Package ‘ASSA’

May 14, 2018

Version 1.0

Date 2018-05-08

Title Applied Singular Spectrum Analysis (ASSA)

Description
Functions to model and decompose time series into principal components using singular spectrum analysis (de Carvalho and Rua (2017) <doi:10.1016/j.ijforecast.2015.09.004>; de Carvalho et al (2012) <doi:10.1016/j.econlet.2011.09.007>).

Author Miguel de Carvalho [aut, cre],
Gabriel Martos [aut]

Depends R (>= 3.0.1)

Maintainer Miguel de Carvalho <miguel.decarvalho@ed.ac.uk>

License GPL (>= 3)

Repository CRAN

Imports stats

NeedsCompilation yes

Date/Publication 2018-05-14 12:51:57 UTC

R topics documented:

`ASSA-package` *Applied Singular Spectrum Analysis*

Description

The package **ASSA** is an add-on tool for R that implements time series decomposition and modeling methods based on singular spectrum analysis (SSA) and multivariate singular spectrum analysis (MSSA). The current version of the package includes tools tailored for extracting business cycles, and for computing trendlines.

For a complete list of functions, data sets and documentation, type `help.start()` and follow the link to **ASSA** on the Package Index.

Funding from the project INTERSTATA (Interdisciplinary Statistics in Action), by the International Research and Partnership Fund, is gratefully acknowledged.

`bmssa` *Multivariate Singular Spectrum Business Cycle Indicator*

Description

Computes a business cycle indicator using multivariate singular spectrum analysis.

Usage

```
bmssa(y, l = 32)
```

Arguments

`y`	multivariate time series of economic activity data from which the cycle is to be extracted; the first column is reserved to Gross Domestic Product (GDP).
`l`	window length; by default, `l = 32`.

Details

The business cycle indicator produced using this routine is based on methods proposed in de Carvalho and Rua (2017). A quick summary of the method is as follows. Multivariate singular spectrum analysis is used to decompose a multivariate time series (`y`) into principal components, and a Fisher g statistic automatically selects elementary reconstructed components (`erc`) within business cycle frequencies. The indicator results from adding elementary reconstructed components within business cycle frequencies. The `plot` method depicts the resulting business cycle indicator, and the `print` method reports the business cycle indicator along with the components selected by the Fisher g statistic.

Value

`cycle`	time series with the business cycle indicator.
`sfisher`	vector with indices of elementary reconstructed components selected with Fisher g statistic; see details.
`erc`	time series with elementary reconstructed components resulting from targeted grouping based on a Fisher g statistic.
`l`	window length.

Author(s)

Miguel de Carvalho.

References

de Carvalho, M., Rodrigues, P., and Rua, A. (2012). Tracking the US business cycle with a singular spectrum analysis. *Economics Letters*, **114**, 32–35.

de Carvalho, M. and Rua, A. (2017). Real-time nowcasting the US output gap: Singular spectrum analysis at work. *International Journal of Forecasting*, **33**, 185–198.

See Also

See `combplot` for a chart of the selected elementary reconstructed components from which the business cycle indicator results. See `bssa` for a univariate version of the method.

Examples

```
## Tracking the US Business Cycle (de Carvalho et al, 2017; Fig. 6)
data(GDPIP)
fit <- bmssa(log(GDPIP))
plot(fit)
print(fit)
```

`brexit` *Brexit Poll Tracker*

Description

The data consist of 267 polls conducted before the June 23 2016 EU referendum, which took place in the UK.

Usage

```
brexit
```

Format

A dataframe with 272 observations on six variables. The object is of class `list`.

Source

Financial Times (FT) Brexit poll tracker.

References

de Carvalho, M. and Martos, G. (2018). Brexit: Tracking and disentangling the sentiment towards leaving the EU. Submitted.

Examples

```
## Leave--stay plot (de Carvalho and Martos, 2018; Fig. 1)
data(brexit)
attach(brexit)
par(pty = "s")
plot(leave[(leave > stay)], stay[(leave > stay)],
     xlim = c(22, 66), ylim = c(22, 66), pch = 16, col = "red",
     xlab = "Leave", ylab = "Stay")
points(leave[(stay > leave)], stay[(stay > leave)],
       pch = 16, col = "blue")
points(leave[(stay == leave)], stay[(stay == leave)],
       pch = 24)
abline(a = 0, b = 1, lwd = 3)
```

---

bssa *Singular Spectrum Business Cycle Indicator*

Description

Computes a business cycle indicator using singular spectrum analysis.

