

# Package ‘dlnm’

November 4, 2009

**Type** Package

**Title** Distributed Lag Non-linear Models

**Version** 1.1.1

**Date** 2009-11-03

**Author** Antonio Gasparrini and Ben Armstrong

**Maintainer** Antonio Gasparrini <antonio.gasparrini@lshtm.ac.uk>

**Imports** stats, graphics, grDevices, splines, tsModel

**Depends** stats, graphics, splines

**Suggest** mgcv, NMMAPSlite

**Description** The package dlnm contains functions to specify basis and cross-basis matrices in order to run distributed lag models and their non-linear extension, then to predict and graph the results for a fitted model.

**License** GPL (>= 2)

**LazyLoad** yes

**Repository** CRAN

**Date/Publication** 2009-11-04 17:51:48

## R topics documented:

dlnm-package . . . . .	2
chicagoNMMAPS . . . . .	3
crossbasis . . . . .	4
crossplot . . . . .	8
crosspred . . . . .	10

<b>Index</b>	<b>13</b>
--------------	-----------

## Description

The package **dlnm** contains functions to specify basis and cross-basis matrices in order to run distributed lag models (DLM) and their non-linear extension (DLNM), then to predict and plot the results for a fitted model.

## Details

Distributed lag non-linear models (DLNM) represent a modelling framework to describe simultaneously non-linear and delayed dependencies in time-series data. This methodology is based on the definition of a *cross-basis*, a bi-dimensional space of functions specifying the dependency along the space of the predictor and along lags. The cross-basis functions are built combining the basis functions for the two dimensions, chosen among a set of possible bases. This family includes simple distributed lag models (DLM) as a special case. The DLNM framework is developed for time series data: the potential applications to other study designs are currently under evaluation and will be explored in next releases.

Given a series of observations ordered and equally spaced in time, `crossbasis` creates a matrix object of class "crossbasis" containing the transformed variables to be included in the model formula. The estimation is obtained by the default model command `glm`. The accuracy of the results using alternative commands is not guaranteed. After the model fitting, `crosspred` predicts the results for a set of suitable values of the original predictor and stores them in a "crosspred" object. Finally, `crossplot` offers a set of choices to plot the results.

Use `citation("dlnm")` to cite this package.

A list of changes included in the current and previous versions can be found typing `file.show(system.file("Change", package = "dlnm"))`.

For further information on DLNM, see the references below. For a detailed description of the capabilities of the package, refer to:

`vignette("dlnmOverview")`

## Author(s)

Antonio Gasparrini and Ben Armstrong

Maintainer: Antonio Gasparrini, <antonio.gasparrini@lshtm.ac.uk>

## References

Armstrong, B. Models for the relationship between ambient temperature and daily mortality. *Epidemiology*. 2006, **17**(6):624-31.

## See Also

`crossbasis` to create the basis and cross-basis matrices. `crosspred` to predict the effects after model fitting. `crossplot` to plot several type of graphs.

---

`chicagoNMMAPS`*Daily mortality weather and pollution data for Chicago*

---

## Description

The dataset contains daily mortality (all causes, CVD, respiratory), weather (temperature, dew point temperature, relative humidity) and pollution data (PM10 and ozone) for Chicago in the period 1987-2000 from the National Morbidity, Mortality and Air Pollution Study (NMMAPS)

## Usage

```
data(chicagoNMMAPS)
```

## Format

A data frame with 5114 observations on the following 14 variables.

```
date Date in the period 1987-2000
time The sequence of observations
year Year
month Month (numeric)
doy Day of the year
dow Day of the week (factor)
death Counts of all cause mortality excluding accident
cvd Cardiovascular Deaths
resp Respiratory Deaths
temp Mean temperature (in Celsius degrees)
dptp Dew point temperature
rhum Mean relative humidity
pm10 PM10
o3 Ozone
```

## Details

These data represents a subsample of the variables included in the NMMAPS dataset for Chicago.

The variable `temp` is derived from the original `tmpd` after a transformation from Fahrenheit to Celsius. The variables `pm10` and `o3` are an approximated reconstruction of the original series, adding the de-trended values and the median of the long term trend. This is the reason they include negative values. See `vignette('PollutantProcess')` from the packages **NMMAPSdata** or **NMMAPSlite**.

