Using the R package collin to visualize the effects of collinearity in distributed lag models

(collin version 0.0.1)

Jose Barrera-Gómez*1,2,3 and Xavier Basagaña†1,2,3

1ISGlobal
2Universitat Pompeu Fabra (UPF)
3CIBER Epidemiología y Salud Pública (CIBERESP)

February 11, 2021

Contents

1 Introduction 1
2 Getting started 2
3 How does the package work? 2
4 Illustrative examples 4
  4.1 Example 1: Windows of susceptibility in a cohort study . . . . . . . . . . . . . . . . 4
  4.2 Example 2: Time series study with linear effects . . . . . . . . . . . . . . . . . . . . 11
  4.3 Example 3: Time series study with nonlinear effects . . . . . . . . . . . . . . . . . . 16

Bibliography 26

1 Introduction

This document is a user’s guide for the R1 package collin for the visualization of the effects of collinearity in distributed lag models (DLNM). The package usage is based on two elements provided by the user: a model including a crossbasis created with the dlnm (https://cran.r-project.org/web/packages/dlnm/), and a set of hypothesized true effects. Then, collin performs a simulation study and provides a visualization of results to assess whether the actual results of the study could be driven by collinearity, as described in the original work by Basagaña and Barrera-Gómez[1]. The illustrative examples used there are reproduced here.

*jose.barrera@isglobal.org
†xavier.basagana@isglobal.org
1R is a free and open source software and it is available at CRAN (http://cran.r-project.org/).
2 Getting started

The last version released on CRAN can be installed directly within an R session by:

```
install.packages("collin")
```

A brief overview of the package is obtained by:

```
library(collin)
```

````
##
## This is collin 0.0.1. For details, use:
## > help(package = 'collin')
## and
## > browseVignettes('collin')
##
## To cite the methods in the package use:
## > citation('collin')
```

```
help(package = "collin")
```

Once the package has been installed, the vignettes, including the most recent version of this document, as well as the corresponding R code, are available through

```
browseVignettes("collin")
```

3 How does the package work?

The package works in a two-step procedure.

In the first step, the `collindlnm` function is used to simulate results from a DLNM created with the `dlnm` package\(^2\) and a hypothetical effect pattern, both provided by the user. The main arguments of the `collindlnm` function are:

- **model**: the fitted DLNM, which includes a crossbasis, to be evaluated. Currently, models allowed are those of class `glm` (i.e. a generalized linear model) or `lme` (i.e. a linear mixed effects model).
- **x**: a matrix or a vector, depending on whether the hypothetical effect to be explored is linear or non-linear, including the values of the predictor under study.
- **cb**: an object of class `crossbasis`, included in the model under study (`model`).
- **at**: the increase(s) in the predictor under study to be considered to report the effects of the variable. If the hypothetical effect to be analyzed is linear, then it must be a single number. If the hypothetical effect to be analyzed is non-linear, it must be a vector with at least two different values, in order to approximate the shape of the effect.
• cen: the reference value of the predictor of interest, used to calculate effects. If the effect is linear, the value of cen is irrelevant (and it is internally set to 0).

• effect: if the effect is linear, a vector of length (maximum(lag) + 1) including the linear effect at each lag. If the effect is non-linear, a matrix including the effect at each lag (column) for each value provided in at (rows).

• type: if type = "coef" (default), the hypothetical effect is supposed to be in the linear predictor scale (i.e. it is considered as values of regression coefficient in model). If type = "risk", the effect is supposed to be in terms of relative risks (i.e. exp(coef), as ORs or RR in logistic or Poisson families, respectively). If model is of class lme, then it must be type = "coef" (default).

• shape: the shape of the relationship between the linear predictor and the outcome. Default is, "linear". The case shape = "nonlinear" is currently implemented only if model is of class glm.

• nsim: the number of simulations. Default is 100.

• seed: the seed for reproducibility of results. Default is seed = NULL (no seed).

In the second step, a visualization of the simulation study is displayed using the specific plot() method, which allows to assessing whether the results of the original fitted model are compatible with collinearity problems observed when considering the alternative hypothetical effect pattern. The arguments for plot() depend on the hypothetical effect pattern being linear or non-linear.

For the case of a linear effect, the plot() method requires only two arguments:

• x: a result of the collindlm.

• lags: indicator of the lags where the results are displayed. Default is lags = NULL, in which case all lags are displayed.

For the case of a non-linear effect, the plot() method requires three additional arguments to allow the user to set how the plots associated at each value of at are shown:

• show: default option, show = "manual", requires the user to manually set the numbers of rows and columns to arrange the plots in a single array of plot, using the par function and setting the value of mfrow. This is the most flexible option to arrange the visualization in a document. The option show = "auto" is the same than show = "manual" except that the value of mfrow is automatically set by the package. The option show = "sequence" shows the plots sequentially, waiting for the user’s input before moving to the next plot.

• addlegend and varlegend: to add a label indicating, in each plot, the name of the predictor under analysis and the value of at.
4 Illustrative examples

For further details on the following illustrative examples, see the original work\cite{1}. Datasets mempm25 and rhospno2, included in the collin package and used in sections 4.1 and 4.2 of this document, are synthetic datasets generated with the R package synthpop\cite{2}, based on real datasets used in the original work\cite{1}. Hence, results shown in this document can (and should) differ from the original results.

