Package ‘fractalRegression’

August 19, 2023

Type Package

Title Performs Fractal Analysis and Fractal Regression

Version 1.2

Date 2023-08-19

Description
Various functions for performing fractal and multifractal analysis including performing fractal regression. Please refer to Peng and colleagues (1994) <doi:10.1103/physreve.49.1685>, Kantelhardt and colleagues (2002)<doi:10.1016/S0378-4371(02)01383-3>, and Likens and colleagues (2019) <doi:10.1016/j.physa.2019.121580>.

License GPL (>= 3)

Depends R (>= 3.5.0)

Suggests knitr, markdown, rmarkdown, ggplot2, crqa, MFDFA, fracdiff, tseries, fields, gridExtra, qpdf

Encoding UTF-8

Imports Rcpp (>= 1.0.0), colorRamps

LazyData true

LinkingTo Rcpp, RcppArmadillo, RcppEigen

RoxygenNote 7.2.3

NeedsCompilation yes

Author Aaron Likens [aut, cre] (<https://orcid.org/0000-0002-6535-5772>), Travis Wiltshire [aut] (<https://orcid.org/0000-0001-7630-2695>)

Maintainer Aaron Likens <alikens@unomaha.edu>

Repository CRAN

Date/Publication 2023-08-19 20:52:33 UTC

R topics documented:

`dcca` *Detrended Cross-Correlation Analysis*

Description

Fast function for computing detrended cross-correlation analysis (DCCA) on long time series, which is a bivariate extension of detrended fluctuation analysis (DFA).

Usage

```
dcca(x, y, order, scales)
```

Arguments

`x`	A real valued vector (i.e., time series data) to be analyzed.
`y`	A real valued vector (i.e., time series data) to be analyzed.
`order`	is an integer indicating the polynomial order used for detrending the local windows (e.g, 1 = linear, 2 = quadratic, etc.). There is not a pre-determined limit on the order of the polynomial order but the user should avoid using a large polynomial on small windows. This can result in overfitting and non-meaningful estimates.
`scales`	An integer vector of scales over which to compute correlation. Unlike univariate DFA, MRA does not require that scales be in log units. Scale intervals can be sequential, for example, when the analysis is exploratory and no a priori hypotheses have been made about the scale of correlation. A small subset of

targeted scales may also be investigated where scale-specific research questions exist. We have found that windows smaller than say 8 observations create stability problems due to overfitting. This is espcially when the order of the fitting polynomial is large.

Details

Details of the algorithm are specified in Podobnik and Stanley (2008) and in Zebende (2011). In general, the output of the algorithm are estimates of ρDCCA, which range from -1 to 1 and can generally be interpreted as follows:

- $\rho DCCA = -1.0- >$ perfect anti-cross-correlation
- $\rho DCCA = 0.0- >$ no cross-correlation
- $\rho DCCA = 1.0- >$ perfect cross-correlation

Value

The object returned from the function is a list including the following:

- `scales` indicates the values of the scales used for estimates ρDCCA
- `rho` includes the scale-wise estimates of ρDCCA

References

Podobnik, B., & Stanley, H. E. (2008). Detrended cross-correlation analysis: a new method for analyzing two nonstationary time series. Physical review letters, 100(8), 084102.

Zebende, G. F. (2011). DCCA cross-correlation coefficient: Quantifying level of cross-correlation. Physica A: Statistical Mechanics and its Applications, 390(4), 614-618.

//'

Examples

```
# Here is a simple example for running DCCA using a white noise and pink noise time series.
# For more detailed examples, see the vignette.

noise <- rnorm(5000)

pink.noise <- fgn_sim(n = 5000, H = 0.9)

scales <- logscale(scale_min = 10, scale_max = 1250, scale_ratio = 1.1)

dcca.out <- dcca(noise, pink.noise, order = 1, scales = scales)
```

`dcca.plot` *Detrended Cross Correlation Plot*

Description

A plotting method for constructing scalewise correlation plot

Usage

```
dcca.plot(
  rhos,
  order = 1,
  ci = FALSE,
  iterations = NULL,
  return.ci = FALSE,
  loess.rho = FALSE,
  loess.ci = FALSE
)
```