**Source**

The complete dataset is available at the Internet-based Health and Air Pollution Surveillance System (iHAPSS) website:

<http://www.ihapss.jhsph.edu>

or through the packages **NMMAPSdata** or **NMMAPSlite**.

---

crossbasis

*Generate a cross-basis matrix for a DLNM*

---

**Description**

Generate the basis functions for the two spaces of predictor and lags, choosing among a set of possible bases. Then, these functions are combined in order to create the related cross-basis matrix, which can be included in a model formula to fit a distributed lag non-linear model (DLNM).

**Usage**

```
crossbasis(var, vartype="ns", vardf=1, vardegree=1, varknots=NULL,
varbound=range(var), varint=FALSE, cen=TRUE, cenvalue=mean(var),
maxlag=0, lagtype="ns", lagdf=1, lagdegree=1, lagknots=NULL,
lagbound=c(0,maxlag), lagint=TRUE)

## S3 method for class 'crossbasis':
summary(object, ...)
```

**Arguments**

The arguments below define two set of basis functions calling the internal functions `mkbasis` and `mklagbasis`. The first one is applied to `var`, in order to describe the relationship in the space of the predictor. The second one is applied to a new vector `0:maxlag`, in order to describe the relationship in the space of lags. Many arguments refer to the specific basis for each space (with stub `var-` or `lag-`). Then, the two set of basis functions are combined in order to create the related cross-basis functions.

the predictor variable, defined as a numeric vector of ordered observations.

`vartype`, `lagtype`

`var` type of basis. See Details below for the list of possible choices.

`vardf`, `lagdf`

dimension of the basis, equivalent to number of degrees of freedom spent to specify the relationship in each space. They depend on `knots` if provided, or on `degree` for `type="poly"`.

`vardegree`, `lagdegree`

degree of polynomial. Used only for `type` equal to `"bs"` (degree of the piecewise polynomial for the B-spline) or `"poly"` (degree of the polynomial).

<code>varknots</code> , <code>lagknots</code>	knots location for the basis. They specify the position of the internal knots for "ns" and "bs", the cut-off points for "strata" (defining right-open intervals) and the threshold(s)/cut-off points for "lthr", "hthr" and "dthr". They must be set within the range of <code>var</code> and <code>0:maxlag</code> , respectively, and if provided, are automatically ordered and made unique, determining the value of <code>df</code> . If only <code>df</code> is provided, <code>varknots</code> are placed at equally spaced quantiles (in the space of predictor), and <code>lagknots</code> at equally spaced values on the log scale of lags.
<code>varbound</code> , <code>lagbound</code>	boundary knots (sometimes called external knots). Used only for <code>type</code> equal to "ns" and "bs".
<code>varint</code> , <code>lagint</code>	logical. If TRUE and <code>df</code> >1, an 'intercept' is included in the basis. The default values should not be changed: see Warnings below.
<code>cen</code>	logical. If TRUE, the basis functions for the space of predictor are centered. See Note below.
<code>cenvalue</code>	centering value, used as a reference point for the predicted effects.
<code>maxlag</code>	a positive value defining the maximum lag.
<code>object</code>	a object of class "crossbasis".
<code>...</code>	additional arguments to be passed to <code>summary</code> .

## Details

The value in `type` defines the basis for each space (predictor and lags). It must be one of:

"ns": natural cubic B-splines (constrained to be linear beyond the boundary knots). Specified by `knots` (internal knots) and `bound` (boundary or external knots). See the functions `ns` for additional information. If `knots` is provided, the dimension `df` is set to `length(knots)+1+int`. An intercept is included if `int=T`. The transformed variables can be centered at `cenvalue`.

"bs": B-splines characterized by `degree` (degree of the piecewise polynomial). Specified by `knots` (internal knots) and `bound` (boundary or external knots). See the functions `bs` for additional information. If `knots` is provided, the dimension `df` is set to `length(knots)+degree+int`; if not, `df` must be higher than `degree+int`. An intercept is included if `int=T`. The transformed variables can be centered at `cenvalue`.

