Package ‘bfast’

February 17, 2012

Version 1.4.3

Date 2012-02-13

Title Breaks For Additive Season and Trend (BFAST)

Author Jan Verbesselt [aut, cre], Achim Zeileis [aut], Rob Hyndman [ctb], Rogier De Jong [ctb]

Maintainer Jan Verbesselt <Jan.Verbesselt@wur.nl>

Description BFAST integrates the decomposition of time series into trend, seasonal, and remainder components with methods for detecting and characterizing abrupt changes within the trend and seasonal components. BFAST can be used to analyze different types of satellite image time series and can be applied to other disciplines dealing with seasonal or non-seasonal time series, such as hydrology, climatology, and econometrics. The algorithm can be extended to label detected changes with information on the parameters of the fitted piecewise linear models. BFAST monitoring functionality is added based on a paper that has been submitted to Remote Sensing of Environment. BFAST monitor provides functionality to detect disturbance in near real-time based on BFAST-type models.

Depends R (>= 2.0.0), graphics, stats, strucchange, MASS, forecast,zoo, raster, sp

Imports graphics, stats, strucchange, zoo, raster

License GPL (>= 2)

URL http://bfast.R-Forge.R-project.org/

LazyLoad yes

LazyData yes

Repository CRAN

Repository/R-Forge/Project bfast

Repository/R-Forge/Revision 369

Date/Publication 2012-02-17 17:31:41

R topics documented:

`bfast-package` *Breaks For Additive Season and Trend (BFAST)*

Description

BFAST integrates the decomposition of time series into trend, seasonal, and remainder components with methods for detecting and characterizing abrupt changes within the trend and seasonal components. BFAST can be used to analyze different types of satellite image time series and can be applied to other disciplines dealing with seasonal or non-seasonal time series,such as hydrology, climatology, and econometrics. The algorithm can be extended to label detected changes with information on the parameters of the fitted piecewise linear models.

Additionally monitoring disturbances in BFAST-type models at the end of time series (i.e., in near real-time) is available: Based on a model for stable historical behaviour abnormal changes within newly acquired data can be detected. Different models are available for modeling the stable historical behavior. A season-trend model (with harmonic seasonal pattern) is used as a default in the regresssion modelling.

Details

The package contains:

- `bfast`: Main function for iterative decomposition and break detection as described in Verbesselt et al (2010ab).
- `bfastmonitor`: Monitoring approach for detecting disturbances in near real-time (see Verbesselt et al. 2011, submitted to Remote Sensing and Environment).
- `bfastpp`: Data pre-processing for BFAST-type modeling.
- Functions for plotting and printing, see `bfast`.
- `simts`: Artificial example data set.
- `harvest`: NDVI time series of a P. radiata plantation that is harvested.
- `som`: NDVI time series of locations in the south of Somalia to illustrate the near real-time disturbance approach

Author(s)

Jan Verbesselt [aut, cre], Achim Zeileis [aut], Rob Hyndman [ctb], Rogier De Jong [ctb]

References

Verbesselt J, Hyndman R, Newnham G, Culvenor D (2010). Detecting Trend and Seasonal Changes in Satellite Image Time Series. *Remote Sensing of Environment*, **114**(1), 106–115. http://dx.doi.org/10.1016/j.rse.2009.08.014

Verbesselt J, Hyndman R, Zeileis A, Culvenor D (2010). Phenological Change Detection while Accounting for Abrupt and Gradual Trends in Satellite Image Time Series. *Remote Sensing of Environment*, **114**(12), 2970–2980. http://dx.doi.org/10.1016/j.rse.2010.08.003

Verbesselt J, Zeileis A, Herold M (2011). Near Real-Time Disturbance Detection in Terrestrial Ecosystems Using Satellite Image Time Series: Drought Detection in Somalia. Working Paper 2011-18. Working Papers in Economics and Statistics, Research Platform Empirical and Experimental Economics, Universitaet Innsbruck. http://EconPapers.RePEc.org/RePEc:inn:wpaper:2011-18. Submitted to Remote Sensing and Environment.

