Package 'OLScurve'

August 29, 2016

Type Package

Title OLS growth curve trajectories

Version 0.2.0

Date 2014-02-20

Maintainer Phil Chalmers <rphilip.chalmers@gmail.com>

Description Provides tools for more easily organizing and
plotting individual ordinary least square (OLS) growth curve
trajectories.

Depends lattice

Suggests testthat, knitr, lavaan

ByteCompile yes

LazyLoad yes

LazyData yes

Repository CRAN

License GPL (>= 2)

URL https://github.com/philchalmers/OLScurve

Collate 'OLScurve.R' 'parplot.R' 'subjplot.R' 'OLScurve-package.R'

Author Phil Chalmers [cre, aut],
Carrie Smith [ctb],
Matthew Sigal [ctb]

NeedsCompilation no

Date/Publication 2014-02-20 18:27:03

R topics documented:

`OLScurve-package` *OLS growth curve trajectories*

Description

OLS growth curve trajectories

Details

Provides tools for organizing, calculating, and plotting ordinary least squares growth curve trajectories. These type of models are typically used to diagnose the effectiveness of a specified time functional form at the individual level rather than at the group level (where FIML estimation is required).

Author(s)

Phil Chalmers <`rphilip.chalmers@gmail.com`> and Carrie Smith <`carrie.elizabeth@gmail.com`> and Matthew Sigal <`matthewsigal@gmail.com`>

`gender` *Description of gender*

Description

A vector identifying whether the repeated measure row was Male or Female for the `nonlin.example` dataset.

Author(s)

Phil Chalmers <`rphilip.chalmers@gmail.com`>

`nonlin.example` *Description of nonlin.example data*

Description

An artificial dataset simulated by the parameters given in the vignette file for `OLScurve`.

Arguments

`gender` a vector identifying whether the repeated measure row was Male or Female

Author(s)

Phil Chalmers <`rphilip.chalmers@gmail.com`>

OLScurve *Ordinary least squares growth curve trajectories*

Description

The OLScurve provides a simple way of specifying ordinary least squares (OLS) growth curve models in R. Individual OLS trajectories are fit to each case and an OLScurve object is returned which can be passed to several graphical and summary function within the package.

Usage

```
OLScurve(formula, data,
  time = data.frame(time = 0:(ncol(data) - 1)), ...)

 ## S3 method for class 'OLScurve'
print(x, group = NULL, SE = TRUE,
   adjust = FALSE, digits = 3, ...)

 ## S3 method for class 'OLScurve'
plot(x, group = NULL, sep = FALSE,
   ...)
```

Arguments

formula	a formula specifying how the functional form of time should be coded. By default time is the only predictor but can be modified to, and any typical additive R formula may be used (e.g., powers, square roots, and exponentials)
data	a data frame in the wide (one subject per row) format containing only the time related variables. Can be of class matrix or data.frame
time	a data.frame object specifying the relative spacing between time points. The default is for equal spacing and this variable is name time.
x	an OLScurve object
group	a factor grouping variable used to partition the results
SE	logical; print a list containing the standard errors?
digits	number of digits to round
sep	logical; should the plots be separated?
...	additional arguments to be passed
adjust	logical; apply adjustment to make the variances unbiased? Only applicable for simple linear trajectories. Unadjusted valuse can be interpreted as upper bounds of the true variance parameters

Details

As Bollen and Curran (2006) note, there are a variety of advantages to using the case-by-case approach for estimating trajectory parameters. First of all, OLS estimation is intuitively appealing, making it a good pedagogical tool for introducing how to model trajectories, and illuminates many essential conditions and assumptions necessary for LCMs. Second, prediction of the parameters for individual trajectory estimates are calculated for each case in the sample, which can lead to several diagnostics by statistical and graphical means. Also, summary statistics can be computed for these estimates (which can also be graphically portrayed) and if need be these estimates can be analyzed further by other statistical frameworks.

Unfortunately there are also several limitation to OLS estimation for LCMs, namely: overall tests of fit are not readily available, the structure of the error variances must be unrealistically constrained to estimate a pooled standard error, the latent factors cannot be regressed without error on other exogenous or time-varying variables, and analytic significance tests are often not readily available (Bollen & Curran, 2006). However, OLS estimation may still be useful in the preliminary stages of latent curve modeling for (a) selecting appropriate functional forms of growth, (b) examining unconditional population homogeneity, (c) observing whether the relationship between growth factors are linear, and for (d) detecting influential outliers (Carrig et al., 2004).

