Package ‘dpa’

February 19, 2015

Version 1.0-3

Date 2010-10-15

Title Dynamic Path Approach

Author Emile Chappin <e.j.l.chappin@tudelft.nl>

Maintainer Emile Chappin <e.j.l.chappin@tudelft.nl>

Description A GUI or command-line operated data analysis tool, for analyzing time-dependent simulation data in which multiple instantaneous or time-lagged relations are assumed. This package uses Structural Equation Modeling (the sem package). It is aimed to deal with time-dependent data and estimate whether a causal diagram fits data from an (agent-based) simulation model.

Depends R (>= 2.9.0), tcltk, sem, igraph

License LGPL (>= 2.0)

URL http://www.chappin.com - http://www.r-project.org

Repository CRAN

Date/Publication 2012-10-29 08:58:35

NeedsCompilation no

SystemRequirements Tcl/Tk package BWidget.

R topics documented:

`dpa-package` *Dynamic Path Approach for Analyzing time-dependent simulation data*

Description

A GUI or command-line operated data analysis tool, for analyzing time-dependent simulation data in which multiple instantaneous or time-lagged relations are assumed. This package uses Structural Equation Modeling (the sem package). It is aimed to deal with time-dependent data and estimate whether a causal diagram fits data from an (agent-based) simulation model.

Details

Package:	dpa
Type:	Package
Version:	1.0-03
Date:	2010-10-15
License:	LGPL Version 2 or later.
LazyLoad:	yes

~~ An overview of how to use the package, including the most important ~~ ~~ functions ~~

Author(s)

Emile J.L. Chappin <E.J.L.Chappin@TuDelft.NL>

References

http://www.chappin.com

Examples

```
#Start the GUI with:
dpa.start()
```

`dpa.analysis.options` *Select the data and interval for analysis*

Description

The function provides an option to perform analysis on the whole data at a time or for every time step or for a specific time interval. The function also gives you an option of performing analysis for specific time steps and not for all time steps in the data.

Usage

```
dpa.analysis.options()
```

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.analysis.options()
```

dpa.analysis.performDPA *Perfrom DPA analysis*

Description

This function first converts the relations and data entered by the user in a format which is acceptable to SEM. Then the analysis is performed to give the results which can be saved by clicking on the saveDPA button. In addition, path diagram graphs are automatically created.

Usage

```
dpa.analysis.performDPA()
```

Details

Based on settings, the analysis is performed once for the full data set, using the prepared relations or performs the analysis for each time step.

Value

The basic resuls are assigned to sem.DPA. From that a number of statistics are drawn and put into sem.standardized (the standardized parameter estimates), sem.results.parameters (the parameters), sem.results.statistics (the stats regarding the fit and iterations), and sem.results.coefficients (the coefficients for all analyses if performed for each time step),

Author(s)

Emile J.L. CHappin

References

http://www.chappin.com

See Also

`dpa.data.loadDataFromDatabase`, `dpa.data.loadDataFromDisk`

Examples

```
#dpa.analysis.performDPA()
```

dpa.analysis.saveDPA *Save DPA result*

Description

This function allows you to save the DPA result in a text file to a destination folder on disk

Usage

```
dpa.analysis.saveDPA(dpaFileName=NULL)
```

Arguments

`dpaFileName`	file that it is saved to.

Details

If dpaFileName is not specified, file selection will open in the GUI. Otherwise, dpaFileName is used.

Value

Result is saved to the specified file.

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.analysis.saveDPA(dpaFileName=NULL)
```

`dpa.data.authenticationCancel` *Cancel the authetication window*

Description

You can return to the load data page by clicking cancel button on authentication page at any moment.

