The case time series design

A case study for applications in clinical epidemiology

Antonio Gasparrini

29 September 2021

Contents

This document was originally presented as eAppendix 1 of the article "*The case time series design*," accepted for publication in Epidemiology (Gasparrini 2021), and it reproduces the analysis presented as the first case study. An updated version of this document and related material are available at the GitHub page and at the personal website of the author. The material includes the Rmarkdown files to compile the document, plus scripts with the embedded R code. Note that the code is profiled for clarity, not for speed, with the aim of illustrating the steps of the analysis and the features of the design. It can (probably should) be modified when re-used in real analyses.

This case study illustrates the application of the case time series design in clinical studies. Specifically, the example describes an analysis of the association between acute respiratory infection (flu) and acute myocardial infarction (AMI) using a cohort reconstructed from linked electronic health records. The sample includes 3,927 subjects who experienced a (first) AMI event and had at least one primary care consultation for flu during a pre-determined follow-up period. The analysis illustrates an application of the case time series design with a non-repeated event outcome and binary indicators of exposure episodes. These data were originally presented and analysed with an alternative method in previous publications (Warren-Gash et al. 2012). The code shown below creates and uses simulated data to reproduce the features of the original dataset, which cannot be made publicly available, and the steps and (approximate) results of the application of the case time series design.

Preparation

The following packages are loaded in the session, and need to be installed to run the R code:

```
library(dlnm) ; library(gnm) ; library(mgcv) ; library(pbs)
library(data.table) ; library(scales)
```

We first set a seed to ensure the exact replicability of the results, as the code includes expressions with random number generation, and we also set the graphical parameter `las` for the plots:

```
set.seed(13041975)
par(las=1)
```

Simulating the original data

The data used in this case study are simulated directly in this section. The user can skip it if not of interest, and start with the following section for the data analysis. First, we set the parameters, namely the number of subjects `n` and the date of start and end of follow-up. Note that we reduce the follow-up period to one year, in order to obtain a more manageable dataset. The code:

```
n <- 3927
dstart <- as.Date("2007-01-01")
dend <- as.Date("2007-12-31")
```

Then we generate the time variables across the follow-up period, namely `date` (calendar days), `time` (a sequence of integers starting from 1), `month` (months in numbers), and `doy` (days of the year). In addition, we randomly generate `dob` (date of birth) for each subject, with age at start between 35 and 100 years old.

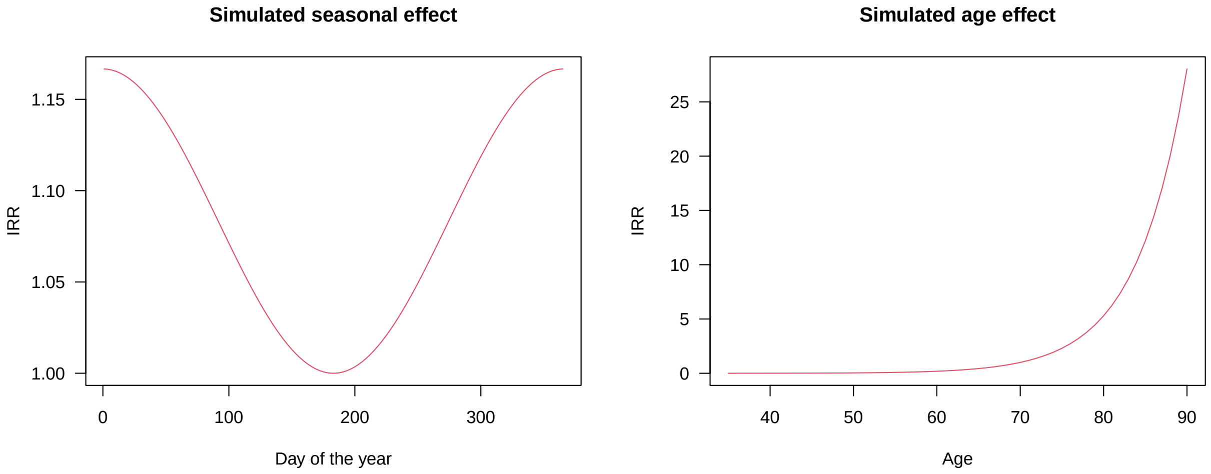
```
date <- seq(dstart, dend, by=1)
times <- seq(length(date))
month <- month(date)
doy <- yday(date)
dob <- sample(seq(dstart-round(100*365.25), dstart-round(35*365.25), by=1), n)
```