Arguments

`rhos`	an object containing results from detrended cross correlation analysis. The object should be returned from the `dcca` function of this package.
`order`	integer representing the detrending order used in the dcca calculation. Default is 1.
`ci`	a logical indicating whether confidence intervals should be computed using the `iaafft` function from this package. NOTE: with long time series (» than N = 1,000), this can greatly reduce processing speed. Confidence intervals can be used for conventional significance testing of scale-wise correlation coefficients.
`iterations`	integer that specifies the the number of surrogate time series to be generated for the purpose of confidence intervals. Default = 19. Larger number of surrogates will slow computational speed but produce better confidence interval estimates.
`return.ci`	logical indicating whether the confidence intervals should be returned
`loess.rho`	logical indicating whether a loess fit should be used for displaying multiscale regression coefficient trajectories
`loess.ci`	logical indicating whether a loess fit should be used to smooth confidence intervals

`detrend_cov` *Detrended Covariance Functional that returns the detrended covariance between two vectors*

Description

Detrended Covariance Functional that returns the detrended covariance between two vectors

Usage

```
detrend_cov(x, y, m)
```

Arguments

`x`	a real valued column vector
`y`	is a real valued column vector
`m`	is the detrending order

`dfa` *Detrended Fluctuation Analysis*

Description

Fast function for computing detrended fluctuation analysis (DFA), a widely used method for estimating long-range temporal correlations in time series data. DFA is also a form of mono-fractal analysis that indicates the degree of self-similarity across temporal scales.

Usage

```
dfa(x, order, verbose, scales, scale_ratio = 2)
```

Arguments

`x`	A real valued vector (i.e., time series data) to be analyzed.
`order`	An integer indicating the polynomial order used for detrending the local windows (e.g, 1 = linear, 2 = quadratic, etc.). There is not a pre-determined limit on the order of the polynomial order but the user should avoid using a large polynomial on small windows. This can result in overfitting and non-meaningful estimates.
`verbose`	If the value of verbose = 1, then a list object is returned that includes: `log_scales` the log of all included scales, `log_rms` the log root mean square error (RMS) per scale, and `alpha` the overall α estimate. If the value of verbose = 0, then a list containing only 'alpha' will be returned.

scales — An integer valued vector indicating the scales one wishes to resolve in the analysis. Best practice is to use scales which are evenly spaced in the logarithmic domain e.g., `scales = 2^(4:(N/4))`, where N is the length of the time series. Other, logarithmic bases may also be used to give finer resolution of scales while maintaining ~= spacing in the log domain e.g, `scales = unique(floor(1.1^(30:(N/4))))`. Note that fractional bases may produce duplicate values after the necessary floor function.

scale_ratio — A scaling factor by which successive window sizes were created. The default is 2 but should be addressed according to how scales were generated for example using `logscale(16, 100, 1.1)`, where 1.1 is the scale ratio.

Details

Details of the algorithm are specified in detail in Peng et al. (1994) and visualized nicely in Kelty-Stephen et al. (2016). The output of the algorithm is an α (alpha) estimate which is a generalization of the Hurst Exponent. Conventional interpretation of α is:

- $\alpha < 0.5 =$ anti-correlated
- $\alpha\ = 0.5 =$ uncorrelated, white noise
- $\alpha > 0.5 =$ temporally correlated
- $\alpha\ = 1 =$ 1/f-noise, pink noise
- $\alpha > 1 =$ non-stationary and unbounded
- $\alpha\ = 1.5 =$ fractional brownian motion

We recommend a few points of consideration here in using this function. One is to be sure to verify there are not cross-over points in the logScale- logFluctuation plots (Peng et al., 1995; Perakakis et al ., 2009). Cross- over points (or a visible change in the slope as a function of of scale) indicate that a mono-fractal characterization does not sufficiently characterize the data. If cross-over points are evident, we recommend proceeding to using the mfdfa() to estimate the multi-fractal fluctuation dynamics across scales.

While it is common to use only linear detrending with DFA, it is important to inspect the trends in the data to determine if it would be more appropriate to use a higher order polynomial for detrending, and/or compare the DFA output for different polynomial orders (see Kantelhardt et al., 2001).

