

Package ‘dynCorr’

February 14, 2012

Version 0.1-1

Date 2009-01-09

Title Dynamic Correlation Package

Author Joel Dubin, Dandi Qiao, Hans-Georg Mueller.

Maintainer Joel Dubin <jdubin@uwaterloo.ca>

Depends lpridge

Description Computes dynamical correlation estimates and percentile bootstrap confidence intervals for pairs of longitudinal responses, including consideration of lags and derivatives.

License GPL-2

Repository CRAN

Date/Publication 2009-02-03 20:03:23

R topics documented:

bootstrapCI	2
dynamicCorrelation	4
dynCorrData	8
Index	9

bootstrapCI	<i>Bootstrap Confidence Interval</i>
-------------	--------------------------------------

Description

Computes percentile bootstrap (BS) confidence intervals for dynamical correlation for pairs of longitudinal responses, including consideration of lags and derivatives, following a local polynomial regression smoothing step.

Usage

```
bootstrapCI(dataFrame, depVar, indepVar, subjectVar, function.choice, width.range, width.place, bound)
```

Arguments

dataFrame	The data frame that contains the dependent variables/responses, the independent variable (often time), and the subject/individual identification; there should be one row entry for each combination of subject/individual and indepVar (often time).
depVar	Dependent variables/responses; at least two are necessary for purposes of calculating at least one dynamical correlation estimate of interest; there should be a unique column within depVar for each response.
indepVar	Independent variable, typically the discrete recorded time points at which the dependent variables were collected; note that this is the independent variable for purposes of curve creation leading into estimating the dynamical correlations between pairs of dependent variables; must be contained in a single column.
subjectVar	Column name of the individuals; there should be one row entry for each combination of subject/individual and indepVar.
function.choice	A vector of length 3 to indicate which derivatives desired for local polynomial curve smoothing; 1st entry is for 0th derivative (i.e., original function), 2nd entry is for 1st derivative, 3rd is for 2nd derivative; 1=yes, 0=no. e.g., c(1,0,1) would be specified if interest is in looking at derivative 0 and derivative 2, c(1,0,0) for looking at original function (0th derivative) only, etc.
width.range	Bandwidth for local polynomial regression curve smoothing for each dependent variable/response; it can be a list that specifies a distinct bandwidth two-element vector for each response, or a single two-element vector that is used for each of the responses — the program is currently set up to allow linearly increasing or decreasing bandwidths, specified by this two-element vector with the increase (or decrease) occurring from the first argument in width.place to its second argument; the lpepa function within the lpridge package is called, which uses Epanechnikov kernel weighting, and the specifications of bandwidth in width.range will be used there; the default bandwidth is the range of indepVar (usually time) divided by 4, i.e., a constant global bandwidth.

<code>width.place</code>	Endpoints for width change assuming a non-constant bandwidth requested — the program is currently set up to allow linearly increasing or decreasing bandwidths, specified in <code>width.range</code> , with the increase (or decrease) occurring from the first argument in <code>width.place</code> to its second argument; it can be a list that specifies different endpoints for each response, or a single vector of endpoints that is used for each of the responses; default is no endpoints specified which means constant global bandwidth throughout the range of <code>indepVar</code> .
<code>boundary.trunc</code>	Indicate the boundary of <code>indepVar</code> that should be truncated after smoothing; this may be done in case of concerns about estimating dynamical correlation at the boundaries of <code>indepVar</code> ; this is a two element vector, where the first argument is how much to truncate from the right of the minimum value of <code>indep.var</code> and the second argument is how much to truncate from the left of the maximum value of <code>indep var</code> , within an individual and specific response; default is no truncation, i.e., <code>c(0,0)</code> .
<code>byOrder</code>	A vector that specifies the order of the variables/responses and derivatives (if any) to be the leading variable in the calculations; this will have an effect on how <code>lag.input</code> is to be interpreted; default is to use the order as specified in the <code>depVar</code> argument.
<code>max.dynCorrLag</code>	A specified lag value at which you would like to obtain a percentile BS dynamical correlation interval estimate; only single lag values are allowed (due to computational considerations), and the lag value considered might be that which is output from a lag analysis using the <code>dynamicCorrelation</code> function (see Example 2 on the <code>dynamicCorrelation</code> help page).
<code>B</code>	The number of samples used for the bootstrap.
<code>percentile</code>	The percentile used to construct confidence intervals for dynamic correlations.
<code>by.deriv.only</code>	If TRUE, the inter-dynamical correlations between different derivatives are not computed, which can save computation time (e.g., when <code>function.choice=c(1,0,1)</code> is specified and <code>by.deriv.only=T</code> , then only dynamical correlations will be calculated within (and not across) the 0th and 2nd derivative estimates, respectively); default is TRUE.
<code>seed</code>	The seed used in generating the BS samples.