These variables are used for simulating the temporal variation in the underlying risk of AMI, with a cyclic seasonal trend and a long-term change by age modelled by a cosine function and polynomials, respectively. These effects are defined as a incident rate ratio (IRR), and created by the following code:

```
frrseas <- function(doy) (cos(doy*2*pi / 366) + 1) / 12 + 1
frrage <- function(age) exp((age - 70) / 6)
```

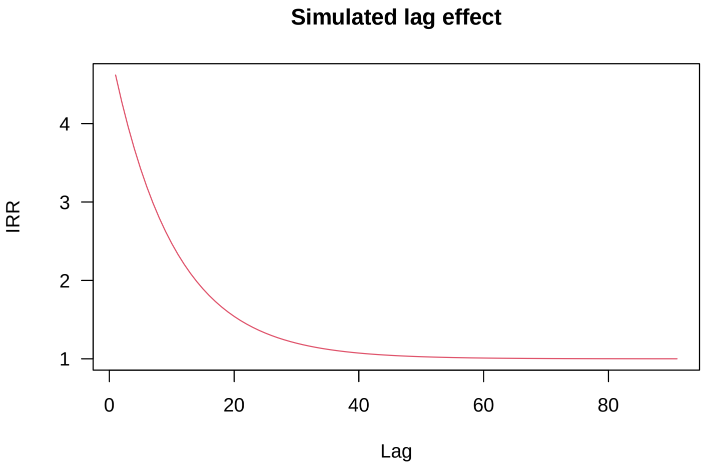
These temporal variations in risk along day of the year and age are represented in the graphs below:

```
plot(1:365, frrseas(1:365), type="l", col=2, ylab="IRR", xlab="Day of the year",
  main="Simulated seasonal effect")
plot(35:90, frrage(35:90), type="l", col=2, ylab="IRR", xlab="Age",
  main="Simulated age effect")
```



Now we create a function to define the IRR along the *lag dimension*. In this case, this dimension represents the risk after a flu episode, with the lag unit defined by day. Similarly, we illustrate the phenomenon graphically:

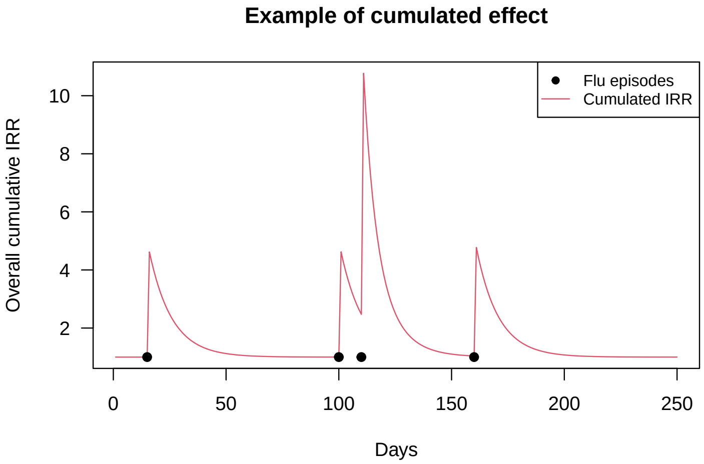
```
frrlag <- function(lag) exp(-(lag/10)) * 4 + 1
plot(1:91, frrlag(1:91), type="l", col=2, ylab="IRR", xlab="Lag",
  main="Simulated lag effect")
```



The graph indicates that, within a lag period of 3 months (1 to 91 days of lag) as in the original analyses, the risk is much increased in the first days after the flu episode, but then it attenuates and tends to null after approximately one month.

