Package ‘wavemulcor’

October 12, 2022

Type Package

Title Wavelet Routines for Global and Local Multiple Regression and Correlation

Version 3.1.2

Description Wavelet routines that calculate single sets of wavelet multiple regressions and correlations, and cross-regressions and cross-correlations from a multivariate time series. Dynamic versions of the routines allow the wavelet local multiple (cross-)regressions and (cross-)correlations to evolve over time.

License GPL-3

Depends R (>= 3.4.0), waveslim (>= 1.7.5)

Imports plot3D, RColorBrewer

Suggests covr, knitr, markdown, rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Author Javier Fernandez-Macho [aut, cre] (<https://orcid.org/0000-0002-5970-4382>)

Maintainer Javier Fernandez-Macho <javier.fernandezmacho@ehu.es>

Repository CRAN

Date/Publication 2021-09-03 12:40:02 UTC

R topics documented:

`wavemulcor-package` *Wavelet Routines for Global and Local Multiple Regression and Correlation*

Description

Wavelet routines that calculate single sets of wavelet multiple regressions and correlations, and cross-regressions and cross-correlations from a multivariate time series. Dynamic versions of the routines allow the wavelet local multiple (cross-)regressions and (cross-)correlations to evolve over time.

Details

Wavelet routines that calculate single sets of wavelet multiple regressions and correlations (WMR and WMC), and cross-regressions and cross-correlations (WMCR and WMCC) from a multivariate time series. Dynamic versions of the routines allow the wavelet local multiple (cross-)regressions (WLMR and WLMCR) and (cross-)correlations (WLMC and WLMCC) to evolve over time. The output from these Wavelet statistics can later be plotted in single graphs, as an alternative to trying to make sense out of several sets of wavelet correlations or wavelet cross-correlations. The code is based on the calculation, at each wavelet scale, of the square root of the coefficient of determination in a linear combination of variables for which such coefficient of determination is a maximum. The code provided here is based on the wave.correlation routine in Brandon Whitcher's waveslim R package Version: 1.6.4, which in turn is based on wavelet methodology developed in Percival and Walden (2000), Gençay, Selçuk and Whitcher (2002) and others. Version 2 incorporates wavelet local multiple correlations (WLMC). These are like the previous global WMC but consisting in one single set of multiscale correlations along time. That is, at each time t, they are calculated by letting a window of weighted wavelet coefficients around t move along time. Six weight functions are provided. Namely, the uniform window, Cleveland's tricube window, Epanechnikov's parabolic window, Bartlett's triangular window and Wendland's truncated power window and the Gaussian window. Version 2.2 incorporates auxiliary functions that calculate local multiple correlations and cross-correlations (LMC, LMCC). They are calculated by letting move along time a window of weighted time series values around t. Any of the six weight functions mentioned above can be used. They also feed a new routine to compute wavelet local multiple cross-correlation (WLMCC). Version 3 extends all the previous correlation routines (WMC, WMCC, LMC, WLMC, WLMCC) to handle wavelet regressions (WMR, WMCR, LMR, WLMR, WLMCR) that provide regression coefficients and statistics across wavelet scales. Auxiliary plot_ and heatmap_ routines are also provided to visualize the wavmulcor statistics.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2012. Wavelet multiple correlation and cross-correlation: A multiscale analysis of Eurozone stock markets. Physica A: Statistical Mechanics and its Applications 391, 1097–1104. <DOI:10.1016/j.physa.2011.11.002>

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

`heatmap_wave.local.multiple.correlation` *Auxiliary routine for heatmaping wave local multiple correlations*

Description

Produces a heatmap of wave local multiple correlations.

Usage

```
heatmap_wave.local.multiple.correlation(Lst, xaxt="s", ci=NULL, pdf.write=NULL)
```

Arguments

`Lst`	A list from wave.local.multiple.regression.
`xaxt`	An optional vector of labels for the "x" axis. Default is 1:n.
`ci`	value to plot: "center" value of confidence interval (*i.e.* the estimated correlation), the "lower" bound, or the "upper" bound. Default is "center".
`pdf.write`	Optional name leader to save files to pdf format. The actual name of the file is "heat_<pdf.write>_WLMC.pdf".

Details

The routine produces a time series vs. wavelet periods heatmap of wave local multiple correlations.

Value

Heat map.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

`heatmap_wave.local.multiple.cross.correlation` *Auxiliary routine for heatmaping wave local multiple cross-correlations*

Description

Produces heatmaps of wave local multiple cross-correlations.

Usage

```
heatmap_wave.local.multiple.cross.correlation(Lst, lmax,
     lag.first=FALSE, xaxt="s", ci=NULL, pdf.write=NULL)
```

Arguments

- `Lst` A list from wave.local.multiple.cross.regression.
- `lmax` maximum lag (and lead).
- `lag.first` if TRUE, it produces lag-lead pages with $J+1$ wavelet heatmaps each. Otherwise (default) it gives wavelet pages with $2*lmax+1$ lag-lead heatmaps each.
- `xaxt` An optional vector of labels for the "x" axis. Default is 1:n.
- `ci` value to plot: "center" value of confidence interval (*i.e.* the estimated cross-correlation), the "lower" bound, or the "upper" bound. Default is "center".
- `pdf.write` Optional name leader to save files to pdf format. The actual name of the file is either "heat_<pdf.write>_WLMCC_lags.pdf" or, "heat_<pdf.write>_WLMCC_levels.pdf".

Details

The routine produces a set of time series vs. wavelet periods heatmaps of wave local multiple cross-correlations at different lags and leads.

Value

Heat map.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

`heatmap_wave.multiple.cross.correlation`

Auxiliary routine for heatmaping wave multiple cross-correlations

---

Description

Produces heatmaps of wave multiple cross-correlations.

Usage

```
heatmap_wave.multiple.cross.correlation(Lst, lmax, by=3, ci=NULL, pdf.write=NULL)
```

Arguments

- `Lst` — A list from wave.multiple.cross.regression or wave.multiple.cross.correlation.
- `lmax` — maximum lag (and lead).
- `by` — labels are printed every lmax/by. Default is 3.
- `ci` — value to plot: "center" value of confidence interval (*i.e.* the estimated cross-correlation), the "lower" bound, or the "upper" bound. Default is "center".
- `pdf.write` — Optional name leader to save files to pdf format. The actual name of the file is either "heat_<pdf.write>_WLMCC_lags.pdf" or, "heat_<pdf.write>_WLMCC_levels.pdf".

Details

The routine produces a set of time series vs. wavelet periods heatmaps of wave local multiple cross-correlations at different lags and leads.

Value

Heat map.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2012. Wavelet multiple correlation and cross-correlation: A multiscale analysis of Eurozone stock markets. Physica A: Statistical Mechanics and its Applications 391, 1097–1104. <DOI:10.1016/j.physa.2011.11.002>

`local.multiple.correlation` *Routine for local multiple correlation*

Description

Produces an estimate of local multiple correlations (as defined below) along with approximate confidence intervals.

Usage

```
local.multiple.correlation(xx, M, window="gauss", p = .975, ymaxr=NULL)
```

Arguments

`xx`	A list of n time series, *e.g.* xx <- list(v1, v2, v3)
`M`	length of the weight function or rolling window.
`window`	type of weight function or rolling window. Six types are allowed, namely the uniform window, Cleveland or tricube window, Epanechnikov or parabolic window, Bartlett or triangular window, Wendland window and the gaussian window. The letter case and length of the argument are not relevant as long as at least the first four characters are entered.
`p`	one minus the two-sided p-value for the confidence interval, *i.e.* the cdf value.
`ymaxr`	index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level lmc chooses the one maximizing the multiple correlation.

Details

The routine calculates a time series of multiple correlations out of n variables. The code is based on the calculation of the square root of the coefficient of determination in that linear combination of locally weighted values for which such coefficient of determination is a maximum.

