty=3)
lines(c(t2, t2), c(0, pinf_stop * n), lty=3)
```

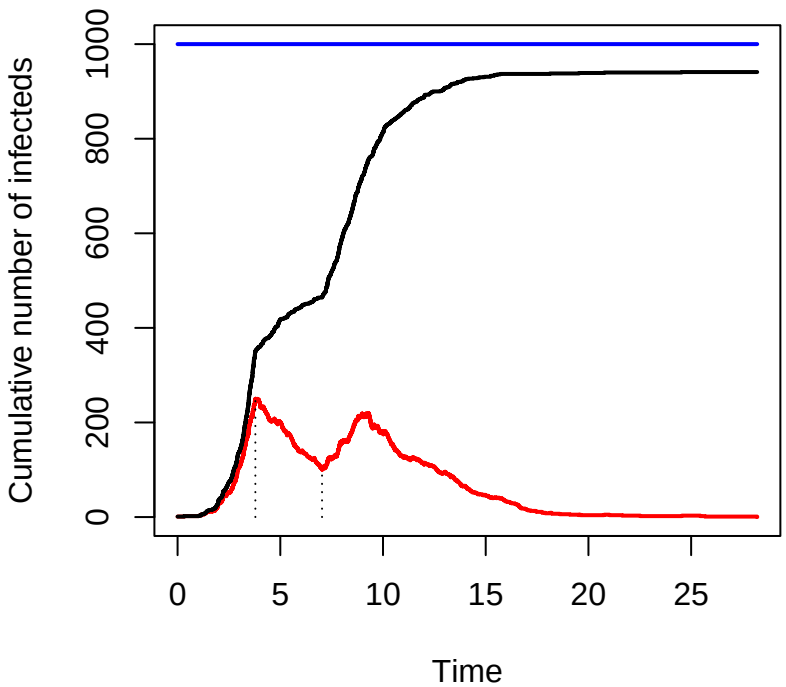


Figure 6: Stacked plot showing the number of susceptible, infected and recovered individuals over time, with intervention start and end indicated.

We see that after the intervention has been suspended the number of infections rises again, almost to 25%, but it decreases again just after that. We can again use the `SIR2surv` function to convert to data for use with Cox and Poisson modelling.

```
dlong <- SIR2surv(dfr)
head(dlong)
```

```
# A tibble: 6 x 9
     id     w tstart  time status  pinf logpinf fuptime logfuptime
  <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>
1     2     1  0     0.323      1 0.001   -6.91   0.323     -1.13
2     3     1  0     0.323      0 0.001   -6.91   0.323     -1.13
3     3     1  0.323 1.03       1 0.002   -6.21   0.711     -0.341
4     4     1  0     0.323      0 0.001   -6.91   0.323     -1.13
5     4     1  0.323 1.03       0 0.002   -6.21   0.711     -0.341
6     4     1  1.03  1.14       1 0.003   -5.81   0.106     -2.24
```

```
tail(dlong)
```

```
# A tibble: 6 x 9
     id     w tstart  time status  pinf logpinf fuptime logfuptime
  <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl>   <dbl>   <dbl>      <dbl>
1   942    59   21.0  21.6      0 0.004   -5.52   0.627    -0.467
2   942    59   21.6  22.5      0 0.003   -5.81   0.879    -0.129
3   942    59   22.5  24.5      0 0.002   -6.21   2.02      0.701
4   942    59   24.5  25.5      0 0.003   -5.81   0.910    -0.0939
5   942    59   25.5  25.7      0 0.002   -6.21   0.220    -1.51
6   942    59   25.7  28.2      0 0.001   -6.91   2.54      0.933
```

```r
dim(dlong)
```

```
[1] 746220      9
```

```r
# Make concise version
dlong <- as_tibble(dlong) %>%
  group_by(tstart, time, status) %>%
  summarize(pinf = min(pinf), fuptime = min(fuptime),
            w = sum(w)) %>%
  mutate(logpinf = log(pinf), logfuptime = log(fuptime)) %>%
  ungroup()
```

```
`summarise()` has grouped output by 'tstart', 'time'. You can override using
the `.groups` argument.
```

```r
head(dlong)
```

```
# A tibble: 6 x 8
  tstart  time status  pinf fuptime     w logpinf logfuptime
   <dbl> <dbl>  <dbl> <dbl>   <dbl> <dbl>   <dbl>      <dbl>
1  0     0.323      0 0.001   0.323   998   -6.91     -1.13
2  0     0.323      1 0.001   0.323     1   -6.91     -1.13
3  0.323 1.03       0 0.002   0.711   997   -6.21     -0.341
4  0.323 1.03       1 0.002   0.711     1   -6.21     -0.341
5  1.03  1.14       0 0.003   0.106   996   -5.81     -2.24
6  1.03  1.14       1 0.003   0.106     1   -5.81     -2.24
```

```r
tail(dlong)
```

```
# A tibble: 6 x 8
  tstart  time status  pinf fuptime     w logpinf logfuptime
   <dbl> <dbl>  <dbl> <dbl>   <dbl> <dbl>   <dbl>      <dbl>
1   21.6  22.5      0 0.003   0.879    60   -5.81    -0.129
2   22.5  24.5      0 0.002   2.02     59   -6.21     0.701
3   22.5  24.5      1 0.002   2.02      1   -6.21     0.701
4   24.5  25.5      0 0.003   0.910    59   -5.81    -0.0939
5   25.5  25.7      0 0.002   0.220    59   -6.21    -1.51
6   25.7  28.2      0 0.001   2.54     59   -6.91     0.933
```

```r
dim(dlong)
```

```
[1] 2821    8
```

Note that we have included 59 subjects that were at risk for infection and never got infected during the epidemic. In the data it occurs as one subject with weight $w = 59$.

Let's see whether we can pick up the changes in infection rates without knowing the times at which the interventions were in effect. Our first attempt is with a non-parametric estimate of the cumulative rate, using `coxph()`, shown in Figure 7.

```r
c0 <- coxph(Surv(tstart, time, status) ~
              offset(logpinf),
            data=dlong, weights=w)
bh0 <- basehaz(c0, centered=FALSE)
bh0 <- bh0[!duplicated(bh0$hazard), ]
# Divide by weighted average of offset variable
scaling_factor <- exp(weighted.mean(dlong$logpinf, dlong$w))
bh0$hazard <- bh0$hazard / scaling_factor
plot(c(0, bh0$time), c(0, bh0$hazard), type="s",
     xlab="Time", ylab="Cumulative hazard")
lines(c(t1, t1), c(0, 11), lty=3)
lines(c(t2, t2), c(0, 12), lty=3)
```

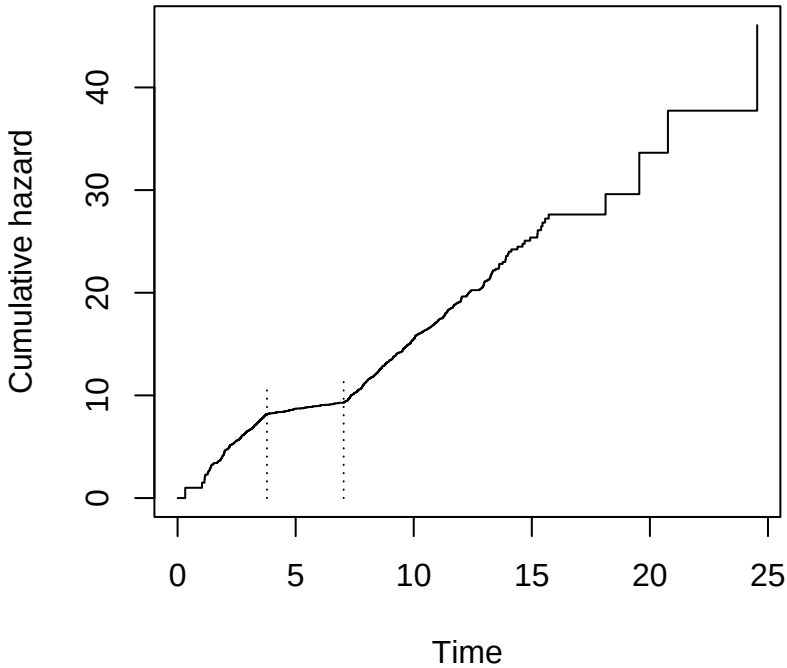


Figure 7: Baseline hazard of the Cox model under intervention.

Knowing the time of the intervention, a decreasing rate of infection can be seen, including an increase to the rate before intervention after the intervention has been suspended. We can also use Poisson regression again, with natural splines (same choice of knots as before, based on quartiles), to obtain a smooth estimate of $\beta(t)$ (note, however, that the true $\beta(t)$ is piecewise constant, not smooth).