First, we set the number of simulations and the seed that will be applied to all examples. We set the number of simulations \texttt{mynsim} = 50 to reduce the computation time needed to reproduce all examples in this vignette. However, in a particular study, the number of simulations can be increased (for instance, to the default value 100).

\begin{verbatim}
mynsim <- 50 # number of simulations
myseed <- 23984 # seed
\end{verbatim}

Additional packages required for the examples are:

\begin{verbatim}
library(nlme) # lme
library(dlnm)
## This is dlnm 2.3.9. For details: help(dlnm) and vignette('dlnmOverview').
library(splines) # ns
\end{verbatim}

4.1 Example 1: Windows of susceptibility in a cohort study

Here, we used data from a study by Rivas et al.\cite{3}, which aimed to estimate the association between air pollution exposure (PM$_{2.5}$, in $\mu g/m^3$) during the prenatal period and the first seven postnatal years on working memory tests taken at age 8 in a cohort of 2221 children. Exposure matrix contains the exposure to PM$_{2.5}$ at pregnancy, and from years 1 to 7.

\begin{verbatim}
# data summary:
summary(mempm25)
## id session school sex agecen
## 0001 : 4 1:2221 07 : 584 female:4280 Min. :-1.87694
## 0002 : 4 2:2221 17 : 496 male :4604 1st Qu.:-0.75990
## 0003 : 4 3:2221 25 : 472 Median :-0.06722
## 0004 : 4 4:2221 05 : 440 Mean : 0.00000
## 0005 : 4 32 : 420 3rd Qu.: 0.71306
## 0006 : 4 09 : 412 Max. : 3.16891
## (Other):8860 (Other):6060 NA's :13
## educ resses pm25y0
## university :5224 Min. :-0.385097 Min. : 7.169
## secondary :2628 1st Qu.:-0.159291 1st Qu.:14.731
## primary or less than primary: 988 Median : 0.034258 Median :16.113
## NA's : 44 Mean : 0.007765 Mean :16.423
## 3rd Qu.: 0.163290 3rd Qu.:17.932
\end{verbatim}

\url{https://cran.r-project.org/web/packages/synthpop/index.html}
The correlation between exposure to PM$_{2.5}$ at different periods, shown in Table 1 is high, with 18% of values exceeding 0.9. Children took the working memory tests in four repeated occasions throughout a year and children were nested in schools, so a 3-level mixed effects model framework was used. We used the distributed lag nonlinear model framework to model the effect of PM$_{2.5}$. We reproduced the original analyses by considering a linear effect of PM$_{2.5}$ and restricting the lagged effects with a quadratic $b$-spline with two equally-spaced internal knots. The model was further adjusted for age, sex, maternal education and residential neighborhood socioeconomic status. First, we start with the estimation from single-lag models.
<table>
<thead>
<tr>
<th>Pregnancy</th>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
<th>Year 4</th>
<th>Year 5</th>
<th>Year 6</th>
<th>Year 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pregnancy</td>
<td>1.00</td>
<td>0.92</td>
<td>0.89</td>
<td>0.88</td>
<td>0.74</td>
<td>0.61</td>
<td>0.60</td>
</tr>
<tr>
<td>Year 1</td>
<td>0.92</td>
<td>1.00</td>
<td>0.96</td>
<td>0.92</td>
<td>0.79</td>
<td>0.67</td>
<td>0.65</td>
</tr>
<tr>
<td>Year 2</td>
<td>0.89</td>
<td>0.96</td>
<td>1.00</td>
<td>0.92</td>
<td>0.74</td>
<td>0.65</td>
<td>0.59</td>
</tr>
<tr>
<td>Year 3</td>
<td>0.88</td>
<td>0.92</td>
<td>0.92</td>
<td>1.00</td>
<td>0.82</td>
<td>0.61</td>
<td>0.69</td>
</tr>
<tr>
<td>Year 4</td>
<td>0.74</td>
<td>0.79</td>
<td>0.74</td>
<td>0.82</td>
<td>1.00</td>
<td>0.81</td>
<td>0.80</td>
</tr>
<tr>
<td>Year 5</td>
<td>0.61</td>
<td>0.67</td>
<td>0.65</td>
<td>0.61</td>
<td>0.81</td>
<td>1.00</td>
<td>0.76</td>
</tr>
<tr>
<td>Year 6</td>
<td>0.60</td>
<td>0.65</td>
<td>0.59</td>
<td>0.69</td>
<td>0.80</td>
<td>0.76</td>
<td>1.00</td>
</tr>
<tr>
<td>Year 7</td>
<td>0.56</td>
<td>0.62</td>
<td>0.58</td>
<td>0.56</td>
<td>0.82</td>
<td>0.91</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Table 1: Correlation between PM\(_{2.5}\) concentrations at different lags.

```r
# set the exposure increase:
pm25change <- 10

# data.frame to store effects and CI:
pm25effects <- data.frame(lower = rep(NA, nlagspm25),
estimate = rep(NA, nlagspm25),
upper = rep(NA, nlagspm25))

# fit models:
for (i in 1:nlagspm25) {
  # select exposure lag:
  Ei <- Qpm25[, i]
  # fit model for that single lag:
  modi <- lme(wmemo ~ Ei + sex + agecen + educ + resses,
              data = mempm25,
              weights = ~ wei,
              random = ~ 1|school/id,
              na.action = na.omit,
              control = lmeControl(opt = "optim"))
  # get effect estimate (for Echange units increase):
  pm25effects[i, ] <- pm25change * intervals(modi)$fixed["Ei", ]
}
rm(Ei, modi)
```