General recommendations for choosing the min and max scale are an sc_min = 10 and sc_max = (N/4), where N is the number of observations. See Eke et al. (2002) and Gulich and Zunino (2014) for additional considerations.

Value

The object returned can take the following forms:

- If the value of verbose = 1, then a list object is returned that includes: `log_scales` the log of all included scales, `log_rms` the log root mean square error (RMS) per scale, and `alpha` the overall α estimate.
- If the value of verbose = 0, then a list containing only ‘alpha‘ the estimated scaling exponent α will be returned.

References

Eke, A., Herman, P., Kocsis, L., & Kozak, L. R. (2002). Fractal characterization of complexity in temporal physiological signals. Physiological measurement, 23(1), R1-R38.

Gulich, D., & Zunino, L. (2014). A criterion for the determination of optimal scaling ranges in DFA and MF-DFA. Physica A: Statistical Mechanics and its Applications, 397, 17-30.

Kantelhardt, J. W., Koscielny-Bunde, E., Rego, H. H., Havlin, S., & Bunde, A. (2001). Detecting long-range correlations with detrended fluctuation analysis. Physica A: Statistical Mechanics and its Applications, 295(3-4), 441-454.

Kelty-Stephen, D. G., Stirling, L. A., & Lipsitz, L. A. (2016). Multifractal temporal correlations in circle-tracing behaviors are associated with the executive function of rule-switching assessed by the Trail Making Test. Psychological assessment, 28(2), 171-180.

Peng C-K, Buldyrev SV, Havlin S, Simons M, Stanley HE, and Goldberger AL (1994), Mosaic organization of DNA nucleotides, Physical Review E, 49, 1685-1689.

Peng C-K, Havlin S, Stanley HE, and Goldberger AL (1995), Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series, Chaos, 5, 82-87.

Perakakis, P., Taylor, M., Martinez-Nieto, E., Revithi, I., & Vila, J. (2009). Breathing frequency bias in fractal analysis of heart rate variability. Biological psychology, 82(1), 82-88.

Examples

```
noise <- rnorm(5000)

scales <- c(16,32,64,128,256,512,1024)

dfa.noise.out <- dfa(
    x = noise,
    order = 1,
    verbose = 1,
    scales = scales,
    scale_ratio = 2)

pink.noise <- fgn_sim(n = 5000, H = 0.9)

dfa.pink.out <- dfa(
    x = pink.noise,
    order = 1,
    verbose = 1,
    scales = scales,
    scale_ratio = 2)

anticorr.noise <- fgn_sim(n = 5000, H = 0.25)

dfa.anticorr.out <- dfa(
    x = anticorr.noise,
    order = 1,
    verbose = 1,
    scales = scales,
    scale_ratio = 2)
```

dfa.plot *Detrended Fluctuation Plot*

Description

Plot method for monofractal detrended fluctuation analysis

Usage

```
dfa.plot(x)
```

Arguments

`x`	is an object returned from the `dfa` function of this package. Plot parameters are chosen automatically,

dlcca *Multiscale Lagged Regression Anlaysis Fast function for computing MLRA on long time series*

Description

Multiscale Lagged Regression Anlaysis Fast function for computing MLRA on long time series

Usage

```
dlcca(x, y, order, scales, lags, direction)
```

Arguments

`x`	is a real valued vector of time series data
`y`	is a real valued vector of time series data
`order`	is an integer indicating the polynomial order used for detrending the local windows
`scales`	integer vector of scales over which to compute correlation. Performance is best when scales are evenly spaced in log units. Choosing a logarithm base between 1 and 2 may also improve performance of regression.
`lags`	integer indicating the maximum number of lags to include
`direction`	string indicating a positive ('p') or negative ('n') lag

Value

The object returned from the dlcca() function is a list containing rho coefficients for each lag at each of the scales.

`fgn_sim` *Simulate fractional Gaussian Noise.*

Description

Simulate fractional Gaussian Noise.

Usage

```
fgn_sim(n = 1000, H = 0.7)
```

Arguments

`n`	integer indicating length of desired series
`H`	Hurst exponent ranges between 0 and 1

Value

A numeric vector of length n.

`fractaldata` *A whitenoise, monofractal, and multifractal timeseries*

Description

These data include three simulated data to be used for understanding the differences between the various univariate methods in the dataset to compare whitenoise, monofractal, and multifractal data.

