Package: hysteresis (via r-universe)

August 28, 2024

Title Tools for Modeling Rate-Dependent Hysteretic Processes and Ellipses

Version 2.7.2

Date 2024-02-26

Author Spencer Maynes, Fan Yang, and Anne Parkhurst

Maintainer Spencer Maynes <smaynes89@gmail.com>

Depends car, msm, MASS

Suggests knitr, markdown

VignetteBuilder knitr

Description Fit, summarize and plot sinusoidal hysteretic processes using: two-step simple harmonic least squares, ellipse-specific non-linear least squares, the direct method, geometric least squares or linear least squares. See Yang, F and A. Parkhurst, ``Efficient Estimation of Elliptical Hysteresis with Application to the Characterization of Heat Stress" <DOI:10.1007/s13253-015-0213-6>.

License GPL (>= 2)

NeedsCompilation no

Date/Publication 2024-02-28 08:30:02 UTC

Repository https://spencerm89.r-universe.dev

RemoteUrl https://github.com/cran/hysteresis

RemoteRef HEAD

RemoteSha e2be83de621e115d1385be0ee880e98fe7faeab2

Contents

`hysteresis-package` *Modeling Rate-Dependent Hysteretic Processes*

Description

Fit, summarize and plot sinusoidal hysteretic processes using two step harmonic least squares. If the process is elliptical, other methods such as a geometric method, Halir and Flusser's direct specific least squares, ordinary least squares, and ellipse-specific non-linear least squares are also available.

Details

Package:	hysteresis
Type:	Package
Version:	2.7.2
Date:	2024-02-26
License:	gpl (>= 2)

Fits input and output variables x and y that form a hysteresis loop based on the generalized transcendental equation

$$x_t = b.x * cos(2pi * t/period + phase.angle) + cx + e_{x,t}$$

$$y_t = b.y*cos(2pi*t/period+phase.angle)^n+retention*sin(2pi*t/period+phase.angle)^m+cy+e_{y,t}$$

where

$$t = 0, ..., n.points - 1 if times =' equal'$$

The functions `mloop` and `floop` can be used to simulate, fit, and obtain derived parameter estimates (see `loop.parameters` or `ellipse.parameters`) along with delta method standard errors for hysteresis loops.. Additionally `summary.fittedloop` can be used to bootstrap results in order to produce less biased standard errors for derived parameters and obtain a model fit that is not dependent on the assumption of independent and normally distributed errors. If m=1 and n=1 then the hysteresis loop will form an ellipse which can be simulated with `mel`, fitted using 5 different available methods with `fel`, and bootstrapped using the function method `summary.ellipsefit`. If the upper and lower halves of the loop are structered differently, then the functions `mloop2r`, `floop2r` and `summary.loop2r` should be used. These functions fit a model with two values of retention for when the curve is above and below the split line. Studentized residuals are also available ((see `residuals.ellipsesummary`).

Author(s)

Spencer Maynes, Fan Yang, and Anne Parkhurst.

Maintainer: Spencer Maynes <smaynes89@gmail.com>

References

Yang, F and A. Parkhurst, "Efficient Estimation of Elliptical Hysteresis with Application to the Characterization of Heat Stress" DOI:10.1007/s13253-015-0213-6

See Also

Check out the vignette browseURL(system.file('doc/index.html',package='hysteresis'))
For simulating hysteresis loops, `mloop` and `mel`.
For fitting hysteresis loops, `floop` and `fel`.
For summarizing hysteresis loops, `summary.fittedloop` and `summary.ellipsefit`.
For bootstrapping ellipses, `summary.ellipsefit`. For fitting multiple hysteresis loops at once, can use `fel` and `floop` or `fel.repeated` and `floop.repeated` which can be easier to use for studies involving repeated measures.
Miscellaneous `plot.ellipsefit`, `plot.ellipsefitlist`, `plot.ellipsesummary`, `residuals.ellipsesummary`.

Examples

```
###Take a look at the vignette.
#browseURL(system.file('doc/index.html',package='hysteresis'))

### Simulate and fit a hysteresis loop with m=3 and n=5.
 loop1 <- mloop(sd.x=0.05,sd.y=0.05,n=5,m=3)
 model <- floop(loop1$x,loop1$y,n=5,m=3)
 model                #Gives estimate with delta standard errors
 model$Estimates      #Gives estimates
 model$Std.Errors     #Lists delta standard errors

### Plot hysteresis loop.
 plot(model,main="Simulated Hysteresis Loop n=5 m=3")

### Bootstrap estimates and standard errors (Seed is necessary if want to reproduce results)
 booted.loop <- floop(loop1$x,loop1$y,,n=5,m=3,boot=TRUE, seed=1523)
 booted.loop                 #Gives boot estimates, boot bias, boot SE and boot quartiles
 booted.loop$Boot.Estimates    #Gives boot estimates
 booted.loop$Boot.Std.Errors   #Gives boot standard errors
plot(booted.loop,main="Simulated Bootstrapped Loop n=5, m=3",putNumber=TRUE)


### Simulate and fit an ellipse.
ellipse1 <- mel(sd.x=0.2,sd.y=0.04)
ellipse1.fit <- fel(ellipse1$x,ellipse1$y)
ellipse1.fit            #Gives estimates with delta standard errors and 95%CI
ellipse1.fit$Estimates      #Gives all estimates
ellipse1.fit$Std.Errors     #Lists delta standard errors

### Plot ellipse
```

```
plot(ellipse1.fit,xlab="Input",ylab="Output",main="Simulated Ellipse")

### Bootstrap estimates and standard errors (Seed is necessary if want to reproduce results)
booted.ellipse <- fel(ellipse1$x,ellipse1$y,boot=TRUE, seed=123)
booted.ellipse              #Gives boot estimates, boot bias, boot SE and boot quartiles
booted.ellipse$Boot.Estimates  #Gives boot estimates
booted.ellipse$Boot.Std.Errors #Gives boot standard errors
```

`EllipseData` *Simulated Ellipse Data for 6 Ellipses with period=24.*

Description

Three subjects with two replications = 6 ellipses created by mel. All 6 ellipses are centered around the origin with a phase angle of pi/2 and differ by subject in terms of their retention (0.4,0.8,0.4) and b.x saturation point (0.6,0.6,1). Errors in both the input and the output are given a standard deviation of 0.1 for all ellipses. Used in the help page for `fel.repeated`.

Usage

```
data(EllipseData)
```

Format

A data frame with 144 observations on the following 4 variables.

`X` a numeric vector

`Y` a numeric vector

`subjects` subject, one of "A", "B", or "C".

`repeated` which ellipse for subject. Either 1 or 2.

Examples

```
## Data is created using the following code
set.seed(1)
ellip1 <- mel(method = 2, retention = 0.4, b.x = 0.6, b.y = 0.8, cx = 0, cy = 0,
    sd.x = 0.1, sd.y = 0.1, period = 24, n.points = 48, phase.angle = pi/2)
ellip2 <- mel(method = 2, retention = 0.8, b.x = 0.6, b.y = 0.8, cx = 0, cy = 0,
    sd.x = 0.1, sd.y = 0.1, period = 24, n.points = 48, phase.angle = pi/2)
ellip3 <- mel(method = 2, retention = 0.4, b.x = 1, b.y = 0.8, cx = 0, cy = 0,
    sd.x = 0.1, sd.y = 0.1, period = 24, n.points = 48, phase.angle = pi/2)
X <- c(ellip1$x, ellip2$x, ellip3$x)
Y <- c(ellip1$y, ellip2$y, ellip3$y)
subjects <- c(rep("A", length(ellip1$x)), rep("B", length(ellip2$x)), rep("C",
    length(ellip3$x)))
    repeated <- rep(c(1,2),each=24,times=3)

##Use data file to fit 6 ellipses.
```

```
data(EllipseData)
six.models <- fel.repeated(EllipseData$X, EllipseData$Y, method = "harmonic2",
subjects = EllipseData$subjects,repeated=EllipseData$repeated)
six.models

#Model fit for B-1
six.models$models["B",1]
par(mfrow=c(2,3))
plot(six.models,xlab="X input",ylab="Y output")
par(mfrow=c(1,1))
```

`fel` *Fitting Ellipses*

Description

Fit a sinusoidal hysteretic (elliptical) process between an input and an output.

