Package 'nlRtsa'

January 22, 2016

Type Package

Title nonlineaR time seRies analysis.

Version 0.1

Date 2015-12-11

Description
Companion to the course Dynamics of Complex Systems given at the Graduate School Behavioural Science (Research Master Behavioural Science) at the Radboud University Nijmegen.

Depends R (>= 3.2.2)

LazyData true

License GPL-2

Imports plyr, dplyr, lattice, latticeExtra, grid, gridExtra, ggplot2, pracma, signal, oce, forecast, fractal

Suggests tseries, matlab, ifultools, knitr, rmarkdown, roxygen2

RoxygenNote 5.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Fred Hasselman [aut, cre],
Maarten Wijnants [aut]

Maintainer Fred Hasselman <f.hasselman@pwo.ru.nl>

R topics documented:

`fd.dfa` *Detrended Fluctuation Analysis (DFA)*

Description

fd.dfa

Usage

```
## S3 method for class 'dfa'
fd(y, fs = NULL, dtrend = "poly1", normalize = FALSE,
  sum.order = 1, scale.max = trunc(length(y)/4), scale.min = 4,
  scale.ratio = 2^(1/4), overlap = 0, plot = FALSE)
```

Arguments

`y`	A numeric vector or time series object.
`normalize`	Normalize the series (default).
`plot`	Return the log-log spectrum with linear fit (default).
`detrend`	Subtract linear trend from the series (default).
`dmethod`	Method to use for detrending, see `DFA`.

Value

Estimate of Hurst exponent (slope of `log(bin)` vs. `log(RMSE)`) and an FD estimate based on Hasselman(2013) A list object containing:

- A data matrix `PLAW` with columns `freq.norm`, `size` and `bulk`.
- Estimate of scaling exponent `sap` based on a fit over the standard range (`fullRange`), or on a user defined range `fitRange`.
- Estimate of the the Fractal Dimension (`FD`) using conversion formula's reported in Hasselman(2013).
- Information output by various functions.

Author(s)

Fred Hasselman

References

Hasselman, F. (2013). When the blind curve is finite: dimension estimation and model inference based on empirical waveforms. Frontiers in Physiology, 4, 75. http://doi.org/10.3389/fphys.2013.00075

See Also

Other FD estimators: `fd.psd`, `fd.sda`

fd.psd *Power Spectral Density Slope (PSD).*

Description

Estimate Alpha, Hurst Exponent and Fractal Dimension through log-log slope.

Usage

```
## S3 method for class 'psd'
fd(y, fs = NULL, normalize = TRUE, dtrend = TRUE,
  plot = FALSE)
```

Arguments

`y`	A numeric vector or time series object.
`normalize`	Normalize the series (default).
`plot`	Return the log-log spectrum with linear fit (default).
`detrend`	Subtract linear trend from the series (default).

Details

Calls function `SDF` to estimate the scaling exponent of a timeseries based on the periodogram frequency spectrum. After detrending and normalizing the signal (if requested), `SDF` is called using a Tukey window (`raised cosine taper`).

A line is fitted on the periodogram in log-log coordinates. Two fit-ranges are used: The 25% lowest frequencies and the Hurvich-Deo estimate (`HDEst`).

Value

A list object containing:

- A data matrix `PLAW` with columns `freq.norm`, `size` and `bulk`.
- Estimate of scaling exponent `alpha` based on a fit over the lowest 25% frequencies (`low25`), or using the HD estimate `HD`.
- Estimate of the the Fractal Dimension (`FD`) using conversion formula's reported in Hasselman(2013).
- Information output by various functions.

Author(s)

Fred Hasselman

References

Hasselman, F. (2013). When the blind curve is finite: dimension estimation and model inference based on empirical waveforms. Frontiers in Physiology, 4, 75. http://doi.org/10.3389/fphys.2013.00075

See Also

Other FD estimators: fd.dfa, fd.sda

Examples

```
fd.psd(rnorm(2048), plot = TRUE)
```

---

fd.sda *Standardised Dispersion Analysis (SDA).*

Description

Standardised Dispersion Analysis (SDA).

Usage

```
## S3 method for class 'sda'
fd(y, fs = NULL, normalize = TRUE, dtrend = FALSE,
  scales = dispersion(y)$scale, fitRange = c(scales[1],
  scales[length(scales) - 2]), plot = FALSE)
```

Arguments

y	A numeric vector or time series object.
normalize	Normalize the series (default).
plot	Return the log-log spectrum with linear fit (default).