Details

This function will provide smooth estimates (curves/functions or their derivatives) of responses of interest, then generate dynamical correlation estimates for each pair of responses. Lags of interest can be specified, using the `lag.input` argument. For smoothing, the function uses local polynomial smoothing using Epanechnikov kernel weighting (by calling the function `lpepa` within the `lpridge` package). The default global bandwidth is generated by taking the range of `indepVar` (usually time) and dividing by 4. This, by default, will be a constant global bandwidth, but proper specification of the `width.range` and `width.place` arguments can allow for a more flexible bandwidth choice, including different specification for each response in `depVar`.

Whereas the `dynamicCorrelation` program, which produces only point estimates, is fast, the `bootstrapCI` program is slow in its current form, as quite a bit of processing is required for each bootstrap sample. Details of the two-stage bootstrap algorithm can be found in Dubin and Muller (2005). We will attempt to boost computing speed in future versions.

In addition, as pointed out in Dubin and Muller (2005), a downward shift of the two-stage bootstrap CI toward 0 is intentional, in order to account for error in the curve creation step. However, it should be noted that greater than expected downward shifts may occur for dynamical correlations that are high in magnitude. A future version of this function will attempt to make this bootstrap approach more robust in this situation.

Author(s)

Joel A. Dubin, Dandi Qiao, Hans-Georg Muller

See Also

[lpepa](#)

Examples

```
data(dynCorrData)

## Example 1: using default smoothing parameters, obtain bootstrap CI estimates
##           for all three pairs of responses, for original function only

examp1.bs <- bootstrapCI(dataFrame=dynCorrData,
                        depVar=c('resp1', 'resp2', 'resp3'),
                        indepVar='time',
                        subjectVar = 'subject',
                        function.choice = c(1,0,0),
                        B = 200, percentile=c(0.025, 0.975), seed = 5)

examp1.bs

## Example 2: using default smoothing parameters, obtain bootstrap CI estimates
##           for all three pairs of responses, for original function only, at -10 days lag,
##           at .01 and .99 percentiles

examp2.bs <- bootstrapCI(dataFrame=dynCorrData,
                        depVar=c('resp1', 'resp2', 'resp3'),
                        indepVar='time',
                        subjectVar = 'subject',
                        function.choice = c(1,0,0), max.dynCorrLag = -10,
                        B = 200, percentile=c(0.01, 0.99), seed = 7)

examp2.bs
```

dynamicCorrelation *Dynamic Correlation*

Description

Computes dynamical correlation estimates for pairs of longitudinal responses, including consideration of lags and derivatives, following a local polynomial regression smoothing step.

Usage

```
dynamicCorrelation(dataFrame, depVar, indepVar, subjectVar, function.choice, width.range, width.place)
```