In the presence of multiple exposure episodes, lagged effects can cumulate in time, depending on the *exposure profile* of an individual. In this case, the risk at a given day is determined by the *exposure history* to flu, with potentially multiple flu episodes contributing at different lags for the same day. For instance, the code below shows an example with the risk associated with four flu episodes in a 250-day period, with the cumulated risk being the product of lag-specific contributions:

```
expprof <- as.numeric(seq(250) %in% c(15,100,110,160))
exphist <- exphist(expprof, lag=c(1,91), fill=0)
rrflu <- apply(exphist, 1, function(x) prod(frrlag(1:91)[x==1]))
plot(seq(250), rrflu, type="l", col=2, ylab="Overall cumulative IRR", xlab="Days",
  main="Example of cumulated effect")
points(c(15,100,110,160), rep(1,4), pch=19)
legend("topright", c("Flu episodes","Cumulated IRR"), pch=c(19,NA), lty=c(NA,1),
  col=1:2, cex=0.8)
```



We have now all the information required for simulating the original data. These will consist of individual records with the following variables, with age measured in days:

- `id`: the identifier of the subject
- `dob`: date of birth
- `start`: the age of the subject at the start of follow-up
- `end`: the age of the subject at the end of follow-up
- `event`: the age of the subject at the occurrence of the AMI event
- `flu*`: multiple variables defining the age(s) of the subject at each flu episode

The data are simulated by looping in a list, producing the observations for each subject, and then binding them in a dataframe. Each of the blocks of code in the loop performs the following steps for each subject:

1. Sample the number of flu episode(s); define the risk of having a flu episode in each day; sample the flu episodes and create an exposure profile
2. Create the exposure history of flu for each day for a given lag period; compute the overall cumulative AMI risk due to flu for each day
3. Define the total AMI risk for each day, dependent on age, season, and flu; sample the unique AMI event
4. Put the information together in a dataframe; add the flu episodes, setting them to NA if less than the sampled maximum of 10

Here is the R code (it takes less than a minute):

```
dlist <- lapply(seq(n), function(i) {

  nflu <- rpois(1,1) + 1
  expprof <- drop(rmultinom(1, nflu, frrseas(doy))) > 0 + 0

  exphist <- exphist(expprof, lag=c(1,91), fill=0)
  rrflu <- apply(exphist, 1, function(x) prod(frrlag(1:91)[x==1]))

  rrtot <- frrage(as.numeric((date-dob[i])/365.25)) * frrseas(doy) * rrflu
  devent <-  date[drop(rmultinom(1, 1, rrtot))==1]

  data <- data.frame(id = paste0("sub", sprintf("%03d", i)), dob = dob[i],
    start = as.numeric(dstart - dob[i]), end = as.numeric(dend - dob[i]),
    event = as.numeric(devent - dob[i]))
  flu <- as.numeric(date[expprof == 1] - dob[i])
  for(j in seq(10)) data[paste0("flu", j)] <- if(j>nflu) NA else flu[j]

  return(data)
})
dataorig <- do.call(rbind, dlist)
```

Specifically, the total number of flu episodes `nflu` are sampled from a Poisson distribution with mean of 1 (plus one to ensure at least one episode). The occurrence of these flu episodes is sampled at random from a multinomial distribution, with probabilities varying by day of the year, thus determining a confounding effect by season. The AMI event for each subject in `devent` is then sampled from a multinomial distribution, with risk for each day `rrtot` defined by flu episodes (with lag), age, and season. Note that that in `rmultinom` probabilities are determined from IRRs by rescaling them internally.

The final line of code binds together all the records. This dataset has a simple form with one record per subject, but it contains all the information for conducting the case time series analysis in the next sections.