"strata": strata variables (dummy parameterization) determined by internal cut-off values specified in `knots`, which represent the lower boundaries for the right-open intervals. Intervals containing no observation are automatically discarded. If `knots` is provided, the dimension `df` is set to `length(knots)+int`. A dummy variable for the reference stratum (the first one by default) is included if `int=T`, generating a full rank basis. Never centered.

"poly": polynomial with power specified by `degree`. The dimension `df` is set to `degree+int`. An intercept, corresponding to a vector of 1's (the power 0 of the polynomial) is included if `int=T`. The transformed variables can be centered at `cenvalue`.

"integer": strata variables (dummy parameterization) for each integer values, expressly created to specify an unconstrained function in the space of lags. `df` is set automatically to the number of integer values minus 1 plus `int`. A dummy variable for the reference stratum (the first one by default) is included if `int=T`, generating a full rank basis. Never centered.

"*hthr*", "*lthr*": high and low threshold parameterization, with a linear relationship above or below the threshold, respectively, and flat otherwise. The threshold is chosen by *knots*: if more than one is provided, a piecewise linear relationship is applied above the first knot or below the last one, respectively, with the slope changing at each further knot. *df* is automatically set to  $\text{length}(\text{knots}) + \text{int}$ . An intercept (corresponding to a vector of 1's) is included if  $\text{int} = \text{T}$ . Never centered.

"*dthr*": double threshold parameterization (2 independent linear relationships above the second and below the first threshold, flat between them). The thresholds are chosen by *knots*. If only one is provided, the threshold is unique (V-model). If more than 2 are provided, the first and the last ones are chosen. *df* is automatically set to  $2 + \text{int}$ . An intercept (corresponding to a vector of 1's) is included if  $\text{int} = \text{T}$ . Never centered.

"*lin*": linear relationship (untransformed apart from optional centering). *df* is automatically set to  $1 + \text{int}$ . An intercept (corresponding to a vector of 1's) is included if  $\text{int} = \text{T}$ . It can be centered at *cenvalue*.

Some arguments can be automatically changed for not sensible combinations, or set to `NULL` if not required.

For a detailed overview of the options, see:

```
vignette("dlnmOverview")
```

## Value

A matrix object of class "`crossbasis`" which can be included in a model formula in order to fit a DLNM. It contains the attributes `crossdf` (global number of degrees of freedom) and `range` (range of the original vector of observations). Additional attributes are returned that correspond to the arguments to `crossbasis`, and explicitly give `type`, `df`, `degree`, `knots`, `bound`, `cen`, `cenvalue` and `maxlag` related to the corresponding basis (with stub `var-` or `lag-`) for use of `crosspred`. The function `summary.crossbasis` returns a summary of the cross-basis matrix and the related attributes, and can be used to check the options for the bases chosen for the two dimensions.

## Warnings

It is strongly recommended to avoid the inclusion of an intercept in the basis for `var`, otherwise the presence of the additional intercept (when included) in the model used to fit the data will cause some of the cross-basis variables to be excluded. Conversely, an intercept should always be included in the basis for the space of lags when `lagtype` is equal to "`ns`", "`bs`", "`strata`" or "`poly`".

## Note

The values in `var` are expected to be equally spaced (with that space defining a lag unit) and ordered in time. `NA` values are allowed.

The name of the crossbasis object will be used by `crosspred` in order to extract the related estimated parameters. This name must not match the names of other predictors in the model formula. In addition, if more than one variable is transformed by cross-basis functions in the same model, different names must be specified.

For continuous functions specified with `vartype` equal to "`ns`", "`bs`", "`poly`" or "`lin`", the reference for the effects predicted by `crosspred` is set at `cenvalue`. For the other choices, the

reference is automatic: for vartype equal to "strata" and "integer", the reference is the first interval, while for vartype equal to "hthr", "lthr" and "dthr", the reference is the region of null effect below, above or between the threshold(s), respectively.

### Author(s)

Antonio Gasparrini, <antonio.gasparrini@lshtm.ac.uk>

### References

Armstrong, B. Models for the relationship between ambient temperature and daily mortality. *Epidemiology*. 2006, **17**(6):624-31.