Value

List of four elements:

`val:`	numeric vector (rows = #observations) providing the point estimates for the local multiple correlation.
`lo:`	numeric vector (rows = #observations) providing the lower bounds of the confidence interval.
`up:`	numeric vector (rows = #observations) providing the upper bounds of the confidence interval.
`YmaxR:`	numeric vector (rows = #observations) giving, at each value in time, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, *lmc* chooses at each value in time the variable maximizing the multiple correlation.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

Examples

```
## Based on  Figure 4 showing correlation structural breaks in Fernandez-Macho (2018).

library(wavemulcor)
options(warn = -1)

xrand1 <- wavemulcor::xrand1
xrand2 <- wavemulcor::xrand2
N <- length(xrand1)
b <- trunc(N/3)
t1 <- 1:b
t2 <- (b+1):(2*b)
t3 <- (2*b+1):N

wf <- "d4"
M <- N/2^3 #sharper with N/2^4
window <- "gaussian"

J <- trunc(log2(N))-3

# ---------------------------

cor1 <- cor(xrand1[t1],xrand2[t1])
cor2 <- cor(xrand1[t2],xrand2[t2])
cor3 <- cor(xrand1[t3],xrand2[t3])
cortext <- paste0(round(100*cor1,0),"-",round(100*cor2,0),"-",round(100*cor3,0))

ts.plot(cbind(xrand1,xrand2),col=c("red","blue"),xlab="time")

xx <- data.frame(xrand1,xrand2)

# ---------------------------

xy.mulcor <- local.multiple.correlation(xx, M, window=window)

val <- as.matrix(xy.mulcor$val)
lo  <- as.matrix(xy.mulcor$lo)
up  <- as.matrix(xy.mulcor$up)
YmaxR <- as.matrix(xy.mulcor$YmaxR)

# ---------------------------

old.par <- par()

# ##Producing line plots with CI

title <- paste("Local Multiple Correlation")
sub <- paste("first",b,"obs:",round(100*cor1,1),"% correlation;","middle",b,"obs:",
             round(100*cor2,1),"%","rest:",round(100*cor3,1),"%")
xlab <- "time"
ylab <- "correlation"
```

```
matplot(1:N,cbind(val,lo,up),
        main=title, sub=sub,
        xlab=xlab, ylab=ylab, type="l", lty=1, col= c(1,2,2), cex.axis=0.75)
abline(h=0) ##Add Straight horiz and vert Lines to a Plot

#reset graphics parameters
par(old.par)
```

`local.multiple.cross.correlation`
Routine for local multiple cross-correlation

Description

Produces an estimate of local multiple cross-correlations (as defined below) along with approximate confidence intervals.

Usage

```
local.multiple.cross.correlation(xx, M, window="gauss", lag.max=NULL, p=.975, ymaxr=NULL)
```

Arguments

`xx`	A list of n time series, *e.g.* xx <- list(v1, v2, v3)
`M`	length of the weight function or rolling window.
`window`	type of weight function or rolling window. Six types are allowed, namely the uniform window, Cleveland or tricube window, Epanechnikov or parabolic window, Bartlett or triangular window, Wendland window and the gaussian window. The letter case and length of the argument are not relevant as long as at least the first four characters are entered.
`lag.max`	maximum lag (and lead). If not set, it defaults to half the square root of the length of the original series.
`p`	one minus the two-sided p-value for the confidence interval, *i.e.* the cdf value.
`ymaxr`	index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level lmc chooses the one maximizing the multiple correlation.

Details

The routine calculates a set of time series of multiple cross-correlations, one per lag and lead) out of n variables.

Value

List of four elements:

- `vals:` numeric matrix (rows = #observations, cols = #lags and leads) providing the point estimates for the local multiple cross-correlation.
- `lower:` numeric vmatrix (rows = #observations, cols = #lags and leads) providing the lower bounds from the confidence interval.
- `upper:` numeric matrix (rows = #observations, cols = #lags and leads) providing the upper bounds from the confidence interval.
- `YmaxR:` numeric matrix (rows = #observations, cols = #lags and leads) giving, at each value in time, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, *lmcc* chooses at each value in time the variable maximizing the multiple correlation.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

Examples

```
## Based on  Figure 4 showing correlation structural breaks in Fernandez-Macho (2018).

library(wavemulcor)

data(exchange)
returns <- diff(log(as.matrix(exchange)))
returns <- ts(returns, start=1970, freq=12)
N <- dim(returns)[1]

M <- 30
window <- "gauss"
lmax <- 1

demusd <- returns[,"DEM.USD"]
jpyusd <- returns[,"JPY.USD"]

set.seed(140859)

xrand <- rnorm(N)

xx <- data.frame(demusd, jpyusd, xrand)
##exchange.names <- c(colnames(returns), "RAND")
```

```
Lst <- local.multiple.cross.correlation(xx, M, window=window, lag.max=lmax)
val <- Lst$vals
low.ci <- Lst$lower
upp.ci <- Lst$upper
YmaxR <- Lst$YmaxR

# ---------------------------

##Producing correlation plot

colnames(val) <- paste("Lag",-lmax:lmax)
xvar <- seq(1,N,M)
par(mfcol=c(lmax+1,2), las=1, pty="m", mar=c(2,3,1,0)+.1, oma=c(1.2,1.2,0,0))
ymin <- -0.1
if (length(xx)<3) ymin <- -1
for(i in c(-lmax:0,lmax:1)+lmax+1) {
  matplot(1:N,val[,i], type="l", lty=1, ylim=c(ymin,1), #xaxt="n",
          xlab="", ylab="", main=colnames(val)[i])
  # if(i==lmax+1) {axis(side=1, at=seq(0,N+50,50))}
  #axis(side=2, at=c(-.2, 0, .5, 1))
  abline(h=0)              ##Add Straight horiz
  lines(low.ci[,i], lty=1, col=2) ##Add Connected Line Segments to a Plot
  lines(upp.ci[,i], lty=1, col=2)
  text(xvar,1, labels=names(xx)[YmaxR][xvar], adj=0.25, cex=.8)
}
par(las=0)
mtext('time', side=1, outer=TRUE, adj=0.5)
mtext('Local Multiple Cross-Correlation', side=2, outer=TRUE, adj=0.5)
```

---

local.multiple.cross.regression

Routine for local multiple cross-regression

Description

Produces an estimate of local multiple cross-regressions (as defined below) along with approximate confidence intervals.

Usage

```
local.multiple.cross.regression(xx, M, window="gauss", lag.max=NULL, p=.975, ymaxr=NULL)
```

Arguments

xx	A list of n time series, *e.g.* xx <- list(v1, v2, v3)
M	length of the weight function or rolling window.

`window`	type of weight function or rolling window. Six types are allowed, namely the uniform window, Cleveland or tricube window, Epanechnikov or parabolic window, Bartlett or triangular window, Wendland window and the gaussian window. The letter case and length of the argument are not relevant as long as at least the first four characters are entered.
`lag.max`	maximum lag (and lead). If not set, it defaults to half the square root of the length of the original series.
`p`	one minus the two-sided p-value for the confidence interval, *i.e.* the cdf value.
`ymaxr`	index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level lmc chooses the one maximizing the multiple correlation.

Details

The routine calculates a set of time series of multiple cross-regressions, one per lag and lead) out of n variables.

Value

List of four elements:

`cor:` List of three elements:

- vals: numeric matrix (rows = #observations, cols = #lags and leads) providing the point estimates for the local multiple cross-correlation.
- lower: numeric vmatrix (rows = #observations, cols = #lags and leads) providing the lower bounds from the confidence interval.
- upper: numeric matrix (rows = #observations, cols = #lags and leads) providing the upper bounds from the confidence interval.

`reg:` List of seven elements:

- rval: numeric array (1st_dim = #observations, 2nd_dim = #lags and leads, 3rd_dim = #regressors+1) of local regression estimates.
- rstd: numeric array (1st_dim = #observations, 2nd_dim = #lags and leads, 3rd_dim = #regressors+1) of their standard deviations.
- rlow: numeric array (1st_dim = #observations, 2nd_dim = #lags and leads, 3rd_dim = #regressors+1) of their lower bounds.
- rupp: numeric array (1st_dim = #observations, 2nd_dim = #lags and leads, 3rd_dim = #regressors+1) of their upper bounds.
- rtst: numeric array (1st_dim = #observations, 2nd_dim = #lags and leads, 3rd_dim = #regressors+1) of their t statistic values.
- rord: numeric array (1st_dim = #observations, 2nd_dim = #lags and leads, 3rd_dim = #regressors+1) of their index order when sorted by significance.
- rpva: numeric array (1st_dim = #observations, 2nd_dim = #lags and leads, 3rd_dim = #regressors+1) of their p values.

`YmaxR:` numeric matrix (rows = #observations, cols = #lags and leads) giving, at each value in time, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, *lmcr* chooses at each value in time the variable maximizing the multiple correlation.

`data:` dataframe (rows = #observations, cols = #regressors) of original data.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

Examples

```
## Based on Figure 4 showing correlation structural breaks in Fernandez-Macho (2018).

library(wavemulcor)
data(exchange)
returns <- diff(log(as.matrix(exchange)))
returns <- ts(returns, start=1970, freq=12)
N <- dim(returns)[1]

M <- 30
window <- "gauss"
lmax <- 1

demusd <- returns[,"DEM.USD"]
jpyusd <- returns[,"JPY.USD"]

set.seed(140859)

xrand <- rnorm(N)

# ---------------------------

xx <- data.frame(demusd, jpyusd, xrand)
##exchange.names <- c(colnames(returns), "RAND")

Lst <- local.multiple.cross.regression(xx, M, window=window, lag.max=lmax)

# ---------------------------

##Producing correlation plot

plot_local.multiple.cross.correlation(Lst, lmax) #, xaxt="s")

##Producing regression plot
```

```
plot_local.multiple.cross.regression(Lst, lmax) #, nsig=2, xaxt="s")
```

`local.multiple.regression`

Routine for local multiple regression

Description

Produces an estimate of local multiple regressions (as defined below) along with approximate confidence intervals.