```
tinf <- sort(dfr$T[dfr$ev == 1])
t.kn <- summary(tinf)[-3] # excluding median
poisfit2 <- glm(status ~ Ns(time, knots=t.kn) +
                  offset(logfuptime) + offset(logpinf),
                family="poisson", data=dlong,
                weights = w)
summary(poisfit2)
```

```
Call:
glm(formula = status ~ Ns(time, knots = t.kn) + offset(logfuptime) +
    offset(logpinf), family = "poisson", data = dlong, weights = w)

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)               2.4397     0.2722   8.961  < 2e-16 ***
```

```
Ns(time, knots = t.kn)1  -3.0403     0.2567 -11.843  < 2e-16 ***
Ns(time, knots = t.kn)2   1.3552     0.3225   4.202 2.64e-05 ***
Ns(time, knots = t.kn)3  -5.7026     0.7405  -7.701 1.35e-14 ***
Ns(time, knots = t.kn)4  -5.7078     1.2121  -4.709 2.49e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 14045  on 2820  degrees of freedom
Residual deviance: 13828  on 2816  degrees of freedom
AIC: 15718

Number of Fisher Scoring iterations: 9
```

This time some of the non-constant effects of the splines are significant, pointing towards a non-constant infection rate. Let us make a plot of the infection rate $\beta(t)$. The true $\beta(t)$ is shown as dotted lines, in Figure 8.

```r
tt <- seq(0, 25, by=0.05)
nd <- data.frame(time=tt, logfuptime=0, logpinf=0)
lambda <- ci.pred(poisfit2, nd) # the rates from the spline model
par(mfrow=c(1, 1))
matshade(tt, lambda, plot=TRUE, col="blue", lwd=2, ylim = c(0, 12),
         xlab="Time", ylab="Infection rate")
lines(c(0, t1), rep(beta, 2), lty=3)
lines(c(t1, t2), rep(beta * effect, 2), lty=3)
lines(c(t2, max(tt)), rep(beta, 2), lty=3)
lines(c(t1, t1), c(0, 0.0002), lty=3)
lines(c(t2, t2), c(0, 0.0002), lty=3)
```

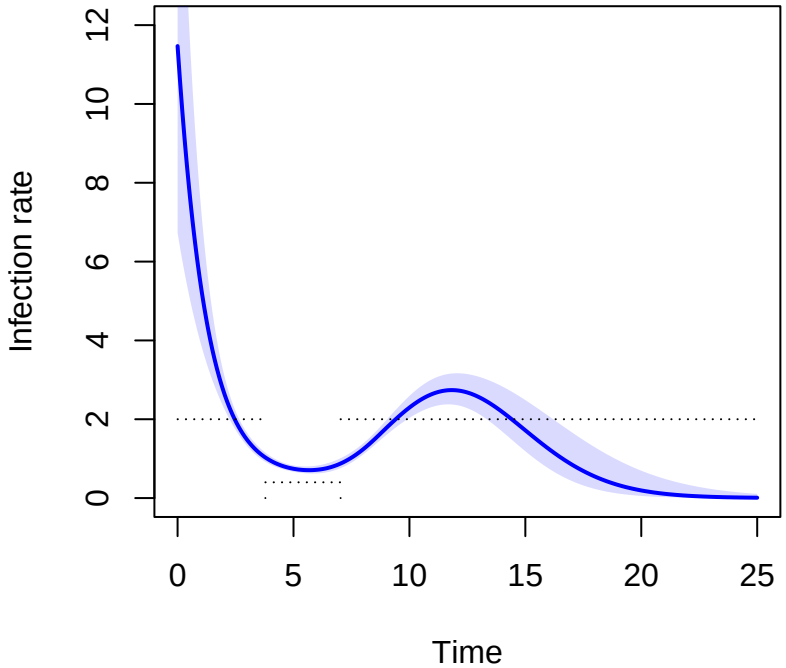


Figure 8: Time course of infection rate.

Because the natural splines try to impose a smooth line through a time-dependent rate that is inherently piecewise constant, the fitted curve is smoothly decreasing from above $\beta = 2$ to about $\beta\eta = 2 * 0.2 = 0.4$, and back to $\beta = 2$.

To be able to study the effect of the interventions, knowing when they were put in place and later suspended, we have to add information about the interventions to the data.

```
dlong$intervention <- 0
dlong$intervention[dlong$tstart > t1] <- 1
dlong$intervention[dlong$tstart > t2] <- 2
dlong$interventioncat <- factor(dlong$intervention,
                                levels = 0:2,
                                labels = c("No", "Intervention", "End"))
```

We start again with the epidemiological occurrence to exposure, this time by intervention period.

```
OE <- dlong %>%
  group_by(interventioncat) %>%
  summarise(O = sum(status * w),
            E = sum(pinf * fuptime * w)) %>%
```

```r
    mutate(rate = O / E,
           lograte = log(rate),
           SElograte = 1 / sqrt(O),
           lower = exp(lograte - qnorm(0.975) * SElograte),
           upper = exp(lograte + qnorm(0.975) * SElograte))
  OE
```

```
# A tibble: 3 x 8
  interventioncat     O     E  rate lograte SElograte lower upper
  <fct>           <dbl> <dbl> <dbl>   <dbl>     <dbl> <dbl> <dbl>
1 No                349  166. 2.10    0.743    0.0535 1.89  2.33
2 Intervention      115  329. 0.349  -1.05     0.0933 0.291 0.419
3 End               476  235. 2.02    0.705    0.0458 1.85  2.21
```

We nicely see that the estimated rates correspond reasonably to the infection rates of 2, 0.4, and 2 in the three periods.

A weighted Cox regression with intervention as categorical covariate (and logarithm of number of infected individual as offset), gives `NA`'s for estimates and standard errors (not shown). This is because the intervention is applied for all subjects in the same period, and hence the intervention effect is confounded with the baseline hazard. The intervention effect would be identifiable if different subjects would experience the interventions at different points in time.

The intervention effect is also identifiable with parametric restrictions on the baseline hazard, like a piecewise constant assumption. We now use weighted Poisson regression, again with log of interval time and log of proportion of infected individuals as intercept, and intervention as categorical covariate.

```r
  poisfit <- glm(status ~ interventioncat + offset(logfuptime) + offset(logpinf),
                 family="poisson", data=dlong,
                 weights = w)
  summary(poisfit)
```

```
Call:
glm(formula = status ~ interventioncat + offset(logfuptime) +
    offset(logpinf), family = "poisson", data = dlong, weights = w)

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)                 0.74272    0.05353  13.875   <2e-16 ***
interventioncatIntervention -1.79436    0.10752 -16.688   <2e-16 ***
interventioncatEnd          -0.03771    0.07047  -0.535    0.593
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 14045  on 2820  degrees of freedom
Residual deviance: 13571  on 2818  degrees of freedom
AIC: 15457