Data expansion

Now that we have the data, we can start our analysis using the case time series design. The first step is to expand the data to recover the individual series. You can appreciate that this leads back to the same data structure used to simulate the original dataset in the section above. We start by showing the process for a given subject (number 3), with data:

```
(sub <- dataorig[3,])
```

```
##       id        dob start   end event  flu1  flu2  flu3 flu4 flu5 flu6 flu7
## 3 sub003 1916-08-18 33008 33372 33274 33105 33273 33276   NA   NA   NA   NA
##   flu8 flu9 flu10
## 3   NA   NA    NA
```

Specifically, we reconstruct the daily series of outcome `y` (AMI event), `flu` (flu indicator), and all the time variables, including them in a new dataframe:

```
date <- as.Date(sub$start:sub$end, origin=sub$dob)
datasub <- data.frame(
  id = sub$id,
  date = date,
  times = seq(length(date)),
  age = as.numeric(date-sub$dob)/365.25,
  y = as.numeric(date-sub$dob) %in% sub$event + 0,
  flu = as.numeric(date-sub$dob) %in% na.omit(as.numeric(sub[6:15])) + 0,
  month = month(date),
  doy = yday(date)
)
```

These expanded data correspond to an individual series of outcome and predictors (therefore the name *case time series* for this design). We can have a look at the first observations for subject 3:

```
head(datasub)
```

```
##       id       date times      age y flu month doy
## 1 sub003 2007-01-01     1 90.37098 0   0     1   1
## 2 sub003 2007-01-02     2 90.37372 0   0     1   2
## 3 sub003 2007-01-03     3 90.37645 0   0     1   3
## 4 sub003 2007-01-04     4 90.37919 0   0     1   4
## 5 sub003 2007-01-05     5 90.38193 0   0     1   5
## 6 sub003 2007-01-06     6 90.38467 0   0     1   6
```

In addition, we create the exposure history for each observation within the follow-up period of the same subject, applying the function `exphist()` on the exposure series over the lag period 1-91:

```
exphistsub <- exphist(datasub$flu, lag=c(1,91), fill=0)
```

You can notice from above that subject 3 had the first flu episode at day 33,105, corresponding to time 98 of the series. We can check the exposure history matrix around those times:

```
timeflu1 <- sub$flu1-sub$start+1
exphistsub[timeflu1 + 0:5, 1:10]
```

```
##     lag1 lag2 lag3 lag4 lag5 lag6 lag7 lag8 lag9 lag10
## 98     0    0    0    0    0    0    0    0    0     0
## 99     1    0    0    0    0    0    0    0    0     0
## 100    0    1    0    0    0    0    0    0    0     0
## 101    0    0    1    0    0    0    0    0    0     0
## 102    0    0    0    1    0    0    0    0    0     0
## 103    0    0    0    0    1    0    0    0    0     0
```

The diagonal pattern of 1’s identifies days corresponding to lags after this specific exposure episode.

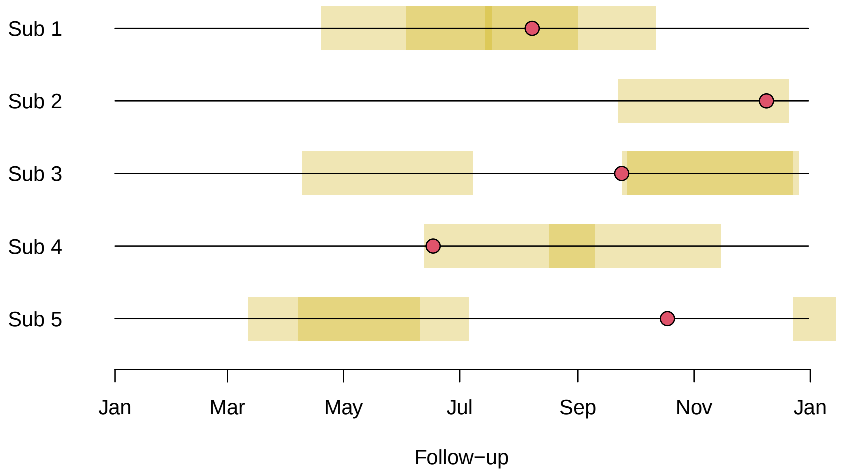
We can now apply this expansion to all the subjects by repeating the steps above, and obtain the final data, including the exposure histories. Here is the code (it takes less than a minute):

```
dlist <- lapply(seq(n), function(i) {

  sub <- dataorig[i,]

  date <- as.Date(sub$start:sub$end, origin=sub$dob)
  data <- data.frame(
    id = sub$id,
    date = date,
    times = seq(length(date)),
    age = as.numeric(date-sub$dob)/365.25,
    y = as.numeric(date-sub$dob) %in% sub$event + 0,
    flu = as.numeric(date-sub$dob) %in% na.omit(as.numeric(sub[6:15])) + 0,
    month = month(date),
    doy = yday(date)
  )

  exphist <- exphist(data$flu, lag=c(1,91), fill=0)

  return(data.table(cbind(data, exphist)))
})
data <- do.call(rbind, dlist)
```

