

# Package ‘crosslag’

May 8, 2026

**Type** Package

**Title** Perform Linear or Nonlinear Cross Lag Analysis

**Version** 0.1.0

**Maintainer** Yaxin Li <LYX010308@163.com>

**Description** Linear or nonlinear cross-lagged panel model can be built from input data. Users can choose the appropriate method from three methods for constructing nonlinear cross lagged models. These three methods include polynomial regression, generalized additive model and generalized linear mixed model. In addition, a function for determining linear relationships is provided.

Relevant knowledge of cross lagged models can be learned through the paper by Fredrik Falkenström (2024) <[doi:10.1016/j.cpr.2024.102435](https://doi.org/10.1016/j.cpr.2024.102435)> and the paper by A Gasparini (2010) <[doi:10.1002/sim.3940](https://doi.org/10.1002/sim.3940)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.1

**Depends** R (>= 4.3.0)

**Imports** gamm4 (>= 0.2.6), ggplot2 (>= 3.5.0), lavaan (>= 0.6.17), mgcv (>= 1.9.1), rms (>= 6.8.0), ggpubr (>= 0.6.0), stats (>= 4.3.2), utils (>= 3.1.0)

**NeedsCompilation** no

**Author** Yaxin Li [aut, cre]

**Repository** CRAN

**Date/Publication** 2024-05-17 10:10:03 UTC

## Contents

adjust_target . . . . .	2
clpm_gamm4_c . . . . .	3
clpm_gamm4_r . . . . .	4
clpm_gam_c . . . . .	5

clpm_gam_r . . . . .	6
clpm_poly_c . . . . .	6
clpm_poly_r . . . . .	7
crlog_line . . . . .	8
judge_line . . . . .	9
test_data1 . . . . .	10
test_data2 . . . . .	11

<b>Index</b>	<b>12</b>
--------------	-----------

---

adjust_target	<i>Title Generate covariates' residuals</i>
---------------	---

---

## Description

Title Generate covariates' residuals

## Usage

```
adjust_target(X, y1, y2 = NULL, cor_vars)
```

## Arguments

X	A dataset containing y1, y2, and covariates. The row name is the name of x,y and covariate
y1	If cross lagged analysis is used between x and y, 'y1' is the name of x.
y2	y2 can be NULL. If cross lagged analysis is used between x and y, 'y2' is the name of y. 'y1' and 'y2' come from the same time point.
cor_vars	the name of covariate

## Value

a list containing the covariate-adjusted value of y1 and y2 (If y2 is not NULL)

## Examples

```
data(test_data1)
result <- adjust_target(X=test_data1,y1="ASI",y2=NULL,cor_vars=c("HDL_C","LDL_C"))
```

---

clpm\_gamm4\_c

*Title Cross-lag analysis based on generalized additive mixture models*


---

## Description

Title Cross-lag analysis based on generalized additive mixture models

## Usage

```
clpm_gamm4_c(xname, yname, data.x1, data.y1, data.xt, data.yt, z)
```

## Arguments

xname	If cross lagged analysis is used between x and y, 'xname' is the name of x
yname	If cross lagged analysis is used between x and y, 'yname' is the name of y
data.x1	A numeric variable.
data.y1	A numeric variable. 'data.x1' and 'data.y1' comes from the first time point
data.xt	A numeric variable.
data.yt	A numeric variable. 'data.xt' and 'data.yt' comes from the another time point 'data.x1', 'data.y1', 'data.xt' and 'data.yt' can be the data processed by the function 'adjust_target()'
z	In the generalized additive mixture model, a random intercept is specified and the random effects are grouped by z. 'z' and 'data.x1', 'data.y1', 'data.xt', 'data.yt' are all numeric variables that should ideally correspond to each other.

## Value

A dataframe containing the result of generalized additive mixture model (consists of a smoothing term, a linear term and a stochastic intercept).