Usage

```
data(fractaldata)
```

Format

An object of class `data.frame` with 8000 rows and 4 columns.

Source

https://www.ntnu.edu/inb/geri/software

References

Ihlen, E. A. F. (2012). Introduction to Multifractal Detrended Fluctuation Analysis in Matlab. Frontiers in Physiology, 3. https://doi.org/10.3389/fphys.2012.00141

iaafft *Iterated Amplitude Adjusted Fourier Transform*

Description

Iterated Amplitude Adjusted Fourier Transform

Usage

```
iaafft(signal, N = 1)
```

Arguments

signal	is a real valued time serires
N	is the number of desired surrogates. Default is 1

lm_c *Simplef bivariate regression written in c++*

Description

Simplef bivariate regression written in c++

Usage

```
lm_c(xs, yr)
```

Arguments

xs	a real valued column vector
yr	is a real valued column vector

`logscale` *logscale*

Description

Create logarithmically spaced scales

Usage

```
logscale(scale_min, scale_max, scale_ratio)
```

Arguments

`scale_min`	an integer indicating the minimum scale to be resovled
`scale_max`	an integer indicating the maximum scale to be resolved
`scale_ratio`	a double indicating the ratio by which scale successive scales. For example, scale_ratio = 2 would create a scales increasing by a power of 2.

Value

A vector of of logarithmically spaced scales.

Examples

```
scales <- logscale(scale_min = 16, scale_max = 1024, scale_ratio = 2)
```

`mBm_mGn` *Multifractional Brownian motion and multifractional Gaussian noise*

Description

Simulate multifractional Brownian motion and multifractional Gaussian noise.

Usage

```
mBm_mGn(N, Ht)
```

Arguments

`N`	The length of sample time series to simulate.
`Ht`	The N by 1 vector of the time evolving H(t).

Details

This is an algorithm that simulates discrete time multifractional Brownian motion and multifractional Gaussian noise, which can useful for testing various functions within the ‘fractalRegression‘ package. H(t) should take on any values between 0 and 1. It is meant to capture time varying fractal properties. The example code given below shows a slow evolving Hurst exponent involving a sinusoidal change.

Value

The object returned from the function includes:

- mBm: multifractional Brownian motion
- mGn: multifractional Gaussian noise

Examples

```
t <- 1:1024
Ht <- 0.5+0.5*(sin(0.0025*pi*t))
sim <- mBm_mGn(1024,Ht)
```

`mc_ARFIMA` *Mixed-correlated ARFIMA processes*

Description

Simulate various types of correlated noise processes.

Usage

```
mc_ARFIMA(
  process,
  n,
  rho,
  d1 = NULL,
  d2 = NULL,
  d3 = NULL,
  d4 = NULL,
  alpha = NULL,
  beta = NULL,
  delta = NULL,
  gamma = NULL,
  theta = NULL,
  theta1 = NULL,
  theta2 = NULL
)
```

Arguments

`process`	specifies the type of correlated noise process to simulate and includes 'Noise_rho', 'ARFIMA_rho','ARFIMA_AR','AR_rho', 'Mixed_ARFIMA_ARFIMA','Mixed_ARFIMA_AR',and 'Mixed_ARFIMA_noise'.
`n`	is a numeric value specifying the length of the time-series.
`rho`	specifies the strength of the correlation with values -1 - 1.
`d1`	is a numeric fractional difference parameter for x specifying long term memory.
`d2`	is a numeric fractional difference parameter for x specifying long term memory.
`d3`	is a numeric fractional difference parameter for y specifying long term memory.
`d4`	is a numeric fractional difference parameter for y specifying long term memory.
`alpha`	see Kristoufek (2013) for details.
`beta`	see Kristoufek (2013) for details.
`delta`	see Kristoufek (2013) for details.
`gamma`	see Kristoufek (2013) for details.
`theta`	see Kristoufek (2013) for details.
`theta1`	see Kristoufek (2013) for details.
`theta2`	see Kristoufek (2013) for details.