Usage

```
local.multiple.regression(xx, M, window="gauss", p=.975, ymaxr=NULL)
```

Arguments

`xx`	A list of n time series, *e.g.* xx <- list(v1, v2, v3)
`M`	length of the weight function or rolling window.
`window`	type of weight function or rolling window. Six types are allowed, namely the uniform window, Cleveland or tricube window, Epanechnikov or parabolic window, Bartlett or triangular window, Wendland window and the gaussian window. The letter case and length of the argument are not relevant as long as at least the first four characters are entered.
`p`	one minus the two-sided p-value for the confidence interval, *i.e.* the cdf value.
`ymaxr`	index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level lmc chooses the one maximizing the multiple correlation.

Details

The routine calculates a set of time series of multiple regression coefficients out of n variables.

Value

List of four elements:

`cor:` List of three elements:

- val: numeric vector (rows = #observations) of point estimates for the local multiple correlation.
- lo: numeric vector (rows = #observations) of lower bounds of the confidence interval.
- up: numeric vector (rows = #observations) of upper bounds of the confidence interval.

`reg:` List of seven elements:

- rval: numeric matrix (rows = #observations, cols = #regressors+1) of local regression estimates.
- rstd: numeric matrix (rows = #observations, cols = #regressors+1) of their standard deviations.
- rlow: numeric matrix (rows = #observations, cols = #regressors+1) of their lower bounds.
- rupp: numeric matrix (rows = #observations, cols = #regressors+1) of their upper bounds.
- rtst: numeric matrix (rows = #observations, cols = #regressors+1) of their t statistic values.
- rord: numeric matrix (rows = #observations, cols = #regressors+1) of their index order when sorted by significance.
- rpva: numeric matrix (rows = #observations, cols = #regressors+1) of their p values.

`YmaxR:` numeric vector (rows = #observations) giving, at each value in time, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, *lmr* chooses at each value in time the variable maximizing the multiple correlation.

`data:` dataframe (rows = #observations, cols = #regressors) of original data.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

Examples

```
## Based on  Figure 4 showing correlation structural breaks in Fernandez-Macho (2018).

library(wavemulcor)
options(warn = -1)

xrand1 <- wavemulcor::xrand1
xrand2 <- wavemulcor::xrand2
N <- length(xrand1)
b <- trunc(N/3)
t1 <- 1:b
t2 <- (b+1):(2*b)
t3 <- (2*b+1):N

wf <- "d4"
M <- N/2^3 #sharper with N/2^4
window <- "gaussian"

J <- trunc(log2(N))-3

# ---------------------------
```

```
cor1 <- cor(xrand1[t1],xrand2[t1])
cor2 <- cor(xrand1[t2],xrand2[t2])
cor3 <- cor(xrand1[t3],xrand2[t3])
cortext <- paste0(round(100*cor1,0),"-",round(100*cor2,0),"-",round(100*cor3,0))

ts.plot(cbind(xrand1,xrand2),col=c("red","blue"),xlab="time")

xx <- data.frame(xrand1,xrand2)

# ---------------------------

Lst <- local.multiple.regression(xx, M, window=window) #, ymax=1)

# ---------------------------

##Producing correlation plot
plot_local.multiple.correlation(Lst)

##Producing regression plot
plot_local.multiple.regression(Lst)
```

---

plot_local.multiple.correlation

Auxiliary routine for plotting local multiple correlations

Description

Produces a plot of local multiple correlations.

Usage

```
plot_local.multiple.correlation(Lst, xaxt="s")
```

Arguments

`Lst`	A list from local.multiple.regression or local.multiple.correlation.
`xaxt`	An optional vector of labels for the "x" axis. Default is 1:n.

Details

The routine produces a time series plot of local multiple correlations with its confidence interval. Also, at every upturn and downturn, the name of the variable that maximizes its multiple correlation against the rest is shown. Note that the routine is optimize for *local.multiple.regression*. If you want to use output from the legacy *local.multiple.correlation* function then you must create an empty list and put that output into a list element named *cor* like this: Lst <- list(); Lst$cor <- local.multiple.cross.correlation(xx, M, window=window, lag.max=lmax); Lst$YmaxR <- Lst2$cor$YmaxR; Lst$cor$YmaxR <- NULL.

Value

Plot.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

---

plot_local.multiple.cross.correlation

Auxiliary routine for plotting local multiple cross-correlations

---

Description

Produces a plot of local multiple cross-correlations.

Usage

```
plot_local.multiple.cross.correlation(Lst, lmax, xaxt="s")
```

Arguments

`Lst`	A list from local.multiple.cross.regression or local.multiple.cross.correlation.
`lmax`	maximum lag (and lead).
`xaxt`	An optional vector of labels for the "x" axis. Default is 1:n.

Details

The routine produces a set of time series plots of local multiple cross-correlations, one per lag and lead, each with its confidence interval. Also, at every upturn and downturn, the name of the variable that maximizes its multiple correlation against the rest is shown. Note that the routine is optimize for *local.multiple.cross.regression*. If you want to use output from *local.multiple.cross.correlation* function then you must create an empty list and put that output into a list element named *cor* like this: Lst <- list(); Lst$cor <- local.multiple.cross.correlation(xx, M, window=window, lag.max=lmax); Lst$YmaxR <- Lst2$cor$YmaxR; Lst$cor$YmaxR <- NULL.

Value

Plot.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

`plot_local.multiple.cross.regression` *Auxiliary routine for plotting local multiple cross-regressions*

Description

Produces a plot of local multiple cross-regressions.

Usage

```
plot_local.multiple.cross.regression(Lst, lmax, nsig=2, xaxt="s")
```

Arguments

`Lst`	A list from local.multiple.cross.regression.
`lmax`	maximum lag (and lead).
`nsig`	An optional value for the number of significant variables to plot_ Default is 2.
`xaxt`	An optional vector of labels for the "x" axis. Default is 1:n.

Details

The routine produces time series plots of local multiple cross-regressions with their confidence interval for every lag and lead. Also, at every upturn and downturn of the corresponding local multiple cross-correlation, the name of the variable that maximizes that multiple correlation against the rest is shown on top. The others are named ordered by significance when they are relevant.

Value

Plot.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

`plot_local.multiple.regression`
Auxiliary routine for plotting local multiple regressions

Description

Produces a plot of local multiple regressions.

Usage

```
plot_local.multiple.regression(Lst, nsig=2, xaxt="s")
```

Arguments

`Lst`	A list from local.multiple.regression.
`nsig`	An optional value for the number of significant variables to plot_ Default is 2.
`xaxt`	An optional vector of labels for the "x" axis. Default is 1:n.

Details

The routine produces a time series plot of local multiple regressions with its confidence interval. Also, at every upturn and downturn of the corresponding local multiple correlation, the name of the variable that maximizes that multiple correlation against the rest is shown on top. The others are named ordered by significance when they are relevant.

Value

Plot.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

`plot_wave.local.multiple.correlation`
Auxiliary routine for plotting wave local multiple correlations

Description

Produces a plot of wave local multiple correlations.

Usage

```
plot_wave.local.multiple.correlation(Lst, xaxt="s")
```

Arguments

`Lst`	A list from local.multiple.regression.
`xaxt`	An optional vector of labels for the "x" axis. Default is 1:n.

Details

The routine produces time series plots of wave local multiple correlations with their confidence intervals. Also, at every upturn and downturn, the name of the variable that maximizes its multiple correlation against the rest is shown.

Value

Plot.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

`plot_wave.local.multiple.cross.correlation`
Auxiliary routine for plotting wave local multiple cross-correlations

Description

Produces a plot of wave local multiple cross-correlations.

Usage

```
plot_wave.local.multiple.cross.correlation(Lst, lmax,
     lag.first=FALSE, xaxt="s", pdf.write=NULL)
```

Arguments

`Lst`	A list from local.multiple.cross.regression.
`lmax`	maximum lag (and lead).
`lag.first`	if TRUE, it produces lag-lead pages with $J + 1$ wavelet plots each. Otherwise (default) it gives wavelet pages with $2 * lmax + 1$ lag-lead plots each.
`xaxt`	An optional vector of labels for the "x" axis. Default is 1:n.
`pdf.write`	Optional name leader to save files to pdf format. The actual filename of each page is either "plot_<pdf.write>_WLMCC_<i>.pdf", where "i" is the lag/lead $i = -lmax ... + lmax$, or, "plot_<pdf.write>_WLMCC_<j>.pdf", where "j" is the wavelet level $j = 1...(J + 1)$.

