# Package 'ASSA'

October 12, 2022

**Version** 2.0

**Date** 2020-11-05

**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 (>= 3.6.0)

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

**License** GPL (>= 3)

**Repository** CRAN

**Imports** stats

**LazyData** true

**NeedsCompilation** yes

**Date/Publication** 2020-11-20 10:40:13 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. (2020). Brexit: Tracking and disentangling the sentiment towards leaving the EU. *International Journal of Forecasting*, **36**, 1128–1137.

## 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"))
```

isst *Interval Singular Spectrum Trendlines*

## Description

Computes the trendline estimation for interval time series data using singular spectrum analysis.

## Usage

```
isst(y, l= 'automatic' , m = 'automatic')
```

## Arguments

| | |
|---|---|
| y | itsmframe data corresponding to univariate interval time series data. |
| l | window length; the string 'automatic' sets the default option l = ceiling(y$n + 1) / 2. |
| m | number of leading eigentriples; the string "automatic" yields an automatic criterion for choosing m based on the cumulative periodogram of the residuals; 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 "trendline", "components", "cpgram", and "screeplot", 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 | itsframe object with interval trendline estimation from targeted grouping based on a cumulative periodogram criterion (or according to the number of components specified in m). |
| l | window length. |
| 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". |
| residuals | itsframe object with the residuals from targeted grouping based on a cumulative periodogram criterion (or according to the number of components specified in m). |
| svd | Singular value decomposition corresponding to the trajectory matrix. |
| erc | elementary reconstructed components. |
| observations | itsframe object with the raw data. |

## References

de Carvalho, M. and Martos, G. (2020). Modeling Interval Trendlines: Symbolic Singular Spectrum Analysis for Interval Time Series. Submitted (available on arXiv).

## See Also

See misst for a version of the routine for multivariate interval value time series.

## Examples

```
# Merval data example:
data(merval)
id.data <-  rev(which(merval[,1]>'2015-12-31' & merval[,1]<'2020-10-01') )
y <-  itsframe(date=merval[id.data,1], a=merval[id.data,2], b=merval[id.data,3]);

isst_output <- isst(y ,l = 'automatic', m = 'automatic')
print(isst_output)

# Estimated trendlines
plot(isst_output)

## Scree-plot
plot(isst_output, options = list(type = "screeplot", ncomp = 1:10),
     type = "b", pch = 20, lwd = 2)

# Elementary reconstructed components
plot(isst_output, options=list(type='components',ncomp=1:3),
     xlab='Time')

# cpgram's ('a=low' and 'b=high')
plot(isst_output, options = list(type='cpgram'))
# Setting m = 'automatic' (default option) to obtain cpgrams inside the bandwiths.

#################################
###  Forecasting with isst     ###
#################################
pred <- predict(isst_output, p = 5)
head(pred$forecast,3) # Forecasted interval data.

attributes(pred)
pred$coefficients[1:5] # linear recurrence parameters.
# End
```

---

# itsframe *Interval Time Series Frame Objects*

---

## Description

The function itsframe creates a univariate interval time series object to be used in combination with the functions in the package ASSA.

## Usage

```
itsframe(dates, a, b)
```

## Arguments

| | |
|---|---|
| dates | dates at which observations took place. |
| a | vector with lower interval time-series values sorted in ascendant way (first element in 'a' corresponds to the oldest value of the interval series and last element in 'a' corresponds to the newest value). |
| b | vector with upper interval time-series values sorted in ascendant way (first element in 'b' corresponds to the oldest value of the interval series and last element in 'b' corresponds to the newest value). |

## References

de Carvalho, M. and Martos, G. (2020). Modeling Interval Trendlines: Symbolic Singular Spectrum Analysis for Interval Time Series. Submitted (available on arXiv).

## Examples

```
data(merval)
id.data <-  rev(which(merval[,1]>'2015-12-31' & merval[,1]<'2020-10-01') )
y <-  itsframe(date=merval[id.data,1], a=merval[id.data,2], b=merval[id.data,3]);

plot(y, main = 'MERVAL')
```

---

# merval *MERVAL interval data*

---

## Description

Raw interval data series corresponding to weekly minimum and maximum values of MERVAL index ranging from January 1st 2016 to September 30th 2020.

## Usage