Details

This function includes multiple options simulating various types of correlated noise processes including mixed-correlated ARFIMA processes with power-law cross-correlations, These functions were originally written by Ladislav Kristoufek and posted on his website. They go with the paper presented in Kristoufek (2013). The 'process' argument specifies the type of noise to be generated.

- 'Noise_rho' - Generates two correlated noise series and requires arguments: n, rho.
- 'ARFIMA_rho' - Generates two ARFIMA processes with correlated innovations and requires arguments: n, d1, d2, rho.
- 'ARFIMA_AR' - Generates ARFIMA and AR(1) processes with correlated innovations and requires arguments: n, d1, theta, rho.
- 'AR_rho' - Generates two AR(1) processes with correlated innovations and requires arguments: n, theta1, theta2, rho.
- 'Mixed_ARFIMA_ARFIMA' - Generates MC-ARFIMA process with long-range correlation and long-range cross-correlation (Kristoufec, 2013 Model 1) and requires arguments: alpha, beta, gamma, delta, n, d1, d2, d3, d4, rho.
- 'Mixed_ARFIMA_AR' - Generates MC-ARFIMA process with long-range correlation and short-range cross-correlation (Kristoufec, 2013 Model 2) and requires arguments: alpha, beta, gamma, delta, n, d1, d2, theta, rho.
- 'Mixed_ARFIMA_noise' - Generates MC-ARFIMA process with long-range correlation and simple correlation (Kristoufec, 2013 Model 3) and requires arguments: alpha, beta, gamma, delta, n, d1, d2, rho.

Value

The object returned is a matrix of length n with a time series (x,y) in column 1 and 2.

References

Kristoufek, L. (2013). Mixed-correlated ARFIMA processes for power-law cross-correlations. Physica A: Statistical Mechanics and its Applications, 392(24), 6484-6493.

Examples

```
set.seed(987345757)

sim1 <- mc_ARFIMA(process='Mixed_ARFIMA_ARFIMA', alpha = 0.2,
beta = 1, gamma = 1, delta = 0.2, n = 10000, d1 = 0.4, d2 = 0.3,
d3 = 0.3, d4=0.4, rho=0.9)

plot(sim1[,1],type='l', ylab= "Signal Amplitude", xlab='Time',
main = "MC-ARFIMA with LRC and LRCC")

lines(sim1[,2], col='blue')
```

---

mfdfa *Multifractal Detrended Fluctuation Analysis*

Description

Fast function for computing multifractal detrended fluctuation analysis (MF-DFA), a widely used method for estimating the family of long-range temporal correlations or scaling exponents in time series data. MF-DFA is also a form of multifractal analysis that indicates the degree of interaction across temporal scales.

Usage

```
mfdfa(x, q, order, scales, scale_ratio)
```

Arguments

`x`	A real valued vector (i.e., time series data) to be analyzed.
`q`	A real valued vector indicating the statistical moments (q) to use in the analysis. q must span negative and positive values e.g., -3:3, otherwise and error may be produced.
`order`	is an integer indicating the polynomial order used for detrending the local windows (e.g, 1 = linear, 2 = quadratic, etc.). There is not pre-determined limit on the order of the polynomial order but the user should avoid using a large polynomial on small windows. This can result in overfitting and non-meaningful estimates.

`scales`	An integer valued vector indicating the scales one wishes to resolve in the analysis. Best practice is to use scales which are evenly spaced in the logarithmic domain e.g., `scales = 2^(4:(N/4))`, where N is the length of the time series. Other, logarithmic bases may also be used to give finer resolution of scales while maintaining ~= spacing in the log domain e.g, `scales = unique(floor(1.1^(30:(N/4))))`. Note that fractional bases may produce duplicate values after the necessary floor function.
`scale_ratio`	A scaling factor by which successive window sizes were created. The default is 2 but should be addressed according to how scales were generated for example using `logscale(16, 100, 1.1)`, where 1.1 is the scale ratio.

Details

Details of the algorithm are specified in detail in Kantelhardt et al. (2001; 2002) and visualized nicely in Kelty-Stephen et al. (2016).