A graphical representation of the effects under single-lag models is shown in Figure 1, which has been generated with the following code:

```r
par(las = 1)
xvalues <- 0:(nlagspm25 - 1)
with(pm25effects,
    plot(xvalues, estimate, ylim = range(pm25effects), pch = 19,
    xlab = "Year", ylab = "Change in mean working memory"))
with(pm25effects, segments(xvalues, lower, xvalues, upper))
abline(h = 0, lty = 2)
```

According to Figure 1, models including only PM\(_{2.5}\) from a single period showed negative associations between PM\(_{2.5}\) and working memory across all periods. Now, we fit the distributed lag model:
Figure 1: Estimated effect and 95% confidence intervals of a 10 $\mu g/m^3$ increase in PM$_{2.5}$ exposure in working memory score across the different time periods, obtained from single-lag models.

```r
# create crossbasis:
df <- 5
ekn <- equalknots(x = c(0, nlagspm25 - 1),
                   nk = NULL,
                   fun = "bs",
                   df = df,
                   degree = 2,
                   intercept = TRUE)
cbpm25 <- crossbasis(x = Qpm25,
                      lag = c(0, nlagspm25 - 1),
                      argvar = list(fun = "lin"),
                      arglag = list(fun = "bs", degree = 2, df = df, knots = ekn))

# fit model:
modmempm25 <- lme(wmemo ~ cbpm25 + sex + agecen + educ + resses,
                      data = mempm25,
                      weights = ~ wei,
                      random = ~ 1|school/id,
                      na.action = na.exclude,
                      control = lmeControl(opt = "optim"))

# predict effects at different lags
predmempm25 <- crosspred(basis = cbpm25, model = modmempm25, cen = 0, at = pm25change)
```

A graphical representation of the effects under the previous distributed lag model is shown in Figure 2, which has been generated with the following code:
Figure 2: Estimated effect and 95% confidence intervals of a 10 $\mu g/m^3$ increase in PM$_{2.5}$ exposure in working memory score across the different time periods, obtained from a distributed lag model.

According to Figure 2, the distributed lag model estimates strong opposing effects. Next, we will check if collinearity is a potential explanation for these results. We first try if the obtained pattern is consistent with a constant effect that has the same cumulative effect than the one obtained. The cumulative effect estimated by the fitted model is stored in the object `allfit` within the output `predmempm25`, which was obtained using the `crosspred` function above. We just need to divide that cumulative effect by the number of lags and use it as the common hypothetical effect at all lags:

```r
# constant effect (divide cumulative by number of lags):
(conseffpm25 <- rep(predmempm25$allfit / nlagspm25, nlagspm25))
## 10 10 10 10 10 10 10 10
```

Now we will pass the hypothetical effect to the `collindlm` function. Since `crosspred` above was applied to a linear model (specifically, of class `lme`), the results of the `crosspred` function are expressed in terms of the regression coefficients of the model. Hence, we need to use `collindlm` with `type = "coef"`, which is the default option, so we don’t need to specify it. Also, we don’t need to set the argument `shape` because in this case the hypothetical effect is linear, which is the default option for `shape`. Hence, the first step of the procedure is:

```r
simconseffpm25 <- collindlm(model = modmempm25, # the original fitted model
                         x = Qpm25, # matrix with PM2.5 values at each lag
                         cb = cbpm25, # the crossbasis included in the model
                        x = Qpm25, # matrix with PM2.5 values at each lag
                         cb = cbpm25, # the crossbasis included in the model
```

8
at = pm25change,  # increase in PM2.5 to compute effects
effect = conseffpm25,  # hypothetical effect
nsim = mynsim,
seed = myseed)

## .........10.........20.........30.........40.........50
##
## Simulations done.

The second step of the procedure uses the `plot()` method to visualize the results, as shown in Figure 3 using the following call to the `plot()` method:

```r
par(las = 1)
plot(simconseffpm25, xlab = "Year", ylab = "Change in mean working memory")
```

![Figure 3](image)

**Figure 3:** Estimated effect of a 10 $\mu$g/m$^3$ increase in PM$_{2.5}$ exposure across the different time periods over 50 simulations. Estimates from the same simulation run are connected with lines. The red thick line represents the effects observed in the real dataset (i.e. original fitted model). Results obtained when simulating a constant effect across all lags, with the cumulative effect being equal to the estimated using the real data.

According to Figure 3, the observed pattern is not consistent with a constant effect at all lags, with the same cumulative effect. We try now another pattern, one in which PM$_{2.5}$ only has a (negative) effect at lag 5, and has no effect on the other lags. The effect is four times the observed cumulative effect:

```r
lag5seffpm25 <- rep(0, nlagspm25)
lag5seffpm25[6] <- 4 * predmempm25$allfit
round(lag5seffpm25, 2)
```

```r
## [1] 0.00 0.00 0.00 0.00 0.00 -115.86 0.00 0.00
```
New simulations under that hypothetical effect:

```r
simlag5effpm25 <- collindlnm(model = modmempm25,
                          x = Qpm25, 
                          cb = cbpm25, 
                          at = pm25change, 
                          effect = lag5seffpm25, 
                          nsim = mynsim, 
                          seed = myseed)
```

```
# .........10.........20.........30.........40.........50
##
# Simulations done.
```

And the results, shown in Figure 4, are obtained using the `plot()` method:

```r
par(las = 1)
plot(simlag5effpm25, xlab = "Year", ylab = "Change in mean working memory")
```

![Figure 4](image.png)

**Figure 4:** Estimated effect of a 10 µg/m³ increase in PM$_{2.5}$ exposure across the different time periods over 50 simulations. Estimates from the same simulation run are connected with lines. The red thick line represents the effects observed in the real dataset (i.e. original fitted model). Results obtained when simulating a real effect of year 5 (four times the size of the cumulative effect estimated by the original model) and no effect of all other periods.