Usage

```
fel(x, y=NULL, method = "harmonic2", period = NULL, subjects = NULL,
  times="unknown",subset = NULL,na.action= getOption("na.action"),
  control=nls.control(), boot=FALSE,...)
```

Arguments

`x`	input
`y`	output
`method`	the method to be used for fitting; one of either the default `method="harmonic2"`, `method="nls"`, `method="direct"`, `method="lm"` or `method="geometric"`.
`period`	an optional number that defines the length of the `period`.
`subjects`	an optional factor or list of factors, each of the same length as x. Use to identify several different ellipses to fit at once, in which case `fel` returns an object of class `ellipsefitlist` instead of `ellipsefit`. If `subjects` is a list of factors each combination of the factors must be present in the data or an error will be produced.
`times`	either a numeric vector of length `nrow(x)` or one of the two options "equal" or the default "unknown". If the times at which ellipse observations are taken are known, a numeric vector can be used to give those times. If not, predicted values are found by minimizing geometric distances from the fitted ellipse to the observations. If "equal", time points are assumed to be equally spaced in a counterclockwise fashion. Do not use the "harmonic2" method unless times are either known or are known to be equal. Bootstrapping results are also more accurate if correct times are used.
`subset`	an optional vector specifying a subset of observations to be used in the fitting process.

control	optional and only used if method="nls" or "geometric". See nls.control for method="nls".
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, the factory-fresh default is na.omit. Value na.exclude can be useful.
boot	logical, if TRUE results will be bootstrapped by summary.ellipsefit.
...	other othrt optional arguments passed to summary.ellipsefit if boot=TRUE.

Details

Where the response y is a sinusoidal process with an element of randomness that lags the controlling input x, which is also a stochastic sinusoidal process, an ellipse can be used to fit the relationship between x and y.

The values of parameters such as area, lag, retention, coercion, split angle and hysteresis.y are estimated from this ellipse. See loop.parameters.

The harmonic2 method is a two step harmonic least squares model using generalized transcendental equations presented by Lapshin (1995). Yang and Parkhurst provide the efficient estimates for parameters and as such "harmonic2" is used as the default. Direct specific least squares (method="direct") based on the work of Radim Halir and Jan Flusser is also available although work on delta method standard errors is still in progress. The geometric method is based on the work of Gander, Golub and Strebel and uses the results of an initial direct method to produce an ellipse that minimizes the sum of the squared geometric distances. Finally method="lm" and ellipse specific non-linear least squares (method="nls") are included as well.

If x and y contain more than 1 ellipse that needs to be fit, the argument subjects can be used to identify a period of data to fit separate ellipses.

Bootstrapped estimates for parameter values are provided with summary.ellipsefit. These bootstrapped estimates are generally less biased than those provided by fel in isolation.

Value

fel returns an object of class ellipsefit or ellipsefitlist.

call	the function call.
fit	information dependent on the fitting method used.
method	the method used.
x	the input x used.
y	the output y used.
pred.x	the fitted values for x.
pred.y	the fitted values for y.
period.time	a vector that contains times converted to radians for observations, either estimated after the ellipse has been fitted or given beforehand by times.
fit.statistics	rudimentary measures, based on the "harmonic2" method, include the Multivariate Final Prediction Error (MFPE) and the AIC for both the output alone and the two variables in combination. Although degree of freedom adjustments are made for other methods, measures of fit require further study.

values	a named vector of parameter estimates. See loop.parameters, same as Estimates here.
Estimates	a named vector of parameter estimates. See loop.parameters, same as values.
Std.Errors	Delta standard errors produced by the delta method.
residuals	algebraic residuals from the model. The function residuals.ellipsefit can produce other types of residuals from an ellipsefit object.

if boot==TRUE fel returns an object of class ellipsesummary by making a call to summary.ellipsefit. See summary.ellipsefit.

For bootstrapping

Boot.Estimates	bootstrapped estimates.
Boot.Std.Errors	bootstrap standard errors.

If multiple ellipses are fit simultaneously there will be three arguments to the response, models which will contain the separate model fits for each ellipse, Estimates which will have all of the parameter estimates in matrix form, and Std.Errors which will have all of the delta method standard errors in matrix form. See fel.repeated.

Author(s)

Spencer Maynes, Fan Yang, and Anne Parkhurst.

References

Yang, F and A. Parkhurst, "Efficient Estimation of Elliptical Hysteresis with Application to the Characterization of Heat Stress" DOI:10.1007/s13253-015-0213-6

See Also

plot.ellipsefit for plotting and summary.ellipsefit for summarizing and bootstrapping an ellipsefit object. Also residuals.ellipsefit.

Examples

```
### Simulate and fit a Single ellipse.
Sellipse <- mel(method=2,sd.x=0.2,sd.y=0.04)
Sellipse.fit <- fel(Sellipse$x,Sellipse$y)
Sellipse.fit  #Gives estimates, delta standard errors and 95% CI
Sellipse.fit$Estimates

### Bootstrap estimates and standard errors (Seed is necessary if want to reproduce results)
booted.Sellipse <- fel(Sellipse$x,Sellipse$y,boot=TRUE, seed=123)
booted.Sellipse   #Gives boot estimates, boot bias, boot SE and boot quartiles
plot(booted.Sellipse,main="Simulated Bootstrap Ellipse Loop",xlab="X Input",
ylab="Y Output",values="ellipse.all")
```

`fel.repeated` *Methods for easily fitting multiple ellipses from repeated measures designs.*

Description

Fit a sinusoidal hysteretic process between an input and an output variable across multiple loops separated by subjects and repeated.

Usage

```
fel.repeated(x,y=NULL,subjects=NULL,repeated=NULL,subjects.in="all",repeated.in="all",...)
## S3 method for class 'ellipsefitlist'
summary(object,N=1000,boot=TRUE,seed=NULL,...)
```

Arguments

`x`	numeric input vector.
`y`	numeric output vector.
`subjects`	factor of the same length as x that represents experimental units.
`repeated`	factor of the same length as x that represents the repeated measure.
`subjects.in`	a vector of characters, the levels of `subjects` to be included. Default is "all".
`repeated.in`	a vector of characters, the levels of `repeated` to be included. Default is "all".
`object`	an `ellipsefitlist` object.
`N`	Number of bootstrap replicates.
`boot`	whether to use bootstrapping to obtain standard errors and less biased parameter estimates.
`seed`	for generating random numbers. See `summary.fittedloop`.
`...`	extra arguments to either `fel` or `summary.ellipsefitlist`.

Details

Fits multiple ellipses with one call, separated by the factors `subjects` and `repeated`. The arguments `subjects.in` and `repeated.in` are used to select subsets of the factors `subjects` and `repeated`.