Usage

```
bssa(y, l = 32)
```

Arguments

`y`	time series of economic activity data from which the cycle is to be extracted.
`l`	window length; by default, `l = 32`.

Details

The business cycle indicator produced using this routine is based on methods proposed in de Carvalho et al (2012) and de Carvalho and Rua (2017). A quick summary of the method is as follows: Singular spectrum analysis is used to decompose a GDP time series (`y`) into principal components, and a Fisher g statistic automatically selects elementary reconstructed components (`erc`) within business cycle frequencies. The indicator results from adding principal components within business cycle frequencies. The `plot` method depicts the resulting business cycle indicator, and the `print` method reports the business cycle indicator along with the components selected by the Fisher g statistic.

Value

`cycle`	time series with the business cycle indicator.
`sfisher`	vector with indices of principal components selected with Fisher g statistic; see details.
`erc`	time series with elementary reconstructed components resulting from targeted grouping based on a Fisher g statistic.
`l`	window length.

Author(s)

Miguel de Carvalho.

References

de Carvalho, M., Rodrigues, P., and Rua, A. (2012) Tracking the US business cycle with a singular spectrum analysis. *Economics Letters*, **114**, 32–35.

de Carvalho, M. and Rua, A. (2017) Real-time nowcasting the US output gap: Singular spectrum analysis at work. *International Journal of Forecasting*, **33**, 185–198.

See Also

See `combplot` for a chart of the selected elementary reconstructed components from which the business cycle indicator results. See `bmssa` for a multivariate version of the method.

Examples

```
## Tracking the US Business Cycle (de Carvalho et al, 2017; Fig. 6)
data(GDPIP)
fit <- bssa(log(GDPIP[, 1]))
plot(fit)
print(fit)
```

`combplot` *Comb-plot*

Description

Produces a comb-plot for visualizing what principal components are used for producing the (multivariate) singular spectrum business cycle indicator.

Usage

```
combplot(fit)
```

Arguments

`fit`	a `bssa` or a `bmssa` object.

Details

`combplot` yields a comb-plot indentifying the indices of the components selected according to the Fisher g statistic, along with the corresponding principal components; see de Carvalho and Rua (2017, p. 190) for a definition.

Author(s)

Miguel de Carvalho.

References

de Carvalho, M. and Rua, A. (2017). Real-time nowcasting the US output gap: Singular spectrum analysis at work. *International Journal of Forecasting*, **33**, 185–198.

See Also

`bssa`.

Examples

```
## Tracking the US Business Cycle (de Carvalho and Rua, 2017; Fig. 5)
data(GDPIP)
fit <- bssa(log(GDPIP[, 1]))
combplot(fit)
```

`GDPIP` *A Real-time Vintage of GDP and IP for the US Economy*

Description

US GDP (Gross Domestic Product) and IP (Industrial Production) ranging from from 1947 (Q1) to 2013 (Q4); the data correspond to a real-time vintage.

Usage

```
GDPIP
```

Format

A bivariate time series with 268 observations on two variables. The object is of class `mts`.

Source

Federal Reserve Bank of Philadelphia.

References

de Carvalho, M. and Rua, A. (2017). Real-time nowcasting the US output gap: Singular spectrum analysis at work. *International Journal of Forecasting*, **33**, 185–198.

Examples

```
## Plotting GDP and IP (de Carvalho and Rua, 2017; Fig. 4)
data(GDPIP)
par(mar = c(5, 4, 4, 5) + .1)
plot(GDPIP[, 1], type = "l",
     xlab = "Time", ylab = "Gross Domestic Product (GDP)",
     lwd = 3, col = "red", cex.lab = 1.4, cex.axis = 1.4)
par(new = TRUE)
plot(GDPIP[, 2], type = "l", xaxt = "n", yaxt = "n",
     xlab = "", ylab = "", lwd = 3, col = "blue", cex.axis = 1.4)
axis(4)
mtext("Industrial Production (IP)", side = 4, line = 3, cex = 1.4)
legend("topleft", col = c("red", "blue"),
       lty = 1, lwd = 3, legend = c("GDP", "IP"))
```

`msst` *Multivariate Singular Spectrum Trendlines*

Description

Computes trendlines for multivariate time series data using multivariate singular spectrum analysis.