Number of Fisher Scoring iterations: 10
```

```r
ci.exp(poisfit)
```

```
                            exp(Est.)      2.5%    97.5%
(Intercept)                 2.1016431 1.8923226 2.334118
interventioncatIntervention 0.1662333 0.1346466 0.205230
interventioncatEnd          0.9629888 0.8387570 1.105621
```

This retrieves the occurrence/exposure estimates, albeit with a different parametrization. The β in the pre-intervention period is the exponent of the intercept; the estimated regression parameter -1.794 is the contrast between the $\log \beta$ of the intervention period and the $\log \beta$ of the pre-intervention period (the intercept). The β in the intervention period, given by 0.349, can therefore be retrieved as the exponent of 0.743 + -1.794, which indeed equals 0.349. The β in the post-intervention period can be determined similarly.

The baseline infection rate, as well as the intervention effect, `effect` = 0.2 is nicely recovered.

4.3 Additive hazards models

An interesting application where additive hazards occur very naturally is one where the population can be sub-divided into groups, such as age groups, occupation, households, schools, for instance, and where infections can occur between susceptibles and infected individuals within the same group or across groups.

The figure below shows the number of transmissions across different age groups in the Netherlands in March 2022. It nicely shows the number of transmissions are highest across similar age groups and after that across age groups differing by one generation (most probably transmissions within the same household). But they are numbers, some of which could be higher or lower simply because the groups sizes are different. The objective would be to estimate the infection rates across age groups.

We thus consider multiple groups (for instance age groups) that can infect each other. For group $j = 1, \ldots, J$ define $S_j(t)$, $I_j(t)$, $R_j(t)$ to be the total number of susceptible, infected and recovered individuals in group j at time t, and denote the total number of subjects in group j by $n_j = S_j(t) + I_j(t) + R_j(t)$. We again assume that the groups are closed.

Within group j, new infections happen with rate $S_j(t) \sum_{k=1}^{J} \beta_{kj}(t) \overline{I}_k(t)$, with $\overline{I}_k(t) = I_k(t)/n_j$. The idea is that each of the susceptibles in group j at time t, of which there are $S_j(t)$ can be infected by an infected individual from within group j itself or from within one of the other group. The (potentially time-varying) infection rate parameter $\beta_{kj}(t)$ describes the intensity at which contacts are made between an infected individual from group k and a susceptible individual from group j leading to a new infection. Finally we define the counting processes $N_j(t)$, $j = 1, \ldots, J$, counting the total number of susceptibles in group j becoming infected within $(0, t]$.

Interest is in estimating the (possibly time-varying) transmission parameters $\beta_{kj}(t)$. To illustrate methods for estimating β_{kj}, we are choosing a setup with two groups (think of it as young and old), where the transmission parameter for susceptibles and infectious from the same group are higher ($\beta_{11} = 3$, $\beta_{22} = 2$) than across groups ($\beta_{12} = 0.25$, $\beta_{21} = 0.5$). The following code generates data from this model, with total population sizes of $n_1 = 1000$ in group 1 and $n_2 = 750$ in group 2. The choice for γ is 0.5 in each of the groups. The model starts with one susceptible in each group and $n_j - 1$ susceptibles in group j, at time $t = 0$, and then generates new events (infection in group 1, infection in group 2, recovery in group 1, recovery in group 2, denoted with `ev` = 1, 2, 3, 4, respectively) according to a Poisson process. The first 12 lines of the resulting data are shown.

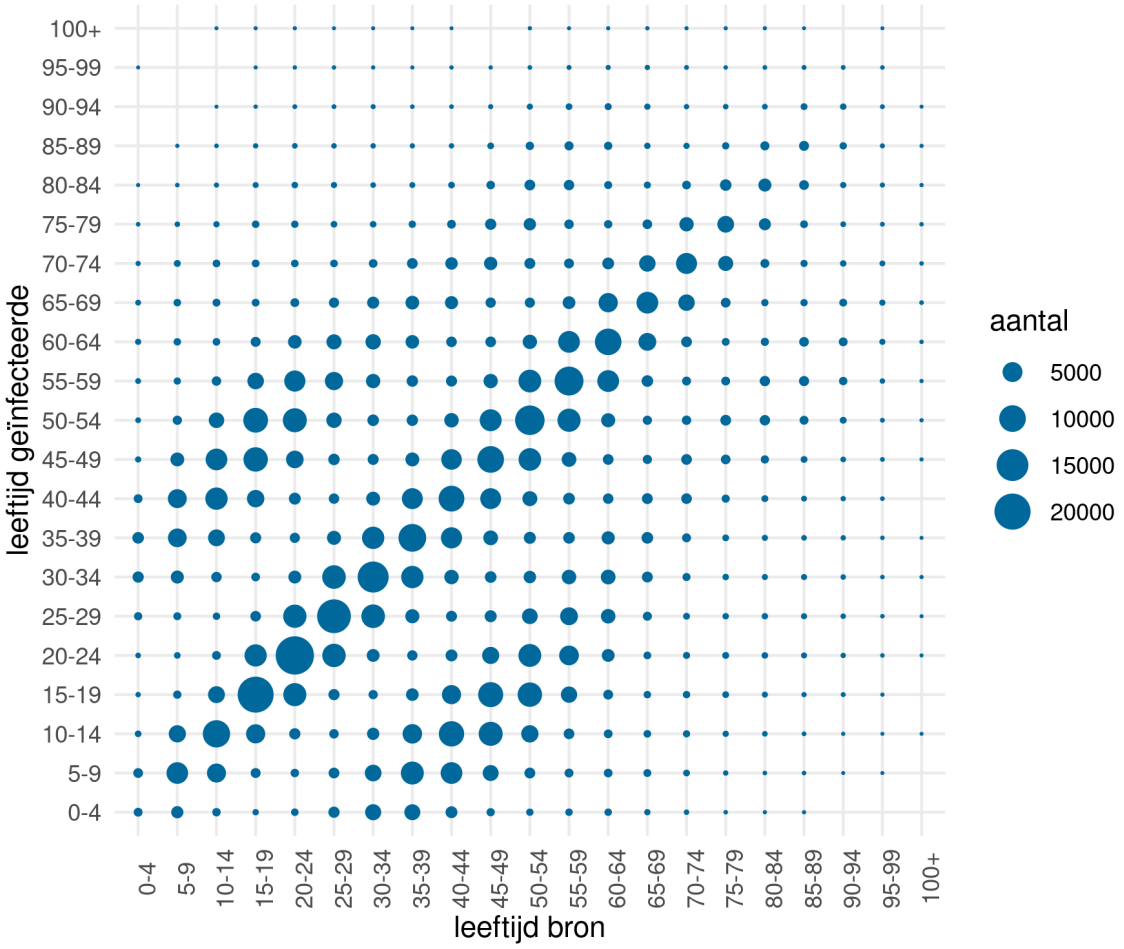


Figure 9: Transmissions across age groups in March 2022 in the Netherlands (leeftijd = age, bron = infector, geinfecteerde = infectee, aantal = number)

```r
#
# Function to generate data from an SIR model with two groups
#
gen_SIR_twogroups <- function(beta, gamma, S0, I0, R0) {
  # Start with S0 susceptible, I0 infected and R0 recovered, at time 0
  T <- 0
  I <- I0
  S <- S0
  R <- R0
  dfr <- matrix(0, 2*sum(n), 8) # time, S, I, R (group 1), S, I, R (group 2), ev
  dfr[1, ] <- c(T, S[1], I[1], R[1], S[2], I[2], R[2], 1)
  i <- 1
  while (sum(I) > 0) {
    i <- i + 1
    # currently I infected, S susceptibles
    rate_inf1 <- (beta[1, 1] * I[1] + beta[2, 1] * I[2]) * S[1] / n[1]
    rate_rem1 <- gamma[1] * I[1]
    rate_inf2 <- (beta[1, 2] * I[1] + beta[2, 2] * I[2]) * S[2] / n[2]
    rate_rem2 <- gamma[2] * I[2]
    rate_tot <- rate_inf1 + rate_inf2 + rate_rem1 + rate_rem2
    # time point of new event
    Tev <- rexp(1, rate_tot)
    ev <- sample(1:4, size=1, prob=c(rate_inf1, rate_inf2, rate_rem1, rate_rem2))
    T <- T + Tev
    if (ev==1) { # new infection in group 1
      S[1] <- S[1] - 1
      I[1] <- I[1] + 1
    }
    if (ev==2) { # new infection in group 2
      S[2] <- S[2] - 1
      I[2] <- I[2] + 1
    }
    if (ev==3) { # removal from group 1
      I[1] <- I[1] - 1
      R[1] <- R[1] + 1
    }
    if (ev==4) { # removal from group 2
      I[2] <- I[2] - 1
      R[2] <- R[2] + 1
    }
    dfr[i, ] <- c(T, S[1], I[1], R[1], S[2], I[2], R[2], ev)
  }
  dfr <- as.data.frame(dfr)
  names(dfr) <- c("T", "S1", "I1", "R1", "S2", "I2", "R2", "ev")
  dfr <- subset(dfr, I1+I2+ev>0)
  return(dfr)
}

# Parameters
beta <- matrix(c(3, 0.25, 0.5, 2), 2, 2)
```

```r
beta
```

```
     [,1] [,2]
[1,] 3.00  0.5
[2,] 0.25  2.0
```

```r
gamma <- c(0.5, 0.5)
# Total population (possibly different sample sizes)
n <- c(1000, 750)
# Starting values
I0 <- c(1, 1) # one susceptible each
S0 <- c(n[1]-1, n[2]-1) # all but one in each group are susceptible
R0 <- c(0, 0) # no one removed/recovered (yet)
# Generate
set.seed(2023)
dfr <- gen_SIR_twogroups(beta, gamma, S0, I0, R0)
head(dfr, n=12)
```

```
           T  S1 I1 R1  S2 I2 R2 ev
1  0.0000000 999  1  0 749  1  0  1
2  0.1195245 998  2  0 749  1  0  1
3  0.4502207 997  3  0 749  1  0  1
4  0.5043289 996  4  0 749  1  0  1
5  0.5096176 996  4  0 748  2  0  2
6  0.5215754 995  5  0 748  2  0  1
7  0.5630131 995  4  1 748  2  0  3
8  0.5828954 995  4  1 747  3  0  2
9  0.6033017 994  5  1 747  3  0  1
10 0.6312626 993  6  1 747  3  0  1
11 0.6473166 993  6  1 746  4  0  2
12 0.6602079 992  7  1 746  4  0  1
```

Figure 10 shows a plot (shown until $t = 2$) of the number of infected (solid line) and recovered (difference between dashed and solid lines) for group 1 (black) and group 2 (red) separately.