Value

`fel.repeated` returns an object of class `ellipsefitlist`.

`models`	Separate model fits for each ellipse, see `fel`.
`Estimates`	Parameter estimates for all ellipses in matrix form.

`Std.Errors` Delta standard errors for all ellipses in matrix form.

When `boot=TRUE` `fel.repeated` returns an object of class `ellipsesummarylist` which consists of

`models` a vector of separate model summaries for each ellipse, see `summary.ellipsefit`.

`values` Bootstrapped parameter estimates, standard errors, quantiles, and more for each ellipse.

`Boot.Estimates` Bootstrapped parameter estimates with reduced bias.

`Boot.Std.Errors` Standard errors provided by bootstrapping.

Author(s)

Spencer Maynes, Fan Yang, and Anne Parkhurst.

References

Yang, F. and A. Parkhurst, Efficient Estimation of Elliptical Hysteresis (submitted)

See Also

`fel` for a more general way to fit multiple ellipses, or for fitting just one ellipse. `plot.ellipsefit` for plotting and `summary.ellipsefit` for summarizing and bootstrapping an `ellipsefitlist` object. Also `residuals.ellipsefitlist`.

Examples

```
## Select 2 subjects with 2 replications and fit 4 ellipses
data(EllipseData)
emodels.rep <- fel.repeated(EllipseData$X, EllipseData$Y, method = "harmonic2",
subjects = EllipseData$subjects,subjects.in=c("A","C"),
repeated=EllipseData$repeated)
emodels.rep               #Gives estimates and delta standard errors
emodels.rep$Estimates     #List estimates only
emodels.rep$Std.Errors    #List delta standard errors
par(mfrow=c(2,2))
plot(emodels.rep, main="Repeated Ellipses",xlab="X",ylab="Y")
par(mfrow=c(1,1))

### Bootstrap estimates and standard errors (Seed is necessary if want to reproduce results)
boot.rep.ellipse<-fel.repeated(EllipseData$X,EllipseData$Y,method = "harmonic2",
subjects = EllipseData$subjects,subjects.in=c("A","C"),
repeated=EllipseData$repeated,boot=TRUE,seed=123)
boot.rep.ellipse  #Gives boot estimates, boot bias, boot SE and boot quartiles
par(mfrow=c(2,2))
plot(boot.rep.ellipse, main="Repeated Ellipses",xlab="X",ylab="Y",values="ellipse")
par(mfrow=c(1,1))

##Can write results to a file. First set your directory from the file tab.
#Change file path in command below to coincide with where you want to store data files
```

```
#setwd("C:/Users.............")
#write.table(boot.rep.ellipse$Boot.Estimates,"Ellipes.eg.repbootvalues.txt")
#test.fel=read.table("Ellipes.eg.repbootvalues.txt",header=TRUE)
#head(test.fel)
```

`floop` *Fit a Hysteresis Loop*

Description

Fits a hysteresis loop given values of n and m chosen by the user. `floop2r` fits an asymetric loop with different values for retention above and below the split line.

Usage

```
floop(x,y=NULL,n=1,m=1,times="equal",period=NULL,
subjects=NULL, subset=NULL,na.action=getOption("na.action"),
extended.classical=FALSE,boot=FALSE,method="harmonic2",
...)
floop2r(x,y=NULL,n=1,m=1,times="equal",period=NULL,
subjects=NULL, subset=NULL,na.action=getOption("na.action"),
extended.classical=FALSE,boot=FALSE,method="harmonic2",
...)
```

Arguments

`x`	numeric input vector.
`y`	numeric output vector.
`n`	a positive integer. Shape parameter regulating the central "plateau" of the hysteresis loop. Default is 1, which makes loop an ellipse when m is also equal to 1. See details.
`m`	an odd positive number. Bulging parameter of the hysteresis loop. Default is 1, which makes loop an ellipse when n is also equal to 1. In this case floop will automatically make a call to `fel`. See details.
`period`	length of time required to make a full loop. Reciprocal of frequency, and if times = "equal", the number of points needed to make a full loop.
`subjects`	an optional factor or list of factors, each of the same length as x. Use to identify a list of different loops to be fit from one set of data, in which case `floop` returns an object of class `fittedlooplist` instead of `fittedloop`.
`times`	either a numeric vector of length nrow(x) or the default "equal". If the times at which loop observations are taken are known, a numeric vector can be used to give those times. If the default "equal" is used instead, time points are assumed to be equally spaced in a counterclockwise fashion.
`subset`	an optional vector specifying a subset of observations to be used in the fitting process.

`na.action` a function which indicates what should happen when the data contain NAs. The default is set by the `na.action` setting of `options`, the factory-fresh default is `na.omit`. Value `na.exclude` can be useful.

`extended.classical`
logical. If true, uses

$$y_t = sign(cos(2*pi*t/period))*b.y*abs(cos(2*pi*t/period))^n+retention*sin(2*pi*t/period)^m+c$$

instead of

$$y_t = b.y*cos(2*pi*t/period)^n+retention*sin(2*pi*t/period)^m+cy+e_{y,t}$$

Allows the user to fit sinusoidal hysteresis loops with any positive real value of n > 1 instead of just odd numbered n. Default is false.

`boot` logical, if TRUE results will be bootstrapped by the default arguments of `summary.fittedloop`.

`method` if the default "harmonic2" is used, times along with m and n are either supplied by the user or given default values. Otherwise a non-linear "geometric" method that minimizes the sum of squared geometric residuals will be utilized.

`...` other optional arguments such as `seed`, N=number of realizations, `cbb` for circular block bootstrapping, are passed to `summary.fittedloop` if `boot=TRUE`.

Details

Fits sinusoidal input and output variables x and y that form a hysteresis loop of the form

$$x_t = b.x * cos(2pi * t/period + phase.angle) + cx + e_{x,t}$$

$$y_t = b.y*cos(2pi*t/period+phase.angle)^n+retention*sin(2pi*t/period+phase.angle)^m+cy+e_{y,t}$$

where

$$t = 0, ..., (n.points - 1) if times =' equal'$$

and the error terms, e, are independently and normally distributed. Also produces a vector of derived values. If `floop2r` is used, retention is assumed to be different above and below the split line that seperates the upper and lower loop trajectories, and addition terms are included in the model.

$$y_t = b.y*cos(2pi*t/period+phase.angle)^n+retention.above*sin(2pi*t/period+phase.angle)^m*I(0 < 2pi*t/period$$

where retention.above and retention.below are retention above and below the split line.

Value

`floop` returns an object of class `fittedloop` while `floop2r` returns an object of class `splitloop`.

`call` the function call.

`fit` information dependent on the fitting method used.

`x` the input.

`y` the output.

`pred.x` fitted x values.

`pred.y` fitted y values.

`period.time` time vector used to fit x and y.

`residuals` residuals measured by Euclidean distance.The function `residuals.fittedloop` can produce other types of residuals

`extended.classical` whether or not an extended loop is fit.

`fit.statistics` rudimentary measures, based on the "harmonic2" method, include the Multivariate Final Prediction Error (MFPE) and the AIC for both the output alone and the two variables in combination. Although degree of freedom adjustments are made for other methods, measures of fit require further study.

`values` a named vector of parameter estimates. See `loop.parameters`, same as Estimates here.

`Estimates` a named vector of parameter estimates. See `loop.parameters`, same as values.

`Std.Errors` standard errors for parameters derived using the delta method.

`method` fitting method used.

if `boot==TRUE` `floop` returns an object of class `loopsummary` by making a call to `summary.fittedloop`. See `summary.fittedloop`.