According to Figure 4, the estimated effect pattern using the real data is relatively similar to the one that would be obtained under a truth in which only year 5 has an effect. Under that scenario, due to the correlation between exposure at different times (collinearity) and to the specific choice to constraint lagged associations, most replications of the study would report non-existent negative effects at lags 1 and 6 and positive effects at lags 3 and 4. In a scenario like that, with a single period having an effect, constraining lagged effects with a spline function would not be a good choice, as
it imposes smooth changes across lags. In reality, however, no one knows the true data generating mechanism, which makes the choice of the appropriate lag function difficult.

4.2 Example 2: Time series study with linear effects

In this example, we analyzed the relationship between the daily number of hospital admissions for respiratory causes and ambient NO\textsubscript{2} concentrations (in µg/m\textsuperscript{3}) in the city of Barcelona (Spain) for years 2006-2015:

```
summary(rhospno2)
```

```
# date t year dow
# 1st Qu.:2008-07-01 1st Qu.:1279 1st Qu.:2008 Monday :522
# Median :2010-12-31 Median :2192 Median :2010 Tuesday :522
# Mean :2010-12-31 Mean :2192 Mean :2010 Wednesday:522
# 3rd Qu.:2013-07-01 3rd Qu.:3104 3rd Qu.:2013 Thursday :522
# Max. :2015-12-31 Max. :4017 Max. :2015 Friday :521
# SUnday :521
```

```
## temp no2 hresp
## Min. : 1.40 Min. : 3.00 Min. : 8.00
## 1st Qu.:11.78 1st Qu.: 46.00 1st Qu.: 26.00
## Median :16.80 Median : 60.00 Median : 34.00
## Mean :16.96 Mean : 61.34 Mean : 35.88
## 3rd Qu.:22.40 3rd Qu.: 74.00 3rd Qu.: 43.00
## Max. :30.40 Max. :159.00 Max. :103.00
## NA's :77
```

We used the DLNM framework with a generalized linear model with the quasi-Poisson family to allow for overdispersion. In particular, we assumed the effect of NO\textsubscript{2} to be linear (in the log scale), explored lagged effects of up to 14 days, and constrained the lag function to follow a natural spline with three internal knots equally-spaced in the log scale. The model was further adjusted for day of the week, temperature (using a crossbasis with a natural spline with 4 equally-spaced internal knots to model the non-linear effects of temperature, and a natural spline with 3 internal knots equally-spaced on the log scale to model the lag structure up to lag 21), and for trend and seasonality (using a natural spline of time with 7 degrees of freedom per year).

First, we need to create the matrix of the lagged values of the exposure, which can be done using the \texttt{lagpad} function. This function has two arguments: \texttt{x}, the numeric vector to be lagged, and \texttt{k}, the number of lags to be applied:

```
# create matrix with lagged data:
nlagsno2 <- 15  # number of lags considered (14 + 1)
Qno2 <- matrix(NA, nrow = dim(rhospno2)[1], ncol = nlagsno2)
for (i in 1:nlagsno2)
  Qno2[, i] <- lagpad(x = rhospno2$no2, k = i - 1)
```

```
# correlation between exposures:
corQno2 <- cor(Qno2, use = "complete.obs")
```
rownames(corQno2) <- colnames(corQno2) <- paste0("lag", 0:(nlagsno2 - 1))

print(corQno2, digits = 2)

Given day -1 d. -2 d. -3 d. -4 d. -5 d. -6 d. -7 d. -8 d. -9 d. -10 d. -11 d. -12 d. -13 d. -14 d.
-1 d. 1.00 0.62 0.33 0.22 0.18 0.21 0.30 0.38 0.27 0.13 0.08 0.09 0.12 0.23 0.32
-2 d. 0.62 1.00 0.62 0.33 0.22 0.19 0.21 0.31 0.38 0.27 0.14 0.09 0.09 0.13 0.24
-3 d. 0.33 0.62 1.00 0.62 0.34 0.23 0.20 0.22 0.31 0.39 0.28 0.15 0.10 0.11 0.11
-4 d. 0.22 0.33 0.62 1.00 0.62 0.34 0.23 0.20 0.22 0.31 0.40 0.29 0.15 0.11
-5 d. 0.21 0.19 0.23 0.34 0.62 1.00 0.62 0.35 0.23 0.20 0.22 0.32 0.40 0.29 0.16
-6 d. 0.30 0.21 0.19 0.23 0.34 0.62 1.00 0.62 0.35 0.23 0.20 0.22 0.32 0.40 0.30
-7 d. 0.38 0.31 0.21 0.20 0.23 0.35 0.62 1.00 0.62 0.35 0.24 0.20 0.23 0.33 0.41
-8 d. 0.27 0.38 0.31 0.22 0.20 0.23 0.35 0.62 1.00 0.62 0.34 0.23 0.20 0.23 0.33
-9 d. 0.13 0.27 0.39 0.31 0.22 0.20 0.23 0.35 0.62 1.00 0.62 0.34 0.23 0.20 0.23 0.23
-10 d. 0.08 0.14 0.28 0.39 0.31 0.22 0.20 0.24 0.34 0.62 1.00 0.62 0.35 0.23 0.20
-11 d. 0.09 0.09 0.14 0.28 0.40 0.32 0.22 0.20 0.23 0.34 0.62 1.00 0.63 0.35 0.24
-12 d. 0.12 0.09 0.09 0.15 0.29 0.40 0.32 0.23 0.20 0.23 0.35 0.63 1.00 0.63 0.35
-13 d. 0.21 0.13 0.09 0.10 0.15 0.29 0.40 0.33 0.23 0.20 0.23 0.35 0.63 1.00 0.63
-14 d. 0.32 0.24 0.13 0.10 0.11 0.16 0.30 0.41 0.33 0.23 0.20 0.24 0.35 0.63 1.00