```
merval
```

## Format

A dataframe with 353 observations. The object is of class `list`.

## Source

Yahoo Finance.

## References

de Carvalho, M. and Martos, G. (2020). Modeling Interval Trendlines: Symbolic Singular Spectrum Analysis for Interval Time Series. Submitted (available on arXiv).

## Examples

```
data(merval)
head(merval,3)
```

---

`misst` *Multivariate Interval Singular Spectrum Trendlines*

---

## Description

Computes a trendline for multivariate interval data using singular spectrum analysis.

## Usage

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

## Arguments

| | |
|---|---|
| `y` | object of class `mitsframe` (multivariate interval time series data). |
| `l` | window length; the string `"automatic"` sets the default option `l = ceiling(y$n + 1) / (y$D+1)` 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 interval 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 | mitsframe object with interval trendline estimation from targeted grouping based on a cumulative periodogram criterion (or according to the number of components specified in vector m). |
| l | window length. |
| m | vector with number of components selected on each dimension. |
| vertical | flag indicating if the trajectory matrices where stacked vertically. |
| residuals | mitsframe object with the interval residuals from targeted grouping based on a cumulative periodogram criterion (or according to the number of components specified in vector m). |
| svd | the Singular Value Decomposition of the trajectory matrix. |
| erc | list with elementary reconstructed components. |
| observations | mitsframe object with the raw data y. |

## References

de Carvalho, M. and Martos, G. (2020). Modeling Interval Trendlines: Symbolic Singular Spectrum Analysis for Interval Time Series. Submitted (available on arXiv).

## See Also

See msst for a similar routine yielding trendlines for standard multivariate time series of data.

## Examples

```
muX.a = function(t){ 8 + t + sin(pi*t) }    ;    muX.b = function(t){ muX.a(t) + 2 }
muY.a = function(t){sqrt(t) + cos(pi*t/2) }    ;  muY.b = function(t){ 2*muY.a(t) + 2 }
N = 100; t=seq(0.1,2*pi,length = N);
set.seed(1)
e.x = rnorm(100); e.y = rnorm(100);
a.X = muX.a(t) + e.x; b.X = a.X + 2
a.Y = muY.a(t) + e.y        ; b.Y = 2*a.Y + 2

A <- cbind(a.X, a.Y); B <- cbind(b.X, b.Y)
y <- mitsframe(dates=t, A=A, B = B)

fit <- misst(y)
fit$l;
fit$m;
fit$vertical

# Estimated trendlines:
head(fit$trendlines$A,5)
head(fit$trendlines$B,5)

## Estimated interval trendlines
plot(fit)
```

```
## Scree-plot
plot(fit, options = list(type = "screeplots"))

## Per
plot(fit, options = list(type = "cpgrams"))

## ERC
plot(fit, options=list(type='components',ncomp=1:3))



##################################
### Forecasting with misst     ###
##################################
pred = predict(fit, p = 5)
pred$forecasts # Forecast organized in an array.
# End
```

---

`mitsframe` *Multivariate Interval Valued Time Series Frame Objects*

---

## Description

The function `mitsframe` is used to create interval-valued multivariate time series objects.

## Usage

```
mitsframe(dates, A, B)
```

## Arguments

| | |
|---|---|
| `dates` | a vector of dates at which observations took place. |
| `A` | matrix with time series in columns (dimensions) and observations in rows corresponding to the lowest values. Data must be sorted in ascendant way (first row in 'A' corresponds to the oldest values of the series and last row in 'A' corresponds to the newest values). |
| `B` | matrix with time series in columns (dimensions) and observations in rows corresponding to the lowest values. Data must be sorted in ascendant way (first row in 'A' corresponds to the oldest values of the series and last row in 'A' corresponds to the newest values). |

## Author(s)

Gabriel Martos and Miguel de Carvalho.

## References

de Carvalho, M. and Martos, G. (2020). Modeling Interval Trendlines: Symbolic Singular Spectrum Analysis for Interval Time Series. Submitted (available on arXiv).

## Examples

```
muX.a = function(t){ 8 + t + sin(pi*t) }    ;    muX.b = function(t){ muX.a(t) + 2 }
muY.a = function(t){sqrt(t) + cos(pi*t/2) }    ;  muY.b = function(t){ 2*muY.a(t) + 2 }
N = 100; t=seq(0.1,2*pi,length = N);
set.seed(1)
e.x = rnorm(100); e.y = rnorm(100);
a.X = muX.a(t) + e.x; b.X = a.X + 2
a.Y = muY.a(t) + e.y        ; b.Y = 2*a.Y + 2