## Examples

```
data(test_data1)
data(test_data2)
result <- clpm_gamm4_c("ASI", "PWRI", test_data1$ASI, test_data1$PWRI,
  data.xt = test_data2$ASI, data.yt = test_data2$PWRI, z=test_data1$time)
```

---

clpm_gamm4_r	<i>Title Cross-lag analysis based on generalized additive mixture models: autoregression</i>
--------------	--

---

## Description

Title Cross-lag analysis based on generalized additive mixture models: autoregression

## Usage

```
clpm_gamm4_r(xname, yname, data.x1, data.y1, data.xt, data.yt, z)
```

## Arguments

xname	If cross lagged analysis is used between x and y, 'xname' is the name of x
yname	If cross lagged analysis is used between x and y, 'yname' is the name of y
data.x1	A numeric variable.
data.y1	A numeric variable. 'data.x1' and 'data.y1' comes from the first time point
data.xt	A numeric variable.
data.yt	A numeric variable. 'data.xt' and 'data.yt' comes from the another time point 'data.x1', 'data.y1', 'data.xt' and 'data.yt' can be the data processed by the function 'adjust_target()'
z	In the generalized additive mixture model, a random intercept is specified and the random effects are grouped by z. 'z' and 'data.x1', 'data.y1', 'data.xt', 'data.yt' are all numeric variables that should ideally correspond to each other.

## Value

A dataframe containing the result of autoregression based on generalized additive mixture model: 'Xt~X1' 'Yt~Y1'.

## Examples

```
data(test_data1)
data(test_data2)
result <- clpm_gamm4_r("ASI", "PWRI", test_data1$ASI, test_data1$PWRI,
  data.xt = test_data2$ASI, data.yt = test_data2$PWRI, z=test_data1$time)
```

---

`clpm_gam_c`*Title Nonlinear Cross Lag Analysis*

---

**Description**

Title Nonlinear Cross Lag Analysis

**Usage**`clpm_gam_c(xname, yname, data.x1, data.y1, data.xt, data.yt)`**Arguments**

<code>xname</code>	If nonlinear cross lagged analysis is used between x and y, 'xname' is the name of x
<code>yname</code>	If nonlinear cross lagged analysis is used between x and y, 'yname' is the name of y
<code>data.x1</code>	A numeric variable.
<code>data.y1</code>	A numeric variable. 'data.x1' and 'data.y1' comes from the first time point
<code>data.xt</code>	A numeric variable.
<code>data.yt</code>	A numeric variable. 'data.xt' and 'data.yt' comes from the another time point

**Value**

A list containing two elements

`clpm_gam_result`

a dataframe containing the result of nonlinear cross lag analysis

`clpm_gam_plot`

a list of plot generated by 'ggplot' and 'ggarrange', you can use 'ggarrange(clpm\_gam\_plot[[2]])' to plot it

**Examples**

```
data(test_data1)
data(test_data2)
result <- clpm_gam_c(xname="ASI", yname = "PWRI",
                    data.x1 = test_data1$ASI, data.y1 = test_data1$PWRI,
                    data.xt = test_data2$ASI, data.yt = test_data2$PWRI)
```

---

 clpm\_gam\_r

*Title Nonlinear Cross Lag Analysis: autoregression*


---

**Description**

Title Nonlinear Cross Lag Analysis: autoregression

**Usage**

```
clpm_gam_r(xname, yname, data.x1, data.y1, data.xt, data.yt)
```

**Arguments**

xname	If cross lagged analysis is used between x and y, 'xname' is the name of x
yname	If cross lagged analysis is used between x and y, 'yname' is the name of y
data.x1	A numeric variable.
data.y1	A numeric variable. 'data.x1' and 'data.y1' comes from the first time point
data.xt	A numeric variable.
data.yt	A numeric variable. 'data.xt' and 'data.yt' comes from the another time point

**Value**

A dataframe containing the result of autoregression: 'Xt~X1' 'Yt~Y1'

