

Package ‘wavemulcor’

June 8, 2018

Title Wavelet Routines for Global and Local Multiple Correlation

Version 2.2.1

Description Wavelet routines that calculate single sets of wavelet multiple correlations (WMC) and cross-correlations (WMCC) out of n variables.

They 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) <DOI:10.1017/CBO9780511841040>;

Gençay, Selçuk and Whitcher (2002) <DOI:10.1016/B978-012279670-8.50013-6> 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 an auxiliary function that calculates local multiple correlations (LMC). 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.

License GPL (>= 2)

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

Suggests `plot3D`

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Author Javier Fernandez-Macho [aut, cre]

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

Repository CRAN

Date/Publication 2018-06-08 21:32:58 UTC

R topics documented:

local.multiple.correlation	2
wave.local.multiple.correlation	4
wave.multiple.correlation	8
wave.multiple.cross.correlation	10
xrand	12
xrand1	13
xrand2	14

Index **15**

local.multiple.correlation

Routine for local multiple correlation

Description

Produces an estimate of the multiscale local multiple correlation (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 (multiscaled) time series, usually the outcomes of dwt or modwt, <i>i.e.</i> <code>xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)</code>
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>i.e.</i> 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:	vector with as many rows as observations providing the point estimates for the local multiple correlation.
lo:	vector with as many rows as observations providing the lower bounds from the confidence interval.
up:	vector with as many rows as observations providing the upper bounds from the confidence interval.
YmaxR:	numeric vector 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, <i>wlmc</i> chooses at each weach value in time the variable maximizing the multiple correlation.

Author(s)

Javier Fernández-Macho, Dpt. of Econometrics and Statistics, & Instituto de Economía Pública, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho@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 (2017).

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 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("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)

```

```

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>i.e.</i> <code>xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)</code>
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>i.e.</i> 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 wlmr 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 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 $n(n-1)/2$ $[JxT]$ 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:	matrix with as many rows as observations and as many columns as levels in the wavelet transform object providing the point estimates for the wavelet local multiple correlation.
lo:	matrix with as many rows as observations and as many columns as levels in the wavelet transform object providing the lower bounds from the confidence interval.

- up: matrix with as many rows as observations and as many columns as levels in the wavelet transform object providing the 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, *wlmc* 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 Econometrics and Statistics, & Instituto de Economía Pública, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho@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 (2017).

library(wavemulcor)
library(plot3D)
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 <- "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")

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"

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 = 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.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>i.e.</i> <code>xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)</code>
N	length of the time series
p	one minus the two-sided p-value for the confidence interval, <i>i.e.</i> 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: 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 Econometrics and Statistics, & Instituto de Economía Pública, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho@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)
wf <- "d4"
J <- 6

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

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

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, ymaxr = NULL)
```

Arguments

xx	A list of n (multiscaled) time series, usually the outcomes of dwt or modwt, <i>i.e.</i> <code>xx <- list(v1.modwt.bw, v2.modwt.bw, v3.modwt.bw)</code>
lag.max	maximum lag. If not set, it defaults to half the square root of the length of the original series.
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 $n(n-1)/2 \cdot J$ 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: 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.

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 Econometrics and Statistics, & Instituto de Economía Pública, University of the Basque Country, Agirre Lehendakari etorb. 83, E48015 BILBAO, Spain. (email: javier.fernandezmacho@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)
wf <- "d4"
J <- 6
lmax <- 36
n <- dim(returns)[1]

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

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

Lst <- wave.multiple.cross.correlation(xx, lmax)
returns.cross.cor <- as.matrix(Lst$xy.mulcor[1:J,])
```

```

YmaxR <- Lst$YmaxR

exchange.names <- c("DEM.USD", "JPY.USD", "RAND")
rownames(returns.cross.cor)<-rownames(returns.cross.cor,
  do.NULL = FALSE, prefix = "Level ")
lags <- length(-lmax:lmax)

lower.ci <- tanh(atanh(returns.cross.cor) - qnorm(0.975) /
sqrt(matrix(trunc(n/2^(1:J)), nrow=J, ncol=lags)- 3))
upper.ci <- tanh(atanh(returns.cross.cor) + qnorm(0.975) /
sqrt(matrix(trunc(n/2^(1:J)), nrow=J, ncol=lags)- 3))

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:(2*lmax+1)),returns.cross.cor[i,], type="l", lty=1, ylim=c(-1,1),
  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))
lines(lower.ci[i,], lty=1, col=2) ##Add Connected Line Segments to a Plot
lines(upper.ci[i,], lty=1, col=2)
abline(h=0,v=lmax+1) ##Add Straight horiz and vert Lines to a Plot
text(1,1, labels=exchange.names[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)

```

xrand

Correlation structural breaks data

Description

Simulated data showing correlation structural breaks in Figure 4 of Fernández-Macho (2017).

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., 2017. Time-localized wavelet multiple regression and correlation, (mimeo).

Examples

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

xrand1	<i>Correlation structural breaks variable 1</i>
--------	---

Description

Simulated data showing correlation structural breaks in Figure 4 of Fernández-Macho (2017).

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., 2017. Time-localized wavelet multiple regression and correlation, (mimeo).

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 (2017).

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., 2017. Time-localized wavelet multiple regression and correlation, (mimeo).

Examples

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

Index

- *Topic **correlation**
 - wave.multiple.correlation, 8
 - *Topic **cross-correlation**
 - wave.multiple.cross.correlation, 10
 - *Topic **datasets**
 - xrand, 12
 - xrand1, 13
 - xrand2, 14
 - *Topic **local correlation**
 - local.multiple.correlation, 2
 - wave.local.multiple.correlation, 4
 - *Topic **local regression**
 - local.multiple.correlation, 2
 - wave.local.multiple.correlation, 4
 - *Topic **moving regression**
 - local.multiple.correlation, 2
 - wave.local.multiple.correlation, 4
 - *Topic **multivariate**
 - local.multiple.correlation, 2
 - wave.local.multiple.correlation, 4
 - wave.multiple.correlation, 8
 - wave.multiple.cross.correlation, 10
 - *Topic **statistics**
 - local.multiple.correlation, 2
 - wave.local.multiple.correlation, 4
 - wave.multiple.correlation, 8
 - wave.multiple.cross.correlation, 10
 - *Topic **wavelet**
 - wave.local.multiple.correlation, 4
 - wave.multiple.correlation, 8
 - wave.multiple.cross.correlation, 10
- lmc (local.multiple.correlation), 2
- local.multiple.correlation, 2
- wave.local.multiple.correlation, 4
- wave.multiple.correlation, 8
- wave.multiple.cross.correlation, 10
- wlmc (wave.local.multiple.correlation), 4
- wmc (wave.multiple.correlation), 8
- wmcc (wave.multiple.cross.correlation), 10
- xrand, 12
- xrand1, 13
- xrand2, 14