Analysis

Now that we have obtained the final dataset, we can start the data analysis. First, we have a look at the follow-up of the first five subjects, including the time of AMI event (red circle) and exposure periods in the 1-91 days after flu episodes:

```
plot(unique(data$date), unique(data$date), ylim=c(0.5,5+0.5), yaxt="n",
  ylab="", xlab="Follow-up", frame.plot=F)
axis(2, at=5:1, labels=paste("Sub",1:5), lwd=0, las=1)
for(i in 5:1) {
  sub <- subset(data, id==unique(data$id)[i])
  flu <- sub$date[sub$flu==1]
  rect(flu+1, rep(i-0.3,length(flu)), flu+91, rep(i+0.3,length(flu)), border=NA,
    col=alpha("gold3",0.3))
  lines(sub$date, rep(i, nrow(sub)))
  points(sub$date[sub$y==1], i, pch=21, bg=2, cex=1.5)
}
```



While many of the subjects only have a single flu episode, four of them in this sample have multiple ones, and these episodes are so close that they generate overlapping exposure windows. This could not be dealt with in the original self-controlled case series analysis (Warren-Gash et al. 2012), while as shown below the case time series design can appropriately account for cumulative effects.

Now, we replicate the main case time series analysis illustrated in the original article (Gasparrini 2021). We first derive the terms to control for age and season using natural cubic and cyclic splines, respectively. We use the wrapper function `onebasis()` that simplifies the prediction and plotting of these associations, to be performed later. We call the function `pbs` from the package with the same name to generate the basis transformations for the cyclic splines. The code:

```
splage <- onebasis(data$age, "ns", knots=quantile(data$age, c(1,3)*0.25))
splseas <- onebasis(data$doy, "pbs", df=3)
```

Then, we implement the distributed lag model (DLM), defining the cross-basis parameterisation for flu with a lag period from 1 to 91 days, using the exposure histories included as lagged terms in `data`:

```
exphist <- data[,-c(1:8)]
cbspl <- crossbasis(exphist, lag=c(1,91), argvar=list("strata",breaks=0.5),
  arglag=list("ns",knots=c(3,10,29)))
```

The function `crossbasis()` internally calls `strata()` with cut-off at 0.5 to parameterise the exposure-response for flu using a simple indicator, and `ns()` to produce the natural cubic splines with specific knots for the lag-response function (see `help(crossbasis)`).

We now have all the terms for fitting the fixed-effects Poisson regression using the function `gnm()`. The regression model includes all the predictors, and defines the conditional stratification through the argument `eliminate`. This is the code:

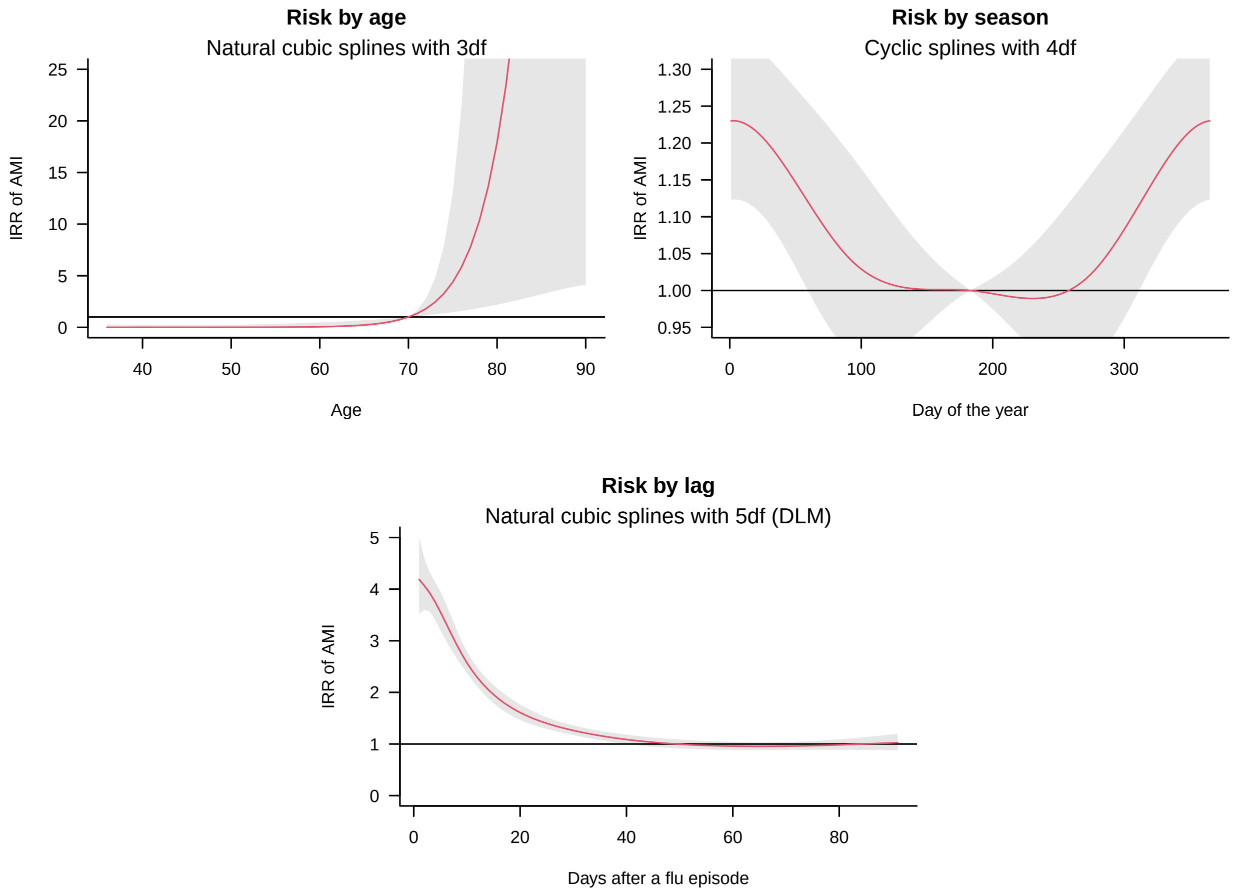
```
mspl <- gnm(y ~ cbspl+splage+splseas, data=data, family=poisson,
  eliminate=factor(id))
```

The estimated coefficients and associated (co)variance matrix of the model can now be used to predict the association of the various terms with the risk of AMI, using the function `crosspred()`:

```
cpspl <- crosspred(cbspl, mspl, at=1)
cpsplage <- crosspred(splage, mspl, cen=70, to=90)
cpsplseas <- crosspred(splseas, mspl, cen=366/2, at=1:365)
```

Finally, we can plot them:

```
plot(cpsplage, col=2, ylab="IRR of AMI", xlab="Age", main="Risk by age",
  ylim=c(0,25))
mtext("Natural cubic splines with 3df", cex=0.8)
plot(cpsplseas, col=2, ylab="IRR of AMI", xlab="Day of the year",
  main="Risk by season", ylim=c(0.95,1.30))
mtext("Cyclic splines with 4df", cex=0.8)
plot(cpspl, var=1, col=2, ylab="IRR of AMI", xlab="Days after a flu episode",
  ylim=c(0,5), main="Risk by lag")
mtext("Natural cubic splines with 5df (DLM)", cex=0.8)
```



Results are similar to those reported in the original article (Gasparrini 2021). Differences in estimated confidence intervals for the risk along with age and after a flu episodes are easily explained by the shorter follow-up period simulated here (one year), which reduces the within-subject age differences and increases the prevalence of exposure to flu.