### See Also

[crosspred](#), [crossplot](#)

### Examples

```
# Example 1. See crosspred and crossplot for other examples

### simple DLM for the effect of PM10 on mortality up to 15 days of lag
### space of predictor: linear effect for PM10
### space of predictor: 5df natural cubic spline for temperature
### lag function: 4th degree polynomial for PM10
### lag function: strata intervals at lag 0 and 1-3 for temperature

data(chicagoNMMAPS)
basis.pm <- crossbasis(chicagoNMMAPS$pm10, vartype="lin", lagtype="poly",
lagdegree=4, cen=FALSE, maxlag=15)
basis.temp <- crossbasis(chicagoNMMAPS$temp, vardf=5, lagtype="strata",
lagknots=1, cenvalue=21, maxlag=3)
summary(basis.pm)
summary(basis.temp)
model <- glm(death ~ basis.pm + basis.temp, family=quasipoisson(), chicagoNMMAPS)
pred.pm <- crosspred(basis.pm, model, at=0:20)

crossplot(pred.pm, "slices", var=10,
title="Effect of a 10-unit increase in PM10 along lags")
# overall effect for a 10-unit increase in PM over 15 days of lag, with CI
pred.pm$allRRfit["10"]
cbind(pred.pm$allRRlow, pred.pm$allRRhigh) ["10",]
crossplot(pred.pm, "overall", ylim=c(0.99,1.04), label="PM10", ci="lines",
title="Overall effect of PM10 over 15 days of lag")

### See the vignette 'dlnmOverview' for a detailed explanation of this example
```

---

 crossplot

*Plot predicted effects for a DLNM*


---

## Description

Plot several graphs (3d, slices and overall effects) of predicted effects from distributed lag non-linear models (DLNM).

## Usage

```
crossplot(crosspred, type="3d", cumul=FALSE, ci="area",
var=NULL, lag=NULL, ylim=NULL, title=NULL, label="var")
```

## Arguments

crosspred	an object of class "crosspred".
type	type of plot. See Details below.
cumul	logical. If TRUE, cumulative effects along lags are reported. Used only if type="slices". See Details
ci	type of confidence intervals representation, one of "area", "bars" or "lines".
var, lag	vectors of predictor values and lags for which specific effects must be plotted. Used only if type="slices".
ylim	numeric vectors of length 2, giving the coordinates ranges for the response axis (y-axis for type equal to "overall" or "slices", z-axis for "3d").
title	the main title on top of the plot.
label	label for predictor to be inserted in the plot.

## Details

Different plots can be obtained choosing the following values for the argument `type`:

"3d": a 3-D plot generated by calling the function [persp](#).

"contour": a contour/level plot generated by calling the function [filled.contour](#).

"overall": a plot of the overall effects (summed up all the single lag contributions).

"slices": a multiple plot of effects at specific values of predictor or lags, chosen by `var` and `lag`, respectively. Up to 4 plots for each dimension are allowed. Cumulative effects along lags are reported if `cumul=TRUE`: in this case, the same option must have been set to obtain the prediction saved in the `crosspred` object (see [crosspred](#)).

For a detailed overview of the options, see:

```
vignette("dlnmOverview")
```

**Note**

All the effects are reported versus a reference value. For continuous functions, this is specified by the centering point defined in the object `crossbasis` (see [crossbasis](#)). Exponentiated effects are automatically returned if `model` has `link` equal to `log` or `logit`.

The values in `var` must match those specified in the object `crosspred` (see [crosspred](#)), while the values in `lag` must be included in the lag period specified by `crossbasis`.

This function creates plots with default settings (i.e. perspective in 3-D plot, colours etc.). Refer to the original estimates stored in the `crosspred` object in order to personalize the output with generic plot commands.

**Author(s)**

Antonio Gasparri, <[antonio.gasparrini@lshtm.ac.uk](mailto:antonio.gasparrini@lshtm.ac.uk)>

**References**

Armstrong, B. Models for the relationship between ambient temperature and daily mortality. *Epidemiology*. 2006, **17**(6):624-31.