Table 2: Correlation between NO$_2$ concentrations at different lags.

The correlation between NO$_2$ concentrations at different lags, shown in Table 2, were lower than in the previous example (Table 1), with highest values around 0.6 for adjacent days.

Now, we start the modelling with the estimates when including single lags in the model:

```
# crossbasis for temperature
# Fixing the knots at equally spaced values of temperature and at equally spaced
# log-values of lag. From:
ktemp <- equalknots(rhospno2$temp, nk = 4)
nlagstemp <- 22 # maximum lag for temperature + 1
klag <- logknots(nlagstemp - 1, nk = 3)

cbtemp <- crossbasis(x = rhospno2$temp,
  argvar = list(knots = ktemp),
  arglag = list(knots = klag),
  lag = nlagstemp - 1)

# number of years for the time spline:
nyears <- diff(range(rhospno2$year)) + 1

# get beta coefficients and CI for each model:
coefsno2single <- data.frame(estimate = rep(NA, nlagsno2),
  lower = rep(NA, nlagsno2),
  upper = rep(NA, nlagsno2))

for (i in 1:nlagsno2) {
  # select exposure lag:
  Ei <- Qno2[, i]
  coefsno2single[i, "estimate"] =
```
# fit model:
modi <- glm(hresp ~ Ei + cbtemp + ns(t, 7 * nyears) + dow,
    data = rhospno2,
    family = quasipoisson,
    na.action = na.exclude)

# get beta estimates and CI:
ints <- confint.default(modi)
coefsno2single$lower[i] <- ints["Ei", "2.5 %"]
coefsno2single$estimate[i] <- summary(modi)$coefficients["Ei", "Estimate"]
coefsno2single$upper[i] <- ints["Ei", "97.5 %"]
}

# set the exposure increase:
no2change <- 10

# compute effects (RRs):
effectno2single <- exp(no2change * coefsno2single)

A graphical representation of the effects under single-lag models is shown in Figure 5, which has been generated with the following code:

par(las = 1)
xvalues <- 0:(nlagsno2 - 1)
with(effectno2single,
    plot(xvalues, estimate, ylim = range(effectno2single), pch = 19, xlab = "Lag", ylab = "RR"))
with(effectno2single, segments(xvalues, lower, xvalues, upper))
abline(h = 1, lty = 2)

![Graphical representation of effects](image)

**Figure 5:** Estimated relative risk (RR) and 95% confidence intervals of hospital admission for respiratory causes for a 10 µg/m³ increase in ambient NO₂ concentration across the different time periods, obtained from single-lag models.
According to Figure 5, single-lag models showed significant increases in risk of respiratory hospital admission (i.e. relative risk, RR > 1) at lags 0 and 6, other periods with elevated non-significant RRs, and a non-significant RR < 1 at lags 1 and 2. Now, we fit the distributed lag model to the data:

```r
# crossbasis for NO2 (linear effect):
lagknots <- logknots(nlagsno2 - 1, nk = 3)
cbn2 <- crossbasis(x = rhospno2$no2,
               lag = c(0, (nlagsno2 - 1)),
               argvar = list(fun = "lin"),
               arglag = list(fun = "ns", knots = lagknots))
```

In this case in which we are going to include two crossbases in the model that will be passed to `collindlm`, it gives problems because of the names:

```r
colnames(cbttemp)
## [1] "v1.11" "v1.12" "v1.13" "v1.14" "v1.15" "v2.11" "v2.12" "v2.13" "v2.14"
## [10] "v3.11" "v3.12" "v3.13" "v3.14" "v3.15" "v4.11" "v4.12" "v4.13"
## [19] "v4.14" "v4.15" "v5.11" "v5.12" "v5.13" "v5.14" "v5.15"

colnames(cbn2)
## [1] "v1.11" "v1.12" "v1.13" "v1.14" "v1.15"

all(colnames(cbn2) %in% colnames(cbttemp))
## [1] TRUE
```

To solve it, we need to change the names of one of the crossbasis:

```r
# change the names of the crossbasis for temperature:
aux <- as.data.frame(cbttemp)
cbttemp <- dim(cbttemp)[2]
crosstempnames <- paste0("crosstemp", 1:cbttemp)
names(aux) <- crosstempnames
rhospno2 <- cbind(rhospno2, aux)
rm(aux)
names(rhospno2)
## [1] "date" "t" "year" "dow" "temp"
## [6] "no2" "hresp" "crosstemp1" "crosstemp2" "crosstemp3"
## [11] "crosstemp4" "crosstemp5" "crosstemp6" "crosstemp7" "crosstemp8"
## [16] "crosstemp9" "crosstemp10" "crosstemp11" "crosstemp12" "crosstemp13"
## [21] "crosstemp14" "crosstemp15" "crosstemp16" "crosstemp17" "crosstemp18"
## [26] "crosstemp19" "crosstemp20" "crosstemp21" "crosstemp22" "crosstemp23"
## [31] "crosstemp24" "crosstemp25"
```