For bootstrapping

`Boot.Estimates` bootstrapped estimates.

`Boot.Std.Errors` bootstrap standard errors.

If multiple loops are fit simultaneously there will be three arguments to the response, models which will contain the separate model fits for each loop, Estimates which will have all of the parameter estimates in matrix form, and Std.Errors which will have all of the delta method standard errors in matrix form. See `floop.repeated`.

Author(s)

Spencer Maynes, Fan Yang, and Anne Parkhurst.

References

Lapshin, R. (1995) Analytical model for the approximation of hysteresis loop and its application to the scanning tunneling microscope.

See Also

Simulate a hysteresis loop with the function `mloop`. Alternatively see `fel` for fitting an ellipse (a hysteresis loop with parameters m=1, n=1) using a variety of methods. Also `residuals.fittedloop`. If a loop is an ellipse, use of fel is strongly recommended instead of floop.

Examples

```
### Simulate and fit a hysteresis loop with n=1 and m=3.
loopf <- mloop(sd.x=0.07,sd.y=0.05,n=3,m=3, retention=.5)
loopf.model <- floop(loopf$x,loopf$y,n=3,m=3)
loopf.model                #Gives estimate and delta standard errors
loopf.model$Estimates      #List estimates only
loopf.model$Std.Errors     #List delta standard errors

### Plot hysteresis loop.
plot(loopf.model,main="Simulated Hysteresis Loop n=3 m=3", values="hysteresis.all")
### Show characteristics of loop on plot
plot(loopf.model,main="Simulated Hysteresis Loop n=3 m=3",values="hysteresis.all",
show=c("retention","coercion"))

### Bootstrap estimates and standard errors (Seed is necessary if want to reproduce results)
booted.loopf <- floop(loopf$x,loopf$y,retention=.5,n=3,m=3,
boot=TRUE, seed=1523)
booted.loopf                 #Gives boot estimates, boot bias, boot SE and boot quartiles
booted.loopf$Boot.Estimates    #Gives boot estimates
booted.loopf$Boot.Std.Errors   #Gives boot standard errors
plot(booted.loopf,main="Simulated Bootstrapped Loop n=1, m=3",
putNumber=TRUE,values="hysteresis.all")
```

`floop.repeated` *Methods for easily fitting multiple loops from repeated measures designs.*

Description

Fit a sinusoidal hysteretic process between an input and an output variable across multiple loops separated by subjects and repeated.

Usage

```
floop.repeated(x,y=NULL,m=1,n=1,subjects=NULL,repeated=NULL,
subjects.in="all",repeated.in="all",...)
floop2r.repeated(x,y=NULL,m=1,n=1,subjects=NULL,repeated=NULL,
subjects.in="all",repeated.in="all",...)
## S3 method for class 'fittedlooplist'
summary(object,N=1000,boot=TRUE,seed=NULL,...)
## S3 method for class 'fittedlooplist2r'
summary(object,N=1000,boot=TRUE,seed=NULL,...)
```

Arguments

x	numeric input vector.
y	numeric output vector.

n	positive integer. Loop shape parameter, see loop.parameters.
m	positive odd integer. Loop bulging parameter, see loop.parameters.
subjects	factor of the same length as x that represents experimental units.
repeated	factor of the same length as x that represents the repeated measure.
subjects.in	a vector of characters, the levels of subjects to be included. Default is "all".
repeated.in	a vector of characters, the levels of repeated to be included. Default is "all".
object	an fittedlooplist object.
N	number of bootstrap replicates. See summary.fittedloop.
boot	whether or not bootstrapping should be performed. See summary.fittedloop.
seed	for generating random numbers. See summary.fittedloop.
...	extra arguments to either floop or summary.fittedloop.

Details

Fits multiple loops with one call, separated by the factors subjects and repeated. The arguments subjects.in and repeated.in are used to select subsets of the factors subjects and repeated.

Value

floop.repeated returns an object of class fittedlooplist.

models	Separate model fits for each loop, see floop.
Estimates	Parameter estimates for all loops in matrix form.
Std.Errors	Delta standard errors for all loops in matrix form.

When boot=TRUE floop.repeated returns an object of class loopsummarylist which consists of

models	Separate model summaries for each ellipse, see summary.fittedloop.
values	Bootstrapped parameter estimates, standard errors, quantiles, and more for each loop.
Boot.Estimates	Bootstrapped parameter estimates with reduced bias.
Boot.Std.Errors	Standard errors provided by bootstrapping.

Author(s)

Spencer Maynes, Fan Yang, and Anne Parkhurst.

References

Yang, F. and A. Parkhurst, Efficient Estimation of Elliptical Hysteresis (submitted)

See Also

floop and summary.fittedloop, also fel.repeated and summary.ellipsefitlist. Also residuals.fittedlooplist.

Examples

```
data(HysteresisData)
loopmodels.rep <- floop.repeated(HysteresisData$X, HysteresisData$Y,
n=5,m=3, subjects = HysteresisData$subjects,subjects.in=c("A","C"),
repeated=HysteresisData$repeated)
loopmodels.rep                #Gives estimates and delta standard errors
loopmodels.rep$Estimates      #List estimates only
loopmodels.rep$Std.Errors     #List delta standard errors

par(mfrow=c(2,2))
plot(loopmodels.rep,main='Simulated Rep Loops',values="hysteresis")
par(mfrow=c(1,1))

loopmodels.rep$models["A",1]       #Select one subject, one replication

### Bootstrap estimates and standard errors (Seed is necessary if want to reproduce results)
boot.rep.loop=floop.repeated(HysteresisData$X, HysteresisData$Y,
n=5,m=3, subjects = HysteresisData$subjects,subjects.in=c("A","C"),
repeated=HysteresisData$repeated,boot=TRUE,seed=123)
boot.rep.loop               #Gives boot estimates, boot bias, boot SE and boot quartiles
boot.rep.loop$Boot.Estimates  #Lists boot estimates
boot.rep.loop$Boot.Std.Errors #Gives boot standard errors

par(mfrow=c(2,2))
plot(boot.rep.loop, main='Simulated Rep Boot Loops', values="hysteresis")
par(mfrow=c(1,1))

##Can write results to a file. First set your directory from the file tab.
## Change file path in command below to coincide with where you want to store data files
##setwd("C:/Users/.....................")
##write.table(boot.rep.loop$Boot.Estimates,"Hys.eg.repbootvalues.txt")
##test.floop=read.table("Hys.eg.repbootvalues.txt",header=TRUE)
##head(test.floop)
```

`floopReflect` *Fitting hysteresis loops for reflected data*

Description

Fits hysteresis loops and ellipses where x and y are flipped and reversed

Usage

```
floopReflect(x,y,...)
felReflect(x,y,...)
```

Arguments

x	the numeric input vector. To fit the loop x is reversed and treated as y.

y	the numeric output vector. To fit the loop y is reversed and treated as x.
...	other arguments to either floop or fel.

Details

A reflected hysteresis loop is one where the output and input are flipped and placed in reverse order.

Value

See floop or fel.

Author(s)

Spencer Maynes, Fan Yang, and Anne Parkhurst.

References

Yang, F. and A. Parkhurst, Estimating Elliptical Hysteresis: A Comparison of Analytic Methods. (submitted)

HysteresisData *Simulated Loop Data for 6 Loops with period=24, n=5 and m=3*

Description

Three subjects with two replications = 6 loops created by mloop. All 6 loops are created with parameters n=5 and m=3 and centered around the origin with a phase angle of pi/2. The 3 subject loops differ in terms of their retention (0.4,0.8,0.6) and b.x saturation point (0.3,0.6,1). Errors in both the input and the output are given a standard deviation of 0.1 for all loops. Used in the help page for floop.repeated.