Details

The routine produces time series plots of wave local multiple cross-correlations with their confidence intervals. Also, at every upturn and downturn, the name of the variable that maximizes its multiple cross-correlation against the rest is shown.

Value

Plot.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

`plot_wave.local.multiple.cross.regression`
Auxiliary routine for plotting wave local multiple cross-regressions

---

Description

Produces a set of plots of wave local multiple cross-regressions.

Usage

```
plot_wave.local.multiple.cross.regression(Lst, lmax, nsig=2,
     xaxt="s", pdf.write=NULL)
```

Arguments

`Lst`	A list from wave.multiplecross.regression.
`lmax`	maximum lag (and lead).
`nsig`	An optional value for the number of significant variables to plot_ Default is 2.
`xaxt`	An optional vector of labels for the "x" axis. Default is 1:n.
`pdf.write`	Optional name leader to save files to pdf format. The actual filename of each page "plot_<pdf.write>_WLMCC_<j>.pdf", where "j" is the wavelet level $j = 1...(J+1)$.

Details

The routine produces $J+1$ pages, one per wavelet level, each with time series plots of wave multiple cross-regressions at different lags and leads, each with their confidence interval. Also, the name of the variable that maximizes that multiple correlation against the rest is shown on top. The others are named with their order of significance when they are relevant.

Value

Plot.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

plot_wave.local.multiple.regression
Auxiliary routine for plotting wave local multiple regressions

Description

Produces a set of plots of wave local multiple regressions.

Usage

```
plot_wave.local.multiple.regression(Lst,nsig=2, xaxt="s")
```

Arguments

`Lst`	A list from wave.multiple.regression.
`nsig`	An optional value for the number of significant variables to plot_ Default is 2.
`xaxt`	An optional vector of labels for the "x" axis. Default is 1:n.

Details

The routine produces $J+1$ time series plots of wave multiple regressions, each with their confidence interval. Also, the name of the variable that maximizes that multiple correlation against the rest is shown on top. The others are named with their order of significance when they are relevant.

Value

Plot.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

plot_wave.multiple.correlation

Auxiliary routine for plotting wave multiple correlations

Description

Produces a plot of wave multiple correlations.

Usage

```
plot_wave.multiple.correlation(Lst)
```

Arguments

Lst	A list from wave.multiple.regression or wave.multiple.correlation.

Details

The routine produces a plot of wave multiple correlations, at each wavelet level, with its confidence interval. Also, at each wavelet level, the name of the variable that maximizes its multiple correlation against the rest is shown.

Value

Plot.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2012. Wavelet multiple correlation and cross-correlation: A multiscale analysis of Eurozone stock markets. Physica A: Statistical Mechanics and its Applications 391, 1097–1104. <DOI:10.1016/j.physa.2011.11.002>

`plot_wave.multiple.cross.correlation` *Auxiliary routine for plotting wave multiple cross-correlations*

Description

Produces a plot of wave multiple cross-correlations.

Usage

```
plot_wave.multiple.cross.correlation(Lst, lmax, by=3)
```

Arguments

`Lst`	A list from wave.multiple.cross.regression or wave.multiple.cross.correlation.
`lmax`	maximum lag (and lead).
`by`	labels are printed every lmax/by. Default is 3.

Details

The routine produces a set of plots of wave multiple cross-correlations, one per wavelet level, with their confidence intervals. Also, at each wavelet level, the name of the variable that maximizes its multiple cross-correlation against the rest is shown.

Value

Plot.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2012. Wavelet multiple correlation and cross-correlation: A multiscale analysis of Eurozone stock markets. Physica A: Statistical Mechanics and its Applications 391, 1097–1104. <DOI:10.1016/j.physa.2011.11.002>

plot_wave.multiple.cross.regression

Auxiliary routine for plotting wave multiple cross-regressions

Description

Produces a plot of wave multiple cross.regressions.

Usage

```
plot_wave.multiple.cross.regression(Lst, lmax, nsig=2, by=3)
```

Arguments

`Lst`	A list from wave.multiple.cross.regression.
`lmax`	maximum lag (and lead).
`nsig`	An optional value for the number of significant variables to plot_ Default is 2.
`by`	labels are printed every lmax/by. Default is 3.

Details

The routine produces a plot of wave multiple regressions, one per wavelet level, with their confidence intervals. Also, the name of the variable that maximizes that multiple correlation against the rest is shown on top. The others are named with their order of significance when they are relevant.

Value

Plot.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

`plot_wave.multiple.regression`
Auxiliary routine for plotting wave multiple regressions

Description

Produces a plot of wave multiple regressions.

Usage

```
plot_wave.multiple.regression(Lst, nsig=2)
```

Arguments

`Lst`	A list from wave.multiple.regression.
`nsig`	An optional value for the number of significant variables to plot_ Default is 2.

Details

The routine produces a plot of wave multiple regressions with their confidence interval. Also, the name of the variable that maximizes that multiple correlation against the rest is shown on top. The others are named with their order of significance when they are relevant.

Value

Plot.

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

wave.local.multiple.correlation

Wavelet routine for local multiple correlation

Description

Produces an estimate of the multiscale local multiple correlation (as defined below) along with approximate confidence intervals.

Usage

```
wave.local.multiple.correlation(xx, M, window="gauss", p = .975, ymaxr=NULL)
```

Arguments

`xx`	A list of n (multiscaled) time series, usually the outcomes of dwt or modwt, *i.e.* xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)
`M`	length of the weight function or rolling window.
`window`	type of weight function or rolling window. Six types are allowed, namely the uniform window, Cleveland or tricube window, Epanechnikov or parabolic window, Bartlett or triangular window, Wendland window and the gaussian window. The letter case and length of the argument are not relevant as long as at least the first four characters are entered.
`p`	one minus the two-sided p-value for the confidence interval, *i.e.* the cdf value.
`ymaxr`	index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level wlmc chooses the one maximizing the multiple correlation.

Details

The routine calculates one single set of wavelet multiple correlations out of n variables that can be plotted in in either single heatmap or J line graphs (the former is usually the best graphic option but the latter is useful if confidence intervals are explicitly needed), as an alternative to trying to make sense out of $n(n-1)/2$ $[J \times T]$ sets of local wavelet correlations. The code is based on the calculation, at each wavelet scale, of the square root of the coefficient of determination in that linear combination of locally weighted wavelet coefficients for which such coefficient of determination is a maximum. The code provided here is based on the wave.multiple.correlation routine in this package which in turn is based on the wave.correlation routine in Brandon Whitcher's *waveslim* R package Version: 1.6.4, which in turn is based on wavelet methodology developed in Percival and Walden (2000); Gençay, Selçuk and Whitcher (2001) and others.

Value

List of four elements:

`val:` numeric matrix (rows = #observations, columns = #levels in the wavelet transform object) providing the point estimates for the wavelet local multiple correlation.

`lo:` numeric matrix (rows = #observations, columns = #levels in the wavelet transform object) providing the lower bounds from the confidence interval.

`up:` numeric matrix (rows = #observations, columns = #levels in the wavelet transform object) providing the upper bounds from the confidence interval.

`YmaxR:` numeric matrix (rows = #observations, columns = #levels in the wavelet transform object) giving, at each wavelet level and time, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, *wlmc* chooses at each wavelet level and value in time the variable maximizing the multiple correlation.

Note

Needs *waveslim* package to calculate *dwt* or *modwt* coefficients as inputs to the routine (also for data in the example).