Value

A list object containing:

- A data matrix PLAW with columns freq.norm, size and bulk.
- Estimate of scaling exponent sap based on a fit over the standard range (fullRange), or on a user defined range fitRange.
- Estimate of the the Fractal Dimension (FD) using conversion formula's reported in Hasselman(2013).
- Information output by various functions.

Author(s)

Fred Hasselman

References

Hasselman, F. (2013). When the blind curve is finite: dimension estimation and model inference based on empirical waveforms. Frontiers in Physiology, 4, 75. http://doi.org/10.3389/fphys.2013.00075

See Also

Other FD estimators: fd.dfa, fd.psd

Examples

```
fd.sda(rnorm(2048))
```

`gg.plotHolder` *gg.plotHolder*

Description

gg.plotHolder

Usage

```
gg.plotHolder(useArial = F, afmPATH = "~/Dropbox")
```

Arguments

`useArial`	Use the Arial font (requires `.afm` font files in the `afmPath`)
`afmPATH`	Path to Arial `.afm` font files.

Value

A blank `ggplot2` object that can be used in concordance with `grid.arrange`.

Examples

```
# Create a plot with marginal distributions.
library(ggplot2)
library(scales)

df <- data.frame(x = rnorm(n = 100), y = rnorm(n = 100), group = factor(sample(x=c(0,1), size = 100, replace =

scatterP <- ggplot(df, aes(x = x, y =y, colour = group)) + geom_point() + gg.theme()
xDense <- ggplot(df, aes(x = x, fill = group)) + geom_density(aes(y= ..count..),trim=FALSE, alpha=.5) + gg.th
yDense <- ggplot(df, aes(x = y, fill = group)) + geom_density(aes(y= ..count..),trim=FALSE, alpha=.5) + coord

library(gridExtra)
grid.arrange(xDense, gg.plotHolder(), scatterP, yDense, ncol=2, nrow=2, widths=c(4, 1.4), heights=c(1.4, 4))
```

`gg.theme` *gg.theme*

Description

gg.theme

Usage

```
gg.theme(type = c("clean", "noax"), useArial = F, afmPATH = "~/Dropbox")
```

Arguments

`type`	One of `"clean"`, or `"noax"`
`useArial`	Use the Arial font (requires `.afm` font files in the `afmPath`)
`afmPATH`	Path to Arial `.afm` font files.

Details

Will generate a `"clean"` ggplot theme, or a theme without any axes (`"noax"`).

Some scientific journals explicitly request the Arial font should be used in figures. This can be achieved by using `.afm` font format (see, e.g. http://www.pure-mac.com/font.html).

Value

A theme for `ggplot2`.

Examples

```
library(ggplot2)
g <- ggplot(data.frame(x = rnorm(n = 100), y = rnorm(n = 100)), aes(x = x, y = y)) + geom_point()
g + gg.theme()
g + gg.theme("noax")
```

---

growth.ac — *Autocatlytic Growth: Iterating difference equations (maps)*

Description

Autocatlytic Growth: Iterating difference equations (maps)

Usage

```
growth.ac(Y0 = 0.01, r = 1, k = 1, N = 100, type = c("driving",
  "damping", "logistic", "vanGeert")[1])
```

Arguments

`Y0`	Initial value.
`r`	Growth rate parameter.
`k`	Carrying capacity.
`N`	Length of the time series.
`type`	One of: "driving" (default), "damping", "logistic", "vanGeert1991".

Value

A timeseries object of length N.

Author(s)

Fred Hasselman

See Also

Other autocatalytic growth functions: growth.ac.cond

Examples

```
# The logistic map in the chaotic regime
growth.ac(Y0 = 0.01, r = 4, type = "logistic")
```

---

growth.ac.cond   *Conditional Autocatlytic Growth: Iterating difference equations (maps)*

Description

Conditional Autocatlytic Growth: Iterating difference equations (maps)

Usage

```
growth.ac.cond(Y0 = 0.01, r = 0.1, k = 2, cond = cbind.data.frame(Y =
  0.2, par = "r", val = 2), N = 100)
```

Arguments

Y0	Initial value
r	Growth rate parameter
k	Carrying capacity
cond	Conditional rules passed as a data.frame of the form: cbind.data.frame(Y = ..., par = ..., val = ...)
N	Length of the time series