**See Also**

[crossbasis](#), [crosspred](#)

**Examples**

```
# Example 3. See crossbasis and crosspred for other examples

### DLNM for the effect of temperature on mortality up to 30 days of lag
### space of predictor: 5df quadratic spline for temperature
### space of predictor: linear effect for PM10
### lag function: 5df natural cubic spline for temperature
### lag function: single strata at lag 0-1 for PM10

data(chicagoNMMAPS)
basis.pm <- crossbasis(chicagoNMMAPS$pm10, vartype="lin", lagtype="strata",
  cen=FALSE, maxlag=1)
basis.temp <- crossbasis(chicagoNMMAPS$temp, vartype="bs", vardf=5, vardegree=2,
  lagdf=5, cenvalue=21, maxlag=30)
summary(basis.pm)
summary(basis.temp)
model <- glm(death ~ basis.pm + basis.temp, family=quasipoisson(), chicagoNMMAPS)
pred.temp <- crosspred(basis.temp, model, at=-26:33, cumul=TRUE)

crossplot(pred.temp, label="Temperature",
  title="3D graph of temperature effect")
crossplot(pred.temp, "contour", label="Temperature",
  title="Contour graph of temperature effect")
crossplot(pred.temp, "overall", label="Temperature",
  title="Overall effect of temperature over 30 days of lag")
crossplot(pred.temp, "slices", var=c(-20,0,27,33),
```

```
lag=c(0,5,15,28), label="Temperature")
crossplot(pred.temp, "slices", cum=TRUE, ci="bars", var=-20,
label="Temperature", title="Cumulative effect for -20C along lags")

### See the vignette 'dlnmOverview' for a detailed explanation of this example
```

---

crosspred

*Generate predicted effects for a DLNM*

---

## Description

Generate predicted effects from a distributed lag non-linear model (DLNM) for a set of values of the original predictor. It returns specific effects for each combination of values and lags, plus overall and (optionally) cumulative effects (summed up along lags).

## Usage

```
crosspred(crossbasis, model, at=NULL,
from=NULL, to=NULL, by=NULL, cumul=FALSE)
```

## Arguments

<code>crossbasis</code>	an object of class "crossbasis".
<code>model</code>	a model object for which the prediction is desired.
<code>at</code>	vector of values used for prediction.
<code>from, to</code>	range of values used for prediction.
<code>by</code>	increment of the sequence.
<code>cumul</code>	logical. If TRUE, cumulative effects are predicted. See details.

## Details

The object `crossbasis` must be the same containing the cross-basis matrix included in `model`, including its attributes and class. The set of values for which the effects must be computed can be specified by `at` or alternatively by `from/to/by`. If specified by `at`, the values are automatically ordered and made unique. By default, `from` and `to` correspond to the range of the original vector of observation stored in the `crossbasis` object (see [crossbasis](#)). If `by` is not provided, 30 equally spaced values are returned.

Matrices with cumulative effects summed upon lags for each values used for prediction are included if `cumul=TRUE`. For a long lag series (i.e. 1000 lags) the routine can be slow. These matrices are required by [crossplot](#) to graph the cumulative effects along lags.

For a detailed overview of the options, see:

```
vignette("dlnmOverview")
```

**Value**

A list object of class "crosspred" with the following components:

<code>predvar</code>	vector of observations used for prediction.
<code>maxlag</code>	a positive value defining the maximum lag.
<code>coef, vcov</code>	related coefficients and variance-covariance matrix from <code>model</code> .
<code>matfit, matse</code>	matrices of effects and related standard errors for each value of <code>predvar</code> and lag.
<code>allfit, allse</code>	vectors of total effects and related standard errors for each value of <code>predvar</code> .
<code>cumfit, cumse</code>	matrices of cumulative effects (along lags) and related standard errors for each value of <code>predvar</code> and lag. Computed if <code>cumul=TRUE</code> .
<code>matRRfit</code>	exponentiated effects from <code>matfit</code> .
<code>matRRlow, matRRhigh</code>	matrices with low and high 95% confidence intervals for <code>matRRfit</code> .
<code>allRRfit</code>	exponentiated total effects from <code>allfit</code> .
<code>cumRRfit</code>	exponentiated effects from <code>cumfit</code> . Computed if <code>cumul=TRUE</code> .
<code>cumRRlow, cumRRhigh</code>	matrices with low and high 95% confidence intervals for <code>cumRRfit</code> . Computed if <code>cumul=TRUE</code> .
<code>model.class</code>	class of the model command used for estimation.
<code>model.link</code>	a specification for the model link function.

