The individual case time series and case crossover design

A case study for applications in environmental epidemiology

Yuantong (Anthony) Sun

24 February 2023

Contents

The case study was originally from Antonio Gasparrni and modified by Anthony Sun. The original version was presented as eAppendix 2 of the article "*The case time series design*", accepted for publication in Epidemiology [@gasparrini2021epidem], and it reproduces the analysis presented as the second 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.

This case study illustrates the application of the case time series design in environmental studies. Specifically, the example describes an analysis of the association between exposure to three different environmental stressors and the risk of respiratory symptoms using a cohort of participants to a smartphone study. The sample includes 1,601 subjects who reported daily the occurrence of respiratory symptoms such as asthma and allergic rhinitis in a smartphone app, and who were assigned exposure levels by linking their geo-located position with high-resolution spatio-temporal maps of pollen, air pollution, and temperature. The analysis illustrates an application of the case time series design with a binary outcome and multiple continuous exposures. The data were collected within the AirRater study, an integrated online platform that combines symptom surveillance, environmental monitoring, and real-time notifications operating in Tasmania [@johnston2018erl]. 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.

Anthony added codes for an individual time-stratified case crossover study design following the CTS 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(data.table) ; library(splines); library(dplyr);
library(tidyr);library(dlnm);library(splines);library(gnm);library(survival)
```

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 study period. Then we create a `date` and related time variables `year`, `month`, `doy` (day of the year), and `dow` (day of the week):

```
n <- 1601
dstart <- as.Date("2015-10-29")
dend <- as.Date("2018-11-19")
date <- seq(dstart, dend, by=1)
year <- year(date)
month <- month(date)
doy <- yday(date)
dow <- factor(wday(date))
```

Then we define follow-up periods for the 1601 subjects, randomly sampling starting dates and length of follow-up, with the constraints that the end of follow-up cannot be later than the end of the study period, and with a length of at least 10 days. The code:

```
fustart <- sample(seq(dstart, dend-10, by=1), n, replace=TRUE)
fuend <- fustart + pmax(pmin(round(exp(rnorm(n, 5.1, 2))), dend-fustart), 10)
sum(fuend-fustart+1)
```

```
## Time difference of 363901 days
```

While the follow-up distribution does not match perfectly the original study, the sampling parameters are set above to generate approximately the same number of total person-days, in this case, 363,901.

Finally, we define some variables used to simulate the distribution of the environmental exposures and, later, the seasonal baseline risk. These variables are the cosine transformation of `doy` and quadratic splines of `date` with 5 degrees of freedom per year. In addition, we simulate 20 random smoke days occurring in the (Australian) summer. The code:

```
cosdoy <- cos(doy*2*pi / 366)
spldate <- bs(date, degree=2, int=TRUE, df=round(length(date)/365.25)*5)
smokeday <- date %in% sample(date[month %in% c(1,2,12)], 20)
```

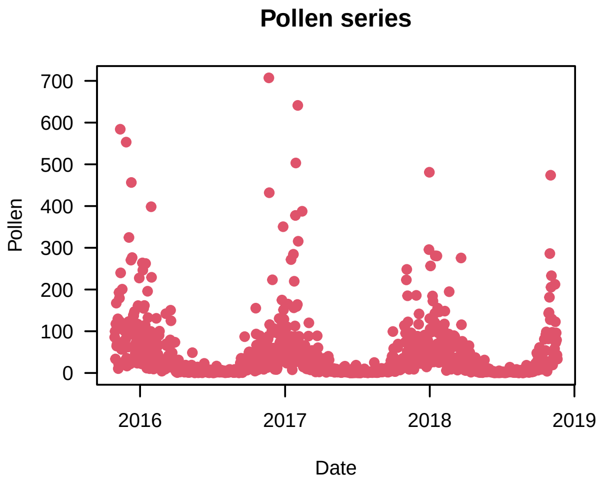
We are now ready to simulate the distribution of the three environmental stressors. In the original study, individual exposure series were reconstructed through the geo-location system of the smartphone by linkage with detailed spatio-temporal exposure maps. In order to simplify the simulation process, we derive here a single series for each stressor, assuming that all the 1601 subjects are exposed to the same levels on the same day. This does not affect the generality of the example, and in real-case settings, individual-level exposure series can nevertheless be used.

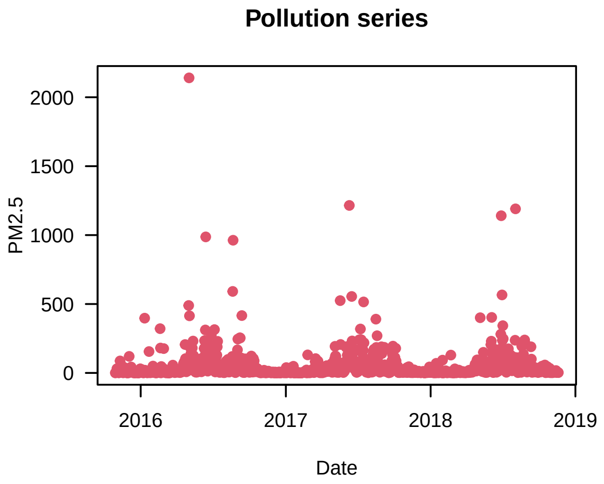
The environmental exposures are created by assuming an underlying seasonal trend, represented by the cosine variable above, plus auto-correlated random normal deviations. Exponentiation is used to produce non-negative values of pollen (grains/m^3) and pollution (PM$_{2.5}$, μgr/m^3), while temperature (°C) is sampled directly. The code:

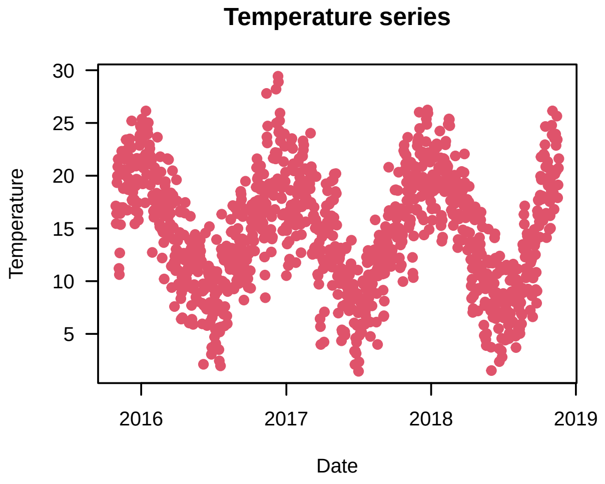
```
pollen <- exp(cosdoy*2+2.5 + arima.sim(list(ar=0.5), length(date), sd=0.8))
pm <- exp((-cosdoy)*1.6+2.5 + smokeday*3.2 +
  arima.sim(list(ar=0.6), length(cosdoy), sd=0.95))
tmean <- cosdoy*6+15 + arima.sim(list(ar=0.6), length(cosdoy), sd=2.6)
envdata <- data.frame(date, pollen, pm, tmean)
```

The variables are included in the dataframe `envdata`. The definitions above provide a realistic distribution of the three exposures, with pollen and temperature peaking in summer, while PM$_{2.5}$ shows higher wintertime levels but with isolated spikes in the summer corresponding to smoke days due to fires. A visual representation is offered by the plots obtained through:

```
plot(date, pollen, xlab="Date", ylab="Pollen", main="Pollen series", col=2,
  pch=19)
plot(date, pm, xlab="Date", ylab="PM2.5", main="Pollution series", col=2,
  pch=19)
plot(date, tmean, xlab="Date", ylab="Temperature", main="Temperature series",
  col=2, pch=19)
```





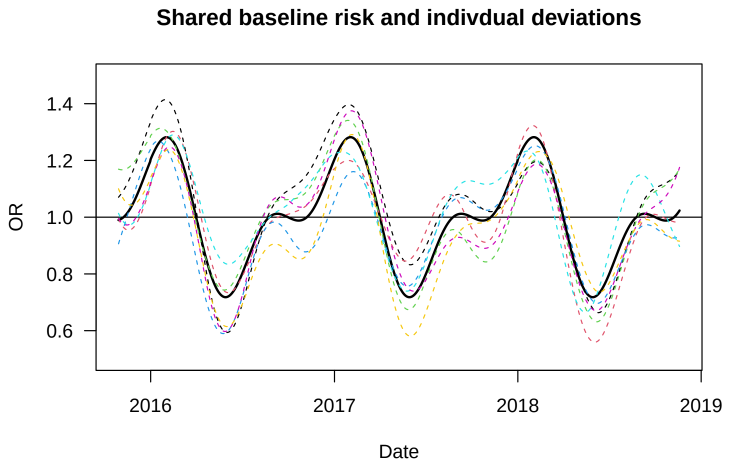


The variables created above can now be used to define individual risk profiles of experiencing allergic symptoms. These profiles will be simulated as risks associated to the three exposures on top of baseline trends. We first simulate the latter as a combination of shared underlying risks and individual-level deviations:

```
fortrend <- function(ind=TRUE) (cosdoy*1.6 + sin(doy*4*pi/366))/8+1 + if(ind)
  spldate %*% runif(ncol(spldate),-0.2,0.2) else 0
```

The function `fortrend()` includes harmonic terms at different periods to define the shared baseline risk common to all subjects, plus optionally individual deviations modelled using random coefficients for the spline of time. These trends are defined as odds ratio (OR). We can graphically represent them using the code below, with the bold black line representing the shared trend and the dashed coloured lines as individual profiles:

```
plot(date, fortrend(ind=F), type="l", lwd=2, ylim=c(0.5,1.5), xlab="Date",
  ylab="OR", main="Shared baseline risk and indivdual deviations")
abline(h=1)
for(i in 1:7) lines(date, fortrend(ind=T), type="l", lty=2, col=i)
```

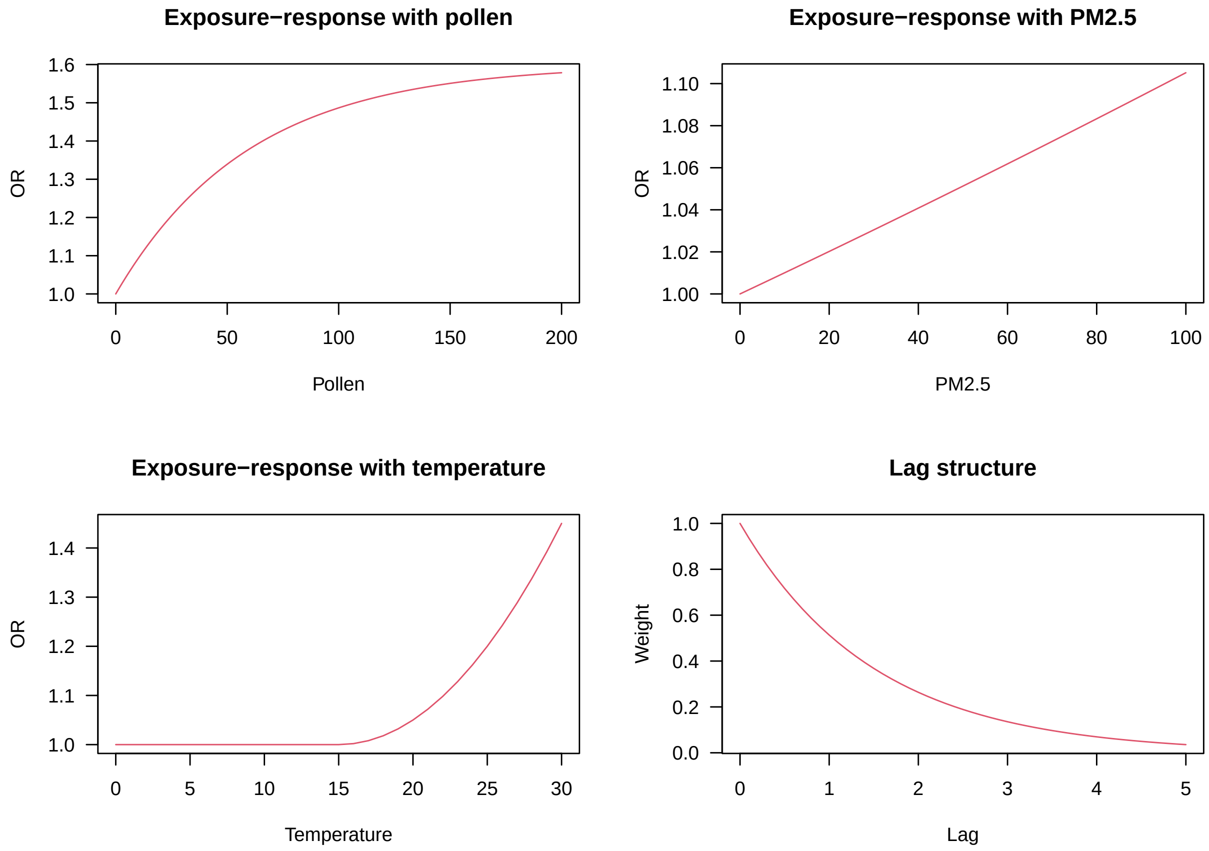


The next step is the definition of the increase in risk due to exposure to the three environmental stressors. Specifically, we define non-linear relationships for pollen and temperature, with effects lagged up to 3 days, and a linear and unlagged association with $PM_{2.5}$. First, we define the three functions to specify the three exposure-response risk shapes and the lag structure:

```
forpoll <- function(x) 1.6 - 0.6*exp(-x/60)
forpm <- function(x) exp(x/1000)
fortmean <- function(x) 1 + ifelse(x>15, 0.002*(x-15)^2, 0)
fwlag <- function(lag) exp(-lag/1.5)
```

These functions define relationships in the OR and lag scales, and can be represented graphically with:

```
plot(0:200, forpoll(0:200), type="l", xlab="Pollen", ylab="OR",
  main="Exposure-response with pollen", col=2)
plot(0:100, forpm(0:100), type="l", xlab="PM2.5", ylab="OR",
  main="Exposure-response with PM2.5", col=2)
plot(0:30, fortmean(0:30), type="l", xlab="Temperature", ylab="OR",
  main="Exposure-response with temperature", col=2)
plot(0:50/10, fwlag(0:50/10), type="l", xlab="Lag", ylab="Weight",
  main="Lag structure", col=2)
```



These shapes are similar to the associations estimated in the original study [@gasparrini2021epidem]. The lag structure is defined as weights, and can be used to represent a decreasing OR proportionally to time after the exposure occurred. As an example, we used the functions above to calculate the net OR in a given day after exposures to pollen of 50, 9, 135, and 93 grains/m^3 in the same and past 3 days (lag 0–3):

```
exp(sum(log(forpoll(c(50,9, 135, 93))) * fwlag(0:3)))
```

```
## [1] 1.647004
```

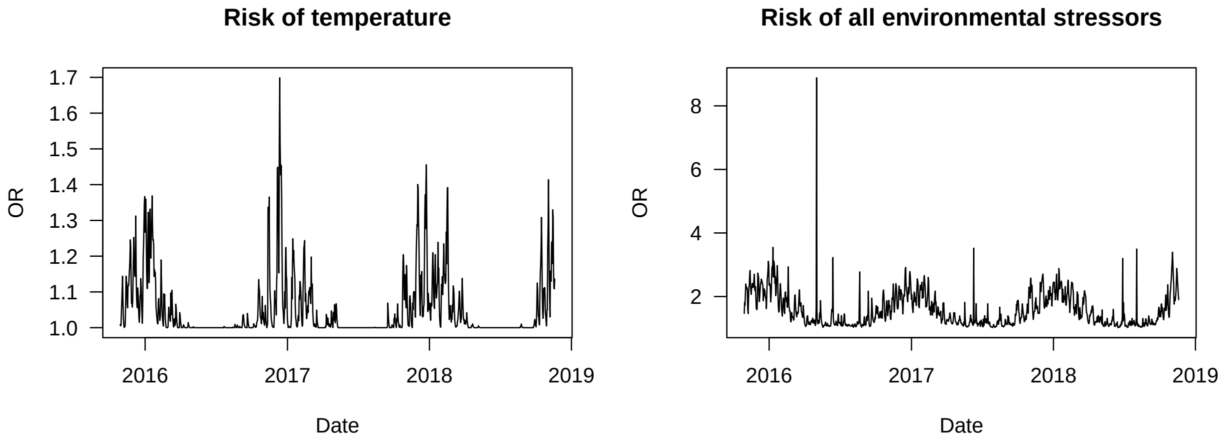
Simply, the expression above computes the log-OR for each exposure occurrence, which are then weighted depending on the lag and then summed and exponentiated to obtain the overall cumulative OR .

We can now apply the same computation to the whole series for the three exposures, using first the function `exphist` to generate the matrix of lagged exposures, and then applying the expression for each row:

```
orpoll <- apply(exphist(pollen, lag=3), 1, function(x)
  exp(sum(log(forpoll(x)) * fwlag(0:3))))
orpm <- forpm(pm)
ortmean <- apply(exphist(tmean, lag=3), 1, function(x)
  exp(sum(log(fortmean(x)) * fwlag(0:3))))
orenv <-  orpoll * orpm * ortmean
```

Note that we assume a lag 0–3 for pollen and temperature, while we simply define a same-day association with no lag for PM$_{2.5}$. The code above computes therefore the OR contribution for each exposure in each day, which are then multiplied to obtain the overall risk associated with all the three environmental stressors in the vector `orenv`. The series for temperature and all the exposures are graphically represented below:

```
plot(date, ortmean, type="l", xlab="Date", ylab="OR",
  main="Risk of temperature")
plot(date, orenv, type="l", xlab="Date", ylab="OR",
  main="Risk of all environmental stressors")
```



We have now all the information required for simulating the original data. These are created 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. Define the follow-up period and identify the related subset of the study period
2. Create the total risk contribution in each follow-up day
3. Sample the occurrence of respiratory symptoms within the follow-up period
4. Put the information together in a dataframe, adding the series of environmental exposures

Here is the R code:

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

  fudate <- seq(fustart[i], fuend[i], by=1)
  sub <- date %in% fudate

  ortot <- fortrend(ind=T)[sub] * orenv[sub] * (1 + wday(fudate) %in% c(2:6)*0.4)

  pbase <- plogis(-3.3 + 14/length(fudate) - 0.0015*length(fudate))
  sympt <- rbinom(sum(sub), 1, plogis(qlogis(pbase) + log(ortot)))

  data <- cbind(data.frame(id=paste0("sub",sprintf("%04d", i)), date=fudate,
    year=year[sub], month=month[sub], dow=dow[sub], y=sympt), envdata[sub, -1])

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

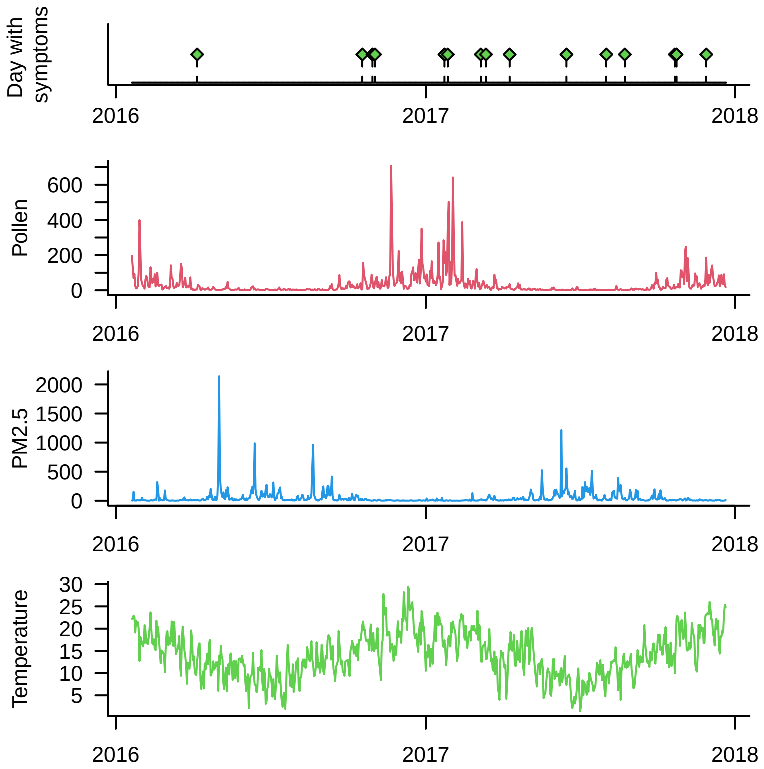
Specifically, the total OR in `ortot` is the product of underlying trends (as the sum of shared seasonal OR plus random individual deviations), the contribution of environmental factors, and a simulated OR of 1.4 for weekdays vs weekends. These are multiplied to a baseline risk in `base` to compute the day and subject-specific odds. The baseline risk varies across individuals, and it is inversely proportional to the follow up period, similary to the real-date example. The indicator of days with respiratory symptoms `sympt` is sampled then from a Bernoulli distribution (binomial with a single trial) with probabilities back-transformed from the logistic scale. Note that this method of sampling does not ensure that all the subjects have at least one day with reported symptoms, and these will be automatically discarded from the analysis and they do not contribute information to the conditional comparison.

The final line of code binds together all the data in a single dataframe. This dataset is already expanded to its case time series format, where the number of rows corresponds to the total person-days of follow-up (363,901). In some situations, it can be more convenient to store the data in multiple datasets, for instance separating individual information and environmental exposures, and then assemble them together for the final analysis.

Statistical analysis with CTS

Now that we have obtained the final dataset, we can replicate the main case time series analysis. First, we have a look at the data for a given subject (identified as sub0036), represented as individual series of daily observations of outcome and predictors (therefore the name *case time series* for this design). The code:

```
dsub <- subset(data, id=="sub0036")
plot(y~date, data=dsub, type="h", lty=2, ylim=c(0,2), yaxt="n", bty="l", xlab="",
  ylab="Day with \nsymptoms", mgp=c(2.2,0.7,0), lab=c(5,3,7))
points(y~date, data=subset(dsub,y>=1), pch=23, bg=3)
plot(pollen~date, data=dsub, type="l", lty=1, bty="l", col=2, xlab="",
  ylab="Pollen")
plot(pm~date, data=dsub, type="l", lty=1, bty="l", col=4, xlab="Date",
  ylab="PM2.5")
plot(tmean~date, data=dsub, type="l", lty=1, bty="l", col=3, xlab="Date",
  ylab="Temperature")
```



We can now define the different terms to be included in the regression model. First, we define a set of splines of time with approximately 8 degrees of freedom per year, and subject/year/month strata indicators, to be used to model the shared seasonal trend and individual deviations, respectively.:

Anthony: different individuals have different follow-up time and even one individual may have an incomplete month series leading to intra-month residual variation. So we need to adjust for time instead of including week in the strata.

```
dftrend <- round(as.numeric(diff(range(data$date))/365.25 * 8))
btrend <- ns(data$date, knots=equalknots(data$date, dftrend-1))
data$stratum <- with(data, factor(paste(id, year, month, sep="-")))
```

We now apply the function `crossbasis()` to parameterise distributed lag linear and non-linear transformations of the environmental variables:

```
cbpoll <- crossbasis(data$pollen, lag=3, argvar=list(knots=c(40,100)),
  arglag=list(knots=1), group=data$id)
cbpm <- crossbasis(data$pm, lag=3, arglag=list("integer"), group=data$id)
cbtmean <- crossbasis(data$tmean, lag=3, argvar=list(knots=1:2*10),
  arglag=list(knots=1), group=data$id)
```

Specifically, the default `ns()` function is used in both the `argvar` and `arglag` arguments to specify natural cubic splines for the exposure-response and lag-response, respectively, of both pollen and temperature, using different knots placements. A default linear exposure-response is defined for $PM_{2.5}$, instead, while the lag-response is parameterised through an unconstrained distributed lag function, namely using indicators for each lag. The lag period is extended to 0–3 for all three exposures. A `group` argument is used to specify that the variables do not represent a unique and complete series, but multiple individual series. See `help(crossbasis)` for more information.

We now have all the terms for fitting the fixed-effects logistic regression using the function `gnm()`, with the strata indicators included in the argument `eliminate`:

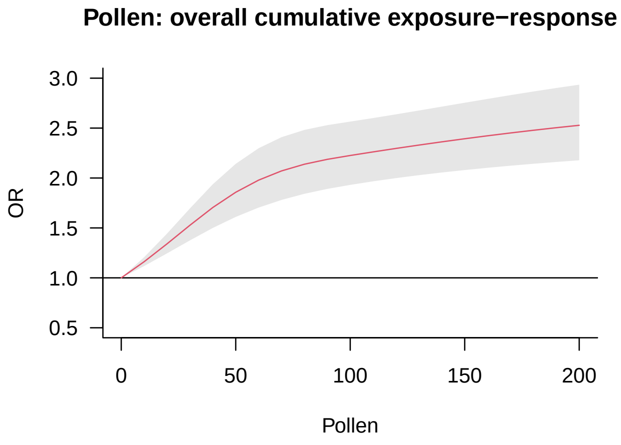
```
mod <- gnm(y ~ cbpoll + cbpm + cbtmean + btrend + dow, eliminate=stratum, data=data,
  family=binomial)
```

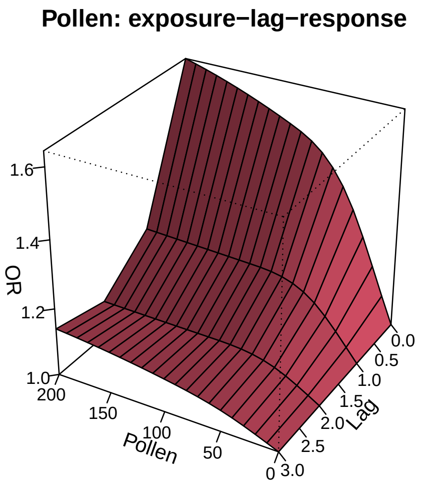
The estimated coefficients and associated (co)variance matrix of the model can now be used to predict the association of the various terms with the risk of respiratory symptoms, using the function `crosspred()`:

```
cppoll <- crosspred(cbpoll, mod, at=0:20*10, cen=0)
cppm <- crosspred(cbpm, mod, at=0:20*5, cen=0)
cptmean <- crosspred(cbtmean, mod, cen=15, by=1.5)
```

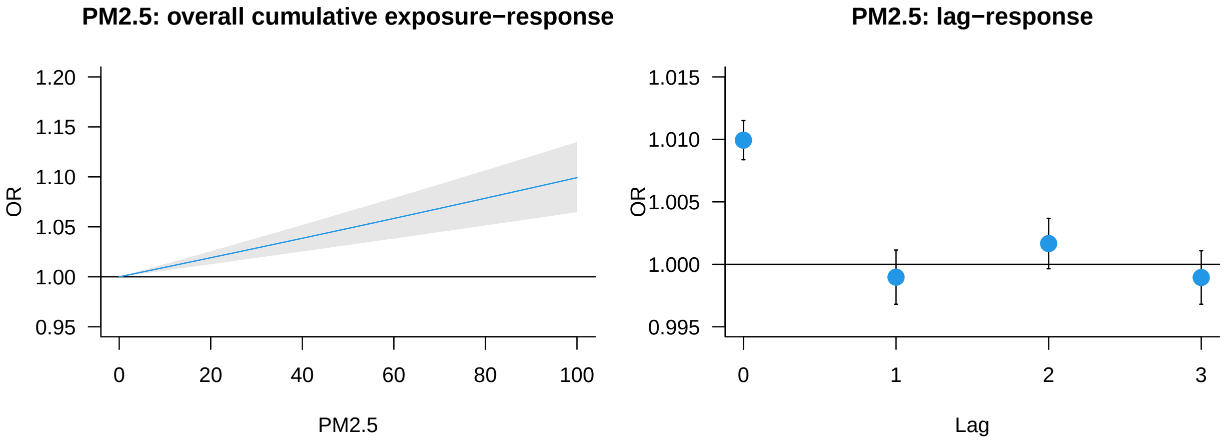
We can now represent graphically the association in both dimensions of exposure intensity and lag. Specifically, the plots below represent the overall cumulative exposure-responses (interpreted as the net associations accounting for the whole lag period), the full bi-dimensional exposure-lag-responses for non-linear relationships of pollen and temperature, and the lag-response corresponding to a 10μgr/m^3 increases in PM$_{2.5}$. The code:

```
plot(cppoll, "overall", xlab="Pollen", ylab="OR", col=2,
  main="Pollen: overall cumulative exposure-response", ylim=c(0.5,3))
plot(cppoll, xlab="Pollen", zlab="OR", main="Pollen: exposure-lag-response",
  cex.axis=0.8, col=2)
```

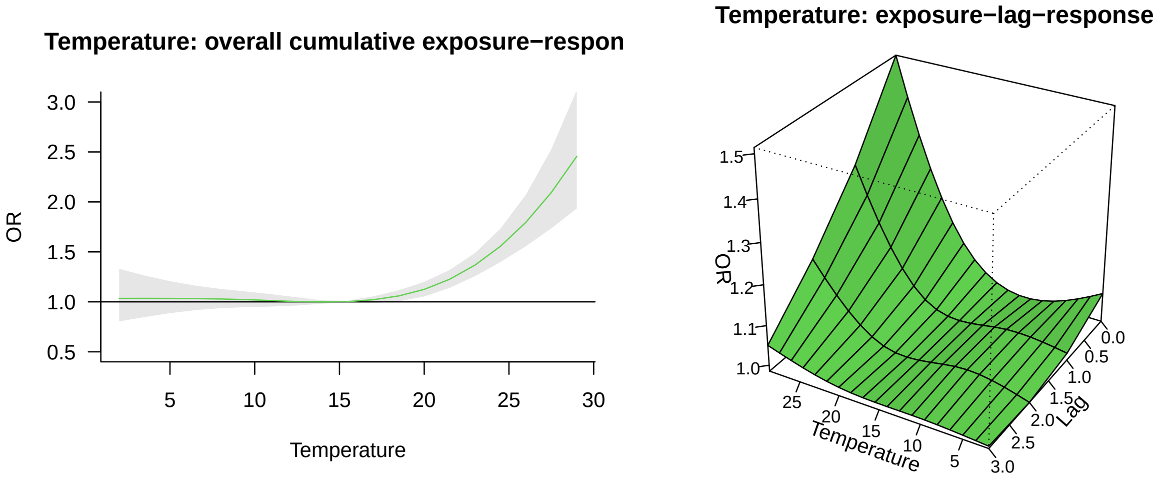

Pollen: overall cumulative exposure−response


Pollen: exposure−lag−response

```
plot(cppm, var=10, "overall", xlab="PM2.5", ylab="OR", col=4,
  main="PM2.5: overall cumulative exposure-response", ylim=c(0.95,1.20))
plot(cppm, var=10, ci="b", type="p", ylab="OR", col=4, pch=19, cex=1.7,
  xlab="Lag", main="PM2.5: lag-response", lab=c(3,5,7), ylim=c(0.995,1.015))
```



```r
plot(cptmean, "overall", xlab="Temperature", ylab="OR", col=3,
  main="Temperature: overall cumulative exposure-response", ylim=c(0.5,3))
plot(cptmean, xlab="Temperature", zlab="OR", ltheta=240, lphi=60, cex.axis=0.8,
  main="Temperature: exposure-lag-response", col=3)
```



The estimated associations are similar to those presented in the original analysis [@gasparrini2021epidem]. The results demonstrate the flexibility of the case time series design to investigate complex relationships with multiple exposures using individual data in a complex cohort setting.

Statistical analysis with individual time-stratified case crossover design

Firstly, we need to select cases in the dataset.

```r
cases <- data_cc %>% filter(y==1) %>% select(-c("pollen","pm","tmean"))
```

Second, generate case crossover data structure.