Usage

```
dpa.data.authenticationCancel()
```

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

`dpa.data.authenticationSubmit` *Authetication of information to establish connection*

Description

In order to access any database, user is required to provide some information such as server type, server name, database name, database table, user name and password. This function is called to verify these information. If the information is found to be correct then the connection to the database will be established and the mentioned data will be loaded. Noyte that this process is surely going to be a time consuming step if the data is large. That is why the user must consider saving the data to the disk in the next step to avoid loading from database next time.

Usage

```
dpa.data.authenticationSubmit()
```

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

`dpa.data.checkData` *Check data*

Description

The data loaded needs to be checked for the missing rows and missing data in the existing rows as that may result to errors in analysis.Also the data need to be sorted before being put further for analysis.

Usage

```
dpa.data.checkData()
```

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.data.checkData()
```

`dpa.data.loadCancel` *cancel loading of data*

Description

It will allow you to cancel the screen which gives an option to select the data loading either from the disk or database.

Usage

```
dpa.data.loadCancel()
```

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

`dpa.data.loadData` *Load data*

Description

It generates a screen which asks you to select one of the two options available i.e. data file loading either from the disk or from the database.

Usage

```
dpa.data.loadData()
```

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.data.loadData()
```

`dpa.data.loadDataFromDatabase` *Load data from database*

Description

The function will open up an authentication window to ask for the information from an user which is required to be verified before connecting to a server.

Usage

```
dpa.data.loadDataFromDatabase()
```

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

`dpa.data.loadDataFromDisk` *Load data from database*

Description

It opens a folder browsing window through which file containing the data can be selected. If no file is selected then a message will be displayed with the text "No file was selected". If there is already some loaded data then a message box is displayed asking to confirm the over-writing of the data.

Usage

```
dpa.data.loadDataFromDisk(dataFileName=NULL)
```

Arguments

`dataFileName`	filename from which the data is loaded

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.data.loadDataFromDisk(dataFileName=NULL)
```

`dpa.data.saveDataToDisk` *Save data to disk*

Description

This function allows you to save the data loaded in the workspace to a destination folder on disk selected through a folder browsing window.

Usage

```
dpa.data.saveDataToDisk(dataFileName=NULL)
```

Arguments

`dataFileName`	filename to which the data is saved

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.saveDataToDisk()
```

`dpa.data.setWorkingDirectory` *Set working directory*

Description

It pops up a folder browsing window in the GUi through which an user can select a particular directory to work with.

Usage

```
dpa.data.setWorkingDirectory()
```

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.data.setWorkingDirectory()
```

`dpa.data.viewOrEditData` *View or Edit data*

Description

This function allows to edit the loaded file in R editor in order to just have a look at the data or to make some changes in the data. Please make a note that the change is applied only to the loaded data and not to the data saved in the disk or database.

Usage

```
dpa.data.viewOrEditData()
```

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.data.viewOrEditData()
```

`dpa.exit` *DPA Exit*

Description

Closes the DPA GUI

Usage

```
dpa.exit()
```

Author(s)

Emile J.L. CHappin

References

http://www.chappin.com

Examples

```
#dpa.exit()
```

`dpa.find.missingRow` *Find missing rows in data*

Description

This function will first sort the loaded data by tick column and job column. There are two cases. One is that the whole data in a particular job is missing or secondly some ticks of a particular job are missing. It will display message for both the cases.

Usage

```
dpa.find.missingRow(dataframe = NULL, tickColumn = NULL, jobColumn = NULL)
```

Arguments

`dataframe`	data that is searched
`tickColumn`	column that represents time
`jobColumn`	column that represents the job or run number

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.find.missingRow(dataframe = NULL, tickColumn = NULL, jobColumn = NULL)
```

`dpa.generate.lag` *Generate lagged parameters in the data*

Description

This function creates lags in the source column and adds the laaged parameter column to the original data.

Usage