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

Examples

```
## Based on  Figure 4 showing correlation structural breaks in Fernandez-Macho (2018).

library(wavemulcor)
options(warn = -1)

data(xrand)
N <- length(xrand1)
b <- trunc(N/3)
t1 <- 1:b
t2 <- (b+1):(2*b)
t3 <- (2*b+1):N

wf <- "d4"
M <- N/2^3 #sharper with N/2^4
window <- "gauss"
J <- 3 #trunc(log2(N))-3

# ---------------------------
```

```
cor1 <- cor(xrand1[t1],xrand2[t1])
cor2 <- cor(xrand1[t2],xrand2[t2])
cor3 <- cor(xrand1[t3],xrand2[t3])
cortext <- paste0(round(100*cor1,0),"-",round(100*cor2,0),"-",round(100*cor3,0))

ts.plot(cbind(xrand1,xrand2),col=c("red","blue"),xlab="time")

xrand1.modwt <- modwt(xrand1, wf, J)
xrand1.modwt.bw <- brick.wall(xrand1.modwt, wf)

xrand2.modwt <- modwt(xrand2, wf, J)
xrand2.modwt.bw <- brick.wall(xrand2.modwt, wf)

xx <- list(xrand1.modwt.bw,xrand2.modwt.bw)

# ---------------------------

xy.mulcor <- wave.local.multiple.correlation(xx, M, window=window)

val <- as.matrix(xy.mulcor$val)
lo  <- as.matrix(xy.mulcor$lo)
up  <- as.matrix(xy.mulcor$up)
YmaxR <- as.matrix(xy.mulcor$YmaxR)

# ---------------------------

old.par <- par()

# ##Producing heat plot

scale.names <- paste0("(",c("2-4","4-8","8-16","16-32","32-64","64-128","128-256","256-512",
                            "512-1024","1024-2048"),"]")
scale.names <- c(scale.names[1:J],"smooth")

title <- paste("Wavelet Local Multiple Correlation")
sub <- paste("first",b,"obs:",round(100*cor1,1),"% correlation;","middle",b,"obs:",
             round(100*cor2,1),"%","rest:",round(100*cor3,1),"%")
xlab <- "time"
ylab <- "periods"

plot3D::image2D(z=val, x=1:nrow(val), y=1:ncol(val),
        main=title, #sub=sub,
        xlab=xlab, ylab=ylab, axes=FALSE, clab = expression(varphi),
        rasterImage = TRUE, contour = list(lwd = 2, col = plot3D::jet.col(11)))
axis(side=1, at=seq(10,nrow(val),by=10), cex.axis=0.75)
axis(side=2, at=1:ncol(val),labels=scale.names, las=1,cex.axis=0.75)

# ---------------------------

##Producing line plots with confidence intervals

colnames(val)[1:J] <- paste0("level",1:J)
par(mfrow=c(3,2), las=1, pty="m", mar=c(2,3,1,0)+.1, oma=c(1.2,1.2,0,0))
```

```
for(i in J:1) {
  matplot(1:N,val[,i], type="l", lty=1, ylim=c(-1,1), xaxt="n",
          xlab="", ylab="", main=colnames(val)[i])
  if(i<3) {axis(side=1, at=seq(10,N,by=10))}
  #axis(side=2, at=c(-.2, 0, .5, 1))
  lines(lo[,i], lty=1, col=2) ##Add Connected Line Segments to a Plot
  lines(up[,i], lty=1, col=2)
  abline(h=0)              ##Add Straight horiz and vert Lines to a Plot
}
par(las=0)
mtext('time', side=1, outer=TRUE, adj=0.5)
mtext('Wavelet Local Multiple Correlation', side=2, outer=TRUE, adj=0.5)

#reset graphics parameters
par(old.par)
```

`wave.local.multiple.cross.correlation`
Wavelet routine for local multiple cross-correlation

Description

Produces an estimate of the multiscale local multiple cross-correlation (as defined below) along with approximate confidence intervals.

Usage

```
wave.local.multiple.cross.correlation(xx, M,
     window="gauss", lag.max=NULL, p=.975, ymaxr=NULL)
```

Arguments

`xx`	A list of n (multiscaled) time series, usually the outcomes of dwt or modwt, *i.e.* xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)
`M`	length of the weight function or rolling window.
`window`	type of weight function or rolling window. Six types are allowed, namely the uniform window, Cleveland or tricube window, Epanechnikov or parabolic window, Bartlett or triangular window, Wendland window and the gaussian window. The letter case and length of the argument are not relevant as long as at least the first four characters are entered.
`lag.max`	maximum lag (and lead). If not set, it defaults to half the square root of the length of the original series.
`p`	one minus the two-sided p-value for the confidence interval, *i.e.* the cdf value.
`ymaxr`	index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level wlmc chooses the one maximizing the multiple correlation.

Details

The routine calculates $J + 1$ sets of wavelet multiple cross-correlations,one per wavelet level, out of n variables, that can be plotted each as lags and leads time series plots.

Value

List of four elements:

- `val:` list of $J+1$ dataframes, each (rows = #observations, columns = #levels) providing the point estimates for the wavelet local multiple correlation.
- `lo:` list of $J+1$ dataframes, each (rows = #observations, columns = #lags and leads) providing the lower bounds from the confidence interval.
- `up:` list of $J+1$ dataframes, each (rows = #observations, columns = #lags and leads) providing the upper bounds from the confidence interval.
- `YmaxR:` numeric matrix (rows = #observations, columns = #levels) giving, at each wavelet level and time, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, *wlmc* chooses at each wavelet level and value in time the variable maximizing the multiple correlation.

Note

Needs *waveslim* package to calculate *dwt* or *modwt* coefficients as inputs to the routine (also for data in the example).

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

Examples

```
## Based on data from Figure 7.9 in Gencay, Selcuk and Whitcher (2001)
## plus one random series.


library(wavemulcor)

data(exchange)
returns <- diff(log(exchange))
returns <- ts(returns, start=1970, freq=12)
N <- dim(returns)[1]
wf <- "d4"
M <- 30
window <- "gauss"
```

```
J <- 3 #trunc(log2(N))-3
lmax <- 2

set.seed(140859)

demusd.modwt <- brick.wall(modwt(returns[,"DEM.USD"], wf, J), wf)
jpyusd.modwt <- brick.wall(modwt(returns[,"JPY.USD"], wf, J), wf)
rand.modwt   <- brick.wall(modwt(rnorm(length(returns[,"DEM.USD"])), wf, J), wf)

##xx <- list(demusd.modwt.bw, jpyusd.modwt.bw)
xx <- list(demusd.modwt, jpyusd.modwt, rand.modwt)
names(xx) <- c("DEM.USD","JPY.USD","rand")

## Not run:
# Note: WLMCC may take more than 10 seconds of CPU time on some systems

Lst <- wave.local.multiple.cross.correlation(xx, M, window=window, lag.max=lmax)
val <- Lst$val
low.ci <- Lst$lo
upp.ci <- Lst$up
YmaxR <- Lst$YmaxR

# ---------------------------

##Producing cross-correlation plot

xvar <- seq(1,N,M)
level.lab <- c(paste("Level",1:J),paste("Smooth",J))
ymin <- -0.1
if (length(xx)<3) ymin <- -1
for(j in 1:(J+1)) {
  par(mfcol=c(lmax+1,2), las=1, pty="m", mar=c(2,3,1,0)+.1, oma=c(1.2,1.2,1.2,0))
  # xaxt <- c(rep("n",lmax),"s",rep("n",lmax))
  for(i in c(-lmax:0,lmax:1)+lmax+1) {
    matplot(1:N,val[[j]][,i], type="l", lty=1, ylim=c(ymin,1), #xaxt=xaxt[i],
            xlab="", ylab="", main=paste("Lag",i-lmax-1))
    abline(h=0)              ##Add Straight horiz
    lines(low.ci[[j]][,i], lty=1, col=2) ##Add Connected Line Segments to a Plot
    lines(upp.ci[[j]][,i], lty=1, col=2)
    text(xvar,1, labels=names(xx)[YmaxR[[j]]][xvar], adj=0.25, cex=.8)
  }
  par(las=0)
  mtext('time', side=1, outer=TRUE, adj=0.5)
  mtext('Local Multiple Cross-Correlation', side=2, outer=TRUE, adj=0.5)
  mtext(level.lab[j], side=3, outer=TRUE, adj=0.5)
}


## End(Not run)
```

`wave.local.multiple.cross.regression`
Wavelet routine for local multiple cross-regression

Description

Produces an estimate of the multiscale local multiple cross-regression (as defined below) along with approximate confidence intervals.

Usage

```
wave.local.multiple.cross.regression(xx, M,
     window="gauss", lag.max=NULL, p=.975, ymaxr=NULL)
```

Arguments

`xx`	A list of n (multiscaled) time series, usually the outcomes of dwt or modwt, *i.e.* xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)
`M`	length of the weight function or rolling window.
`window`	type of weight function or rolling window. Six types are allowed, namely the uniform window, Cleveland or tricube window, Epanechnikov or parabolic window, Bartlett or triangular window, Wendland window and the gaussian window. The letter case and length of the argument are not relevant as long as at least the first four characters are entered.
`lag.max`	maximum lag (and lead). If not set, it defaults to half the square root of the length of the original series.
`p`	one minus the two-sided p-value for the confidence interval, *i.e.* the cdf value.
`ymaxr`	index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level wlmc chooses the one maximizing the multiple correlation.