A <- cbind(a.X, a.Y); B <- cbind(b.X, b.Y)
y <- mitsframe(dates=t, A=A, B = B)

plot(y) # standard plot.
```

---

`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(y$D * (y$n + 1) / (y$D + 1))` in the case of horizontal binding. |
| `m` | vector with the number of leading eigentriples on each dimension. An automatic criterion based on the cumulative periodogram of the residuals is provided by default by using the string `"automatic"`, see details. |
| `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`). |
| `l` | window length. |
| `m` | vector with number of components selected on each dimension. |
| `vertical` | flag indicating if the trajectory matrices where stacked vertically. |
| `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`). |
| `svd` | the Singular Value Decomposition of the trajectory matrix. |
| `erc` | list with elementary reconstructed components. |
| `observations` | mtsframe object with the observations `y`. |

## Author(s)

Gabriel Martos and Miguel de Carvalho

## References

de Carvalho, M. and Martos, G. (2020). Brexit: Tracking and disentangling the sentiment towards leaving the EU. *International Journal of Forecasting*, **36**, 1128–1137.

## See Also

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

## Examples

```
## SIMULATED EXAMPLE
t <- seq(0.05, 5, by = 0.05)
t2 <- seq(0.05, 6, by = 0.05)
p = length(t2)-length(t) # Forecasting horizon parameter:
n = length(t)
Y <- cbind(t^3 - 9 * t^2 + 23 * t + rnorm(n, 0, 1),
           10 * sin(3 * t) / t + rnorm(n, 0, 1))
y <- mtsframe(dates = t, Y)

fit.vertical <- msst(y)

pred.vertical <- predict(fit.vertical, p = p)
print(pred.vertical$forecast)

## 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 components automatically selected
fit$l; fit$m

## 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 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
par(mfrow = c(1, 1))
plot(fit, options = list(type = "screeplot", ncomp.scree = 1:10),
     type = "b", pch = 20, lwd = 2, main='Scree plot')

## Plot elementary reconstructed components
plot(fit, options = list(type = "components", ncomp = 1:2))
```

---

`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)` for `vertical` and `ceiling(y$D * (y$n + 1) / (y$D + 1))` in the case of horizontal binding. |
| `m` | vector with the number of leading eigentriples on each dimension. An automatic criterion based on the cumulative periodogram of the residuals is provided by default by using the string `"automatic"`, 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`). |
| `l` | window length. |
| `m` | vector with number of components selected on each dimension. |
| `vertical` | flag indicating if the trajectory matrices where stacked vertically. |
| `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`). |
| `svd` | the Singular Value Decomposition of the trajectory matrix. |
| `erc` | list with elementary reconstructed components. |
| `observations` | mtsframe object with the observations `y`. |

## Author(s)

Gabriel Martos and Miguel de Carvalho

## References

de Carvalho, M. and Martos, G. (2020). Brexit: Tracking and disentangling the sentiment towards leaving the EU. *International Journal of Forecasting*, **36**, 1128–1137.

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

```
fit <- msstc(y)
# Estimations on the simplex
rowSums(fit$trendlines$Y)

# Forecast also in the simplex
rowSums(predict(fit, p = 5)$forecast)


## Window length and components automatically selected
fit$l; fit$m

## 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 cumulative periodograms (with 95% confidence bands)
par(mfrow = c(1, 3))
plot(fit, options = list(type = "cpgrams") )

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

## Plot elementary reconstructed components
## (de Carvalho and Martos, 2020; Fig. 5)
plot(fit, options = list(type = "components", ncomp = 1:2))
```

---

`mtsframe` *Multivariate Time Series Frame Objects*

---

## Description

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

## Usage

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

## Arguments

| | |
|---|---|
| `dates` | dates at which observations took place. |
| `Y` | matrix with different time-series in columns (dimensions) and observations in rows. Values must be be sorted in ascendant way (first row in 'Y' corresponds to the oldest values of the series and last row in 'Y' corresponds to the newest values). |

## Examples

```
data(brexit); attach(brexit)
head(brexit, 3)
y <- mtsframe(date, Y = brexit[, 1:3])
print(y) # A 'list' with 4 elements: dates, series data matrix, and series length.

head(y$Y, 3)
y$n
y$D

plot(y) # standard plot.