An alternative parameterisation of cross-basis term can be used, specifically using strata functions to represent the risk along lags. The code:

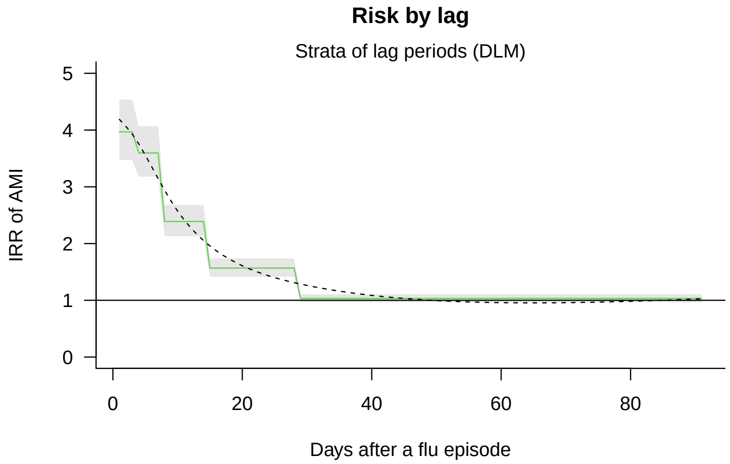
```
cbstr <- crossbasis(exphist, lag=c(1,91), argvar=list("strata",breaks=0.5),
  arglag=list("strata",breaks=c(4,8,15,29)))
```

We can now fit the alternative model:

```
mstr <- gnm(y ~ cbstr+splage+splseas, data=data, family=poisson,
  eliminate=factor(id))
cpstr <- crosspred(cbstr, mstr, at=1)
```

and create the related plot, including the previous fitted relationships as dashed lines:

```
plot(cpstr, var=1, col=3, ylab="IRR of AMI", xlab="Days after a flu episode",
  ylim=c(0,5), main="Risk by lag")
mtext("Strata of lag periods (DLM)", cex=1)
lines(cpspl, var=1, lty=2)
```



This parameterisation can be compared to the original analysis performed with the self-controlled case series design (Warren-Gash et al. 2012), where post-exposure periods consistent with the lag strata were used. However, here the case time series design and DLMs can appropriately handle potentially overlapping exposure periods. We can extract from the predictions the IRR (and 95% confidence intervals) corresponding to the five strata:

```
resstr <- round(with(cpstr, t(rbind(matRRfit,matRRlow,matRRhigh))), 2)
colnames(resstr) <- c("IRR","low95%CI","high95%CI")
resstr[paste0("lag", c(1,4,8,15,29)),]
```

```
##         IRR low95%CI high95%CI
## lag1   3.97     3.47      4.54
## lag4   3.60     3.18      4.07
## lag8   2.39     2.13      2.68
## lag15  1.57     1.42      1.74
## lag29  1.03     0.97      1.11
```

The results demonstrate the flexibility of the case time series design to investigate complex relationships using self-matched comparisons of individual-level data.

References

Gasparrini, Antonio. 2021. “The case time series design.” *Epidemiology* 32 (6): 829–37.

Warren-Gash, Charlotte, Andrew C Hayward, Harry Hemingway, Spiros Denaxas, Sara L Thomas, Adam D Timmis, Heather Whitaker, and Liam Smeeth. 2012. “Influenza infection and risk of acute myocardial infarction in England and Wales: a CALIBER self-controlled case series study.” *Journal of Infectious Diseases* 2006 (11): 1652–59.