---

bfast — *Break Detection in the Seasonal and Trend Component of a Univariate Time Series*

Description

Iterative break detection in seasonal and trend component of a time series. Seasonal breaks is a function that combines the iterative decomposition of time series into trend, seasonal and remainder components with significant break detection in the decomposed components of the time series.

Usage

```
bfast(Yt, h = 0.15, season = c("dummy", "harmonic", "none"),
  max.iter = NULL, breaks = NULL, hpc = "none")
```

Arguments

`Yt` — univariate time series to be analyzed. This should be an object of class "ts" with a frequency greater than one without NA's.

`h` — minimal segment size between potentially detected breaks in the trend model given as fraction relative to the sample size (i.e. the minimal number of observations in each segment divided by the total length of the timeseries.

`season` — the seasonal model used to fit the seasonal component and detect seasonal breaks (i.e. significant phenological change). There are three options: "dummy", "harmonic", or "none" where "dummy" is the model proposed in the first Remote Sensing of Environment paper and "harmonic" is the model used in the second Remote Sensing of Environment paper (See paper for more details) and where "none" indicates that no seasonal model will be fitted (i.e. St = 0). If there is no seasonal cycle (e.g. frequency of the time series is 1) "none" can be selected to avoid fitting a seasonal model.

`max.iter`	maximum amount of iterations allowed for estimation of breakpoints in seasonal and trend component.
`breaks`	integer specifying the maximal number of breaks to be calculated. By default the maximal number allowed by h is used.
`hpc`	A character specifying the high performance computing support. Default is "none", can be set to "foreach". Install the "foreach" package for hpc support.

Details

To be completed.

Value

An object of the class "bfast" is a list with the following elements:

`Yt` equals the Yt used as input.

`output` is a list with the following elements (for each iteration):

`Tt`	the fitted trend component
`St`	the fitted seasonal component
`Nt`	the noise or remainder component
`Vt`	equals the deseasonalized data `Yt - St` for each iteration
`bp.Vt`	output of the `breakpoints` function for the trend model
`ci.Vt`	output of the `breakpoints` confint function for the trend model
`Wt`	equals the detrended data `Yt - Tt` for each iteration
`bp.Vt`	output of the `breakpoints` function for the seasonal model
`ci.Vt`	output of the `breakpoints` confint function for the seasonal model

`nobp` is a list with the following elements:

`nobp.Vt`	logical, TRUE if there are breakpoints detected
`nobp.Wt`	logical, TRUE if there are breakpoints detected

`magnitude` magnitude of the biggest change detected in the trend component

`Time` timing of the biggest change detected in the trend component

Author(s)

Jan Verbesselt

References

Verbesselt J, Hyndman R, Newnham G, Culvenor D (2010). Detecting Trend and Seasonal Changes in Satellite Image Time Series. *Remote Sensing of Environment*, **114**(1), 106–115. http://dx.doi.org/10.1016/j.rse.2009.08.014

Verbesselt J, Hyndman R, Zeileis A, Culvenor D (2010). Phenological Change Detection while Accounting for Abrupt and Gradual Trends in Satellite Image Time Series. *Remote Sensing of Environment*, **114**(12), 2970–2980. http://dx.doi.org/10.1016/j.rse.2010.08.003

See Also

`plot.bfast` for plotting of bfast() results.
`breakpoints` for more examples and background information about estimation of breakpoints in time series.

Examples

```
## Simulated Data
plot(simts) # stl object containing simulated NDVI time series

datats <- ts(rowSums(simts$time.series)) # sum of all the components (season,abrupt,remainder)
tsp(datats) <- tsp(simts$time.series) # assign correct time series attributes
plot(datats)

#fit <- bfast(datats,h=0.15, season="dummy", max.iter=1)
#plot(fit,sim=simts)
#fit # prints out whether breakpoints are detected in the seasonal and trend component

## Real data
## The data should be a regular ts() object without NA's
## See Fig. 8 b in reference
plot(harvest, ylab="NDVI") # MODIS 16-day cleaned and interpolated NDVI time series

(rdist <- 10/length(harvest)) # ratio of distance between breaks (time steps) and length of the time series
#fit <- bfast(harvest,h=rdist, season="harmonic", max.iter=1,breaks=2)
#plot(fit)

## plot anova and slope of the trend identified trend segments
#plot(fit, ANOVA=TRUE)

## plot the trend component and identify the break with the largest magnitude of change
#plot(fit,type="trend",largest=TRUE)

## plot all the different available plots
#plot(fit,type="all")

## output
#niter <- length(fit$output) # nr of iterations
#out <- fit$output[[niter]]  # output of results of the final fitted seasonal and trend models and #nr of breakpoints

## References
citation("bfast")

## For more info
?bfast
```