Now we can fit the model with the two crossbases:
# model formula:
formhosp <- paste0("hresp ~ cbno2 + ",
        paste(crosstempnames, collapse = " + ",
        " + ns(t, 7 * nyears) + dow")
(formhosp <- as.formula(formhosp))
## hresp ~ cbno2 + crosstemp1 + crosstemp2 + crosstemp3 + crosstemp4 +
## crosstemp5 + crosstemp6 + crosstemp7 + crosstemp8 + crosstemp9 +
## crosstemp10 + crosstemp11 + crosstemp12 + crosstemp13 + crosstemp14 +
## crosstemp15 + crosstemp16 + crosstemp17 + crosstemp18 + crosstemp19 +
## crosstemp20 + crosstemp21 + crosstemp22 + crosstemp23 + crosstemp24 +
## crosstemp25 + ns(t, 7 * nyears) + dow

# fit model:
modrhospno2 <- glm(formhosp, family = quasipoisson, na.action = na.exclude, data = rhospno2)

# predict effects at different lags:
predrhospno2 <- crosspred(basis = cbno2, model = modrhospno2, cen = 0, at = no2change)

A graphical representation of the effects under the previous distributed lag model is shown in Figure 6, which has been generated with the following code:

```r
par(las = 1)
plot(predrhospno2, var = no2change, xlim = c(0, nlagsno2 - 1), main = "", xlab = "Day",
     ylab = "RR of hospital admission")
```

![Graphical representation of effects](image)

**Figure 6:** Estimated relative risk (RR) and 95% confidence intervals of hospital admission for respiratory causes for a 10 µg/m³ increase in ambient NO₂ concentration across the different time periods, obtained from a distributed lag model.

According to Figure 6, when fitting the distributed lag model, there was a statistically significant increase in respiratory hospital admissions associated with levels of NO₂ at lag 0, followed by
a statistically significant decreased risk at lags 1 and 2, and a subsequent statistically significant increase around lag 5. The decrease in risk at lags 1 and 2 could be consistent with the harvesting or short-term mortality displacement phenomenon (details in the original work[1]). However, there is also the possibility that this decrease in risk and the subsequent increases around lag 5 could be explained by collinearity since, as we showed above, collinearity can induce estimates with opposing signs. To explore its plausibility, we will analyze a hypothetical truth in which the real effect exists only at lag 0, with the same size as the estimated by the fitted model.

```r
# Effect (RRs) only at lags 0, same as observed
RRveclag0 <- rep(1, nlagsno2)
RRveclag0[1] <- predrhospno2$matRRfit[, "lag0"]
RRveclag0
## [1] 1.006564 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
## [9] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
```

Now we pass the hypothetical effect to `collindlnm`. Since it is given as RRs, we need to set `type = "risk"`:

```r
simlag0effno2 <- collindlnm(model = modrhospno2,
                          x = Qno2,
                          cb = cbno2,
                          at = no2change,
                          effect = RRveclag0,
                          type = "risk",
                          nsim = mynsim,
                          seed = myseed)
```

Results displayed in Figure 7 show that, under a hypothetical truth in which only lag 0 has a real effect, the pattern of the estimated effects bear some similarity to those obtained with the real data (red line), so that collinearity could be involved in these results. I.e. even under the situation in which only lag 0 has a real effect, distributed lag models can suggest a reduction in risk at lags 1-2 and subsequent increases in risk around lag 5. It is important to note that the observed pattern is compatible with many real scenarios, and in particular it is also compatible with a scenario with a real increase in risk at lag 0 and a real decrease in risk at lag 2 (e.g. because of the harvesting phenomenon) (details in the original work[1]).

4.3 Example 3: Time series study with nonlinear effects

In this example, we analyzed the relationship between daily mortality and ambient temperature in Chicago from 1987 to 2000. These data are available as part of the R package `dlnm`:
**Figure 7:** Estimated relative risk (RR) of hospital admission for respiratory cause for a 10 $\mu$g/m$^3$ increase in ambient NO$_2$ concentration across different lags, obtained from a distributed lag model, over 50 simulations. Estimates from the same simulation run are connected with lines. The results were obtained when simulating an effect only at lag 0 and of the same magnitude as the estimated with the real data. The red thick line represents the RRs estimated with the real dataset.

```r
chica <- chicagoNMMAPS[, c("date", "time", "year", "dow", "death", "temp", "pm10")]
summary(chica)
```

```
# date time year dow
# 1st Qu.:1990-07-02 1st Qu.:1279 1st Qu.:1990 Monday :730
# Median :1993-12-31 Median :2558 Median :1994 Tuesday :730
# Mean :1993-12-31 Mean :2558 Mean :1994 Wednesday:730
# 3rd Qu.:1997-07-01 3rd Qu.:3836 3rd Qu.:1997 Thursday :731
# Max. :2000-12-31 Max. :5114 Max. :2000 Friday :731
# Saturday :731
# death temp pm10
# Min. : 69.0 Min. :-26.667 Min. : -3.05
# 1st Qu.:105.0 1st Qu.: 1.667 1st Qu.: 20.77
# Median :114.0 Median : 10.556 Median : 30.25
# Mean :115.4 Mean : 10.107 Mean : 33.74
# 3rd Qu.:124.0 3rd Qu.: 19.444 3rd Qu.: 42.42
# Max. :411.0 Max. :33.333 Max. :356.18
# NA's :251
```