Usage

```
data(HysteresisData)
```

Format

A data frame with 144 observations on the following 4 variables.

`X` a numeric vector

`Y` a numeric vector

`subjects` subject, one of "A", "B", or "C".

`repeated` which replication within subject. Either 1 or 2.

Examples

```
## Data file is created using the following code
set.seed(1)
loop1 <- mloop(n=5,m=3, retention = 0.4, b.x = 0.3, b.y = 0.8, cx = 0, cy = 0,
    sd.x = 0.1, sd.y = 0.1, period = 24, n.points = 48, phase.angle = pi/2)
loop2 <- mloop(n=5,m=3, retention = 0.8, b.x = 0.6, b.y = 0.8, cx = 0, cy = 0,
    sd.x = 0.1, sd.y = 0.1, period = 24, n.points = 48, phase.angle = pi/2)
loop3 <- mloop(n=5,m=3, retention = 0.6, b.x = 1, b.y = 0.8, cx = 0, cy = 0,
    sd.x = 0.1, sd.y = 0.1, period = 24, n.points = 48, phase.angle = pi/2)
X <- c(loop1$x, loop2$x, loop3$x)
Y <- c(loop1$y, loop2$y, loop3$y)
subjects <- c(rep("A", length(loop1$x)), rep("B", length(loop2$x)), rep("C",length(loop3$x)))
    repeated <- rep(c(1,2),each=24,times=3)

##Use data to fit 6 Hysteresis Loops
data(HysteresisData)
six.loops <- floop.repeated(HysteresisData$X, HysteresisData$Y, n=5,m=3,
subjects = HysteresisData$subjects,repeated=HysteresisData$repeated)
six.loops

#Model fit for B-1
six.loops$models["B",1]
par(mfrow=c(2,3))
plot(six.loops)
par(mfrow=c(1,1))
```

`loop.parameters` *Inherent and Derived Parameter Definitions for Hysteresis Loops/Ellipses*

Description

`floop` returns a `fittedloop` object and calculates a variety of hysteresis loop parameters. This is a list of definitions for these parameters, as well as some only available for ellipses through `fel`. If `floop2r` is used a number of these parameters have differing values above and below the split line. The generalized transcendental equations used to fit these loops are

$$x_t = b.x * cos(2pi * t/period + phase.angle) + cx + e_{x,t}$$

$$y_t = b.y{*}cos(2pi{*}t/period{+}phase.angle)^n{+}retention{*}sin(2pi{*}t/period{+}phase.angle)^m{+}cy{+}e_{y,t}$$

where

$$t = 0, ..., n.points - 1 if times =' equal'$$

Value

Specified loop parameters.

`n` Positive integer for the split line parameter. If n=1, split line is linear; If n is even, split line has a u shape; If n is odd and higher than 1, split line has a chair or classical shape.

`m` Positive odd integer for the bulging parameter, indicates degree of outward curving (1=highest level of bulging).

Inherent Parameters

`b.x` saturation point x coordinate. Horizontal distance from the center to the maximum value of the input.

`b.y` saturation point y coordinate. Vertical distance from the center to the point where the input is at its maximum.

`phase.angle` defines the starting point of the loop. The initial angle of the input function at its origin.

`cx` center of input x.

`cy` center of output y.

`retention` split point, representing vertical distance from center to upper loop trajectory. It is the intersection of the loop and the output axis characterizing the distortion in the response at the average input challenge.

Derived Hysteresis Parameters

`coercion` the horizontal distance of the input from the center. It indicates the strain the forcing function places on the output. It is the positive root of intersection between the loop and input axis.

`lag` lag indicates the delay between attributes of the output and the input (such as peak to peak for the ellipse when m=1, n=1).

`area` the area of the hysteresis loop. Can indicate the work done during one cycle or period.

`split.angle` beta, the angle between the tangent to the un-split curve at the center and the input axis.

`hysteresis.x` hysteresis along the input axis. The proportion of coercion due to input saturation b.x.

`hysteresis.y` hysteresis along the output axis. The proportion of retention due to b.y.

Ellipse Parameters

`ampx` Amplitude of the input, equal to b.x.

`ampy` Amplitude of the output.

`rote.deg` and `rote.rad`
theta, counter clockwise angle of rotation between the input axis and the semi-major axis of the loop.In degrees and radians respectively.

`semi.major` half major axis of ellipse, maximum distance from center to perimeter of ellipse.

`semi.minor` half minor axis of ellipse, shortest distance from center to perimeter of ellipse.

`focus.x`, `focus.y`
input x and output y distances of focus points from center.

`eccentricity` Measure of deviation from circle. Zero indicate no deviation from circle.

$$\sqrt{\frac{semi.major^2 - semi.minor^2}{/}semi.major^2}$$

Author(s)

Spencer Maynes, Fan Yang, and Anne Parkhurst.

References

Lapshin, R. (1995) Analytical model for the approximation of hysteresis loop and its application to the scanning tunneling microscope.

See Also

`mloop` for simulating a hysteresis loop and `floop` for fitting a hysteresis loop.

Examples

```
theloop<-mloop(sd.x=0.05,sd.y=0.05,n=2,m=3)
loopmodel<-floop(theloop$x,theloop$y,n=2,m=3)
loopmodel
plot(loopmodel,main="Hysteresis Loop n=2 m=3",values="hysteresis")

#Ellipse Parameters
ellipse.eig <- mel(semi.major=7,semi.minor=4,rote.deg=30)
ellip.eigen.fit <- fel(ellipse.eig$x,ellipse.eig$y)
ellip.eigen.fit$Estimates
plot(ellip.eigen.fit,main="Ellipse from Eigenvalue Parameters",
show=c("semi.major","semi.minor","rote.deg"),values="ellipse")
```

---

mel *Simulate (Make) an Ellipse*

Description

Produces an ellipse based on 1 of 4 possible formulations: 1-Eigenvalues, 2-Hysteresis Coefs, 3-Amplitudes and 4-Algebraic Coefs.

Usage

```
mel(method=1,seed=NULL,...)
mel1(cx=32,cy=39,rote.deg=2,semi.major=7,semi.minor=0.23,
phase.angle=0,n.points=24,period=24,sd.x=0,sd.y=0)
mel2(cx=32,cy=39,b.x=6.99,b.y=0.244,retention=0.23,
phase.angle=0,n.points=24,period=24,sd.x=0,sd.y=0)
mel3(cx=32,cy=39,ampx=6.99,ampy=0.335,lag=2.888,phase.angle=0,
n.points=24,period=24,sd.x=0,sd.y=0)
mel4(x2=0.002293,xy=-.06960,y2=0.9976,x=2.567,y=-75.58,int=1432.7,
phase.angle=0,n.points=24,period=24,sd.x=0,sd.y=0)
```