Details

The routine calculates $J + 1$ sets of wavelet multiple cross-regressions, one per wavelet level, out of n variables, that can be plotted each as lags and leads time series plots.

Value

List of four elements:

`cor:` List of three elements:

- val: list of $J + 1$ dataframes, each (rows = #observations, columns = #levels) providing the point estimates for the wavelet local multiple correlation.
- lo: list of $J+1$ dataframes, each (rows = #observations, columns = #lags and leads) providing the lower bounds from the confidence interval.

- up: list of $J+1$ dataframes, each (rows = #observations, columns = #lags and leads) providing the upper bounds from the confidence interval.

`reg`: List of $J+1$ elements, one per wavelet level, each with:

- rval: numeric array (1st_dim = #observations, 2nd-dim = #lags and leads, 3rd_dim = #regressors+1) of local regression estimates.
- rstd: numeric array (1st_dim = #observations, 2nd-dim = #lags and leads, 3rd_dim = #regressors+1) of their standard deviations.
- rlow: numeric array (1st_dim = #observations, 2nd-dim = #lags and leads, 3rd_dim = #regressors+1) of their lower bounds.
- rupp: numeric array (1st_dim = #observations, 2nd-dim = #lags and leads, 3rd_dim = #regressors+1) of their upper bounds.
- rtst: numeric array (1st_dim = #observations, 2nd-dim = #lags and leads, 3rd_dim = #regressors+1) of their t statistic values.
- rord: numeric array (1st_dim = #observations, 2nd-dim = #lags and leads, 3rd_dim = #regressors+1) of their index order when sorted by significance.
- rpva: numeric array (1st_dim = #observations, 2nd-dim = #lags and leads, 3rd_dim = #regressors+1) of their p values.

`YmaxR`: dataframe (rows = #observations, columns = #levels) giving, at each wavelet level and time, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, *wlmcr* chooses at each wavelet level and value in time the variable maximizing the multiple correlation.

`data`: dataframe (rows = #observations, cols = #regressors) of original data.

Note

Needs *waveslim* package to calculate *dwt* or *modwt* coefficients as inputs to the routine (also for data in the example).

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

Examples

```
## Based on data from Figure 7.9 in Gencay, Selcuk and Whitcher (2001)
## plus one random series.

library(wavemulcor)
```

```
data(exchange)
returns <- diff(log(exchange))
returns <- ts(returns, start=1970, freq=12)
N <- dim(returns)[1]
wf <- "d4"
M <- 30
window <- "gauss"
J <- 3 #trunc(log2(N))-3
lmax <- 2

set.seed(140859)

demusd.modwt <- brick.wall(modwt(returns[,"DEM.USD"], wf, J), wf)
jpyusd.modwt <- brick.wall(modwt(returns[,"JPY.USD"], wf, J), wf)
rand.modwt   <- brick.wall(modwt(rnorm(length(returns[,"DEM.USD"])), wf, J), wf)

xx <- list(demusd.modwt, jpyusd.modwt, rand.modwt)
names(xx) <- c("DEM.USD","JPY.USD","rand")

## Not run:
# Note: WLMCR may take more than 10 seconds of CPU time on some systems

Lst <- wave.local.multiple.cross.regression(xx, M, window=window, lag.max=lmax) #, ymaxr=1)

# ---------------------------

##Producing cross-correlation plot

plot_wave.local.multiple.cross.correlation(Lst, lmax, lag.first=FALSE) #, xaxt="s")

##Producing cross-regression plot

plot_wave.local.multiple.cross.regression(Lst, lmax, nsig=2) #, xaxt="s")

## End(Not run)
```

wave.local.multiple.regression
Wavelet routine for local multiple regression

Description

Produces an estimate of the multiscale local multiple regression (as defined below) along with approximate confidence intervals.

Usage

```
wave.local.multiple.regression(xx, M, window="gauss", p = .975, ymaxr=NULL)
```

Arguments

`xx` A list of n (multiscaled) time series, usually the outcomes of dwt or modwt, *i.e.* xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)

`M` length of the weight function or rolling window.

`window` type of weight function or rolling window. Six types are allowed, namely the uniform window, Cleveland or tricube window, Epanechnikov or parabolic window, Bartlett or triangular window, Wendland window and the gaussian window. The letter case and length of the argument are not relevant as long as at least the first four characters are entered.

`p` one minus the two-sided p-value for the confidence interval, *i.e.* the cdf value.

`ymaxr` index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level wlmc chooses the one maximizing the multiple correlation.

Details

The routine calculates one single set of wavelet multiple regressions out of n variables that can be plotted in in J line graphs with explicit confidence intervals.

Value

List of four elements:

`cor:` List of $J + 1$ elements, one per wavelet level, each with:

- val: numeric matrix (rows = #observations, columns = #levels in the wavelet transform object) providing the point estimates for the wavelet local multiple correlation.
- lo: numeric matrix (rows = #observations, columns = #levels in the wavelet transform object) providing the lower bounds from the confidence interval.
- up: numeric matrix (rows = #observations, columns = #levels in the wavelet transform object) providing the upper bounds from the confidence interval.

`reg:` List of $J + 1$ elements, one per wavelet level, each with:

- rval: numeric matrix (rows = #observations, cols = #regressors+1) of local regression estimates.
- rstd: numeric matrix (rows = #observations, cols = #regressors+1) of their standard deviations.
- rlow: numeric matrix (rows = #observations, cols = #regressors+1) of their lower bounds.
- rupp: numeric matrix (rows = #observations, cols = #regressors+1) of their upper bounds.
- rtst: numeric matrix (rows = #observations, cols = #regressors+1) of their t statistic values.
- rord: numeric matrix (rows = #observations, cols = #regressors+1) of their index order when sorted by significance.
- rpva: numeric matrix (rows = #observations, cols = #regressors+1) of their p values.

`YmaxR:` dataframe (rows = #observations, columns = #levels in the wavelet transform object) giving, at each wavelet level and time, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, *wlmr* chooses at each wavelet level and value in time the variable maximizing the multiple correlation.

`data:` dataframe (rows = #observations, cols = #regressors) of original data.

Note

Needs *waveslim* package to calculate *dwt* or *modwt* coefficients as inputs to the routine (also for data in the example).

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

Examples

```
## Based on data from Figure 7.8 in Gencay, Selcuk and Whitcher (2001)
## plus two random series.

library(wavemulcor)

data(exchange)
returns <- diff(log(as.matrix(exchange)))
returns <- ts(returns, start=1970, freq=12)
N <- dim(returns)[1]

wf <- "d4"
M <- 30
window <- "gauss"
J <- 3 #trunc(log2(N))-3

set.seed(140859)

demusd.modwt <- brick.wall(modwt(returns[,"DEM.USD"], wf, J), wf)
jpyusd.modwt <- brick.wall(modwt(returns[,"JPY.USD"], wf, J), wf)
xrand.modwt <- brick.wall(modwt(rnorm(length(returns[,"DEM.USD"])), wf, J), wf)

# ---------------------------

xx <- list(demusd.modwt, jpyusd.modwt, xrand.modwt)
names(xx) <- c("DEM.USD","JPY.USD","rand")
```

```
Lst <- wave.local.multiple.regression(xx, M, window=window) #, ymaxr=1)

# ---------------------------

##Producing line plots with CI

plot_wave.local.multiple.correlation(Lst) #, xaxt="s")

##Producing regression plots

plot_wave.local.multiple.regression(Lst) #, xaxt="s")
```

`wave.multiple.correlation` *Wavelet routine for multiple correlation*

Description

Produces an estimate of the multiscale multiple correlation (as defined below) along with approximate confidence intervals.

Usage

```
wave.multiple.correlation(xx, N, p = 0.975, ymaxr=NULL)
```

Arguments

`xx`	A list of n (multiscaled) time series, usually the outcomes of dwt or modwt, *i.e.* xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)
`N`	length of the time series
`p`	one minus the two-sided p-value for the confidence interval, *i.e.* the cdf value.
`ymaxr`	index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level wmc chooses the one maximizing the multiple correlation.

Details

The routine calculates one single set of wavelet multiple correlations out of n variables that can be plotted in a single graph, as an alternative to trying to make sense out of $n(n-1)/2$ sets of wavelet correlations. The code is based on the calculation, at each wavelet scale, of the square root of the coefficient of determination in the linear combination of variables for which such coefficient of determination is a maximum. The code provided here is based on the *wave.correlation* routine in Brandon Whitcher's *waveslim* R package Version: 1.6.4, which in turn is based on wavelet methodology developed in Percival and Walden (2000); Gençay, Selçuk and Whitcher (2001) and others.

Value

List of two elements:

xy.mulcor: numeric matrix with as many rows as levels in the wavelet transform object. The first column provides the point estimate for the wavelet multiple correlation, followed by the lower and upper bounds from the confidence interval.