Author(s)

Fred Hasselman

See Also

Other autocatalytic growth functions: growth.ac

Examples

```
# Plot with the default settings
xyplot(growth.ac.cond())

# The function such that it can take a set of conditional rules and apply them sequentially during the iterati
# The conditional rules are passed as a `data.frame`
(cond <- cbind.data.frame(Y = c(0.2, 0.6), par = c("r", "r"), val = c(0.5, 0.1)))
xyplot(growth.ac.cond(cond=cond))

# Combine a change of `r` and a change of `k`
(cond <- cbind.data.frame(Y = c(0.2, 1.99), par = c("r", "k"), val = c(0.5, 3)))
xyplot(growth.ac.cond(cond=cond))
```

```
# A fantasy growth process
(cond <- cbind.data.frame(Y = c(0.1, 1.99, 1.999, 2.5, 2.9), par = c("r", "k", "r", "r","k"), val = c(0.3, 3, 
xyplot(growth.ac.cond(cond=cond))
```

---

in.IT *Initialise It*

Description

Load and/or install R packages

Usage

```
in.IT(need = NULL, inT = TRUE)
```

Arguments

need	A vector of package names to be loaded. The wrapper functions have a predefinded need list and can be used as shortcuts (see details).
inT	Logical. If TRUE (default), packages in need wil be installed if they are not available on the system.

Details

in.IT will check if the Packages in the list argument need are installed on the system and load them. If inT=TRUE (default), it will first install the packages if they are not present and then proceed to load them.

Author(s)

Fred Hasselman

See Also

Other initialise packages: un.IT

Examples

```
in.IT(c("reshape2", "plyr", "dplyr"))
```

`plotRP.crqaOutput` *plot.crqaOutput*

Description

Creates a recurrence plot from the sparse matrix output generated by `crqa`.

Usage

```
## S3 method for class 'crqaOutput'
plotRP(crqaOutput)
```

Arguments

`crqaOutput`	List output from `crqa`

Value

A recurrence plot.

Author(s)

Fred Hasselman

`sa2fd.dfa` *Informed Dimension estimate from DFA slope (H)*

Description

Conversion formula: Detrended Fluctuation Analysis (DFA) estimate of the Hurst exponent (a self-affinity parameter `sa`) to an informed estimate of the (fractal) dimension (FD).

Usage

```
## S3 method for class 'dfa'
sa2fd(sa)
```

Arguments

`sa`	Self-Afinity parameter estimate based on DFA slope (e.g., `fd.sda`)).

Details

The DFA slope (H) will be converted to a dimension estimate using:

$$D_{DFA} \approx 2 - (\tanh(\log(3) * sa))$$

Value

An informed estimate of the Fractal Dimension, see Hasselman(2013) for details.

Author(s)

Fred Hasselman

References

Hasselman, F. (2013). When the blind curve is finite: dimension estimation and model inference based on empirical waveforms. Frontiers in Physiology, 4, 75. http://doi.org/10.3389/fphys.2013.00075

See Also

Other SA to FD converters: sa2fd.psd, sa2fd.sda

Examples

```
# Informed FD of white noise
sa2fd.dfa(0.5)

# Informed FD of Pink noise
sa2fd.dfa(1)

# Informed FD of blue noise
sa2fd.dfa(0.1)
```

---

sa2fd.psd *Informed Dimension estimate from Spectral Slope (aplha)*

Description

Conversion formula: From periodogram based self-affinity parameter estimate (sa) to an informed estimate of the (fractal) dimension (FD).

Usage

```
## S3 method for class 'psd'
sa2fd(sa)
```

Arguments

sa	Self-Affinity parameter estimate based on PSD slope (e.g., fd.psd)).

Details

The spectral slope will be converted to a dimension estimate using:

$$D_{PSD} \approx \frac{3}{2} + \frac{14}{33} * \tanh\left(Slope * \ln(1 + \sqrt{2})\right)$$

Value

An informed estimate of the Fractal Dimension, see Hasselman(2013) for details.