```r
plot(dfr$T, dfr$I1, type="s", lwd=2, ylim=c(0, 2000),
     xlab="Time", ylab="Cumulative number of infecteds",
     xlim=c(0, 10), col = "blue")
lines(dfr$T, dfr$I1 + dfr$R1, type="s", lwd=2, lty=2,
      col = "blue")
lines(dfr$T, dfr$I2, type="s", lwd=2, col="red")
lines(dfr$T, dfr$I2 + dfr$R2, type="s", lwd=2, lty=2, col="red")
lines(dfr$T, dfr$I1 + dfr$I2, type="s", lwd=2)
lines(dfr$T, dfr$I1 + dfr$I2 + dfr$R1 + dfr$R2,
      type="s", lwd=2, lty=2)
legend("topleft", c("Group 1", "Group 2", "Total"), lwd = 2,
       col = c("blue", "red", "black"), bty = "n")
```

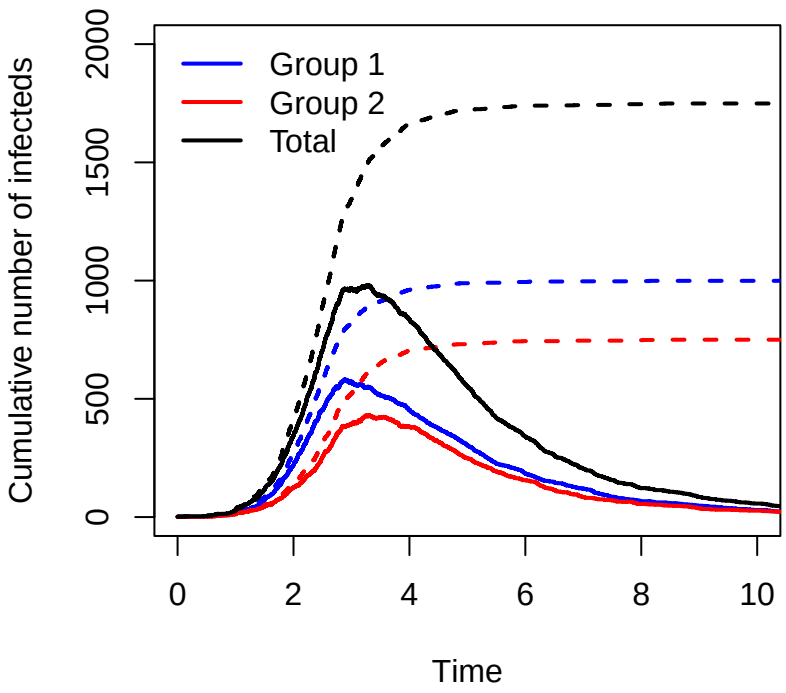


Figure 10: Number of infected and recovered individuals over time within group 1 (blue) and group 2 (red); total numbers in black.

We can adapt the `SIR2surv()` function to the situation of two groups.

```
SIR2surv_groups <- function(SIRdata, group)
{
  if (group == 1)
    n <- SIRdata$S1[1] + SIRdata$I1[1] + SIRdata$R1[1] # first extract the total size
  else if (group == 2)
    n <- SIRdata$S2[1] + SIRdata$I2[1] + SIRdata$R2[1] # first extract the total size
  if (group == 1)
    wh <- which(SIRdata$ev == 1) # select the infection events
  else if (group == 2)
    wh <- which(SIRdata$ev == 2) # select the infection events
  ninf <- length(wh) # number of observed infections in the time window
  tinf <- SIRdata$T[wh]
  d <- data.frame(id = 1:ninf, time = tinf, status = 1)
  d$w <- 1 # give weight 1 to observed infections
  # First is not really an observed event, so remove
  d <- d[-1, ]
```

```
  # Add the rest of the population to the data with number of never infecteds
  d <- rbind(d, data.frame(id=ninf+1, time=max(SIRdata$T), status=0, w=n-ninf))
  # Prepare long data format
  tt <- SIRdata$T
  tt <- tt[tt <= max(tinf)]
  dlong <- survSplit(Surv(time, status) ~ ., data=d, cut=tt[-1])
  # Add proportions of infecteds as time-dependent covariate
  dlong$pinf1 <- SIRdata$I1[match(dlong$tstart, SIRdata$T)] / n
  dlong$logpinf1 <- log(dlong$pinf1)
  dlong$pinf2 <- SIRdata$I2[match(dlong$tstart, SIRdata$T)] / n
  dlong$logpinf2 <- log(dlong$pinf2)
  dlong$fuptime <- dlong$time - dlong$tstart # length of follow-up interval
  dlong$logfuptime <- log(dlong$fuptime)
  dlong <- subset(dlong, w>0)
  return(dlong)
}
dlong1 <- SIR2surv_groups(dfr, group = 1)
dlong2 <- SIR2surv_groups(dfr, group = 2)
```

Again we can look from the individual perspective, define a counting process for each individual i from each group j, $N_{ji}(t)$, having rate $Y_{ji}(t)\sum_{k=1}^{J}\beta_{kj}(t)\overline{I}_k(t)$, where $\overline{I}_k(t) = I_k(t)/n_j$, with n_j the size of group j. This is an *additive hazards model* without intercept, and J time-dependent covariates $\overline{I}_k(t)$, where $Y_{ji}(t)$ is the at risk indicator of subject i in group j for being susceptible to infection.

To simplify notation, consider one group j of susceptibles. We will fix that group, suppress j in the notation everywhere, and let n be the size of that group. Within this group, individual i has rate

$$\lambda_i(t) = Y_i(t)\sum_{k=1}^{J}\beta_k(t)\overline{I}_k(t). \tag{8}$$

This rate conforms to the additive hazards model with rate $Y_i(t)\{\beta_0(t) + \sum_{k=1}^{J}\beta_k(t)X_{ik}(t)\}$, with two non-standard aspects. The first is that in Equation 8 there is no intercept $\beta_0(t)$, the second is that in Equation 8 for each time point t the time-dependent covariates $X_{ik}(t)$ are the same for all susceptible individuals. This has the important implication that the kth column of the matrix $\mathbf{X}(t)$ used in Equation 4 is of the form $\mathbf{Y}(t)\overline{I}_k(t)$, $\mathbf{Y}(t) = (Y_1(t), \ldots, Y_n(t))^\top$. As a result $\mathbf{X}(t)$ is of rank 1, which implies that the $\beta_k(t)$'s are not identifiable from the data when the $\beta_k(t)$'s are allowed to vary freely. To be able to estimate the $\beta_k(t)$'s one would need some kind of smoothing or restricting the $\beta_k(t)$'s to be constant or piecewise constant on time intervals where for instance interventions are put into place and/or suspended.

In the more restricted model where $\beta_k(t)$'s are assumed to be constant, the β_k's *are* identifiable. Two approaches are possible to estimate the β_k coefficients. The first is maximum likelihood, as an extension of Section 2.2. Define $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)^\top$, $\mathbf{X}_i(t) = (X_{i1}(t), \ldots, X_{iJ}(t))$, which here simplifies to $\mathbf{X}_i(t) = \overline{\mathbf{I}}(t) = (\overline{I}_1(t), \ldots, \overline{I}_J(t))$. The log-likelihood of $\boldsymbol{\beta}$ is given by

$$\ell(\boldsymbol{\beta}) = \int_0^\tau \Big[\log\{S(t)\beta^\top\overline{\mathbf{I}}(t)\}\mathrm{d}N(t) - S(t)\beta^\top\overline{\mathbf{I}}(t)\mathrm{d}t\Big].$$

Taking derivatives with respect to the elements of $\boldsymbol{\beta}$ and setting to zero gives

$$\int_0^\tau \frac{\overline{I}_k(t)}{\beta^\top\overline{\mathbf{I}}(t)}\mathrm{d}N(t) = \int_0^\tau S(t)\overline{I}_k(t)\mathrm{d}t,$$

for $k = 1, \ldots, J$, which can be solved numerically. The second is an extension of the approach of Lin and Ying (1994). The latter allows to use

$$U(\boldsymbol{\beta}) = \sum_{i=1}^{n} \int_0^{\tau} \mathbf{X}_i(t) \left\{ \mathrm{d}N_i(t) - Y_i(t)\boldsymbol{\beta}^\top \mathbf{X}_i(t)\mathrm{d}t \right\}$$

as estimating equations, with $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_J)^\top$, and $\mathbf{X}_i(t) = (X_{i1}(t), \ldots, X_{iJ}(t))$, which here simplifies to $\mathbf{X}_i(t) = \bar{\mathbf{I}}(t) = (\bar{I}_1(t), \ldots, \bar{I}_J(t))$, independent of i.

This leads to

$$\widehat{\boldsymbol{\beta}} = \mathbf{A}^{-1}\mathbf{C},$$

with $\mathbf{A}$ a $J \times J$ matrix and $\mathbf{C}$ a J-vector given by

$$\mathbf{A} = n^{-1} \sum_{i=1}^{n} \int_0^{\tau} Y_i(t)\mathbf{X}_i(t)\mathbf{X}_i(t)^\top \mathrm{d}t,$$

and

$$\mathbf{C} = n^{-1} \sum_{i=1}^{n} \int_0^{\tau} \mathbf{X}_i(t)\mathrm{d}N_i(t).$$

Furthermore the variance-covariance matrix of $\widehat{\boldsymbol{\beta}}$ can be consistently estimated by $n^{-1}\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1}$, with $\mathbf{B}$ a $J \times J$ matrix given by

$$\mathbf{B} = n^{-1} \sum_{i=1}^{n} \int_0^{\tau} \mathbf{X}_i(t)\mathbf{X}_i(t)^\top \mathrm{d}N_i(t).$$

Replacing $\mathbf{X}_i(t)$ by $\bar{\mathbf{I}}(t)$, $Y_i(t)$ by $S_i(t)$, and summing over i, we get simpler versions of the matrices $\mathbf{A}$ and $\mathbf{B}$ and of the vector $\mathbf{C}$, namely

$$\mathbf{A} = n^{-1} \int_0^{\tau} S(t)\bar{\mathbf{I}}(t)\bar{\mathbf{I}}(t)^\top \mathrm{d}t, \quad \mathbf{B} = n^{-1} \int_0^{\tau} \bar{\mathbf{I}}(t)\bar{\mathbf{I}}(t)^\top \mathrm{d}N(t), \quad \mathbf{C} = n^{-1} \int_0^{\tau} \bar{\mathbf{I}}(t)\mathrm{d}N(t).$$

The code below implements these estimators and their standard errors, first in group 1, then in group 2.