Usage

```
msst(y, l = "automatic", m = "automatic", vertical = TRUE)
```

Arguments

`y`	mtsframe object containing raw data.
`l`	window length; the string `"automatic"` sets the default option `l = ceiling(y$n + 1) / y$D` for `vertical` and `ceiling(D * (y$n + 1) / (y$D + 1))`.
`m`	number of leading eigentriples. An automatic criterion based on the cumulative periodogram of the residuals is provided by default by using the string `"automatic"`.
`vertical`	logical; if `TRUE` the trajectory matrices are stacked vertically, otherwise the bind is horizontal.

Details

Multivariate singular spectrum analysis is used to decompose time series data (`y`) into principal components, and a cumulative periodogram-based criterion automatically learns about what elementary reconstructed components (`erc`) contribute to the signal; see de Carvalho and Martos (2018) for details. The trendline results from adding elementary reconstructed components selected by the cumulative periodogram of the residuals. The `plot` method depicts the trendlines, and the `print` method reports the trendlines along with the components selected by the cumulative periodogram-based criterion.

Value

`trendline` mtsframe object with trendline estimation from targeted grouping based on a cumulative periodogram criterion (or according to the number of components specified in vector `m`).

`residuals` mtsframe object with the residuals from targeted grouping based on a cumulative periodogram criterion (or according to the number of components specified in vector `m`).

`erc` list with elementary reconstructed components.

`eigen.val` vector with the singular values of the trajectory matrix.

`l` window length.

`selected.components` vector with number of components selected on each dimension.

`selection.criteria` vector indicating if the null hypothesis of white noise is rejected along the dimensions (0: not rejected, 1: rejected).

`rank` rank of the trajectory matrix.

Author(s)

Gabriel Martos and Miguel de Carvalho

References

de Carvalho, M. and Martos, G. (2018). Brexit: Tracking and disentangling the sentiment towards leaving the EU. Submitted.

See Also

See `msstc` for a similar routine yielding trendlines for multivariate time series of compositional data.

Examples

```
## SIMULATED DATA EXAMPLE
N <- 500
t <- seq(.01, 5, length = N)
Y <- cbind(t^3 - 9 * t^2 + 23 * t + rnorm(N, 0, 1),
           10 * sin(3 * t) / t + rnorm(N, 0, 1))
y <- mtsframe(date = t, Y)
fit <- msst(y)
matplot(t, Y, col = "gray", ylim = c(-10, 31),
        xlab = "time", pch = 16)
lines(t, t^3 - 9 * t^2 + 23 * t)
lines(t, fit$trendline$Y[,1], col = "red")
lines(t, 10 * sin(3 * t) / t, col = "black")
lines(t, fit$trendline$Y[,2], col = "red")

## BREXIT DATA EXAMPLE
```

```
## (de Carvalho and Martos, 2018; Fig. 1)
data(brexit)
attach(brexit)
y <- mtsframe(date, brexit[, 1:3] / 100)
fit <- msst(y)

## Window length and number of components automatically selected on
## each dimension:
fit$l
fit$selected.components

## Plot trendlines (de Carvalho and Martos, 2018; Fig. 1)
plot(fit, options = list(type = "trendlines"), xlab="time",
     col=c("blue", "red", "black"), lwd = 2, lty = c(1, 2, 3))

## Plot elementary reconstructed components
## (de Carvalho and Martos, 2018; Fig. 5)
plot(fit, options = list(type = "components", ncomp = 1:3))

## Plot cumulative periodograms (with 95% confidence bands)
par(mfrow = c(1, 3))
plot(fit, options = list(type = "cpgrams",
          series.names = c('Leave','Stay','Undecided')) )

## Scree-plot (with 95% confidence bands)
par(mfrow = c(1, 1))
plot(fit, options = list(type = "screeplots", ncomp = 1:10),
     type = "b", pch = 20, lwd = 2, main='Scree plot')
```

`msstc` *Multivariate Singular Spectrum Trendlines for Compositional Data*

Description

Computes trendlines on the unit simplex for multivariate time series data using multivariate singular spectrum analysis.