**Examples**

```
data(test_data1)
data(test_data2)
clpm_gam_r(xname="ASI", yname = "PWRI", data.x1 = test_data1$ASI,
           data.y1 = test_data1$PWRI, data.xt = test_data2$ASI, data.yt = test_data2$PWRI)
```

---

 clpm\_poly\_c

*Title Nonlinear Cross Lag Analysis based on Polynomial linear regression*


---

**Description**

Title Nonlinear Cross Lag Analysis based on Polynomial linear regression

**Usage**

```
clpm_poly_c(xname, yname, data1, data2, adjust = FALSE, adjust_name = NULL)
```

**Arguments**

xname	If cross lagged analysis is used between x and y, 'xname' is the name of x
yname	If cross lagged analysis is used between x and y, 'yname' is the name of y
data1	A dataframe containing data of x and y at the same time point. If covariate adjustments are made (adjust=T), data of covariates should also be included. The input 'xname', 'yname' and 'adjust_name' is written in the same way among the 'data1' column names
data2	A dataframe containing data of x and y at the another time point. If covariate adjustments are made (adjust=T), data of covariates should also be included. The input 'xname', 'yname' and 'adjust_name' is written in the same way among the 'data2' column names
adjust	The default value is F. If you want to perform covariate adjustments on x and y based on adjust_target(), please use T.
adjust_name	The name of the covariate, corresponding to the data contained in data1 or data2. If there are multiple covariates, they need to be saved in a vector.

**Value**

A dataframe containing the result of Nonlinear Cross Lag Analysis after polynomial regression

**Examples**

```
data(test_data1)
data(test_data2)
# Not adjusting for covariates
result <- clpm_poly_c(xname = "PWRI", yname = "LDL_C",
                     data1 = test_data1, data2 = test_data2, adjust = FALSE)
# Adjust for covariates
result_ad <- clpm_poly_c(xname = "ASI", yname = "PWRI",
                        data1 = test_data1, data2 = test_data2,
                        adjust = TRUE, adjust_name=c("HDL_C", "LDL_C"))
```

---

clpm_poly_r	<i>Title Nonlinear Cross Lag Analysis based on Polynomial linear regression: autoregression</i>
-------------	---

---

**Description**

Title Nonlinear Cross Lag Analysis based on Polynomial linear regression: autoregression

**Usage**

```
clpm_poly_r(xname, yname, data1, data2, adjust = FALSE, adjust_name = NULL)
```

**Arguments**

xname	If cross lagged analysis is used between x and y, 'xname' is the name of x
yname	If cross lagged analysis is used between x and y, 'yname' is the name of y
data1	A dataframe containing data of x and y at the same time point. If covariate adjustments are made (adjust=T), data of covariates should also be included. The input 'xname', 'yname' and 'adjust_name' is written in the same way among the 'data1' column names
data2	A dataframe containing data of x and y at the another time point. If covariate adjustments are made (adjust=T), data of covariates should also be included. The input 'xname', 'yname' and 'adjust_name' is written in the same way among the 'data2' column names
adjust	The default value is F. If you want to perform covariate adjustments on x and y based on adjust_target(), please use T.
adjust_name	The name of the covariate, corresponding to the data contained in data1 or data2. If there are multiple covariates, they need to be saved in a vector.

**Value**

A dataframe containing the result of Nonlinear Cross Lag Analysis after polynomial regression:  
 $X_t \sim X_1, Y_t \sim Y_1$

**Examples**

```
data(test_data1)
data(test_data2)
# Not adjusting for covariates
result <- clpm_poly_r(xname = "ASI", yname = "PWRI",
  data1 = test_data1, data2 = test_data2, adjust = FALSE)
# Adjust for covariates
result_ad <- clpm_poly_r(xname = "ASI", yname = "PWRI",
  data1 = test_data1, data2 = test_data2,
  adjust = TRUE, adjust_name=c("HDL_C", "LDL_C"))
```