First, we calculate the matrix of lagged values of temperature:

```r
# create matrix with lagged data:
nlagstemp <- 31  # number of lags considered (30 + 1)
Qtemp <- matrix(NA, nrow = dim(chica)[1], ncol = nlagstemp)
```
The correlation between temperature in two consecutive days is 0.94, the correlation is still greater than 0.8 for days separated by 8 days or less, and it is around 0.7 for a 30-day separation. We used the distributed lag nonlinear model framework, with the same specifications used in the vignette of the dlnm package, to model the association between mortality and temperature.[2] Namely, we used a crossbasis for temperature, using a quadratic b-spline with 3 equally-spaced internal knots to model the exposure-response association, and a natural spline with 3 equally-spaced internal knots in the log space to model the lagged association up to lag 30. The quasi-Poisson regression model included as additional covariates day of the week, PM$_{10}$ concentrations (modeled with a crossbasis assuming linear effects and a strata lag structure up to lag 1), and a control for trends and seasonality with a natural spline of time with 7 degrees of freedom per year.

First, we create the crossbasis for PM$_{10}$:

```r
# crossbasis for PM10:
cbpm10 <- crossbasis(x = chica$pm10,
                      lag = 1,
                      argvar = list(fun = "lin"),
                      arglag = list(fun = "strata"))
```

Now, we start the modelling with the estimates when including single lags in the model:

```r
# reference value of temperature for effects calculation:
centemp <- 21

# evaluation points (values of temperature):
attemp <- c(-20, 0, 33)

# get beta coefficients and CI for each model:
coefs <- lower <- upper <- matrix(NA, nrow = dim(Qtemp)[2], ncol = length(attemp))

# number of years for time spline:
yearschica <- diff(range(chica$year, na.rm = TRUE)) + 1

for (i in 1:nlagstemp) {
  Ei <- Qtemp[, i]
  coefs[i, ] <- lower[i, ] <- upper[i, ] <- summary(lm(Ei ~ centemp + attemp))
}
# crossbasis for lag i of temperature:
cbi <- onebasis(Ei, fun = "bs", knots = ktemp, degree = 2)

# fit model:
modi <- glm(death ~ cbi + baspm + ns(time, 7 * nyearschica) + dow,
data = chica,
family = quasipoisson)

# get effect estimates and CI:
predi <- crosspred(basis = cbi, model = modi, at = attemp, cen = centemp)
lower[i, ] <- t(predi$matRRlow)
coefs[i, ] <- t(predi$matRRfit)
upper[i, ] <- t(predi$matRRhigh)

A graphical representation of the effects under single-lag models is shown in Figure 8, which has been generated with the following code:

```r
par(las = 1, mfrow = c(3, 1), mar = c(4, 4, 0, 2) + 0.1)
for (i in 1:length(attemp)) {
  plot(0:(nlagstemp - 1), coefs[, i], pch = 19, ylim = c(min(lower[, i]), max(upper[, i])),
       xlab = "", ylab = "RR")
  segments(0:(nlagstemp - 1), lower[, i], 0:(nlagstemp - 1), upper[, i])
  abline(h = 1, lty = 2)
  legend("topright", paste0("Temp = ", attemp[i]))
  mtext("Lag", side = 1, line = 2, cex = 0.7)
}
```

Figure 8 shows that mortality risk increased with cold temperatures for lags < 10 days, except for lag 0, which even showed a protective effect at 0°C (compared to 21°C). For heat, increased risks during the first four days were observed, followed by some lags with protective effects. Now fit the distributed lag model to the data:

```r
# fixing the knots at equally spaced log values of lag:
klag <- logknots(nlagstemp - 1, nk = 3)

# crossbasis matrix for temperature:
cbtemp <- crossbasis(x = chica$temp,
                      argvar = list(fun = "bs", knots = ktemp),
                      arglag = list(knots = klag),
                      lag = nlagstemp - 1)

# fit model:
modtemp <- glm(death ~ cbtemp + baspm + ns(time, 7 * nyearschica) + dow,
               data = chica,
               family = quasipoisson)