Arguments

dataFrame	The data frame that contains the dependent variables/responses, the independent variable (often time), and the subject/individual identification; there should be one row entry for each combination of subject/individual and indepVar (often time)
depVar	Dependent variables/responses; at least two are necessary for purposes of calculating at least one dynamical correlation estimate of interest; there should be a unique column within depVar for each response
indepVar	Independent variable, typically the discrete recorded time points at which the dependent variables were collected; note that this is the independent variable for purposes of curve creation leading into estimating the dynamical correlations between pairs of dependent variables; must be contained in a single column
subjectVar	Column name of the individuals; there should be one row entry for each combination of subject/individual and indepVar
function.choice	A vector of length 3 to indicate which derivatives desired for local polynomial curve smoothing; 1st entry is for 0th derivative (i.e., original function), 2nd entry is for 1st derivative, 3rd is for 2nd derivative; 1=yes, 0=no. e.g., c(1,0,1) would be specified if interest is in looking at derivative 0 and derivative 2, c(1,0,0) for looking at original function (0th derivative) only, etc.
width.range	Bandwidth for local polynomial regression curve smoothing for each dependent variable/response; it can be a list that specifies a distinct bandwidth two-element vector for each response, or a single two-element vector that is used for each of the responses — the program is currently set up to allow linearly increasing or decreasing bandwidths, specified by this two-element vector with the increase (or decrease) occurring from the first argument in width.place to its second argument; the lpepa function within the lpridge package is called, which uses Epanechnikov kernel weighting, and the specifications of bandwidth in width.range will be used there; the default bandwidth is the range of indepVar (usually time) divided by 4, i.e., a constant global bandwidth
width.place	Endpoints for width change assuming a non-constant bandwidth requested — the program is currently set up to allow linearly increasing or decreasing bandwidths, specified in width.range, with the increase (or decrease) occurring from the first argument in width.place to its second argument; it can be a list that specifies different endpoints for each response, or a single vector of endpoints that is used for each of the responses; default is no endpoints specified which means constant global bandwidth throughout the range of indepVar
boundary.trunc	Indicate the boundary of indepVar that should be truncated after smoothing; this may be done in case of concerns about estimating dynamical correlation at the boundaries of indepVar; this is a two element vector, where the first argument is how much to truncate from the right of the minimum value of indep.var and the second argument is how much to truncate from the left of the maximum value

	of indep var, within an individual and specific response; default is no truncation, i.e., c(0,0)
lag.input	Values of lag to be considered; can be a vector of requested lags, for which a dynamical correlation estimate will be produced for each pair of responses at each lag; a positive value of lag.input means that the first entry for the dynamical correlation leads (occurs before) the second entry — conversely, a negative value means that the second entry for the correlation leads the first entry; default is no lag at all considered
byOrder	A vector that specifies the order of the variables/responses and derivatives (if any) to be the leading variable in the calculations; this will have an effect on how lag.input is to be interpreted; default is to use the order as specified in the depVar argument
by.deriv.only	If TRUE, the inter-dynamical correlations between different derivatives are not computed, which can save computation time (e.g., when function.choice=c(1,0,1) is specified and by.deriv.only=T, then only dynamical correlations will be calculated within (and not across) the 0th and 2nd derivative estimates, respectively); default is TRUE
full.lag.output	If TRUE, the dynamical correlation values for each pair of responses and requested derivative will be stored in vectors corresponding to different lag values, which enables plotting the correlations as a function of lag values; all the vectors will be stored in the returned attribute resultMatrix; default is FALSE

Details

This function will provide smooth estimates (curves/functions or their derivatives) of longitudinal responses of interest per individual, then generate dynamical correlation estimates for each pair of responses. Lags of interest can be specified, using the lag.input argument. For smoothing, the function uses local polynomial smoothing using Epanechnikov kernel weighting (by calling the function lpepa within the lpridge package). The default global bandwidth is generated by taking the range of indepVar (usually time) and dividing by 4. This, by default, will be a constant global bandwidth, but proper specification of the width.range and width.place arguments can allow for a more flexible bandwidth choice, including different specification for each response in depVar.

Details of the methodology for dynamical correlation can be found in Dubin and Muller (2005).

Author(s)

Joel A. Dubin, Dandi Qiao, Hans-Georg Muller

See Also

[lpepa](#)

Examples

```
data(dynCorrData)
```



```
width.range = c(40, 40) )  
examp4
```

dynCorrData

An example dataset for use in the example calls in the help files for the dynamicCorrelation and bootstrapCI functions

Description

This dataset contains three longitudinal responses for each subject, each measured every ten days over a varying length of follow-up depending upon subject (for each of 34 subjects). Each row contains all three responses for a given subject and a given follow-up time. The features are that the first and third responses have a moderate-sized positive dynamical correlation, while the first and second, and second and third, respectively, have lesser negative dynamical correlations. The dataset is called in the examples on the dynamicCorrelation and bootstrapCI help pages.

Usage

```
data(dynCorrData)
```

Format

A 648 by 5 data frame.

Index

*Topic **datasets**

dynCorrData, [8](#)

*Topic **multivariate**

bootstrapCI, [2](#)

dynamicCorrelation, [4](#)

bootstrapCI, [2](#)

dynamicCorrelation, [4](#)

dynCorrData, [8](#)

lpepa, [4](#), [6](#)