```
# In group 1
C1 <- sum(dlong1$pinf1 * dlong1$status) / n[1]
C2 <- sum(dlong1$pinf2 * dlong1$status) / n[1]
B11 <- sum(dlong1$pinf1 * dlong1$pinf1 * dlong1$status) / n[1]
B12 <- sum(dlong1$pinf1 * dlong1$pinf2 * dlong1$status) / n[1]
B22 <- sum(dlong1$pinf2 * dlong1$pinf2 * dlong1$status) / n[1]

tmp <- dlong1 %>%
  group_by(time) %>%
  summarise(S11 = sum(pinf1 * pinf1),
            S12 = sum(pinf1 * pinf2),
            S22 = sum(pinf2 * pinf2)) %>%
  mutate(dtime = time - lag(time, default=0))

A11 <- sum(tmp$dtime * tmp$S11) / n[1]
```

```r
A12 <- sum(tmp$dtime * tmp$S12) / n[1]
A22 <- sum(tmp$dtime * tmp$S22) / n[1]

A <- matrix(c(A11, A12, A12, A22), 2, 2)
B <- matrix(c(B11, B12, B12, B22), 2, 2)
C <- c(C1, C2)
betahat <- solve(A) %*% C
Sig <- solve(A) %*% B %*% solve(A) / n[1]
se <- sqrt(diag(Sig))
# Gather results
res1 <- data.frame(truebeta = beta[, 1], betahat = betahat, se = se,
                   lower = betahat - qnorm(0.975) * se,
                   upper = betahat + qnorm(0.975) * se)

# And in group 2
C1 <- sum(dlong2$pinf1 * dlong2$status) / n[2]
C2 <- sum(dlong2$pinf2 * dlong2$status) / n[2]
B11 <- sum(dlong2$pinf1 * dlong2$pinf1 * dlong2$status) / n[2]
B12 <- sum(dlong2$pinf1 * dlong2$pinf2 * dlong2$status) / n[2]
B22 <- sum(dlong2$pinf2 * dlong2$pinf2 * dlong2$status) / n[2]

tmp <- dlong2 %>%
  group_by(time) %>%
  summarise(S11 = sum(pinf1 * pinf1),
            S12 = sum(pinf1 * pinf2),
            S22 = sum(pinf2 * pinf2)) %>%
  mutate(dtime = time - lag(time, default=0))

A11 <- sum(tmp$dtime * tmp$S11) / n[2]
A12 <- sum(tmp$dtime * tmp$S12) / n[2]
A22 <- sum(tmp$dtime * tmp$S22) / n[2]

A <- matrix(c(A11, A12, A12, A22), 2, 2)
B <- matrix(c(B11, B12, B12, B22), 2, 2)
C <- c(C1, C2)
betahat <- solve(A) %*% C
Sig <- solve(A) %*% B %*% solve(A) / n[1]
se <- sqrt(diag(Sig))
# Gather results
res2 <- data.frame(truebeta = beta[, 2], betahat = betahat, se = se,
                   lower = betahat - qnorm(0.975) * se,
                   upper = betahat + qnorm(0.975) * se)
# Show results
# Group 1
res1
```

```
  truebeta   betahat       se     lower     upper
1     3.00  4.232807 0.641617  2.975261 5.4903531
2     0.25 -1.697600 1.009298 -3.675788 0.2805871
```

```
# Group 2
res2
```

```
  truebeta   betahat        se      lower    upper
1      0.5 0.5154944 0.3645516 -0.1990135 1.230002
2      2.0 2.0512047 0.5777674  0.9188014 3.183608
```

Two things are worth noting. The first is that the estimates are reasonably close to the true values, and well within the 95% confidence intervals. The second is that one of the coefficients is estimated to be negative, which of course is not desirable. It would be of interest to estimate the β coefficients under a non-negativity constraint, as pursued in Lu, Goeman, and Putter (2023).

4.4 Hybrid (Cox-Aalen) models

As we have seen, the effect of an intervention is most naturally incorporated as a multiplicative effect. The same goes for the effect of measured characteristics of the susceptible individuals, like gender, or perhaps known risk factors for infection. If we want to incorporate such effects multiplicatively and additionally have a structured population that would call for an additive hazards structure, then hybrid models would be of interest where the hazard of subject i in group j takes the form

$$\lambda_{ji}(t) = \exp(\gamma^\top X_{ji}) \sum_{k=1}^{J} \beta_{kj} \overline{I}_k(t),$$

with X_{ij} a vector of baseline covariates of subject i in group j would be of interest. This type of model is known as a Cox-Aalen model, and has been studied by Scheike and Zhang (2002). We will not pursue this method in this paper.

4.5 Heterogeneity in susceptibility to infection

It is a huge simplification to assume that each susceptible individual is equally susceptible to becoming infected, or that each infected individual is equally likely to infect others. With individual knowledge of covariates, these could be incorporated into the survival analysis models, be they additive hazards, Cox or Poisson models. In the absence of such information, a natural extension to the models considered is to add individual random effects expressing such heterogeneity. In survival analysis, models incorporating such random effects are known under the term *frailty models*, see for instance Balan and Putter (2020). It is possible to fit such frailty models also for SIR models.

The function below generates completely observed data from an SIR model where the transmission parameter associated with a susceptible individual equals βZ, with Z a gamma random variable with mean one and variance `fvar`. Thus, the mean transmission parameter over the population of susceptibles equals β, but susceptible individuals differ in their degree of susceptibility to infection.