YmaxR: numeric vector giving, at each wavelet level, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, *wmc* chooses at each wavelet level the variable maximizing the multiple correlation.

Note

Needs *waveslim* package to calculate *dwt* or *modwt* coefficients as inputs to the routine (also for data in the example).

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2012. Wavelet multiple correlation and cross-correlation: A multiscale analysis of Eurozone stock markets. Physica A: Statistical Mechanics and its Applications 391, 1097–1104. <DOI:10.1016/j.physa.2011.11.002>

Examples

```
## Based on data from Figure 7.8 in Gencay, Selcuk and Whitcher (2001)
## plus one random series.

library(wavemulcor)
data(exchange)
returns <- diff(log(as.matrix(exchange)))
returns <- ts(returns, start=1970, freq=12)
N <- dim(returns)[1]

wf <- "d4"
J <- trunc(log2(N))-3

demusd.modwt <- brick.wall(modwt(returns[,"DEM.USD"], wf, J), wf)
jpyusd.modwt <- brick.wall(modwt(returns[,"JPY.USD"], wf, J), wf)
rand.modwt   <- brick.wall(modwt(rnorm(length(returns[,"DEM.USD"])), wf, J), wf)

xx <- list(demusd.modwt, jpyusd.modwt, rand.modwt)

Lst <- wave.multiple.correlation(xx, N = length(xx[[1]][[1]]))
returns.modwt.cor <- Lst$xy.mulcor[1:J,]
YmaxR <- Lst$YmaxR
```

```
exchange.names <- c("DEM.USD", "JPY.USD", "RAND")

##Producing plot

par(mfrow=c(1,1), las=0, mar=c(5,4,4,2)+.1)
matplot(2^(0:(J-1)), returns.modwt.cor[-(J+1),], type="b",
  log="x", pch="*LU", xaxt="n", lty=1, col=c(1,4,4),
  xlab="Wavelet Scale", ylab="Wavelet Multiple Correlation")
axis(side=1, at=2^(0:7))
abline(h=0)
text(2^(0:7), min(returns.modwt.cor[-(J+1),])-0.03,
  labels=exchange.names[YmaxR], adj=0.5, cex=.5)
```

---

wave.multiple.cross.correlation

Wavelet routine for multiple cross-correlation

Description

Produces an estimate of the multiscale multiple cross-correlation (as defined below).

Usage

```
wave.multiple.cross.correlation(xx, lag.max=NULL, p=.975, ymaxr=NULL)
```

Arguments

`xx`	A list of n (multiscaled) time series, usually the outcomes of dwt or modwt, *i.e.* xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)
`lag.max`	maximum lag (and lead). If not set, it defaults to half the square root of the length of the original series.
`p`	one minus the two-sided p-value for the confidence interval, *i.e.* the cdf value.
`ymaxr`	index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level wmc chooses the one maximizing the multiple correlation.

Details

The routine calculates one single set of wavelet multiple cross-correlations out of n variables that can be plotted as one single set of graphs (one per wavelet level), as an alternative to trying to make sense out of $n(n-1)/2.J$ sets of wavelet cross-correlations. The code is based on the calculation, at each wavelet scale, of the square root of the coefficient of determination in a linear combination of variables that includes a lagged variable for which such coefficient of determination is a maximum.

Value

List of two elements:

- xy.mulcor: numeric matrix with as many rows as levels in the wavelet transform object. The columns provide the point estimates for the wavelet multiple cross-correlations at different lags (and leads). The central column (*lag=0*) replicates the wavelet multiple correlations. Columns to the right (*lag>0*) give wavelet multiple cross-correlations with positive lag, *i.e.* with y=var[Pimax] lagging behind a linear combination of the rest: x[t]hat –> y[t+j]. Columns to the left (*lag<0*) give wavelet multiple cross-correlations with negative lag, *i.e.* with y=var[Pimax] leading a linear combination of the rest: y[t-j] –> x[t]hat.
- ci.mulcor: list of two elements:
  - lower: numeric matrix of the same dimensions as *xy.mulcor* giving the lower bounds of the corresponding $100(1-2(1-p))\%$ confidence interval.
  - upper: *idem* for the upper bounds.
- YmaxR: numeric vector giving, at each wavelet level, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, *wmcc* chooses at each wavelet level the variable maximizing the multiple correlation.

Note

Needs *waveslim* package to calculate *dwt* or *modwt* coefficients as inputs to the routine (also for data in the example).

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2012. Wavelet multiple correlation and cross-correlation: A multiscale analysis of Eurozone stock markets. Physica A: Statistical Mechanics and its Applications 391, 1097–1104. <DOI:10.1016/j.physa.2011.11.002>

Examples

```
## Based on data from Figure 7.9 in Gencay, Selcuk and Whitcher (2001)
## plus one random series.

library(wavemulcor)
data(exchange)
returns <- diff(log(exchange))
returns <- ts(returns, start=1970, freq=12)
N <- dim(returns)[1]

wf <- "d4"
J <- trunc(log2(N))-3
lmax <- 36
n <- dim(returns)[1]
```

```
demusd.modwt <- brick.wall(modwt(returns[,"DEM.USD"], wf, J), wf)
jpyusd.modwt <- brick.wall(modwt(returns[,"JPY.USD"], wf, J), wf)
rand.modwt   <- brick.wall(modwt(rnorm(length(returns[,"DEM.USD"])), wf, J), wf)

##xx <- list(demusd.modwt.bw, jpyusd.modwt.bw)
xx <- list(demusd.modwt, jpyusd.modwt, rand.modwt)

Lst <- wave.multiple.cross.correlation(xx, lmax)
returns.cross.cor <- Lst$xy.mulcor[1:J,]
returns.lower.ci <- Lst$ci.mulcor$lower[1:J,]
returns.upper.ci <- Lst$ci.mulcor$upper[1:J,]
YmaxR <- Lst$YmaxR

# ---------------------------

##Producing correlation plot

rownames(returns.cross.cor) <- rownames(returns.cross.cor, do.NULL = FALSE, prefix = "Level ")
par(mfrow=c(3,2), las=1, pty="m", mar=c(2,3,1,0)+.1, oma=c(1.2,1.2,0,0))
ymin <- -0.1
if (length(xx)<3) ymin <- -1
for(i in J:1) {
 matplot((1:(2*lmax+1)),returns.cross.cor[i,], type="l", lty=1, ylim=c(ymin,1), xaxt="n",
         xlab="", ylab="", main=rownames(returns.cross.cor)[[i]][1])
  if(i<3) {axis(side=1, at=seq(1, 2*lmax+1, by=12), labels=seq(-lmax, lmax, by=12))}
  #axis(side=2, at=c(-.2, 0, .5, 1))
  abline(h=0,v=lmax+1)              ##Add Straight horiz and vert Lines to a Plot
  lines(returns.lower.ci[i,], lty=1, col=2) ##Add Connected Line Segments to a Plot
  lines(returns.upper.ci[i,], lty=1, col=2)
  text(1,1, labels=names(xx)[YmaxR[i]], adj=0.25, cex=.8)
}
par(las=0)
mtext('Lag (months)', side=1, outer=TRUE, adj=0.5)
mtext('Wavelet Multiple Cross-Correlation', side=2, outer=TRUE, adj=0.5)
```

wave.multiple.cross.regression

Wavelet routine for multiple cross-regression

Description

Produces an estimate of the multiscale multiple cross-regression (as defined below).

Usage

```
wave.multiple.cross.regression(xx, lag.max=NULL, p = .975, ymaxr=NULL)
```

Arguments

`xx` A list of n (multiscaled) time series, usually the outcomes of dwt or modwt, *i.e.* xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)

`lag.max` maximum lag (and lead). If not set, it defaults to half the square root of the length of the original series.

`p` one minus the two-sided p-value for the confidence interval, *i.e.* the cdf value.

`ymaxr` index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level wmc chooses the one maximizing the multiple correlation.

Details

The routine calculates one single set of wavelet multiple cross-regressions out of n variables that can be plotted as one single set of graphs (one per wavelet level).

Value

List of four elements:

`xy.mulcor` List of three elements:

- wavemulcor: numeric matrix (rows = #levels, #cols = #lags and leads) with as many rows as levels in the wavelet transform object. The columns provide the point estimates for the wavelet multiple cross-correlations at different lags (and leads). The central column (lag=0) replicates the wavelet multiple correlations. Columns to the right (lag>0) give wavelet multiple cross-correlations with positive lag, *i.e.* with y=var[Pimax] lagging behind a linear combination of the rest: x[t]hat –> y[t+j]. Columns to the left (lag<0) give wavelet multiple cross-correlations with negative lag, *i.e.* with y=var[Pimax] leading a linear combination of the rest: y[t-j] –> x[t]hat.
- lower: numeric matrix (rows = #levels, #cols = #lags and leads) of lower bounds of the confidence interval.
- upper: numeric matrix (rows = #levels, #cols = #lags and leads) of upper bounds of the confidence interval.