Selecting the range of values for q is important. Note that MF-DFA estimates for q = 2 are equivalent to DFA. Larger values of q (q > 2) emphasize larger residuals and smaller values of q (q < 2) emphasis smaller residuals (Kelty-Stephen et al., 2016). For most biomedical signals such as physiological and kinematic, a q range of -5 to 5 is common (Ihlen, 2010). However, in some cases, such as when time series are short (< 3000), it can be appropriate to limit the range of q to positive only. Kelty-Stephen et al. (2016) recommend a positive q range of 0.5 to 10 with an increment of 0.5.

While it is common to use only linear detrending with DFA and MF-DFA, it is important to inspect the trends in the data to determine if it would be more appropriate to use a higher order polynomial for detrending, and/or compare the DFA and MF-DFA output for different polynomial orders (see Ihlen, 2012; Kantelhardt et al., 2001).

General recommendations for choosing the min and max scale are a scale_min = 10 and scale_max = (N/4), where N is the number of observations. See Eke et al. (2002), Gulich and Zunino (2014), Ihlen (2012), and for additional considerations and information on choosing the correct parameters.

Value

The output of the algorithm is a list that includes:

- `log_scale` The log scales used for the analysis
- `log_fq` The log of the fluctuation functions for each scale and q
- `Hq` The q-order Hurst exponent (generalized Hurst exponent)
- `Tau` The q-order mass exponent
- `q` The q-order statistical moments
- `h` The q-order singularity exponent
- `Dh` The dimension of the q-order singularity exponent

References

Ihlen, E. A. F. (2012). Introduction to Multifractal Detrended Fluctuation Analysis in Matlab. Frontiers in Physiology, 3. https://doi.org/10.3389/fphys.2012.00141

Kantelhardt, J. W., Koscielny-Bunde, E., Rego, H. H., Havlin, S., & Bunde, A. (2001). Detecting long-range correlations with detrended fluctuation analysis. Physica A: Statistical Mechanics and its Applications, 295(3-4), 441-454.

Kantelhardt, J. W., Zschiegner, S. A., Koscielny-Bunde, E., Havlin, S., Bunde, A., & Stanley, H. E. (2002). Multifractal detrended fluctuation analysis of nonstationary time series. Physica A: Statistical Mechanics and its Applications, 316(1-4), 87-114.

Kelty-Stephen, D. G., Palatinus, K., Saltzman, E., & Dixon, J. A. (2013). A Tutorial on Multifractality, Cascades, and Interactivity for Empirical Time Series in Ecological Science. Ecological Psychology, 25(1), 1-62. https://doi.org/10.1080/10407413.2013.753804

Kelty-Stephen, D. G., Stirling, L. A., & Lipsitz, L. A. (2016). Multifractal temporal correlations in circle-tracing behaviors are associated with the executive function of rule-switching assessed by the Trail Making Test. Psychological Assessment, 28(2), 171-180. https://doi.org/10.1037/pas0000177

Examples

```
noise <- rnorm(5000)

scales <- c(16,32,64,128,256,512,1024)

mf.dfa.white.out <- mfdfa(
    x = noise, q = c(-5:5),
    order = 1,
    scales = scales,
    scale_ratio = 2)

pink.noise <- fgn_sim(n = 5000, H = 0.9)

mf.dfa.pink.out <- mfdfa(
    x = pink.noise,
    q = c(-5:5),
    order = 1,
    scales = scales,
    scale_ratio = 2)
```

`mfdfa.plot` *Multifractal Spectrum Plot*

Description

Method for plotting various forms of the multifractal spectrum

Usage

```
mfdfa.plot(mf, do.surrogate, nsurrogates = 19, return.ci = FALSE)
```

Arguments

`mf`	an object containing elements related to the mutlifractal spectrum derived from Multifractal Detrended Fluctuation Analysis
`do.surrogate`	logical indicating whether surrogation should be performed on the time series
`nsurrogates`	integer indicating the number of surrogates to be constructed. Default is 19 for 95 surrogates ore more precise but increase computational time.
`return.ci`	logical indicating if confidence intervals derived from surrogate analysis should be returned.

Author(s)

Aaron D. Likens (2022)

References

Kantelhardt et al. (2002). Multifractal detrended fluctuation analys of nonstationary time series. Physica A: Statistical Mechanics and its Applications, 87

`mfdfa_cj` *Multifractal Analysis Chhabra-Jensen Method*

Description

Fast function for computing multifractal analysis using a lesser-known method for estimating the family of long-range temporal correlations or scaling exponents in time series data. This is also a form of multifractal analysis that indicates the degree of interaction across temporal scales.