```
dpa.generate.lag(dataframe = NULL, tickColumn = NULL, sourceColumn = NULL, minLag=1,maxLag=1)
```

Arguments

`dataframe`	The selected data file for which analysis is to be performed.
`tickColumn`	Time column in the data.
`sourceColumn`	The variable (column in the data) in which lag is to be created.
`minLag`	Minimum lag to be created in the source column.
`maxLag`	Maximum lag to be created in the source column.

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.generate.lag(dataframe = NULL, tickColumn = NULL, sourceColumn = NULL, minLag=1,maxLag=1)
```

`dpa.incrementValue` *DPA increment value*

Description

Increments the global value i

Usage

```
dpa.incrementValue(i)
```

Arguments

`i`	The global value that is incremented

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.incrementValue(i)
```

dpa.locate.missing *Locate missing data*

Description

Locates and reports on missing data

Usage

```
dpa.locate.missing(dataframe = NULL, tickColumn = NULL, jobColumn = NULL)
```

Arguments

`dataframe`	Dataset
`tickColumn`	Column that represents time
`jobColumn`	Column that represents run or job

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.locate.missing(dataframe = NULL, tickColumn = NULL, jobColumn = NULL)
```

dpa.relations.addRelations *Adding an entry made to the relations*

Description

Add relation function adds entries (made by the user in the dataframe of relations window) to the relations one at a time.The relations are added one row after the other. The final relation hence created is used further in the analysis and plotting part.

Usage

```
dpa.relations.addRelations(From_column = NULL, To_column = NULL, Lag_in = NULL, minLag = NULL, maxLag
```

Arguments

`From_column`	The character string which is the variable name from which the relation starts (From column).
`To_column`	The character string which is the variable name at which the relation ends (To column).
`Lag_in`	The character string which is the variable name for which lagged parameters are to be created.
`minLag`	Minimum lag which is to be created in Lag_in column.
`maxLag`	Maximum lag which is to be created in Lag_in column.
`Direction`	Specify whether direction is unidirectional or bidirectional.

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#addRelations("a","b","From",0,2,"UniDirectional")
```

`dpa.relations.editRelations`

Edit relations

Description

This function pops up a screen which is designed so as to allow the user to enter the relations. Please see details section to understand how to make an entry.

Usage

```
dpa.relations.editRelations()
```

Details

The variable from which direction arrow starts is to be kept in the From column whereas the other variable to which arrow points is to be placed in To column. The possible values of the entry box are automatically taken from the variables in the data loaded from which an user has to just make a selection. The third column Create lag for is for attaching lag to either of the two variables. A user is to choose from the two options either from or to. The fourth and fifth column of the screen provides the option to make an entry of minimum and maximum lags to be attached to the variable entered in third column before. For ex-if the minimum and maximum lag is entered as 1 and 3 respectively then three variables will be created as variable_L1, variable_L2 and variable_L3. It will become clearer with the example I am taking after giving an idea about the last column. The last column direction is to specify whether the relation is unidirectional or bidirectional i.e. the arrow is single headed or double headed. After making all the entries one needs to click on submit button to include it in the relations (Null in the start). On clicking submit button you can see your entries on the screen below the combo boxes. Now you can make another entry in the same way and click on submit to include it too.

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.relations.editRelations()
```

---

dpa.relations.loadRelations

Load the saved relations from disk

Description

This function opens a folder browsing through which the relations file can be selected from the disk.Relations file already saved before will be loaded in workspace. If no file is selected then a message will be displayed stating that no file was selected. If there is already some loaded relation then a message box is displayed asking to confirm the over-writing of the data.

Usage

```
dpa.relations.loadRelations(loadRelFileName=NULL)
```

Arguments

`loadRelFileName`	File from which the relations are loaded

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.relations.loadRelations(loadRelFileName=NULL)
```

`dpa.relations.saveRelations` *Save the relation*

Description

This function will allow the user to save the relation loaded in the workspace to a destination folder on disk selected through a browsing folder option.

Usage

```
dpa.relations.saveRelations(saveRelFileName=NULL)
```

Arguments

`saveRelFileName`	File to which the relations are saved

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.relations.saveRelations(saveRelFileName=NULL)
```

`dpa.results.generateCoefficientsPlots` *Generate the Coefficients plot*

Description

This function generates the plot of the strengths of the relations over time. This therefore only works if the analysis is performed for multiple time steps. Both a png and pdf are generated. This is the main result of the analysis.