---

crlog\_line

*Title Linear Cross lagged analysis*


---

**Description**

Title Linear Cross lagged analysis

**Usage**

```
crlog_line(xname, yname, data.x1, data.y1, data.xt, data.yt)
```

**Arguments**

xname	If linear cross lagged analysis is used between x and y, 'xname' is the name of x
yname	If linear cross lagged analysis is used between x and y, 'yname' is the name of y
data.x1	A numeric variable.
data.y1	A numeric variable. 'data.x1' and 'data.y1' comes from the first time point
data.xt	A numeric variable.
data.yt	A numeric variable. 'data.xt' and 'data.yt' comes from the another time point

**Value**

a dataframe containing 'Relation','beta','se','z','pvalue','x','y','rmr' and 'cfi'. Relation: This is a string that represents the relationship between the variables. beta: This is the estimated coefficient of the model. p: This is the p-value of the model coefficient, used to test whether the coefficient is significantly different from 0. x: This is the input xname, representing the name of the x variable. y: This is the input yname, representing the name of the y variable. rmr: This is one of the model fit measures, representing the root mean square residual of the model. cfi: This is one of the model fit measures, representing the comparative fit index of the model.

**Examples**

```
data(test_data1)
data(test_data2)
result <- crlog_line(xname="ASI",yname = "PWRI",data.x1 = test_data1$ASI,
                    data.y1 = test_data1$PWRI,data.xt = test_data2$ASI,
                    data.yt = test_data2$PWRI)
```

---

judge_line	<i>Title Determine a linear relationship between x and y by Restricted Cubic Splines(RCS) and plot it</i>
------------	---

---

**Description**

Title Determine a linear relationship between x and y by Restricted Cubic Splines(RCS) and plot it

**Usage**

```
judge_line(xname, yname, Data, parm)
```

**Arguments**

xname	A character, the name of the x. If there are multiple covariates, they need to be saved in a vector
yname	A character, the name of the y. If there are multiple covariates, they need to be saved in a vector
Data	A dataframe containing data of x and y. The input 'xname' and 'yname' is written in the same way among the 'Data' column names
parm	the number of knots of RCS

**Value**

A list containing the information of RCS plots. You can use 'ggpubr::ggarrange()' to plot the list. The list is a list of ggplot objects. Each ggplot object is a plot that represents the relationship between one independent variable (from xname) and one dependent variable (from yname). These plots are based on a linear regression model that uses a restricted cubic spline to handle the independent variable. Each plot includes a line of predicted values (solid blue line) and a confidence interval for these predictions (transparent blue area). The title of each plot includes the p-value for non-linearity, as well as the names of the independent and dependent variables.

**Examples**

```
data(test_data1)

# Get RCS plots
result <- judge_line(xname="ASI",yname="HDL_C",Data = test_data1,param = 4)
## Plot an RCS curve graph
ggpubr::ggarrange(result[[1]])

# Simultaneously determining multiple linear relationships
results <- judge_line("ASI",c("HDL_C","LDL_C"),Data = test_data1,param = 4)
## Plot many RCS curve graphs
## Adjust 'nrow' and 'ncol' according to actual situation
ggpubr::ggarrange(plotlist = results, nrow = 1, ncol = 2)
```

---

test\_data1

*Data for the first time point used for testing*

---

**Description**

Data for the first time point used for testing

**Format**

A dataframe.

**Source**

Generated from case information of a hospital in Hebei province

**Examples**

```
data(test_data1)
```

---

test_data2	<i>Data for the second time point used for testing</i>
------------	--

---

**Description**

Data for the second time point used for testing

**Format**

A dataframe.

**Source**

Generated from case information of a hospital in Hebei province

**Examples**

```
data(test_data2)
```

# Index

`adjust_target`, 2

`clpm_gam_c`, 5

`clpm_gam_r`, 6

`clpm_gamm4_c`, 3

`clpm_gamm4_r`, 4

`clpm_poly_c`, 6

`clpm_poly_r`, 7

`crlog_line`, 8

`judge_line`, 9

`test_data1`, 10

`test_data2`, 11