---

bfastmonitor	*Near Real-Time Disturbance Detection Based on BFAST-Type Models*

Description

Monitoring disturbances in time series models (with trend/season/regressor terms) at the end of time series (i.e., in near real-time). Based on a model for stable historical behaviour abnormal changes within newly acquired data can be detected. Different models are available for modeling the stable historical behavior. A season-trend model (with harmonic seasonal pattern) is used as a default in the regresssion modelling.

Usage

```
bfastmonitor(data, start,
  formula = response ~ trend + harmon, order = 3, lag = NULL, slag = NULL,
  history = c("ROC", "BP", "all"),
  type = "OLS-MOSUM", h = 0.25, end = 10, level = 0.05,
  hpc = "none", verbose = FALSE, plot = FALSE)
```

Arguments

`data`	A time series of class `ts`, or another object that can be coerced to such. For seasonal components, a frequency greater than 1 is required.
`start`	numeric. The starting date of the monitoring period. Can either be given as a float (e.g., `2000.5`) or a vector giving period/cycle (e.g., `c(2000, 7)`).
`formula`	formula for the regression model. The default is `response ~ trend + harmon`, i.e., a linear trend and a harmonic season component. Other specifications are possible using all terms set up by `bfastpp`, i.e., `season` (seasonal pattern with dummy variables), `lag` (autoregressive terms), `slag` (seasonal autoregressive terms), or `xreg` (further covariates). See `bfastpp` for details.
`order`	numeric. Order of the harmonic term, defaulting to `3`.
`lag`	numeric. Order of the autoregressive term, by default omitted.
`slag`	numeric. Order of the seasonal autoregressive term, by default omitted.
`history`	specification of the start of the stable history period. Can either be a character, numeric, or a function. If character, then selection is possible between reverse-ordered CUSUM (`"ROC"`, default), Bai and Perron breakpoint estimation (`"BP"`), or all available observations (`"all"`). If numeric, the start date can be specified in the same form as `start`. If a function is supplied it is called as `history(formula, data)` to compute a numeric start date.
`type`	character specifying the type of monitoring process. By default, a MOSUM process based on OLS residuals is employed. See `mefp` for alternatives.
`h`	numeric scalar from interval (0,1) specifying the bandwidth relative to the sample size in MOSUM/ME monitoring processes.
`end`	numeric. Maximum time (relative to the history period) that will be monitored (in MOSUM/ME processes). Default is 10 times the history period.
`level`	numeric. Significance level of the monitoring (and ROC, if selected) procedure, i.e., probability of type I error.
`hpc`	character specifying the high performance computing support. Default is `"none"`, can be set to `"foreach"`. See `breakpoints` for more details.

`verbose`	logical. Should information about the monitoring be printed during computation?
`plot`	logical. Should the result be plotted?

Details

`bfastmonitor` provides monitoring of disturbances (or structural changes) in near real-time based on a wide class of time series regression models with optional season/trend/autoregressive/covariate terms. See Verbesselt at al. (2011) for details.

Based on a given time series (typically, but not necessarily, with frequency greater than 1), the data is first preprocessed for regression modeling. Trend/season/autoregressive/covariate terms are (optionally) computed using `bfastpp`. Second, the data is split into a history and monitoring period (starting with `start`). Third, a subset of the history period is determined which is considered to be stable (see also below). Fourth, a regression model is fitted to the preprocessed data in the stable history period. Fifth, a monitoring procedure is used to determine whether the observations in the monitoring period conform with this stable regression model or whether a change is detected.