Arguments

method	selects which of the functions mel1, mel2, mel3, mel4 to use to describe the ellipse.
seed	integer, the starting seed.
...	arguments to the functions mel1, mel2, mel3, mel4 described below.
cx	Center of input x.
cy	Center of output y.
phase.angle	defines the starting point of the ellipse. Does not change ellipse shape.
rote.deg	Theta, angle of rotation. In degrees. Only used if method=1.
semi.major	Half length of major axis. Only used if method=1.
semi.minor	Half length of minor axis. Only used if method=1.
b.x	Saturation point x coordinate. Only used if method=2.
b.y	Saturation point y coordinate. Only used if method=2.
retention	another ellipse parameter used if method=2. split point, representing vertical distance from center to upper loop trajectory. It is the intersection of the loop and the output axis characterizing the distortion in the response at the average input challenge.
ampx	The range of the ellipse input values divided by 2. Only used if method=3.
ampy	The range of the ellipse output values divided by 2. Only used if method=3.
lag	The number of points between the location where the input reaches its maximum value and where the output reaches its maximum value. Lag is therefore dependent on the value chosen for period. Only used if method=3.
x2	Coefficient on x^2 in the equation found in details. Only used if method=4.
xy	Coefficient on xy in the equation found in details. Only used if method=4.
y2	Coefficient on y^2 in the equation found in details. Only used if method=4.
x	Coefficient on x in the equation found in details. Only used if method=4.
y	Coefficient on y in the equation found in details. Only used if method=4.
int	Coefficient on the intercept in the equation found in details. Only used if method=4.
n.points	Number of points on ellipse. Equally spaced around circumference of ellipse/period.
period	Number of points required to make a full loop around the ellipse.
sd.x	optional number specifying a normally distributed standard deviation for x.
sd.y	optional number specifying a normally distributed standard deviation for y.

Details

All of the four methods can be used to specify a series of points that make up an ellipse. The function mel uses parameters to form an ellipse and find derived variables such as area, lag, retention, and coercion. Optionally, normally distributed random variation can be introduced in both the x and y directions. The first method is useful alongside the nls, lm and direct fitting methods, while the second is comparable to the harmonic2 ellipse fitting method. The third method for mel is included because it is the easiest to interpret. Finally the fourth method uses the equation 0=a0+a1*x^2+a2*xy+a3*y^2+a4*x+a5*y to form an ellipse. The "a" parameters here are marked as int, x2, xy, y2, x and y in the function itself.

Value

mel returns an object of class ellipsemake.

values	the nine fundamental parameters (cx,cy,rote.deg,semi.major,semi.minor,b.x,b.y,a,phase.angle) of which only four or five are used along with the four derived parameters (area, lag, retention, coercion).
method	the method used.
x	the input x.
y	the output y.

Author(s)

Spencer Maynes, Fan Yang, and Anne Parkhurst.

References

Yang, F. and A. Parkhurst, Efficient Estimation of Elliptical Hysteresis. (submitted)

See Also

fel for fitting observations that form an ellipse and creating an ellipsefit object, plot.ellipsefitfor plotting an ellipsefit object. summary.ellipsefit for summarizing an ellipsefit object, and plot.ellipsesummary for plotting an ellipsesummary object.

Examples

```
ellipseA <- mel(method=3,cx=35, cy=39, ampx=7, ampy=2, lag=3, sd.x=0.2,sd.y=0.04)
 ellipseA.fit <- fel(ellipseA$x,ellipseA$y)
  plot(ellipseA.fit,xlab="Input",ylab="Output",main="Simulated Ellipse",
  putNumber=TRUE)
 boot.ellipseA.fit <- fel(ellipseA$x,ellipseA$y, boot=TRUE, seed=231)
  plot(boot.ellipseA.fit,xlab="Input",ylab="Output",
  main="Bootstrapped Ellipse",values="ellipse.all")

ellipse.eig <- mel(semi.major=7,semi.minor=4,rote.deg=30)
ellip.eigen.fit <- fel(ellipse.eig$x,ellipse.eig$y)
ellip.eigen.fit$Estimates
plot(ellip.eigen.fit,main="Ellipse from Eigenvalue Parameters",
show=c("semi.major","semi.minor","rote.deg"),values="ellipse")
```

---

mloop *Simulate (Make) a Hysteresis Loop*

Description

Simulate a hysteresis loop with a variety of possible parameters.

Usage

```
mloop(cx = 0, cy = 0, retention = 0.2, b.x = 0.6, b.y = 0.8,n = 1, m = 1,
     sd.x = 0, sd.y = 0, phase.angle = 0, n.points = 24,
    period = 24,extended.classical=FALSE,seed=NULL)
mloop2r(cx=0,cy=0,retention.above=0.2,retention.below=0.15,b.x=0.6,b.y=0.8,n=1,
        m=1,sd.x=0,sd.y=0,phase.angle=0,n.points=24,period=24,
        extended.classical=FALSE,seed=NULL)
```

Arguments

n	Positive integer for the split line parameter. If n=1, split line is linear; If n is even, split line has a u shape; If n is odd and higher than 1, split line has a chair or classical shape.
m	Positive odd integer for the bulging parameter, indicates degree of outward curving (1=highest level of bulging).
b.x	number. Saturation point x coordinate. Horizontal distance from the center to the maximum value of the input challenge.
b.y	number. Saturation point y coordinate. Vertical distance from the center to the point where the input is at its maximum.
phase.angle	number in degrees. Defines the starting point of the loop. The initial angle of the input function at its origin.
cx	number. Center of input x.
cy	number. Center of output y.
retention	number. Split point, represents vertical distance from center to upper loop trajectory. It is the intersection of the loop and the output axis characterizing the distortion in the response at the average input challenge. Assumes symmetrical curve above and below split line.
retention.above	number. Retention above the split line. `mloop2r` creates a loop where retention above and below the split line may be different.
retention.below	number. Retention below the split line.
sd.x	number. Standard deviation of the normally distributed variation in the input vector x.
sd.y	number. Standard deviation of the normally distributed variation in the output vector y.
n.points	number of points on loop.
period	number of equally spaced points required to make a full loop.
extended.classical	logical. If true, fit a classical hysteresis loop regardless of n. Uses $y_t = sign(cos(2*pi*t/period))*b.y*abs(cos(2*pi*t/period))^n+retention*sin(2*pi*t/period)^m+c$ instead of $y_t = b.y*cos(2*pi*t/period)^n+retention*sin(2*pi*t/period)^m+cy+e_{y,t}$

	Allows the user to fit classical loops with any n>1 instead of just odd numbered n. Default is false.
seed	integer. Starting seed.

Details

Simulates input and output variables x and y that form a hysteresis loop of the form

$$x_t = b.x * cos(2pi * t/period + phase.angle) + cx + e_{x,t}$$

$$y_t = b.y*cos(2pi*t/period+phase.angle)^n+retention*sin(2pi*t/period+phase.angle)^m+cy+e_{y,t}$$

where

$$t = 0, ..., n.points - 1 if times =' equal'$$

and the error terms e are normally distributed. Also produces a vector of derived values.

Value

mloop returns an object of class `hysteresisloop`.

values	estimated values of various coefficients and derived parameters of the hysteresis loop. See `loop.parameters`
x	the input x.
y	the output y.

Author(s)

Spencer Maynes, Fan Yang, and Anne Parkhurst.

References

Lapshin, R. (1995) Analytical model for the approximation of hysteresis loop and its application to the scanning tunneling microscope.

See Also

Fit a hysteresis loop with the function `floop`.

Examples

```
#Simulate a loop with n=3, m=1, retention=0.9
loop1 <- mloop(cx=5,cy=8,retention=0.9,sd.x=0.01,sd.y=0.05,n=3,m=1)
loopmodel <- floop(loop1$x,loop1$y,n=3,m=1)
loopmodel
##Plot hysteresis loop.
plot(loopmodel,main="Simulated Hysteresis Loop n=3 m=1",xlab="Input",
ylab="Output",values="hysteresis.all")
```

plot.fittedloop *Plot a fitted ellipse or hysteresis loop.*

Description

A scatterplot of x and y fitted with an ellipse or hysteresis loop. Uses objects created by fel, summary.ellipsefit and floop. Can also plot an ellipsefitlist or ellipsesummarylist object that contains multiple ellipses.