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

References

Bollen, K. A. & Curran, P. J. (2006). *Latent Curve Models: A Structural Equation Perspective*. John Wiley & Sons.

Carrig, M. M., Wirth, R. J., & Curran, P. J. (2004). A SAS Macro for Estimating and Visualizing Individual Growth Curves. *Structural Equation Modeling, 11*, 132-149.

See Also

parplot, subjplot

Examples

```
## Not run:
##linear
data <- t(t(matrix(rnorm(1000),200)) + 1:5)
mod1 <- OLScurve(~ time, data = data)
mod1 #unadjusted variances
print(mod1, adjust = TRUE) #adjusted
plot(mod1)

##quadratic
data <- t(t(matrix(rnorm(1000),200)) + (0:4)^2)
mod2 <- OLScurve(~ time + I(time^2), data = data)
mod2
plot(mod2)
```

```
##sqrt
data <- t(t(matrix(rnorm(1000),200)) + 20*sqrt(5:1))
mod3 <- OLScurve(~ sqrt(time), data = data)
mod3
plot(mod3)

##exponential
data <- t(t(matrix(rnorm(1000,0,5),200)) + exp(0:4))
mod4 <- OLScurve(~ exp(time), data = data)
mod4
plot(mod4)

##combination
data <- t(t(matrix(rnorm(1000),200)) + 20*sqrt(1:5))
mod5 <- OLScurve(~ time + sqrt(time), data = data)
mod5
plot(mod5)

##piecewise (global linear trend with linear shift at time point 3)
data <- t(t(matrix(rnorm(1000),200)) + (0:4)^2)
time <- data.frame(time1 = c(0,1,2,3,4), time2 = c(0,0,0,1,2))
mod6 <- OLScurve(~ time1 + time2, data, time=time)
mod6
plot(mod6)

##two group analysis with linear trajectories
data1 <- t(t(matrix(rnorm(500),100)) + 1:5)
data2 <- t(t(matrix(rnorm(500),100)) + 9:5)
data <- rbind(data1,data2)
group <- c(rep('male',100),rep('female',100))

mod <- OLScurve(~ time, data)
print(mod,group)
plot(mod,group)

## End(Not run)
```

`parplot` *Plot distribution of parameters*

Description

A plotting function for displaying the distribution of the OLS parameter estimates.

Usage

```
parplot(object, ...)

## S3 method for class 'OLScurve'
parplot(object, type = "hist",
  group = NULL, breaks = NULL, prompt = TRUE, ...)
```

Arguments

`object`	an object of class `OLScurve`
`type`	type of plot to display; can be `'hist'`, `'boxplot'`, or `'splom'` for a histogram, boxplot, or scatter plot matrix
`group`	a `factor` grouping variable used to partition the results
`breaks`	number of breaks to be used in plotting the histogram
`prompt`	a logical variable indicating whether `devAskNewPage(ask=TRUE)` should be called
`...`	additional arguments to be passed

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

Examples

```
## Not run:
data <- t(t(matrix(rnorm(1000),200)) + 1:5)
group <- rep(c('Male', 'Female'), each=nrow(data)/2)
mod <- OLScurve(~ time, data = data)
parplot(mod)
parplot(mod, type = 'boxplot')
parplot(mod, type = 'splom')

parplot(mod, group=group)
parplot(mod, type='boxplot', group=group)
parplot(mod, type='splom', group=group)

## End(Not run)
```

---

subjplot *Plot individually estimated parameters*

Description

A plotting function for displaying the individuals trajectories and their modelled functional form. Useful for detecting aberrant individual trajectories.

Usage

```
subjplot(object, ...)

## S3 method for class 'OLScurve'
subjplot(object, layout = NULL,
  prompt = TRUE, ...)
```

Arguments

object	an object of class OLScurve
layout	a variable to be passed to xyplot to adjust the graphical layout
prompt	a logical variable indicating whether devAskNewPage(ask=TRUE) should be called
...	additional arguments to be passed

Author(s)

Phil Chalmers <rphilip.chalmers@gmail.com>

Examples

```
## Not run:
data <- t(t(matrix(rnorm(1000),200)) + 1:5)
mod <- OLScurve(~ time, data = data)
subjplot(mod)

##quadratic
data <- t(t(matrix(rnorm(1000),200)) + (0:4)^2)
mod2 <- OLScurve(~ time + I(time^2), data = data)
subjplot(mod2)



## End(Not run)
```

Index