```r
ds <- data.frame(case_date=as.Date(character()),
                 control_date1=as.Date(character()), control_date2=as.Date(character()),
                 control_date3=as.Date(character()), control_date4=as.Date(character()),
                 control_date5=as.Date(character()), control_date6=as.Date(character()),
                 control_date7=as.Date(character()), control_date8=as.Date(character()),
                 control_date9=as.Date(character()), stringsAsFactors = FALSE)
series <- data.frame(case_date=seq(as.Date('2015-01-29'), as.Date('2018-11-19'),'day'))

for(i in seq(nrow(series))){
  cat(i, '')
  ds[i, 1] <- series$case_date[i]
  ds[i, 2] <- series$case_date[i]-28
  ds[i, 3] <- series$case_date[i]-21
  ds[i, 4] <- series$case_date[i]-14
  ds[i, 5] <- series$case_date[i]-7
  ds[i, 6] <- series$case_date[i]
  ds[i, 7] <- series$case_date[i]+7
  ds[i, 8] <- series$case_date[i]+14
  ds[i, 9] <- series$case_date[i]+21
  ds[i, 10] <- series$case_date[i]+28
}

# Transform wide format to long format
ds <- ds %>% gather('var','control_date', control_date1:control_date9) %>%
  filter(month(case_date)==month(control_date)) %>% select(-var) %>%
  mutate(case=case_when(case_date==control_date~1, TRUE~0))
```

Expand the case data and merge with environmental data

```r
cases <- cases %>% select(-y) %>% rename(case_date=date)
dat <- left_join(cases, ds, by="case_date")
envdata <- envdata %>% mutate(lag1_poll=lag(as.numeric(pollen), 1),
                              lag2_poll=lag(as.numeric(pollen), 2),
                              lag3_poll=lag(as.numeric(pollen),3),
                              lag1_pm=lag(as.numeric(pm),1),
                              lag2_pm=lag(as.numeric(pm),2),
                              lag3_pm=lag(as.numeric(pm),3),
                              lag1_tmean=lag(as.numeric(tmean),1),
                              lag2_tmean=lag(as.numeric(tmean),2),
                              lag3_tmean=lag(as.numeric(tmean),3)) %>%
  rename(control_date=date)
dat <- left_join(dat, envdata, by= "control_date")
```

Set model parameters and fit model. Customize the crossbasis function manually.

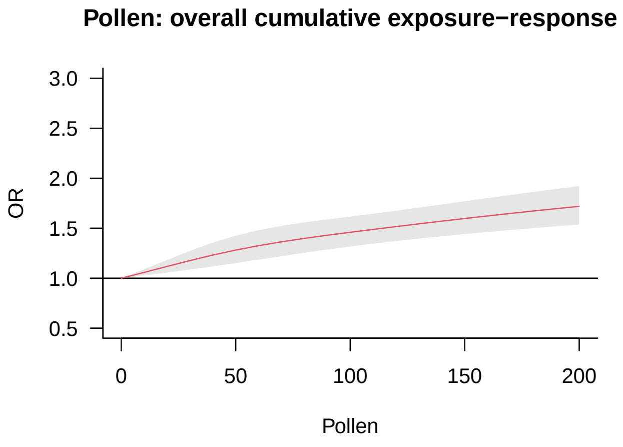
```r
cbpoll <- crossbasis(dat[,c(8,11:13)], lag=3, argvar=list(knots=c(40,100)),
  arglag=list(knots=1))
cbpm <- crossbasis(dat[,c(9,14:16)], lag=3, arglag=list("integer"))
cbtmean <- crossbasis(dat[,c(10,17:19)], lag=3, argvar=list(knots=1:2*10),
  arglag=list(knots=1))

mod <- clogit(case ~ cbpoll + cbpm + cbtmean + strata(id), data=dat)
```

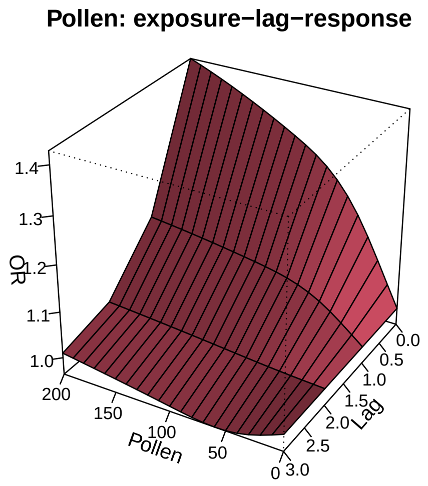
The estimated coefficients and associated (co)variance matrix of the model can now be used to predict the association of the various terms with the risk of respiratory symptoms, using the function `crosspred()`:

```
cppoll <- crosspred(cbpoll, mod, at=0:20*10, cen=0)
cppm <- crosspred(cbpm, mod, at=0:20*5, cen=0)
cptmean <- crosspred(cbtmean, mod, cen=15, by=1.5)
```

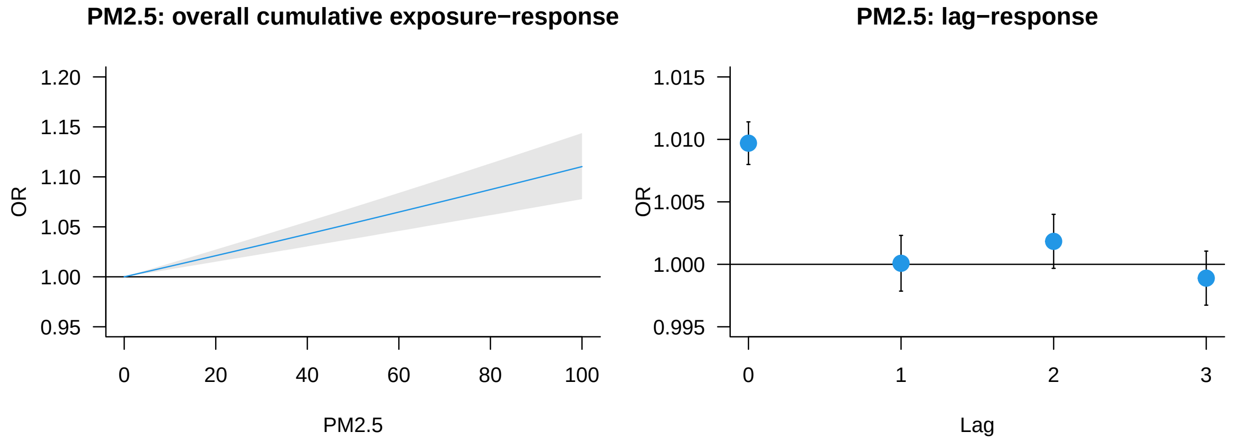
We can now represent graphically the association in both dimensions of exposure intensity and lag. Specifically, the plots below represent the overall cumulative exposure-responses (interpreted as the net associations accounting for the whole lag period), the full bi-dimensional exposure-lag-responses for non-linear relationships of pollen and temperature, and the lag-response corresponding to a 10μgr/m^3 increases in PM$_{2.5}$. The code:

```
plot(cppoll, "overall", xlab="Pollen", ylab="OR", col=2,
  main="Pollen: overall cumulative exposure-response", ylim=c(0.5,3))
plot(cppoll, xlab="Pollen", zlab="OR", main="Pollen: exposure-lag-response",
  cex.axis=0.8, col=2)
```

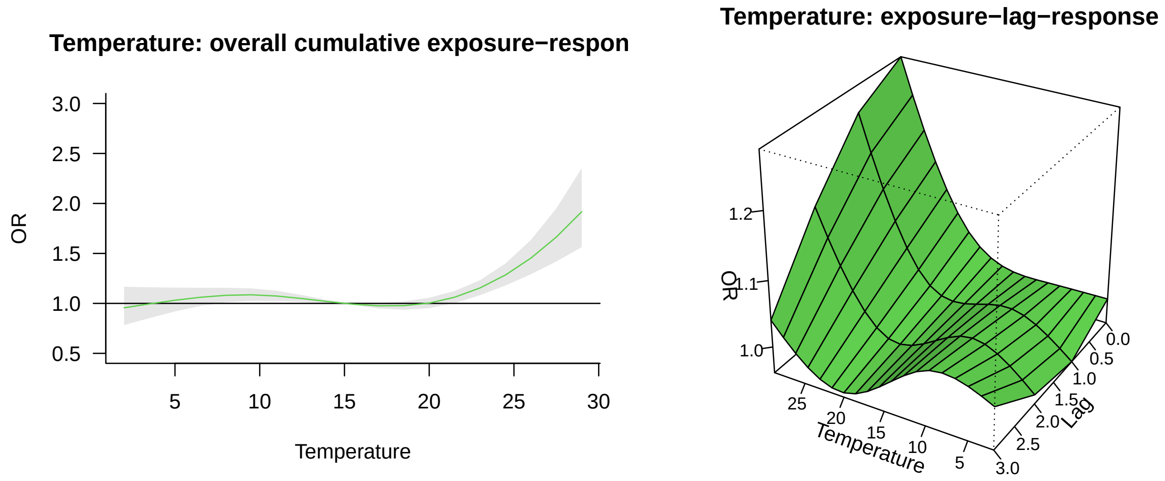




```
plot(cppm, var=10, "overall", xlab="PM2.5", ylab="OR", col=4,
  main="PM2.5: overall cumulative exposure-response", ylim=c(0.95,1.20))
plot(cppm, var=10, ci="b", type="p", ylab="OR", col=4, pch=19, cex=1.7,
  xlab="Lag", main="PM2.5: lag-response", lab=c(3,5,7), ylim=c(0.995,1.015))
```



```
plot(cptmean, "overall", xlab="Temperature", ylab="OR", col=3,
  main="Temperature: overall cumulative exposure-response", ylim=c(0.5,3))
plot(cptmean, xlab="Temperature", zlab="OR", ltheta=240, lphi=60, cex.axis=0.8,
  main="Temperature: exposure-lag-response", col=3)
```



Conclusion

The results from CTS and case crossover design are different, because:

1) The CTS used all information and adjusted day of week in the model. But the case crossover conducted a self-match for each case and one case only had 3-4 self-controls. Case crossover design adjusted temporal variation by design.
2) There is missing exposure data in case crossover dataset, because some cases are very close to the end of study period.
3) I would choose CTS over time stratified case crossover design, as the former has a more flexible data structure and doesn't require an expansion dataset, which saves the computational power.

References

1. Gasparrini, Antonio. 2021. “The case time series design.” Epidemiology 32 (6): 829–37.
2. Johnston, F H, A J Wheeler, G J Williamson, S L Campbell, P J Jones, I S Koolhof, C Lucani, N B Cooling, and D M J S Bowman. 2018. “Using smartphone technology to reduce health impacts from atmospheric environmental hazards.” Environmental Research Letters 13 (4): 044019.