`xy.mulreg:` List of seven elements:

- rval: numeric array (1stdim = #levels, 2nddim = #lags and leads, 3rddim = #regressors+1) of regression estimates.
- rstd: numeric array (1stdim = #levels, 2nddim = #lags and leads, 3rddim = #regressors+1) of their standard deviations.
- rlow: numeric array (1stdim = #levels, 2nddim = #lags and leads, 3rddim = #regressors+1) of their lower bounds.
- rupp: numeric array (1stdim = #levels, 2nddim = #lags and leads, 3rddim = #regressors+1) of their upper bounds.
- rtst: numeric array (1stdim = #levels, 2nddim = #lags and leads, 3rddim = #regressors+1) of their t statistic values.

- rord: numeric array (1stdim = #levels, 2nddim = #lags and leads, 3rddim = #regressors+1) of their index order when sorted by significance.
- rpva: numeric array (1stdim = #levels, 2nddim = #lags and leads, 3rddim = #regressors+1) of their p values.

`YmaxR:` numeric vector giving, at each wavelet level, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, *wmcr* chooses at each wavelet level the variable maximizing the multiple correlation.

`data:` dataframe (rows = #levels, cols = #regressors) of original data.

Note

Needs *waveslim* package to calculate *dwt* or *modwt* coefficients as inputs to the routine (also for data in the example).

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

Examples

```
## Based on data from Figure 7.9 in Gencay, Selcuk and Whitcher (2001)
## plus one random series.

library(wavemulcor)

data(exchange)
returns <- diff(log(exchange))
returns <- ts(returns, start=1970, freq=12)
N <- dim(returns)[1]

wf <- "d4"
J <- trunc(log2(N))-3
lmax <- 36

set.seed(140859)

demusd.modwt <- brick.wall(modwt(returns[,"DEM.USD"], wf, J), wf)
jpyusd.modwt <- brick.wall(modwt(returns[,"JPY.USD"], wf, J), wf)
rand.modwt   <- brick.wall(modwt(rnorm(length(returns[,"DEM.USD"])), wf, J), wf)

# ---------------------------

xx <- list(demusd.modwt, jpyusd.modwt, rand.modwt)
```

```
names(xx) <- c("DEM.USD","JPY.USD","rand")

Lst <- wave.multiple.cross.regression(xx, lmax)

# ---------------------------

##Producing correlation plot

plot_wave.multiple.cross.correlation(Lst, lmax) #, by=2)

##Producing regression plot

plot_wave.multiple.cross.regression(Lst, lmax) #, by=2)
```

`wave.multiple.regression` *Wavelet routine for multiple regression*

Description

Produces an estimate of the multiscale multiple regression (as defined below) along with approximate confidence intervals.

Usage

```
wave.multiple.regression(xx, N, p = 0.975, ymaxr=NULL)
```

Arguments

`xx`	A list of n (multiscaled) time series, usually the outcomes of dwt or modwt, *i.e.* xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)
`N`	length of the time series
`p`	one minus the two-sided p-value for the confidence interval, *i.e.* the cdf value.
`ymaxr`	index number of the variable whose correlation is calculated against a linear combination of the rest, otherwise at each wavelet level wmc chooses the one maximizing the multiple correlation.

Details

The routine calculates one single set of wavelet multiple regressions out of n variables that can be plotted in a single graph.

Value

List of four elements:

`xy.mulcor:` numeric matrix with as many rows as levels in the wavelet transform object. The first column provides the point estimate for the wavelet multiple correlation, followed by the lower and upper bounds from the confidence interval.

`xy.mulreg:` List of seven elements:

- rval: numeric matrix (rows = #levels, cols = #regressors+1) of regression estimates.
- rstd: numeric matrix (rows = #levels, cols = #regressors+1) of their standard deviations.
- rlow: numeric matrix (rows = #levels, cols = #regressors+1) of their lower bounds.
- rupp: numeric matrix (rows = #levels, cols = #regressors+1) of their upper bounds.
- rtst: numeric matrix (rows = #levels, cols = #regressors+1) of their t statistic values.
- rord: numeric matrix (rows = #levels, cols = #regressors+1) of their index order when sorted by significance.
- rpva: numeric matrix (rows = #levels, cols = #regressors+1) of their p values.

`YmaxR:` numeric vector giving, at each wavelet level, the index number of the variable whose correlation is calculated against a linear combination of the rest. By default, *wmc* chooses at each wavelet level the variable maximizing the multiple correlation.

`data:` dataframe (rows = #levels, cols = #regressors) of original data.

Note

Needs *waveslim* package to calculate *dwt* or *modwt* coefficients as inputs to the routine (also for data in the example).

Author(s)

Javier Fernández-Macho, Dpt. of Quantitative Methods, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho at ehu.eus).

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A: Statistical Mechanics, vol. 490, p. 1226–1238. <DOI:10.1016/j.physa.2017.11.050>

Examples

```
## Based on data from Figure 7.8 in Gencay, Selcuk and Whitcher (2001)
## plus one random series.

library(wavemulcor)

data(exchange)
returns <- diff(log(as.matrix(exchange)))
returns <- ts(returns, start=1970, freq=12)
```

```
N <- dim(returns)[1]

wf <- "d4"
J <- trunc(log2(N))-3

set.seed(140859)

demusd.modwt <- brick.wall(modwt(returns[,"DEM.USD"], wf, J), wf)
jpyusd.modwt <- brick.wall(modwt(returns[,"JPY.USD"], wf, J), wf)
xrand.modwt <- brick.wall(modwt(rnorm(length(returns[,"DEM.USD"])), wf, J), wf)

# ---------------------------

xx <- list(demusd.modwt, jpyusd.modwt, xrand.modwt)
names(xx) <- c("DEM.USD","JPY.USD","rand")

Lst <- wave.multiple.regression(xx)

# ---------------------------

##Producing correlation plot
plot_wave.multiple.correlation(Lst)

##Producing regression plot
plot_wave.multiple.regression(Lst)
```

---

xrand *Correlation structural breaks data*

Description

Simulated data showing correlation structural breaks in Figure 4 of Fernández-Macho (2018).

Usage

```
data("xrand")
```

Format

A data frame with 512 observations on the following 2 variables.

`xrand1` a numeric vector

`xrand2` a numeric vector

Details

$xrand1[t]$ and $xrand2[t]$ are highly correlated at low frequencies (long timescales) but uncorrelated at high frequencies (short timescales). However, during a period of time spanning the second third of the sample ($T/3 < t < 2T/3$) that behavior is reversed so that data become highly correlated at short timescales but uncorrelated at low frequencies.

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A, 492, https://doi.org/10.1016/j.physa.2017.11.050

Examples

```
data(xrand)
## maybe str(xrand) ; plot(xrand) ...
```

`xrand1` *Correlation structural breaks variable 1*

Description

Simulated data showing correlation structural breaks in Figure 4 of Fernández-Macho (2018).

Usage

```
data("xrand")
```

Format

A data frame with 512 observations on 1 variables.

`xrand1` a numeric vector

Details

$xrand1[t]$ and $xrand2[t]$ are highly correlated at low frequencies (long timescales) but uncorrelated at high frequencies (short timescales). However, during a period of time spanning the second third of the sample ($T/3 < t < 2T/3$) that behavior is reversed so that data become highly correlated at short timescales but uncorrelated at low frequencies.

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A, 492, https://doi.org/10.1016/j.physa.2017.11.050

Examples

```
data(xrand)
## maybe str(xrand) ; plot(xrand) ...
```

xrand2 *Correlation structural breaks variable 2*

Description

Simulated data showing correlation structural breaks in Figure 4 of Fernández-Macho (2018).

Usage

```
data("xrand")
```

Format

A data frame with 512 observations on 1 variables.

`xrand2` a numeric vector

Details

$xrand1[t]$ and $xrand2[t]$ are highly correlated at low frequencies (long timescales) but uncorrelated at high frequencies (short timescales). However, during a period of time spanning the second third of the sample ($T/3 < t < 2T/3$) that behavior is reversed so that data become highly correlated at short timescales but uncorrelated at low frequencies.

References

Fernández-Macho, J., 2018. Time-localized wavelet multiple regression and correlation, Physica A, 492, https://doi.org/10.1016/j.physa.2017.11.050

Examples

```
data(xrand)
## maybe str(xrand) ; plot(xrand) ...
```

Index