Usage

```
msstc(y, l = 'automatic', m = 'automatic', vertical = TRUE)
```

Arguments

`y`	mtsframe object containing data.
`l`	window length; the string `'automatic'` sets the default option `l = ceiling(y$n + 1) / y$D`.
`m`	number of leading eigentriples; the string `'automatic'` yields a vector containing the number of components in each dimension to be used in the trendline estimation. An automatic criterion based on the cumulative periodogram of the residuals is provided by default; see details.

vertical — logical; if TRUE the trajectory matrices are stacked vertically, otherwise the bind is horizontal.

Details

The trendline produced using this routine is based on the methods proposed in de Carvalho and Martos (2018). A quick summary of the method is as follows. Multivariate singular spectrum analysis is used to decompose time series data (y) into principal components, and a cumulative periodogram-based criterion automatically learns about what elementary reconstructed components (erc) contribute to the signal; see de Carvalho and Martos (2018) for details. The trendline results from adding elementary reconstructed components selected by the cumulative periodogram, and after projecting into the unit simplex. The plot method depicts the trendlines, and the print method reports the trendlines along with the components selected by the cumulative periodogram-based criterion.

Value

trendline — mtsframe object with trendline estimation from targeted grouping based on a cumulative periodogram criterion (or according to the number of components specified in vector m).

residuals — mtsframe object with the residuals from targeted grouping based on a cumulative periodogram criterion (or according to the number of components specified in vector m).

erc — list with elementary reconstructed components.

eigen.val — vector with the singular values of the trajectory matrix.

l — window length.

selected.components — vector with number of components selected on each dimension.

selection.criteria — a vector indicating if the null hypothesis of white noise is rejected along the dimensions (0: not rejected, 1: rejected).

rank — rank of the trajectory matrix.

Author(s)

Gabriel Martos and Miguel de Carvalho

References

de Carvalho, M. and Martos, G. (2018). Brexit: Tracking and disentangling the sentiment towards leaving the EU. Submitted.

See Also

See msst for a similar routine yielding trendlines for multivariate time series, but which does not project the pointwise estimates to the unit simplex.

Examples

```
## Brexit data and MSSA on the simplex
## (de Carvalho and Martos, 2018; Fig. 1)
data(brexit)
attach(brexit)
y <- mtsframe(date, brexit[, 1:3] / 100)
fit <- msstc(y)

## Window length and number of components automatically selected on
## each dimension:
fit$l
fit$selected.components

## Plot trendlines (de Carvalho and Martos, 2018; Fig. 1)
plot(fit, options = list(type = "trendlines"), xlab="time",
     col=c("blue", "red", "black"), lwd = 2, lty = c(1, 2, 3))

## Plot elementary reconstructed components
## (de Carvalho and Martos, 2018; Fig. 5)
plot(fit, options = list(type = "components", ncomp = 1:3))

## Plot cumulative periodograms (with 95% confidence bands)
par(mfrow = c(1, 3))
plot(fit, options = list(type = "cpgrams",
          series.names = c('Leave','Stay','Undecided')) )

## Scree-plot (with 95% confidence bands)
par(mfrow = c(1, 1))
plot(fit, options = list(type = "screeplots", ncomp = 1:10),
     type = "b", pch = 20, lwd = 2, main='Scree plot')
```

---

mtsframe *Multivariate Time Series Frame Objects*

Description

The function `mtsframe` is used to create mutivariate time series objects to be used in combination with the functions in the package ASSA.

Usage

```
mtsframe(dates, Y)
```

Arguments

`dates`	dates at which observations took place (in case of using an index, the format of this field will be taken as numeric.)
`Y`	matrix with time-series in columns and observations in rows.

Examples

```
data(brexit)
attach(brexit)
y <- mtsframe(date, Y = brexit[, 1:3])
print(y)
# Ploting the time series under study (blue = Leave, red = Stay, black = Undecided)
plot(y, col = c('blue', 'red', 'black'), time.format = '%m-%y')
# When 'date' is a time index, the time.format specification is omitted.
```

---

sst *Singular Spectrum Trendline*

Description

Computes a trendline for univariate time series data using singular spectrum analysis.

Usage

```
sst(y, l = "automatic", m = "automatic")
```

Arguments

`y`	mtsframe format data containing data. While `y` can include several time series, a more appropriate method for multivariate time series is `msst`.
`l`	window length; the string `"automatic"` automatic sets the default option `l = ceiling(y$n + 1) / 2`.
`m`	number of leading eigentriples; the string `"automatic"` yields a vector containing the number of components in each dimension to be used in the trendline estimation. An automatic criterion based on the cumulative periodogram of the residuals is provided by default; see details.