**Warnings**

The name of the object `crossbasis` will be used by to extract the related estimated parameters from `model`. This name must not match the names of other predictors in the model formula. In addition, if more than one variable is transformed by cross-basis functions in the same model, different names must be specified.

**Note**

All the effects are reported versus a reference value corresponding to the centering point for continuous functions or to the default values for the other options (see `crossbasis`). Exponentiated effects are included if `model.link` is equal to `log` or `logit`.

**Author(s)**

Antonio Gasparri, <antonio.gasparrini@lshtm.ac.uk>

**References**

Armstrong, B. Models for the relationship between ambient temperature and daily mortality. *Epidemiology*. 2006, **17**(6):624-31.

**See Also**

[crossbasis](#), [crossplot](#)

**Examples**

```
# Example 2. See crossbasis and crossplot for other examples

### DLM with threshold for the effect of O3 on mortality up to 5 days of lag
### space of predictor: linear effect above 40.3 microgr/m3 for O3
### space of predictor: linear effects below 10C and above 25C for temperature
### lag function: integer lag parameterization (unconstrained) for O3
### lag function: strata intervals at lag 0-1, 2-6 and 7-15 for temperature

data(chicagoNMMAPS)
basis.o3 <- crossbasis(chicagoNMMAPS$o3, vartype="hthr", varknots=40.3,
lagtype="integer", maxlag=5)
basis.temp <- crossbasis(chicagoNMMAPS$temp, vartype="dthr", varknots=c(10,25),
lagtype="strata", lagknots=c(2,7), maxlag=15)
summary(basis.o3)
summary(basis.temp)
model <- glm(death ~ basis.o3 + basis.temp, family=quasipoisson(), chicagoNMMAPS)
pred.o3 <- crosspred(basis.o3, model, at=c(0:65,40.3,50.3))

crossplot(pred.o3, "slices", var=50.3,
title="Effect of a 10-unit increase in ozone along lags")
# overall effect for a 10-unit increase in ozone over 15 days of lag, with CI
pred.o3$allRRfit["50.3"]
cbind(pred.o3$allRRlow, pred.o3$allRRhigh)["50.3",]
crossplot(pred.o3, label="Ozone", title="3D graph of ozone effect")
crossplot(pred.o3, "contour", label="Ozone",
title="Contour graph of ozone effect")

### alternatively, a piecewise linear effect above 35 microgr/m3 for o3,
### with an additional change in slope at 50 microgr/m3
### same lag function as before

basis.o3 <- crossbasis(chicagoNMMAPS$o3, vartype="hthr", varknots=c(35,50),
lagtype="integer", maxlag=5)
summary(basis.o3)
model <- update(model)
pred.o3 <- crosspred(basis.o3, model, at=c(0:65))

crossplot(pred.o3,"overall",label="Ozone",
title="Overall effect of ozone over 5 days of lag")

### See the vignette 'dlnmOverview' for a detailed explanation of this example
```

# Index

- \*Topic **datasets**
  - chicagoNMMAPS, 3
- \*Topic **hplot**
  - crossplot, 8
- \*Topic **package**
  - dlnm-package, 2
- \*Topic **smooth**
  - crossbasis, 4
  - crossplot, 8
  - crosspred, 10
- \*Topic **ts**
  - crossbasis, 4
  - crossplot, 8
  - crosspred, 10

bs, 5

chicagoNMMAPS, 3

crossbasis, 2, 4, 9–12

crossplot, 2, 7, 8, 10, 12

crosspred, 2, 6–9, 10

dlnm (*dlnm-package*), 2

dlnm-package, 2

filled.contour, 8

glm, 2

mkbasis, 4

mklagbasis, 4

ns, 5

persp, 8

summary.crossbasis, 6

summary.crossbasis (*crossbasis*), 4