Usage

```
dpa.results.generateCoefficientsPlots(filename=NULL,colors=NULL,indices=NULL,legend=NULL)
```

Arguments

`filename`	The name of the files
`colors`	A range of colors for the lines in the graph
`indices`	A selection of parameters can be entered. In this way, multiple graphs can be made easily if there are too many lines in the graph
`legend`	Where the legend needs to go

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.results.generateCoefficientsPlots(filename=NULL,colors=NULL,indices=NULL,legend=NULL)
```

`dpa.results.generateFitPlots` *Generate the Fit plot*

Description

This function generates the plot of the fit measures from the analysis over time. This therefore only works if the analysis is performed for multiple time steps. Both a png and pdf are generated. This is a main result of the analysis.

Usage

```
dpa.results.generateFitPlots(filename=NULL,colors=NULL,indices=NULL,legend=NULL)
```

Arguments

`filename`	The name of the files
`colors`	A range of colors for the lines in the graph
`indices`	A selection of fit measures can be entered. In this way, multiple graphs can be made easily if there are too many lines in the graph. The fit measures are: iterations, df, GFI, AGFI, RMSEA, SRMR, NFI, and NNFI
`legend`	Where the legend needs to go

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.results.generateFitPlots(filename=NULL,colors=NULL,indices=NULL,legend=NULL)
```

`dpa.results.setGraphDir` *Change results directory*

Description

The user is given an option to change the directory other than the working directory for saving the results.

Usage

```
dpa.results.setGraphDir(graphDir=NULL)
```

Arguments

`graphDir`	Folder to set

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.results.setGraphDir(graphDir=NULL)
```

`dpa.results.viewNodePlots` *Generate graphs in between the parameters of data*

Description

This function designs a screen asking an user for the x and y parameters between which the graph is to be plotted. Also there is an option to plot the graph for all the time steps or for a specific time step as entered. The graph is generated on clicking plot button and saved in the results directory.

Usage

```
dpa.results.viewNodePlots()
```

Details

It uses scatter plot function for generating the graphs. The graph is saved as .png file.

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

`dpa.results.viewRelationsPlots` *Generate the Relations plot*

Description

This function generates the plot of the relations showing every connection between parameters along with their strength. The stronger a relation is the wider will be a line connecting them. The parameter estimates value as got from the analysis result is also written at every connecting line after approximating it to 3 places after decimel.

Usage

```
dpa.results.viewRelationsPlots(tickNumber=NULL)
```

Arguments

`tickNumber`	if the analysis is performed per time step, that is denoted in the graph by specifying this parameter

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

`dpa.sort.data` *Sort data*

Description

Sorts the dataset according to run or job first and time second.

Usage

```
dpa.sort.data(dataframe = NULL, tickColumn = NULL, runColumn = NULL)
```

Arguments

`dataframe`	The dataset to be sorted
`tickColumn`	The column depicting time
`runColumn`	The column depicting job or run

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Examples

```
#dpa.sort.data(dataframe = NULL, tickColumn = NULL, runColumn = NULL)
```

`dpa.start` *DPA Start*

Description

Starts the DPA GUI and resets all the global variables

Usage

```
dpa.start()
```

Author(s)

Emile J.L. CHappin

References

http://www.chappin.com

Examples

```
#starts the GUI
dpa.start()
```

`i` *i*

Description

Container for the time tick

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

`parameters` *Parameters*

Description

Container for the parameters

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

`relations` *Relations*

Description

Container for the relations

Author(s)

Emile J.L. Chappin

References

http://www.chappin.com

Index