Details

Singular spectrum analysis decompose time series data (`y`) into principal components, and a cumulative periodogram-based criterion learn about elementary reconstructed components (`erc`) that contribute to the signal. The trendline results from adding principal components selected by a cumulative periodogram-based criteria; see de Carvalho and Martos (2018, Section 4.1). The `plot` method yields the resulting trendlines along with the data; `options` for the plot method are give by a list including the strings `"trendlines"`, `"components"`, `"cpgrams"`, and `"screeplots"`, along with a set of values (`ncomp`) indicating the components on which these diagnostics are to be depicted (e.g. `plot(fit, options = list(type = "components", ncomp = 1:3))`.

Value

`trendline`	mtsframe object with trendline estimation from targeted grouping based on a cumulative periodogram criterion (or according to the number of components specified in vector `m`).
`residuals`	mtsframe object with the residuals from targeted grouping based on a cumulative periodogram criterion (or according to the number of components specified in vector `m`).
`erc`	list with elementary reconstructed components.
`eigen.val`	vector with the singular values of the trajectory matrix.
`l`	window length.
`selected.components`	vector with components selected on each dimension.
`selection.criteria`	a vector indicating if the hypothesis of white noise residual is rejected along the dimensions (0: not rejected, 1: rejected).
`rank`	rank of the trajectory matrix.

Author(s)

Gabriel Martos and Miguel de Carvalho

References

de Carvalho, M. and Martos, G. (2018). Brexit: Tracking and disentangling the sentiment towards leaving the EU. Submitted.

See Also

See `msst` for a version of the routine for multivariate time series, and see `msstc` for a version of the routine for multivariate time series of compositional data.

Examples

```
## SIMULATED DATA EXAMPLES
## Example 1
N <- 500
t <- seq(.01, 5, length = N)
Y <- t^3 - 9 * t^2 + 23 * t + rnorm(N, 0, 1)
y <- mtsframe(date = t, Y)
fit <- sst(y)
plot(t, Y, col = "gray", ylab = "",
     xlab = "time", pch = 16, ylim = c(-10, 31))
lines(t, t^3 - 9 * t^2 + 23 * t)
lines(t, fit$trendline$Y, col = "red")
## Example 2
N <- 500
t <- seq(.01, 5, length = N)
Y <- 10 * sin(3 * t) / t + rnorm(N, 0, 1)
```

```
y <- mtsframe(date = t, Y)
fit <- msst(y)
fit$selected.components
plot(t, Y, col = "gray", ylab = "",
     xlab = "time", pch = 16, ylim = c(-10, 31))
lines(t, 10 * sin(3 * t) / t, col = "black")
lines(t, fit$trendline$Y, col = "red")

## BREXIT DATA EXAMPLE
## Note: sst also can deal with several time series as input yet the
## most appropriate method for multivariate time series is msst
data(brexit)
attach(brexit)
y <- mtsframe(date, brexit[, 1:3] / 100)
fit <- sst(y)
plot(fit)

## Number of components automatically selected
fit$selected.components

## Chronological plot (de Carvalho and Martos, 2018; Fig. 1)
plot(fit, options = list(type = "trendlines"), xlab = "time",
     col = c("blue", "red", "black"), lwd = 2, lty = c(1, 2, 3))

## Plot elementary reconstructed components
## (de Carvalho and Martos, 2018; Fig. 3)
plot(fit, options = list(type = "components", serie = 1, ncomp = 1:2))
plot(fit, options = list(type = "components", serie = 2, ncomp = 1:2))
plot(fit, options = list(type = "components", serie = 3, ncomp = 1  ))

## Plot cumulative periodograms (with 95% confidence bands)
par(mfrow=c(1,3))
plot(fit, options = list(type = "cpgrams",
           series.names = c('Leave','Stay','Undecided')))

## Scree-plot (with 95% confidence bands)
par(mfrow = c(1, 3))
plot(fit, options = list(type = "screeplots", ncomp = 1:10,
          series.names = c('Leave','Stay','Undecided')),
            type = "b", pch = 20, lwd = 2)
```

Index