```
#
# Function to generate data from an SIR model with heterogeneity
#
gen_SIR_Z <- function(beta, gamma, S0, I0, R0, fvar=1) {
  # Start with S0 susceptible, I0 infected and R0 recovered, at time t=0
  T <- 0
  S <- S0; I <- I0; R <- R0
  n <- S0 + I0 + R0
  dfr <- matrix(0, 2 * n, 5) # time, S, I, R, ev (1 for infection, 0 for recovery)
  # Now generate frailty terms
```

```r
  Z <- rgamma(S0, shape = 1/fvar, rate = 1/fvar) # mean one, variance fvar
  # Initialize
  dfr[1, ] <- c(T, S, I, R, 1)
  i <- 1
  while (I > 0) { # run until no more infecteds left
    i <- i + 1
    # currently I infected, S susceptibles, determine rates
    rate_inf <- beta * I * sum(Z) / n
    rate_rem <- gamma * I
    rate_tot <- rate_inf + rate_rem
    # time point of new event
    Tev <- rexp(1, rate_tot)
    # determine type of event
    ev <- sample(0:1, size = 1, prob = c(rate_rem, rate_inf))
    # if infection (ev=1) then
    T <- T + Tev
    if (ev==1) { # new infection
      # select (at random) which susceptible got infected and remove
      idx <- sample(1:S, size = 1, prob = Z)
      Z <- Z[-idx] # remove the infected individual from Z
      S <- S - 1
      I <- I + 1
    } else { # removal
      I <- I - 1
      R <- R + 1
    }
    dfr[i, ] <- c(T, S, I, R, ev)
  }
  dfr <- as.data.frame(dfr)
  names(dfr) <- c("T", "S", "I", "R", "ev")
  dfr <- subset(dfr, !(T == 0 & ev == 0))
  return(dfr)
}

# Generate
n <- 1000
beta <- 2
gamma <- 0.5
set.seed(2023)
dfr <- gen_SIR_Z(beta, gamma, I0 = 1, S0 = n-1, R0 = 0)
dfr2023 <- dfr # save copy for later
```

This heterogeneity in susceptibility leads to an epidemic with considerably fewer infections compared to one with the same β and γ and no such heterogeneity, as can be seen from Figure 11.

```r
plot(dfr$T, dfr$I, type="s", lwd=2, xlab="Time", ylab="",
     ylim=c(0, n), col="red")
lines(dfr$T, dfr$I + dfr$R, type="s", lwd=2)
lines(dfr$T, dfr$I + dfr$R + dfr$S, type="s", lwd=2, col="blue")
```

```
legend(5, 0.8, c("I", "I + R", "I + R + S"), lwd=2,
       col=c("red", "black", "blue"), bty="n", xjust=1)
```

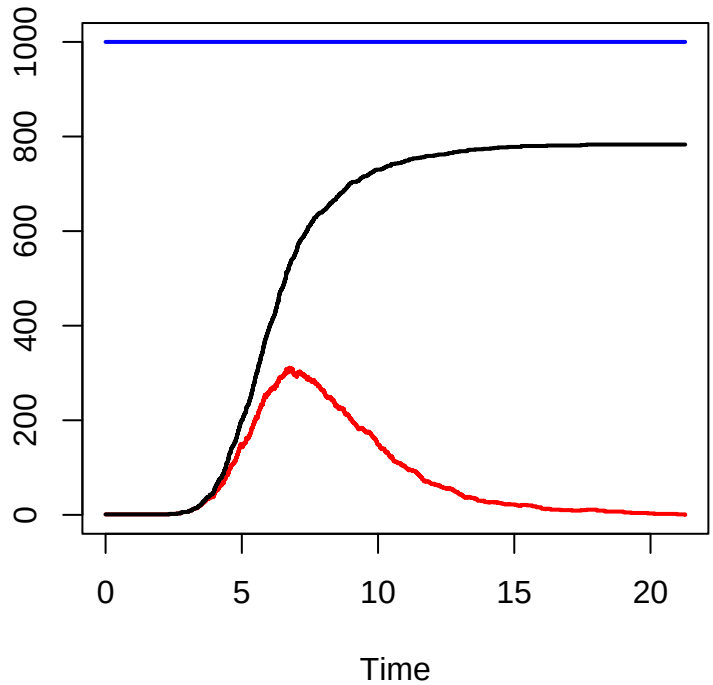


Figure 11: Stacked plot showing the number of susceptible, infected and recovered individuals over time from a stochastic SIR model with heterogeneity in susceptibility.

It can be shown by Jensen's inequality that the assumption that all individuals are equally susceptible leads to an upper bound for the final proportion infected, see for instance Katriel (2012) and Miller (2012).

Let us now estimate β from this generated data, first assuming it is time constant.

```
dlong <- SIR2surv(dfr)
poisfit0 <- glm(status ~ offset(logfuptime) + offset(logpinf),
                family="poisson", data=dlong,
                weights = w)
summary(poisfit0)
```

```
Call:
glm(formula = status ~ offset(logfuptime) + offset(logpinf),
    family = "poisson", data = dlong, weights = w)
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.07530    0.03576   2.106   0.0352 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 11915  on 464706  degrees of freedom
Residual deviance: 11915  on 464706  degrees of freedom
AIC: 13481

Number of Fisher Scoring iterations: 9
```

```r
cipoisfit0 <- confint(poisfit0)
tmp <- c(poisfit0$coef, cipoisfit0)
exp(tmp)
```

```
(Intercept)       2.5 %      97.5 %
   1.078203    1.004388    1.155547
```

The result, 1.08, is considerably lower than the true value 2, caused by the fact that the most susceptible infividuals get infected first, resulting in less susceptible individuals remainin in the susceptible pool over time. As in standard survival analysis, the presence of the frailty terms induces time-varying behaviour of β. This is indeed picked up using Poisson regression with splines.

```r
tinf <- sort(dfr$T[dfr$ev == 1])
t.kn <- summary(tinf)[-3] # excluding median
poisfit1 <- glm(status ~ Ns(time, knots=t.kn) + offset(logfuptime) + offset(logpinf),
                family="poisson", data=dlong,
                weights = w)
summary(poisfit1)
```

```
Call:
glm(formula = status ~ Ns(time, knots = t.kn) + offset(logfuptime) +
    offset(logpinf), family = "poisson", data = dlong, weights = w)

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)              0.06701    0.92263   0.073   0.9421
Ns(time, knots = t.kn)1  0.01332    0.89069   0.015   0.9881
Ns(time, knots = t.kn)2 -1.28876    0.53126  -2.426   0.0153 *
Ns(time, knots = t.kn)3  0.12176    1.92596   0.063   0.9496
Ns(time, knots = t.kn)4 -1.15869    0.53328  -2.173   0.0298 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 11915  on 464706  degrees of freedom
Residual deviance: 11792  on 464702  degrees of freedom
```

```
AIC: 13366

Number of Fisher Scoring iterations: 9
```

We indeed see non-constant $\beta(t)$ in Figure 12.