Author(s)

Fred Hasselman

References

Hasselman, F. (2013). When the blind curve is finite: dimension estimation and model inference based on empirical waveforms. Frontiers in Physiology, 4, 75. http://doi.org/10.3389/fphys.2013.00075

See Also

Other SA to FD converters: sa2fd.dfa, sa2fd.sda

Examples

```
# Informed FD of white noise
sa2fd.psd(0)

# Informed FD of Brownian noise
sa2fd.psd(-2)

# Informed FD of blue noise
sa2fd.psd(2)
```

---

sa2fd.sda *Informed Dimension estimate from SDA slope.*

Description

Conversion formula: Standardised Dispersion Analysis (SDA) estimate of self-affinity parameter (SA) to an informed estimate of the fractal dimension (FD).

Usage

```
## S3 method for class 'sda'
sa2fd(sa)
```

Arguments

sa	Self-afinity parameter estimate based on SDA slope (e.g., fd.sda)).

Details

Note that for some signals different PSD slope values project to a single SDA slope. That is, SDA cannot distinguish between all variaties of power-law scaling in the frequency domain.

Value

An informed estimate of the Fractal Dimension, see Hasselman(2013) for details.

Author(s)

Fred Hasselman

References

Hasselman, F. (2013). When the blind curve is finite: dimension estimation and model inference based on empirical waveforms. Frontiers in Physiology, 4, 75. http://doi.org/10.3389/fphys.2013.00075

See Also

Other SA to FD converters: `sa2fd.dfa`, `sa2fd.psd`

Examples

```
# Informed FD of white noise
sa2fd.sda(-0.5)

# Informed FD of Brownian noise
sa2fd.sda(-1)

# Informed FD of blue noise
sa2fd.sda(-0.9)
```

`scaleR` *ScaleR*

Description

Rescale a vector to a user defined range defined by user.

Usage

```
scaleR(x, mn = min(x, na.rm = T), mx = max(x, na.rm = T), lo = 0,
  hi = 1)
```

Arguments

`x`	Input vector or data frame.
`mn`	Minimum value of original, defaults to `min(x, na.rm = TRUE)`.
`mx`	Maximum value of original, defaults to `max(x, na.rm = TRUE)`.
`lo`	Maximum value to rescale to, defaults to `1`.
`hi`	Minimum value to rescale to, defaults to `0`.

Details

Three uses:

1. scaleR(x) - Scale x to data range: min(x.out)==0; max(x.out)==1
2. scaleR(x,mn,mx) - Scale x to arg. range: min(x.out)==mn==0; max(x.out)==mx==1
3. scaleR(x,mn,mx,lo,hi) - Scale x to arg. range: min(x.out)==mn==lo; max(x.out)==mx==hi

Value

A data frame with rescaled variables in columns (colum names: ‘inputvariablename.resc‘).

Examples

```
# Works on numeric objects
somenumbers <- cbind(c(-5,100,sqrt(2)),c(exp(1),0,-pi))

scaleR(somenumbers)
scaleR(somenumbers,mn=-100)

# Values < mn will return < lo (default=0)
# Values > mx will return > hi (default=1)
scaleR(somenumbers,mn=-1,mx=99)

scaleR(somenumbers,lo=-1,hi=1)
scaleR(somenumbers,mn=-10,mx=101,lo=-1,hi=4)
```

`try.CATCH` *tryCatch both warnings (with value) and errors*

Description

try.CATCH

Usage

```
try.CATCH(expr)
```

Arguments

`expr`	an R expression to evaluate

Details

In longer simulations, aka computer experiments, you may want to 1) catch all errors and warnings (and continue) 2) store the error or warning messages

Here's a solution (see R-help mailing list, Dec 9, 2010):

Catch *and* save both errors and warnings, and in the case of a warning, also keep the computed result.

Value

a list with 'value' and 'warning', where value' may be an error caught.

Author(s)

Martin Maechler; Copyright (C) 2010-2012 The R Core Team `un.IT` *Un-initialise It*

Description

Unload and/or uninstall R packages.

Usage

```
un.IT(loose, unT = FALSE)
```

Arguments

`loose`	A vector of package names to be unloaded.
`unT`	Logical. If `TRUE`, packages in `loose` wil be un-installed if they are available on the system.

Details

`un.IT`will check if the Packages in the list argument `loose` are installed on the system and unload them. If `unT=TRUE` it will first unload the packages if they are loaded, and then proceed to uninstall them.

Author(s)

Fred Hasselman

See Also

Other initialise packages: `in.IT`

Examples

```
## Not run: un.IT(loose = c("reshape2", "plyr", "dplyr"), unT = FALSE)
```

Index