Usage

```
mfdfa_cj(Timeseries, qValues, scales)
```

Arguments

`Timeseries`	is a real valued time series
`qValues`	real valued vector of q-orders
`scales`	unsigned integer vector of scales to be resolved

mlra *Multiscale Lagged Regression Analysis*

Description

Fast function for computing multiscale lagged regression analysis (MLRA) on long time series. Combining DFA with ordinary least square regression, MLRA is a form of fractal regression that can be used to estimate asymmetric and multiscale regression coefficients between two variables at different time-scales and temporal lags.

Usage

```
mlra(x, y, order, scales, lags, direction)
```

Arguments

`x`	A real valued vector (i.e., time series data) to be analyzed.
`y`	A real valued vector (i.e., time series data) to be analyzed.
`order`	is an integer indicating the polynomial order used for detrending the local windows (e.g, 1 = linear, 2 = quadratic, etc.). There is a not pre-determined limit on the order of the polynomial order but the user should avoid using a large polynomial on small windows. This can result in overfitting and non-meaningful estimates.
`scales`	An integer vector of scales over which to compute correlation. Unlike univariate DFA, MRA does not require that scales be in log units. Scale intervals can be sequential, for example, when the analysis is exploratory and no a priori hypotheses have been made about the scale of correlation. A small subset of targeted scales may also be investigated where scale-specific research questions exist. We have found that windows smaller than say 8 observations create stability problems due to overfitting. This is espcially when the order of the fitting polynomial is large.
`lags`	An integer indicating the maximum number of lags to include in the analysis.
`direction`	A character string indicating a positive ('p') or negative ('n') lag.

Details

Mathematical treatment of the MLRA algorithm and its performance can be found in Kristoufek (2015) and Likens et al. (2019).

Use of the direction parameter specifies whether the scale-wise β coefficients for positive or negative lags will be estimated.

Note that under conditions with linear and quadratic trends, Likens et al. (2019) found that there was a systematic positive bias in the β estimates for larger scales. Using a polynomial detrending order of 2 or greater was shown to attenuate this bias.

Value

The object returned from the mlra() function is a list containing `betas` the β coefficients for each lag at each of the scales.

References

Kristoufek, L. (2015). Detrended fluctuation analysis as a regression framework: Estimating dependence at different scales. Physical Review E, 91(2), 022802.

Likens, A. D., Amazeen, P. G., West, S. G., & Gibbons, C. T. (2019). Statistical properties of Multiscale Regression Analysis: Simulation and application to human postural control. Physica A: Statistical Mechanics and its Applications, 532, 121580.

Examples

```
# Here is a simple example for running MLRA using a white noise and pink noise time series.
# For more detailed examples, see the vignette.

noise <- rnorm(5000)

pink.noise <- fgn_sim(n = 5000, H = 0.9)

scales <- logscale(scale_min = 10, scale_max = 1250, scale_ratio = 1.1)

mlra.out <- mlra(
    x = noise,
    y = pink.noise,
    order = 1,
    scales = scales,
    lags = 100, direction = 'p')
```

`mra` *Multiscale Regression Analysis (MRA)*

Description

Fast function for computing multiscale regression analysis (MRA) on long time series. Combining DFA with ordinary least square regression, MRA is a form of fractal regression that can be used to estimate asymmetric and multiscale regression coefficients between two variables.

Usage

```
mra(x, y, order, scales)
```

Arguments

`x` A real valued vector (i.e., time series data) to be analyzed. A key difference between DCCA and MRA is that MRA produces asymmetric estiamtes. That is, x is assumed to be an independent variable and y is assumed to be a dependent variable. MRA should be used when one of the time series in question is usefully cast as the independent variable. That is, x is assumed to effect change in y. If no such causal relationship is anticipated, use DCCA instead.

`y` A real valued vector (i.e., time series data) to be analyzed.

`order` is an integer indicating the polynomial order used for detrending the local windows (e.g, 1 = linear, 2 = quadratic, etc.). There is not a pre-determined limit on the order of the polynomial order but the user should avoid using a large polynomial on small windows. This can result in overfitting and non-meaningful estimates.