# Customized plot (time.format is an additional feature to use when y$date is in 'date' format)
plot(y, time.format = '%Y' , col = c('blue','red','black'), lty = 2, type = 'p',
     pch = 20, main = 'Brexit data', xlab = 'Year', ylab ='Trendline estimations')
```

---

`predict` *Forecasting with Singular Spectrum Trendline*

---

## Description

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

## Usage

```
predict(fitted.model, p = 1)
```

## Arguments

| | |
|---|---|
| `fitted.model` | estimated model using the functions in the package. |
| `p` | the horizon to produce forecasts. |

## Details

predict is a wrapper function for predictions from the results of various singular spectrum model fitting functions on the package. The function invokes particular methods which depend on the class of the first argument (i.e. sst,msst, msstc, isst, misst, etc) .

## Value

| | |
|---|---|
| `forecast` | Matrix containing in columns the dimensions and in rows the forecasts. |
| `a` | Parameters corresponding to the linear recurrence formula. |

## Author(s)

Gabriel Martos and Miguel de Carvalho

## References

de Carvalho, M. and Martos, G. (2020). Brexit: Tracking and disentangling the sentiment towards leaving the EU. *International Journal of Forecasting*, **36**, 1128–1137. de Carvalho, M. and Martos, G. (2020). Modeling Interval Trendlines: Symbolic Singular Spectrum Analysis for Interval Time Series. Submitted (available on arXiv).

## See Also

See `sst`,`msst`, `msstc`, `isst`, `misst` for a version of different models.

## Examples

```
## SIMULATED DATA EXAMPLE
set.seed(1)
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)

# Forecasting:
predict(fit, p = 5)$forecast
```

---

# 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` | tsframe format data containing univariate time series data. 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 an automatic criterion for choosing m based on the cumulative periodogram of the residuals; 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 "trendline", "components", "cpgram", and "screeplot", 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 | tsframe object with trendline estimation from targeted grouping based on a cumulative periodogram criterion (or according to the number of components specified in m). |
| l | window length. |
| 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". |
| residuals | tsframe object with the residuals from targeted grouping based on a cumulative periodogram criterion (or according to the number of components specified in m). |
| svd | Singular value decomposition corresponding to the trajectory matrix. |
| erc | elementary reconstructed components. |
| observations | tsframe object with the raw data. |

## Author(s)

Gabriel Martos and Miguel de Carvalho

## References

de Carvalho, M. and Martos, G. (2020). Brexit: Tracking and disentangling the sentiment towards leaving the EU. *International Journal of Forecasting*, **36**, 1128–1137.

## 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

```
## BREXIT DATA EXAMPLE
data(brexit); attach(brexit)
l <- tsframe(date, brexit[, 1] / 100) # l = leave
fit <- sst(l);
fit$m; fit$l # Number of ERC and parameter l in the model.
plot(fit, col = "red", lwd = 3, xlab = 'Time', ylab = 'Leave')
```

```
points(date, brexit[, 1] / 100, pch = 20)

## Scree-plot
plot(fit, options = list(type = "screeplot", ncomp = 1:10,
                         series.names = c('Leave')), type = "b", pch = 20, lwd = 2)


## Plot cumulative periodogram
par(mfrow=c(1,1), mar=c(4,2,1,1))
plot(fit, options = list(type = "cpgram", series.names = c('Leave')) )

## Elementary Reconstructed Components (ERC) plot:
plot(fit, options = list(type = "components", ncomp = 1:2))
```

---

`tsframe` *Time Series Frame Objects*

---

## Description

The function `tsframe` creates a univariate time series object to be used in combination with the functions in the package ASSA.

## Usage

```
tsframe(dates, y)
```

## Arguments

| | |
|---|---|
| `dates` | dates at which observations took place. |
| `y` | vector with time-series values sorted in ascendant way (first element in 'y' corresponds to the oldest value of the series and last element in 'y' corresponds to the newest value). |

## Examples

```
data(brexit); attach(brexit)
head(brexit, 3)
y <- tsframe(date, y = brexit[, 1]) # data is
print(y) # 'list' with 4 elements: dates, the series data, and serie length.
plot(y, col = 'blue' , lwd = 2, lty = 1)
```

# Index