

Package ‘dynrautoVAR’

July 4, 2023

Title Automated VAR(1) in dynr

Version 0.9.1

Description Automatically fit VAR(1) models based on input data in 'dynr'.

URL <https://github.com/jeksterslab/dynrautoVAR>,
<https://jeksterslab.github.io/dynrautoVAR/>

BugReports <https://github.com/jeksterslab/dynrautoVAR/issues>

License GPL-3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

Depends R (>= 3.5.0)

Imports dynr, fclust, igraph

Suggests knitr, rmarkdown, testthat, rprojroot

RoxygenNote 7.2.3

NeedsCompilation no

Author Jonathan Park [aut, cre, cph]

Maintainer Jonathan Park <JPark@psu.edu>

R topics documented:

demo.dat	2
demo.dynr.var	2
dynr.cookdist	3
dynr.sub	4
dynr.var	5
Index	8

demo.dat

*Demo Data***Description**

Demo Data

Usage

demo.dat

Format

A data frame with 3000 rows and 7 columns.

X1 Observed variable 1.**X2** Observed variable 2.**X3** Observed variable 3.**X4** Observed variable 4.**X5** Observed variable 5.**id** ID variable for each individual (1 to 8).**time** Time variable (1 to 300).

demo.dynr.var

*Demo Output of dynr.var() using demo.dat***Description**

Demo Output of dynr.var() using demo.dat

Usage

demo.dynr.var

Format

A list of length ten. Each element of the list has the following elements

Betas Maybe we can put a description here.**ResidCov** Maybe we can put a description here.**PCC** Maybe we can put a description here.**PDC** Maybe we can put a description here.**Res** Maybe we can put a description here.

dynr.cookdist	<i>Cook's Distance in Dynr</i>
---------------	--------------------------------

Description

Cook's Distance in Dynr

Usage

```
dynr.cookdist(dataframe, id, time, ini.beta, dir = getwd())
```

Arguments

dataframe	a data frame object of data that contain a column of subject ID numbers (i.e., an ID variable), a column indicating subject-specific measurement occasions (i.e., a TIME variable), at least one column of observed values, and any number of covariates. The TIME variable should contain subject-specific sequences of (subsets of) consecutively equally spaced numbers (e.g, 1, 2, 3, ...). That is, the program assumes that the input data.frame is equally spaced with potential missingness. If the measurement occasions for a subject are a subset of an arithmetic sequence but are not consecutive, NAs will be inserted automatically to create an equally spaced data set before estimation. Missing values in the observed variables should be indicated by NA.
id	a character string of the name of the ID variable in the data.
time	a character string of the name of the TIME variable in the data.
ini.beta	the matrix of starting/fixed values for the transition matrix in the specified linear dynamic model.
dir	path for output files.

Author(s)

Jonathan Park

Maybe we can put a description here on how Cook's distance is calculated. Also Jonathan can you give me the data set you used in the example?

I made the following changes to the code.

1. I changed the name to `dynr.cookdist` to match the other functions in the package.
2. I changed the argument data to `dataframe` to match `dynr::dynr.data()`.
3. I changed the argument ID to `id` to match `dynr::dynr.data()`.
4. I set a default value for the argument `dir`.
5. I used `styler::style_pkg()` to style the code for added readability.

References

Add references to the Cook's distance applied to dynr here.

dynr.sub

*Subgrouping Algorithm***Description**

Subgrouping Algorithm

Usage

```

dynr.sub(
  input,
  alpha = 0.05,
  params.cook = NULL,
  params.var = c("Betas", "ResidCov", "PDC", "PCC", "Both"),
  method = c("hard", "fuzz"),
  k = 2,
  m = 2
)

```

Arguments

input	list of <code>dynr::dynr.cook()</code> objects or output from <code>dynr.var()</code> .
alpha	significance level for testing of transition matrix coefficients.
params.cook	parameters for clustering for <code>dynr::dynr.cook()</code> input.
params.var	parameters for clustering for <code>dynr.var()</code> input.
method	Subgrouping method.
k	number of clusters.
m	value of fuzzifier.

Details

I encapsulated dynrautoVAR into a temporary package so I can render the documentation and do some tests.

I made the following changes to the code.