`scales` An integer vector of scales over which to compute correlation. Unlike univariate DFA, MRA does not require that scales be in log units. Scale intervals can be sequential, for example, when the analysis is exploratory and no a priori hypotheses have been made about the scale of correlation. A small subset of targeted scales may also be investigated where scale-specific research questions exist. We have found that windows smaller than say 8 observations create stability problems due to overfitting. This is espcially when the order of the fitting polynomial is large.

Details

Mathematical treatment of the MRA algorithm and its performance can be found in Kristoufek (2015) and Likens et al. (2019).

Note that under conditions with linear and quadratic trends, Likens et al. (2019) found that there was a systematic positive bias in the β estimates for larger scales. Using a polynomial detrending order of 2 or greater was shown to attenuate this bias.

Value

The object returned from the mra() function is a list including the following:

- `scales` indicates the values of the scales used for estimates
- `betas` are the scale specific β estimates of the influence of x on y
- `r2` is the scale specific r-squared value of the model fit (i.e., variance in y accounted for by x at that scale)
- `t_observed` is the estimated t-statistic for a given β at a given scale.

References

Kristoufek, L. (2015). Detrended fluctuation analysis as a regression framework: Estimating dependence at different scales. Physical Review E, 91(2), 022802.

Likens, A. D., Amazeen, P. G., West, S. G., & Gibbons, C. T. (2019). Statistical properties of Multiscale Regression Analysis: Simulation and application to human postural control. Physica A: Statistical Mechanics and its Applications, 532, 121580.

Examples

```
# Here is a simple example for running MRA using a white noise and pink noise time series.
# For more detailed examples, see the vignette.

noise <- rnorm(5000)

pink.noise <- fgn_sim(n = 5000, H = 0.9)

scales <- logscale(scale_min = 10, scale_max = 1250, scale_ratio = 1.1)

mra.out <- mra(x = noise, y = pink.noise, order = 1, scales = scales)
```

`mra.plot` *Multiscale Regression Plot*

Description

A plotting method for constructing scalewise regression plot

Usage

```
mra.plot(
  betas,
  order = 1,
  ci = FALSE,
  iterations = NULL,
  return.ci = FALSE,
  loess.beta = FALSE,
  loess.ci = FALSE
)
```

Arguments

`betas`	an object containing modeling results from multiscale regression analysis. The object should be returned from the `mra` function of this package.
`order`	integer representing the detrending order used in the `mra` calculation. Default is 1.
`ci`	a logical indicating whether confidence intervals should be computed using the `iaafft` function from this package. NOTE: with long time series (» than N = 1,000), this can greatly reduce processing speed. Confidence intervals can be used for conventional significance testing of scale-wise correlation coefficients.

`iterations`	integer that specifies the the number of surrogate time series to be generated for the purpose of confidence intervals. Default = 19. Larger number of surrogates will slow computational speed but produce better confidence interval estimates.
`return.ci`	logical indicating whether the confidence intervals should be returned
`loess.beta`	logical indicating whether a loess fit should be used for displaying multiscale regression coefficient trajectories
`loess.ci`	logical indicating whether a loess fit should be used to smooth confidence intervals

---

`poly_residuals` *Polynomial Residuals Function that fits a polynomial and returns the residuals*

Description

Polynomial Residuals Function that fits a polynomial and returns the residuals

Usage

```
poly_residuals(yr, m)
```

Arguments

`yr`	is a real valued vector
`m`	is the detrending order

`seq_int` *Integer Sequence Function that produces a sequence of integers from 1 to N*

Description

Integer Sequence Function that produces a sequence of integers from 1 to N

Usage

```
seq_int(length)
```

Arguments

`length`	is a positive integer that will produce a sequence from 1:length

---

`seq_int_range` *Sequence of Integer ranges Function that produces a sequece of integers that span a specific range*

Description

Sequence of Integer ranges Function that produces a sequece of integers that span a specific range

Usage

```
seq_int_range(start, stop)
```

Arguments

`start`	is a positive integer and gives the smallest value in the sequence
`stop`	is a positive integer and gives the largest value in a sequence

Index