Usage

```
## S3 method for class 'ellipsefit'
plot(x,putNumber=FALSE,values=NULL,
xlim=NULL,ylim=NULL,main=NULL,newPred=TRUE,show=NULL,split.line=FALSE,...)
## S3 method for class 'ellipsesummary'
plot(x,putNumber=FALSE,values=NULL,
xlim=NULL,ylim=NULL,main=NULL,newPred=TRUE,split.line=FALSE,...)

## S3 method for class 'ellipsefitlist'
plot(x,main=NULL, values=NULL, ...)
## S3 method for class 'ellipsesummarylist'
plot(x,main=NULL, values=NULL, ...)

## S3 method for class 'fittedloop'
plot(x,split.line=TRUE,xlim=NULL,
ylim=NULL,putNumber=FALSE,values=NULL,main=NULL,show=NULL,...)
## S3 method for class 'loopsummary'
plot(x,split.line=TRUE,xlim=NULL,
ylim=NULL,putNumber=FALSE,values=NULL,main=NULL,...)

## S3 method for class 'fittedlooplist'
plot(x,main=NULL,values=NULL,...)
## S3 method for class 'loopsummarylist'
plot(x,main=NULL,values=NULL,...)
## S3 method for class 'fittedlooplist2r'
plot(x,main=NULL,values=NULL,...)
## S3 method for class 'loopsummarylist2r'
plot(x,main=NULL,values=NULL,...)

## S3 method for class 'loop2r'
plot(x,split.line=TRUE,xlim=NULL,ylim=NULL,putNumber=FALSE,main=NULL,values=NULL,...)
## S3 method for class 'loop2rsummary'
plot(x,split.line=TRUE,xlim=NULL,ylim=NULL,putNumber=FALSE,main=NULL,values=NULL,...)
```

Arguments

x	a fitted ellipse or hysteresis loop created by either fel, summary.ellipsefit or floop.
putNumber	optional logical that numbers points from first to last.
values	one of NULL, "hysteresis", "inherent", "derived", "hysteresis.all", "ellipse", or "ellipse.all". Parameter values printed in title. Default is NULL in which case none are printed. See loop.parameters or ellipse.parameters
xlim	limits for x axis.
ylim	limits for y axis.
main	an overall title for the plot.
newPred	draw an ellipse with 100 points. If FALSE use predicted ellipse from ellipsefit object which will result in a rougher shape.
show	a character vector of parameters to be shown in the plot. Possible values are "retention", "coercion", "b.x", "b.y", "semi.major","semi.minor","rote.deg","focus.x",and "focus.y". show is not available for bootstrapped results.
split.line	logical. Whether to include the split line, which is the input output relationship when hysteresis is removed.
...	Arguments to be passed to plot.

Author(s)

Spencer Maynes, Fan Yang, and Anne Parkhurst.

See Also

fel for fitting points that form an ellipse and summary.ellipsefit for bootstrapping and summarizing an ellipsefit object. Also floop and summary.fittedloop for fitting and summarizing hysteresis loops more generally.

Examples

```
##Fit and plot an ellipse
  ellipse1 <- mel(sd.x=0.2,sd.y=0.04)
  ellipse1.fit <- fel(ellipse1$x,ellipse1$y)
plot(ellipse1.fit,main="Simulated Ellipse",xlab="X Input",
ylab="Y Output",show=c("semi.major", "semi.minor"),values="ellipse.all")
### Bootstrapping
  booted.ellipse <- fel(ellipse1$x,ellipse1$y,boot=TRUE, seed=123)
plot(booted.ellipse,xlab="X Input",ylab="Y Output",
main="Simulated Bootstrap Ellipse",values="ellipse")

##Fit and plot a hysteresis loop
  loop1 <- mloop(sd.x=0.05,sd.y=0.05,n=5,m=3)
  loopmodel <- floop(loop1$x,loop1$y,n=5,m=3)
plot(loopmodel,main="Simulated Hysteresis Loop n=5 m=3",
xlab="Input",ylab="Output", values="hysteresis.all")
  booted.loop <- floop(loop1$x,loop1$y,,n=5,m=3,boot=TRUE, seed=1523)
```

```
plot(booted.loop,xlab="Input",ylab="Output",
main="Simulated Bootstrapped Loop n=5, m=3",putNumber=TRUE)
plot(booted.loop,main="Simulated Bootstrapped Loop n=5, m=3",
xlab="Input",ylab="Output",values="hysteresis.all")
```

`residuals.fittedloop` *Residuals, studentized residuals and fitted values for the hysteresis package.*

Description

Extract input, output, geometric and algebraic residuals, studentized residuals and fitted values from fitted loops or ellipses.

Usage

```
##S3 methods for classes 'ellipsefit', 'ellipsesummary', 'fittedloop',
##'loopsummary','ellipsefitlist', 'ellipsesummarylist', 'fittedlooplist',
##'loopsummarylist', 'loop2r', 'fittedlooplist2r',
##'loopsummarylist2r' and 'loop2rsummary'.
## S3 method for class 'ellipsefit'
residuals(object,...)
## S3 method for class 'ellipsefit'
rstudent(model,...)
## S3 method for class 'ellipsefit'
fitted(object,...)
```

Arguments

`object`	an object created by `fel` or `floop`.
`model`	an object created by `fel` or `floop`.
`...`	other arguments.

Details

Geometric residuals are based on the straight line distance between predicted and true values along an x,y cartesian plane, and algebraic residuals are based on the method used to calculate the `ellipsefit` object. If method="harmonic2" (which is always the case if this is a `fittedloop` object) or if bootstrapping has occurred, then, there are no algebraic residuals and `residuals.ellipsefit` replaces these with the geometric residuals.

Studentization for the `rstudent` function is performed as if method="harmonic2" regardless of the method used for fitting the ellipse/loop. Therefore, unless method="harmonic2" and no bootstrapping is performed, these are pseudo-studentized residuals, not true studentized residuals. This is internal scaling studentization. Studentization for bootstrapping in the functions `summary.ellipsefit`/`summary.fittedloop` differs from the studentization performed by `rstudent` in that it only accounts for the influence matrix and does not divide by the standard deviation.

Value

input	a numeric vector. Observed input - fitted input for residuals.
output	a numeric vector. Observed output - fitted output for residuals.
geometric	a numeric vector. Not available with rstudent. See details.
algebraic	a numeric vector. Not available with rstudent or when the 'harmonic2' method is used. See details.

Author(s)

Spencer Maynes, Fan Yang, and Anne Parkhurst.

References

Yang, F. and A. Parkhurst, Estimating Elliptical Hysteresis: A Comparison of Analytic Methods. (submitted)

See Also

fel, floop, summary.ellipsefit and summary.fittedloop.