# effect estimates:
predtemp <- crosspred(basis = cbtemp, model = modtemp, at = attemp, cen = centemp)
```

A graphical representation of the effects under the previous distributed lag model is shown in Figure 9, which has been generated with the following code:
Figure 8: Relative risks (RR) and 95% confidence intervals for the associations between temperature and mortality by lag, using single-lag models. Results are presented for temperatures -20°C, 0°C, 33°C, taking 21°C as a reference. The effect of temperature was modeled using a quadratic b-spline with 3 equally-spaced internal knots.

par(las = 1, mfrow = c(3, 1), mar = c(4, 4, 0, 2) + 0.1)
plot(predtemp, var = attemp[1])
legend("topright", paste0("Temp = ", attemp[1]))

plot(predtemp, var = attemp[2], yaxt = "n", ylim = c(0.94, 1.05))
axis(2, at = c(0.96, 0.98, 1, 1.02, 1.04))
legend("topright", paste0("Temp = ", attemp[2]))

plot(predtemp, var = attemp[3])
legend("topright", paste0("Temp = ", attemp[3]))

According to Figure 9, associations were similar but more precise than those from single-lag models (Figure 8). A protective association at lag 0 was detected at both -20°C and 0°C. Now, we analyze the scenario in which there were no true RRs below one:

RRmattemp <- predtemp$matRRfit
round(RRmattemp, 2)
Figure 9: Relative risks (RR) and 95% confidence intervals for the associations between temperature and mortality by lag, using a distributed lag model. Results are presented for temperatures -20°C, 0°C, 33°C, taking 21°C as a reference. The effect of temperature was modeled using a cross-basis with a quadratic b-spline with 3 equally-spaced internal knots for the exposure-response association and a natural spline with 3 equally-spaced internal knots in the log space to model the lagged association.
# all effects null from lag 8 included
RRmattemp[, paste0("lag", 6:(nlagtemp - 1))] <- 1

# at temp 1:
RRmattemp[attempc[1], paste0("lag", 0:2)] <- 1
RRmattemp[attempc[1], paste0("lag", 3:5)] <- c(1.07, 1.12, 1.06)

# at temp 2:
RRmattemp[attempc[2], paste0("lag", 0:2)] <- 1
RRmattemp[attempc[2], paste0("lag", 3:5)] <- c(1.08, 1.03, 1.01)

# at temp 3:
RRmattemp[attempc[3], paste0("lag", 0:5)] <- c(1.15, 1.20, 1.22, 1.15, 1.10, 1.04)

RRmattemp

Now we need to set type = "risk" (because we have RRs) and shape = "nonlinear":

```r
simchicalag0null <- collindlnm(model = modtemp,
   x = chica$temp,
   cb = cbtemp,
   at = attemp,
   cen = centemp,
   effect = RRmattemp,
   type = "risk",
   shape = "nonlinear",
   nsim = mynsim,
   seed = myseed)
```

The results, shown in Figure 10, are obtained using the plot() method:

```r
par(las = 1, mfrow = c(3, 1), mar = c(4, 4, 2, 2))
plot(simchicalag0null, varlegend = "Temperature")
```
Figure 10: Estimated relative risks (RR) for mortality as a function of temperature obtained from distributed lag models, over 50 simulations. Estimates from the same simulation run are connected with gray lines. The red thick line represents the RRs observed in the real dataset. Results are presented for temperatures -20°C, 0°C, 33°C, taking 21°C as a reference. The results were obtained when simulating data with the following RRs: At temperature -20°C: RR = 1 at lags 0-2, and 6-30, RR = 1.07 at lag 3, RR = 1.12 at lag 4 and RR = 1.06 at lag 5; at temperature 0°C: RR = 1 for lags 0-2 and 6-30, RR = 1.08 at lag 3, RR = 1.03 at lag 4, RR 1.01 at lag 5; at temperature 33°C: RR = 1.15 at lag 0, RR = 1.2 at lag 1, RR 1.22 at lag 2, RR = 1.15 at lag 3, RR = 1.10 at lag 4, RR = 1.04 at lag 5 and RR = 1 at lags 6-30.

The plot() method also allows the user to select a subset of lags to be shown, using the argument `lags` (by default, all lags are shown). Also, we can set `show = "auto"` to let the grid plot be arranged automatically. For instance, the following code produces Figure 11:

```r
par(las = 1)
plot(simchicalag0null, lags = 0:8, show = "auto", varlegend = "Temperature")
```

The gray lines in Figure 10 show the results obtained when data were simulated from a scenario in which there were no true RRs below one. Hence, results obtained in that scenario could be compatible with the estimated effects using the real data, i.e. RR < 1 at lag 0 for cold temperatures and RR < 1 at the second week for hot temperatures. Still, even after exploring several potential
Figure 11: (Same than Figure 10 but showing until lag 10). Estimated relative risks (RR) for mortality as a function of temperature obtained from distributed lag models, over 50 simulations. Estimates from the same simulation run are connected with gray lines. The red thick line represents the RRs observed in the real dataset. Results are presented for temperatures -20°C, 0°C, 33°C, taking 21°C as a reference. The results were obtained when simulating data with the following RRs: At temperature -20°C: RR = 1 at lags 0-2, and 6-30, RR = 1.07 at lag 3, RR = 1.12 at lag 4 and RR = 1.06 at lag 5; at temperature 0°C: RR = 1 for lags 0-2 and 6-30, RR = 1.08 at lag 3, RR = 1.03 at lag 4, RR 1.01 at lag 5; at temperature 33°C: RR = 1.15 at lag 0, RR = 1.2 at lag 1, RR 1.22 at lag 2, RR = 1.15 at lag 3, RR = 1.10 at lag 4, RR = 1.04 at lag 5 and RR = 1 at lags 6-30.

scenarios, the observed results lay at the extreme of the obtained distribution. This, and the fact that single-lag models also show RR < 1 at lag 0 for cold temperatures, suggest that there might be other explanations for this result.
## Warning in par(opar): el parámetro del gráfico "cin" no puede ser especificado
## Warning in par(opar): el parámetro del gráfico "cra" no puede ser especificado
## Warning in par(opar): el parámetro del gráfico "csi" no puede ser especificado
## Warning in par(opar): el parámetro del gráfico "cxy" no puede ser especificado
## Warning in par(opar): el parámetro del gráfico "din" no puede ser especificado
## Warning in par(opar): el parámetro del gráfico "page" no puede ser especificado
Bibliography