```r
tt <- seq(0, 15, by=0.05)
nd <- data.frame(time=tt, logfuptime=0, logpinf=0)
lambda <- ci.pred(poisfit1, nd) # the rates from the spline model
matshade(tt, lambda, plot=TRUE, col="blue", lwd=2,
         xlab="Time", ylab="Infection rate")
```

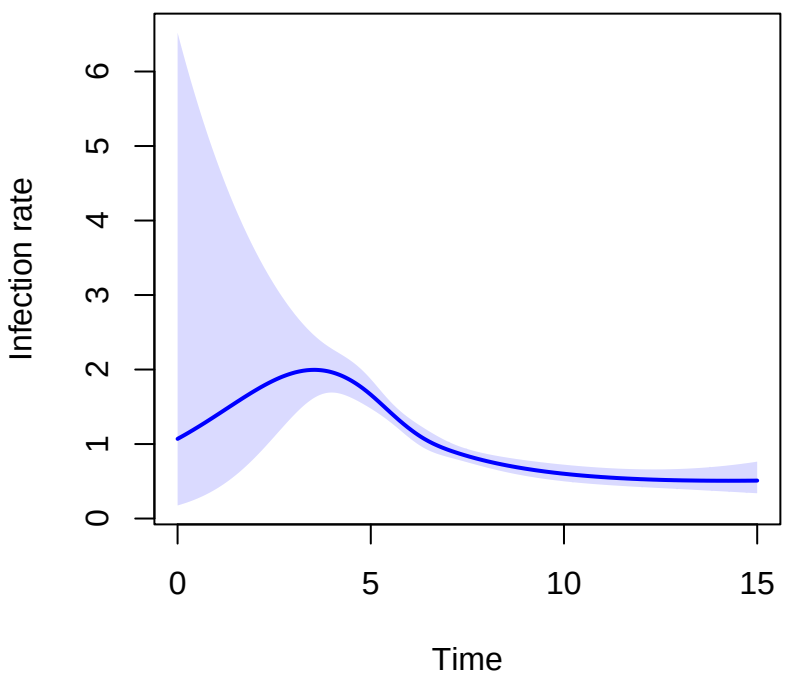


Figure 12: Estimate of $\beta(t)$ obtained using Poisson regression with cubic splines from an SIR model with heterogeneity in susceptibility.

Fitting a Cox model with individual frailty terms will not work, because the individual model with individual frailty is not identifiable in the absence of time-fixed covariates. A Poisson model with individual frailty terms seems like a viable option, but all the methods in https://rpubs.com/kaz_yos/glmm1 throw errors. It should be feasible though to write a dedicated EM-algorithm for this, but this is not pursued here.

5 Challenges

The type of data that we considered in this paper is too idealized in the sense that we typically do not know the time of infections exactly and we do not know the number of infectious and susceptible individuals exactly. Numbers of infections are usually aggregated over days or weeks. Moreover they are typically reported with some delay and they are incomplete. Methodology needs to be extended to deal with these more realistic data settings. We will not cover all of these aspects here, but we will show how the methods in this paper can still be used to estimate the transmission parameter β in case the (correct) probability distribution of the time to recovery after infection is known, and aggregate data on the daily number of newly infected individuals is reported, as was the case for the COVID-19 infection. For now we assume there is no reporting delay or incompleteness.

We start from the fully observed data used before. To make it roughly in line with the recent COVID-19 epidemic, based on Figure 2 it seems reasonable to say that the time unit there is months, and that we have daily updates of the number of new infections. Let us say that after 10 infections the outbreak is "detected" (at day 41), and that we set this to day 0 and start reporting daily new infections after that.

```
dfr <- dfr2023
# Time is now in months, convert time to days and then aggregate in days
dfr$T <- dfr$T * 30
head(dfr, n = 15)
```

```
          T   S  I R ev
1   0.00000 999  1 0  1
2  70.19984 998  2 0  1
3  78.49197 997  3 0  1
4  82.72576 996  4 0  1
5  82.78907 995  5 0  1
6  84.71693 994  6 0  1
7  90.82812 993  7 0  1
8  91.41363 992  8 0  1
9  92.33939 991  9 0  1
10 94.75551 990 10 0  1
11 95.48714 989 11 0  1
12 95.86087 988 12 0  1
13 96.58542 987 13 0  1
14 98.59173 986 14 0  1
15 99.67859 985 15 0  1
```

```
dfr <- subset(dfr, I > 10)
dfr$T <- ceiling(dfr$T)
dfr$T <- dfr$T - min(dfr$T) + 1
dfr <- subset(dfr, ev == 1)
tau <- max(dfr$T)
aggr <- as_tibble(dfr) %>%
  group_by(T) %>%
  summarize(newinf = n()) %>%
  rename(day = T)
aggr
```

```
# A tibble: 232 x 2
     day newinf
   <dbl>  <int>
 1     1      2
 2     2      1
 3     4      1
 4     5      1
 5     7      3
 6     8      1
 7     9      1
 8    10      2
 9    11      1
10    12      4
# i 222 more rows
```

This is what these aggregate data look like, with `newinf` the number of new infections reported each day.

```
aggr
```

```
# A tibble: 232 x 2
     day newinf
   <dbl>  <int>
 1     1      2
 2     2      1
 3     4      1
 4     5      1
 5     7      3
 6     8      1
 7     9      1
 8    10      2
 9    11      1
10    12      4
# i 222 more rows
```

Using the information on time to recovery, each newly infected individual on day d will be counted as an infected individual on day $d + j$ with probability $p_j = P(\text{time to recovery} \geq j)$. The expected number of infected individuals on day $\tilde{d}$ then is a sum of the number of newly infected individuals on day $\tilde{d} - j$, weighted by p_j. We can then create a daily analayis data set where each day is represented by a number of individuals becoming infected with `status` $= 1$ (the number of newly infected individuals on that day), and a number of individuals that are susceptible and did not get infected on that day with `status` $= 0$ (the population size n minus the cumulative number of infected individuals until and including that day).

```
# Vector p, with p[j] containing probability of still being infectious
# j-1 days after infection
p <- 1 - pexp(0:tau, rate = gamma / 30)
# Structure for analysis data, "empty" rows will be deleted later
ana <- matrix(0, 2 * tau, 5) # columns are start, stop, ninf, status, weight
# Initialize
I0 <- 10
```

```r
vnewinf <- cuminf <- I0
for(i in 1:tau) {
  # Number of infected at start of bin (day)
  ninf <- c(crossprod(vnewinf, rev(p[1:i])))
  # Number of newly infected
  idxi <- match(i, aggr$day)
  newinf <- ifelse(is.na(idxi), 0, aggr$newinf[idxi]) # current newly infected
  cuminf <- cuminf + newinf # cumulative number of infected
  vnewinf <- c(vnewinf, newinf) # add newinf to vector vnewinf
  # Construct two rows for analysis data
  ana[2 * (i-1) + 1, ] <- c(i-1, i, ninf, 1, newinf)
  ana[2 * (i-1) + 2, ] <- c(i-1, i, ninf, 0, n - cuminf)
}
# Make into data frame
ana <- as.data.frame(ana)
names(ana) <- c("Tstart", "Tstop", "ninf", "status", "weight")
# Remove "empty" rows, days without new infections, weight will be zero
ana <- subset(ana, weight > 0)
ana <- as_tibble(ana)
```

This is what the data look like.

```r
ana
```

```
# A tibble: 667 x 5
   Tstart Tstop  ninf status weight
    <dbl> <dbl> <dbl>  <dbl>  <dbl>
 1      0     1  10        1      2
 2      0     1  10        0    988
 3      1     2  11.8      1      1
 4      1     2  11.8      0    987
 5      2     3  12.6      0    987
 6      3     4  12.4      1      1
 7      3     4  12.4      0    986
 8      4     5  13.2      1      1
 9      4     5  13.2      0    985
10      5     6  14.0      0    985
# i 657 more rows
```

Note that on days 4 and 5 no new infections occurred, we therefore see no rows on these days with `status` = 1.

Figure 13 shows the expected number of infected individuals over time (in days since the outbreak was detected).

```r
ana$mid <- (ana$Tstart + ana$Tstop) / 2
plot(ana$mid, ana$ninf, type = "l", lwd = 2,
     xlab = "Days since outbreak detection",
     ylab = "Number of infecteds")
```

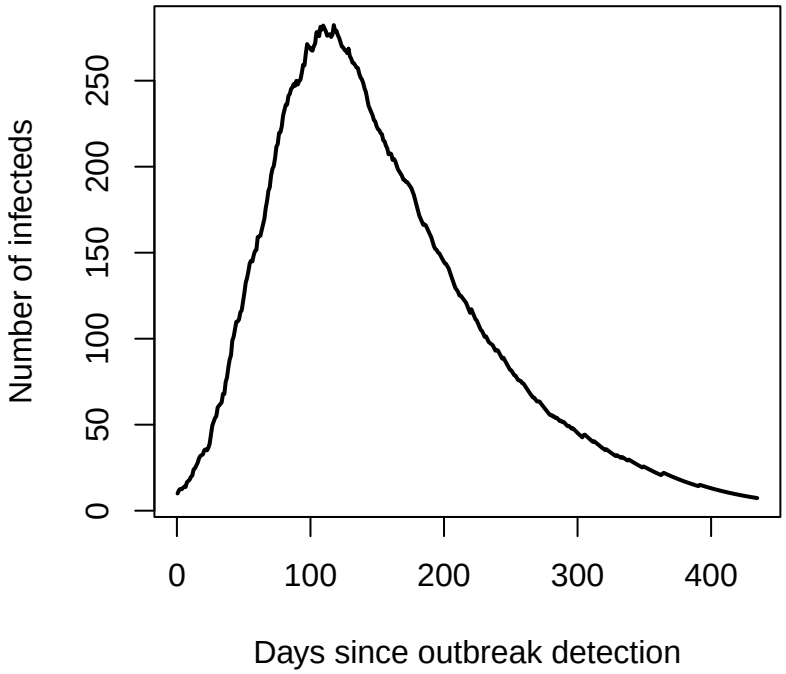


Figure 13: Expected number of infecteds over time.

After calculating `pinf`, the proportion of infected individuals, and `fuptime`, the time between the start and end of the reporting time period, which was one day, 1/30 of a month, we can then use Poisson regression with log link and `logpinf` and `logfuptime` as offsets, to obtain an estimate of β.