Examples

```
##For multiple loops/ellipses
data(HysteresisData)
Mloopmodels.rep <- floop.repeated(HysteresisData$X, HysteresisData$Y,
n=5,m=3, subjects = HysteresisData$subjects,subjects.in=c("A","C"),
repeated=HysteresisData$repeated)
Mloopmodels.rep               #Gives estimates and delta standard errors
residuals(Mloopmodels.rep)  #$input $output $geometric
fitted(Mloopmodels.rep)     #$input $output
scatterplotMatrix(cbind(residuals(Mloopmodels.rep)$input,
residuals(Mloopmodels.rep)$output,residuals(Mloopmodels.rep)$geometric,
fitted(Mloopmodels.rep)$input,fitted(Mloopmodels.rep)$output),
  main='Residuals for Multiple Hysteresis Loops',smooth=FALSE,
  var.labels=c("Input Resid","Output Resid","Geometric Resid",
  "Fitted Input", "Fitted Output"),
  groups=residuals(Mloopmodels.rep)$repeated)
rstudent(Mloopmodels.rep)  #$input $output
scatterplotMatrix(cbind(rstudent(Mloopmodels.rep)$input,
rstudent(Mloopmodels.rep)$output,fitted(Mloopmodels.rep)$input,
fitted(Mloopmodels.rep)$output),main='Studentized Residuals
for Multiple Hysteresis Loops',smooth=FALSE,
var.labels=c("Input Resid","Output Resid", "Fitted Input",
"Fitted Output"),groups=residuals(Mloopmodels.rep)$repeated)

##For single Ellipse
ellipse1 <- mel(sd.x=0.2,sd.y=0.04)
ellipse1.fit <- fel(ellipse1$x,ellipse1$y)
residuals(ellipse1.fit)
fitted(ellipse1.fit)
```

```
scatterplotMatrix(cbind(residuals(ellipse1.fit)$input,
residuals(ellipse1.fit)$output,residuals(ellipse1.fit)$geometric,
fitted(ellipse1.fit)$input,fitted(ellipse1.fit)$output),
main='Residuals for Simulated Ellipse',smooth=FALSE,
var.labels=c("Input Resid","Output Resid","Geometric Resid",
"Fitted Input", "Fitted Output"))

rstudent(ellipse1.fit) #for input and output variables
scatterplotMatrix(cbind(rstudent(ellipse1.fit)$input,
rstudent(ellipse1.fit)$output,fitted(ellipse1.fit)$input,
fitted(ellipse1.fit)$output),main='Studentized Residuals
for Simulated Ellipse',smooth=FALSE,
var.labels=c("Input Resid","Output Resid","Fitted Input",
"Fitted Output"))

plot(ellipse1.fit$pred.y,rstudent(ellipse1.fit)$output,
xlab="Fitted Output",ylab="Output Studentized Residuals",
main="Studentized Residuals:Simulated Ellipse")
abline(h = 0, lty = 2, col = "gray")
qqnorm(rstudent(ellipse1.fit)$output,sub='Output Studentized
Residuals Simulated Ellipse')
qqline(rstudent(ellipse1.fit)$output,col="red") #q-q line
```

`summary.fittedloop` *Summarizing and Bootstrapping Fitted Ellipses or Loops*

Description

summary methods for classes `ellipsefit` and `fittedloop` created by the functions `fel` and `floop`. Can bootstrap results to produce parameter estimates with reduced bias and standard errors.

Usage

```
## S3 method for class 'ellipsefit'
summary(object,boot=TRUE, N = 1000,
studentize=TRUE, center=FALSE, cbb=NULL, joint=FALSE,seed=NULL,...)
## S3 method for class 'fittedloop'
summary(object,boot=TRUE,N=1000,
cbb=NULL,joint=FALSE,seed=NULL,...)
## S3 method for class 'loop2r'
summary(object,boot=TRUE,N=1000,
cbb=NULL,joint=FALSE,seed=NULL,...)
```

Arguments

`object`	an object of class `ellipsefit` or `fittedloop`, a result of a call to `fel` or `floop`.
`boot`	logical. Whether to perform bootstrapping to get standard errors, which is the default TRUE, or to get standard errors through the delta method if FALSE.

N	optional number of bootstrap replicates. Default of 1000.
studentize	studentize the residuals to improve performance. Default is true.
center	center x and y residuals around zero. Default is false. Irrelevant for "harmonic2" method.
cbb	allows for circular block bootstrapping. The default is NULL in which case circular block bootstrapping is not performed. If cbb is an integer greater than 1 which is a divisor of either the number of observations for methods "nls", "lm", "geometric" or the number of observations minus 3 for method="harmonic2" then it is used as the block size for circular block bootstrapping.
joint	logical that defaults to false. Resample input and output residuals paired by observation, instead of separately.
seed	either NULL or a positive integer. Set the random number seed.
...	further arguments passed to or from other methods.

Details

Bootstrap objects created by fitting hysteretic data with one of the functions fel or floop and produce statistical summaries. Bootstrapping reduces the bias on estimates and also gives standard errors. Bootstrap estimates are created by subtracting original estimates from bootstrap means to get a bias estimate, and then subtracting this bias from the original estimate.

Residuals are studentized as if they were produced using the harmonic2 method, regardless of which method was actually used to produce them. However, unpublished simulation studies show that these studentized residuals provide better 95 percent coverages for all methods despite this. This studentization is not true studentization as in rstudent.ellipsefit as it only accounts for the influence matrix and does not divide by the standard deviation.

If residuals are serially correlated than the argument cbb may be used to sample blocks of length cbb instead of individual residuals. Circular block bootstrapping is used, which means that all residuals are equally likely to be included and blocks can be made up of the last points on the ellipse together with the first.

When using the 'nls', 'geometric' or 'lm' methods individual bootstrap replications may occasionally fail to converge, when this occurs an extra replication will take the place of the one that failed to converge and a warning message will be produced.

Value

call	function call for original fit.
method	fitting method used. Only for summary.ellipsefit. See fel.
x	the input x.
y	the output y.
pred.x	the bootstrap fitted values for x.
pred.y	the bootstrap fitted values for y.
values	matrix containing parameter and standard error estimates, bootstrap quantiles, and bootstrapped parameter estimates for a wide variety of parameters. See loop.parameters.

`Delta.Std.Errors`
: the delta method standard errors.

`fit.statistics`
: rudimentary measures, based on the "harmonic2" method, include the Multivariate Final Prediction Error (MFPE) and the AIC for both the output alone and the two variables in combination. Although degree of freedom adjustments are made for other methods, measures of fit require further study

For bootstrapping

`summarycall`
: the function call.

`boot.data`
: parameter estimates from individual bootstrap replications.

`Boot.Estimates`
: bootstrapped estimates.

`Boot.Std.Errors`
: bootstrap standard errors.

Author(s)

Spencer Maynes, Fan Yang, and Anne Parkhurst.

References

Yang, F. and A. Parkhurst, Efficient Estimation of Elliptical Hysteresis (submitted)

Correa, Solange, Extended Bootstrap Bias Correction with Application to Multilevel Modelling of Survey Data under Informative Sampling.

See Also

`fel` for fitting points that form an ellipse and creating an ellipsefit object and `plot.ellipsesummary` for plotting an ellipsesummary object.

Examples

```
#Loop example with circular block bootstrapping
loop1 <- mloop(n=1,m=2,sd.x=0.05,sd.y=0.05)
loop1.fit <- floop(loop1$x,loop1$y,m=2,n=1)
boot.loop1 <- summary(loop1.fit,cbb=3)
boot.loop1
plot(boot.loop1)

#Ellipse example.
ellipse1 <- mel(sd.x=0.2,sd.y=0.04)
ellipse1.fit <- fel(ellipse1$x,ellipse1$y)
boot.ellipse1.fit <- summary(ellipse1.fit)
boot.ellipse1.fit
plot(boot.ellipse1.fit,xlab="Input",ylab="Output",
main="Bootstrapped Ellipse",putNumber=TRUE)
```

Index