1. I changed the name to `dynr.sub()` to match the other functions in the package.
2. I changed the argument inputs to `input`.
3. I removed the argument type and based the type on the class of the input.
4. I changed the argument `p.val` to `alpha`.
5. I changed the argument `params` to `params.cook` and `var.opt` to params.var`.`
6. I used `styler::style_pkg()` to style the code for added readability.

Author(s)

Jonathan Park

This function takes a list of outputs from `dynr::dynr.cook()` or output from `dynr.var()` and performs a subgrouping analysis. The output is a list of clusters.

References

Add references to the subgrouping algorithms here.

<code>dynr.var</code>	<i>Automatically Fit VAR(1) Models Based on Input Data</i>
-----------------------	--

Description

Automatically Fit VAR(1) Models Based on Input Data

Usage

```
dynr.var(
  dataframe,
  nv,
  id = NULL,
  time,
  dir = getwd(),
  alpha = 0.05,
  ini.mu = NULL,
  ini.cov = NULL,
  ini.beta = NULL,
  ini.loadings = NULL,
  use.mi = FALSE,
  aux = NULL
)
```

Arguments

<code>dataframe</code>	a data frame object of data that contain a column of subject ID numbers (i.e., an ID variable), a column indicating subject-specific measurement occasions (i.e., a TIME variable), at least one column of observed values, and any number of covariates. The TIME variable should contain subject-specific sequences of (subsets of) consecutively equally spaced numbers (e.g, 1, 2, 3, ...). That is, the program assumes that the input data.frame is equally spaced with potential missingness. If the measurement occasions for a subject are a subset of an arithmetic sequence but are not consecutive, NAs will be inserted automatically to create an equally spaced data set before estimation. Missing values in the observed variables should be indicated by NA.
<code>nv</code>	number of variables.

<code>id</code>	a character string of the name of the ID variable in the data.
<code>time</code>	a character string of the name of the TIME variable in the data.
<code>dir</code>	path for output files.
<code>alpha</code>	significance level for testing of transition matrix coefficients.
<code>ini.mu</code>	a vector of the starting or fixed values of the initial state vector.
<code>ini.cov</code>	a positive definite matrix of the starting or fixed values of the initial error covariance structure.
<code>ini.beta</code>	the matrix of starting/fixed values for the transition matrix in the specified linear dynamic model.
<code>ini.loadings</code>	matrix of starting or fixed values for factor loadings.
<code>use.mi</code>	if <code>use.mi = TRUE</code> , use <code>dynr::dynr.mi()</code> to address missing data in covariates.
<code>aux</code>	names of the auxiliary variables used in the imputation model if <code>use.mi = TRUE</code> .

Details

I encapsulated dynrautoVAR into a temporary package so I can render the documentation and do some tests.

I made the following changes to the code.

1. I changed the argument `data` to `dataframe` to match `dynr::dynr.data()`.
2. I changed the argument `ID` to `id` to match `dynr::dynr.data()`.
3. I changed the argument `p.val` to `alpha`.
4. I removed the default NULL value on arguments that require explicit values (`dataframe`, `nv`, and `time`).
5. I made all initial condition arguments begin with `ini..`
6. I set a default value for the argument `dir` to remove it from the argument handling section.
7. I cleaned up the argument handling section.
8. I changed `result$estimation.result` to `results$estimation.result`.
9. I gave the output a class of `dynrVar` for subsequent use of the output.
10. I used `styler::style_pkg()` to style the code for added readability.

Note that I mainly based the way I documented the arguments from functions in the `dynr` package used within this function. If you are amenable to the way I documented this function, I will proceed to documenting the rest.

Value

Returns a list of fitted VAR(1) models for each subject.

Author(s)

Jonathan Park

This function takes in a `data.frame`, fits individual models for each subject indicated by the ID variable and stores the fitted models as a list. Note that if `ini.*` arguments are not provided, initial values are set to null matrices or null vectors.

dynr.var

7

References

Add references to [dynr.var\(\)](#) here.

Index

* **data**

demo.dat, [2](#)

demo.dynr.var, [2](#)

demo.dat, [2](#)

demo.dynr.var, [2](#)

dynr.cookdist, [3](#)

dynr.sub, [4](#)

dynr.var, [5](#)

dynr.var(), [4](#), [5](#), [7](#)

dynr::dynr.cook(), [4](#), [5](#)

dynr::dynr.data(), [3](#), [6](#)

dynr::dynr.mi(), [6](#)

styler::style_pkg(), [3](#), [4](#), [6](#)