```
ana <- as_tibble(ana) %>%
  mutate(pinf = ninf / n,
         logpinf = log(pinf),
         fuptime = 1 / 30,
         logfuptime = log(fuptime))

poisreg <- glm(status ~ offset(logpinf) + offset(logfuptime),
               data = ana, weights = weight, family = "poisson")
summary(poisreg)
```

```
Call:
glm(formula = status ~ offset(logpinf) + offset(logfuptime),
    family = "poisson", data = ana, weights = weight)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
```

```
(Intercept)  0.13255    0.03599   3.683 0.000231 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 7952.6  on 666  degrees of freedom
Residual deviance: 7952.6  on 666  degrees of freedom
AIC: 9498.6

Number of Fisher Scoring iterations: 8
```

```r
cipoisreg <- confint(poisreg)
```

```
Waiting for profiling to be done...
```

```r
cipoisreg
```

```
     2.5 %     97.5 %
0.06116516 0.20226594
```

```r
tmp <- c(poisreg$coef, cipoisreg)
exp(tmp)
```

```
(Intercept)       2.5 %      97.5 %
   1.141732    1.063074    1.224174
```

The result again is very close to the true value of $\beta = 2$. The Poisson regression also gives a confidence interval. Note, however, that the estimate of the standard error of $\widehat{\beta}$ is too optimistic, for two reasons. First, assuming known time-to-recovery distribution, the randomness in the actual time-to-recovery is ignored and replaced by their expectations. Second, the time-to-recovery distribution is typically estimated with considerable uncertainty in itself. These sources of randomness need to be taken into account to obtain correct standard errors and confidence intervals. This is outside the scope of this manuscript.

6 Discussion

In this manuscript we have shown how standard methods from survival analysis can be used to estimate pivotal quantities in SIR models. In particular we have focused on estimating the transmission parameter in the SIR model. We have illustrated the use of multiplicative models like the Cox model and Poisson regression, and of the additive hazards model, and we have argued for the usefulness of the Cox-Aalen model, which is a hybrid of multiplicative and additive models. The possibility of using these standard models with the wide availability of software to elucidate underlying pivotal parameters opens possibilities in many situations, for instance in structured and/or clustered data.

Interestingly, in contrast to what seems to be implicit in the literature (Kenah 2011, 2013, 2015), we do not need information on who infected whom, but correct knowledge of the number of susceptible and infectious individuals over time is needed. In most realistic situations this knowledge is not readily available and needs to be further estimated from the available data and assumptions.

Clearly, more work is needed. First and most importantly, the data challenges need to be addressed.

Issues like incompleteness of reporting of infections, reporting delay will severely complicate reliable knowledge of the number of infected and infectious individuals over time. Second, the issue of heterogeneity in susceptibility is of major interest. Using frailty models seems a very promising way of estimating the extent of variability in susceptibility in the population. Third, while multiplicative models have already been used to estimate the effect of interventions in slowing down the spread of an infection, additive hazards models have to the best of our knowledge not been used in infectious disease models. Finally, the use of hybrid multiplicative - additive models such as the Cox-Aalen model seems particularly attractive.

References

Aalen, Odd O. 1980. "A Model for Nonparametric Regression Analysis of Counting Processes." In *Mathematical Statistics and Probability Theory*, edited by Witold Klonecki, Andrzej Kozek, and Jan Rosinski, 1–25. Lecture Notes in Statistics. Springer, New York.

———. 1989. "A Linear Regression Model for the Analysis of Life Times." *Statistics in Medicine* 8 (8): 907–25.

Aalen, Odd O., Ørnulf Borgan, and Håkon K. Gjessing. 2008. *Survival and Event History Analysis: A Process Point of View*. Springer, New York.

Andersen, Per Kragh, and Richard D. Gill. 1982. "Cox's Regression Model for Counting Processes: A Large Sample Study." *Annals of Statistics* 10: 1100–1120.

Anderson, Roy M., and Robert May. 1991. *Infectious Diseases of Humans: Dynamics and Control*. Oxford University Press, Oxford.

Balan, Theodor, and Hein Putter. 2020. "A Tutorial on Frailty Models." *Statistical Methods in Medical Research* 29: 3424–54.

Becker, Niels G. 1989. *Analysis of Infectious Disease Data*. Chapman; Hall, London.

Becker, Niels G., and Tom Britton. 1999. "Statistical Studies of Infectious Disease Incidence." *Journal of the Royal Statistical Society, Series B* 61 (2): 287–307.

Beddington, J. R., C. A. Free, and J. H. Lawton. 1975. "Dynamic Complexity in Predator-Prey Models Framed in Difference Equations." *Nature* 255 (5503): 58–60.

Clayton, David, and Michael Hills. 1993. *Statistical Models in Epidemiology*. Oxford University Press.

Diekmann, Odo, Hans A. P. Heesterbeek, and Tom Britton. 2013. *Mathematical Tools for Understanding Infectious Disease Dynamics*. Princeton University Press.

Donnet, Sophie, and Adeline Samson. 2013. "A Review on Estimation of Stochastic Differential Equations for Pharmacokinetic/Pharmacodynamic Models." *Advanced Drug Delivery Reviews* 65 (7): 929–39.

Ho, David D., Avidan U. Neumann, Alan S. Perelson, Wen Chen, John M. Leonard, and Martin Markowitz. 1995. "Rapid Turnover of Plasma Virions and CD4 Lymphocytes in HIV-1 Infection." *Nature* 373 (6510): 123–26.

Katriel, Guy. 2012. "The Size of Epidemics in Populations with Heterogeneous Susceptibility." *Journal of Mathematical Biology* 65 (2): 237–62.

Kenah, Eben. 2011. "Contact Intervals, Survival Analysis of Epidemic Data, and Estimation of R0." *Biostatistics* 12: 548–66.

———. 2013. "Non-Parametric Survival Analysis of Infectious Disease Data." *Journal of the Royal Statistical Society, Series B* 75: 277–303.

———. 2015. "Semiparametric Relative-Risk Regression for Infectious Disease Transmission Data." *Journal of the American Statistical Association* 110: 313–25.

Kermack, W. O., and A. G. McKendrick. 1927. "A Contribution to the Mathematical Theory of Epidemics." *Proceedings of the Royal Society London* 115: 700–721.

KhudaBuksh, Waiur R., Boseung Choi, Eben Kenha, and Grzegorz A. Rempala. 2019. "Survival Dynamical Systems: Individual-Level Survival Analysis from Population-Level Epidemic Models." *Interface Focus* 10: 20190048.

Lin, Dan-Yu, and Zhiliang Ying. 1994. "Semiparametric Analysis of the Additive Risk Model."

Biometrika 81: 61–71.

Lu, Chengyuan, Jelle Goeman, and Hein Putter. 2023. "Maximum Likelihood Estimation in the Additive Hazards Model." *Biometrics* 79: 1646–56.

Martinussen, Torben, and Thomas H. Scheike. 2006. *Dynamic Regression Models for Survival Data*. Springer, New York.

Miller, Joel C. 2012. "A Note on the Derivation of Epidemic Final Sizes." *Bulletin of Mathematical Biology* 74 (9): 2125–41.

Murdoch, William W., Cheryl J. Briggs, and Roger M. Nisbet. 2013. *Consumer-Resource Dynamics (MPB-36)*. Princeton University Press.

Scheike, Thomas H., and Mei-Jie Zhang. 2002. "An Additive-Multiplicative Cox-Aalen Regression Model." *Scandinavian Journal of Statistics* 29: 75–88.

Wolkewitz, Martin, Markus Dettenkofer, Hartmut Bertz, Marten Schumacher, and Johannes Huebner. 2002. "Statistical Epidemic Modeling with Hospital Outbreak Data." *Statistics in Medicine* 29: 75–88.