The regression model can be specified by the user. The default is to use a linear trend and a harmonic season: `response ~ trend + harmon`. However, all other terms set up by `bfastpp` can also be omitted/added, e.g., `response ~ 1` (just a constant), `response ~ season` (seasonal dummies for each period), etc. Further terms precomputed by `bfastpp` can be `lag` (autoregressive terms of specified order), `slag` (seasonal autoregressive terms of specified order), `xreg` (covariates, if `data` has more than one column).

For determining the size of the stable history period, various approaches are available. First, the user can set a start date based on subject-matter knowledge. Second, data-driven methods can be employed. By default, this is a reverse-ordered CUSUM test (ROC). Alternatively, breakpoints can be estimated (Bai and Perron method) and only the data after the last breakpoint are employed for the stable history. Finally, the user can also supply a function for his/her own data-driven method.

Value

`bfastmonitor` returns an object of class `"bfastmonitor"`, i.e., a list with components as follows.

`data`	original `"ts"` time series,
`tspp`	preprocessed `"data.frame"` for regression modeling,
`model`	fitted `"lm"` model for the stable history period,
`mefp`	fitted `"mefp"` process for the monitoring period,
`history`	start and end time of history period,
`monitor`	start and end time of monitoring period,
`breakpoint`	breakpoint detected (if any).
`magnitude`	median of the difference between the data and the model prediction in the monitoring period.

Author(s)

Achim Zeileis, Jan Verbesselt

References

Verbesselt J, Zeileis A, Herold M (2011). Near Real-Time Disturbance Detection in Terrestrial Ecosystems Using Satellite Image Time Series: Drought Detection in Somalia. Working Paper 2011-18. Working Papers in Economics and Statistics, Research Platform Empirical and Experimental Economics, Universitaet Innsbruck. http://EconPapers.RePEc.org/RePEc:inn:wpaper:2011-18. Submitted to Remote Sensing and Environment.

See Also

monitor, mefp, breakpoints

Examples

```
## See Fig. 6 a and b in Verbesselt et al. (2011)
## for more information about the data time series and acknowledgements

NDVIa <- as.ts(zoo(som$NDVI.a, som$Time))
plot(NDVIa)
## apply the bfast monitor function on the data
## start of the monitoring period is c(2010, 13)
## and the ROC method is used as a method to automatically identify a stable history
mona <- bfastmonitor(NDVIa, start = c(2010, 13))
mona
plot(mona)
## fitted season-trend model in history period
summary(mona$model)
## OLS-based MOSUM monitoring process
plot(mona$mefp, functional = NULL)
## the pattern in the running mean of residuals
## this illustrates the empirical fluctuation process
## and the significance of the detected break.

NDVIb <- as.ts(zoo(som$NDVI.b, som$Time))
plot(NDVIb)
monb <- bfastmonitor(NDVIb, start = c(2010, 13))
monb
plot(monb)
summary(monb$model)
plot(monb$mefp, functional = NULL)

## set the stable history period manually and use a 4th order harmonic model
bfastmonitor(NDVIb, start = c(2010, 13),
  history = c(2008, 7), order = 4, plot = TRUE)

## just use a 6th order harmonic model without trend
mon <- bfastmonitor(NDVIb, formula = response ~ harmon,
    start = c(2010, 13), order = 6, plot = TRUE)
summary(mon$model)

## References
citation("bfast")
```

```
## For more info
?bfastmonitor


## TUTORIAL for processing raster bricks (satellite image time series of 16-day NDVI images)

#if(require("raster")) {
#  modisraster ## properties of the raster brick
# plot(modisraster, 1) ## plot one raster of the 275 NDVI image layers
# plot(bfastts(modisraster[1], dates, type ="16-day")) ## plot one pixel of the raster brick

## derive median NDVI of a NDVI raster brick
# medianNDVI <- calc(modisraster, fun=function(x) median(x, na.rm = TRUE))
# plot(medianNDVI)
#}

# ## helper function to be used with the calc() function
# ## see ? calc for more info
# xbfastmonitor <- function(x,dates) {
#  ndvi <- timeser(x,dates)
#  ndvi <- window(ndvi,end=c(2011,14))
#  ## delete end of the time to obtain a dataset similar to RSE paper (Verbesselt et al.,2012)
#  bfm <- bfastmonitor(data = ndvi,start=c(2010,12),history = c("ROC"))
#  return(cbind(bfm$breakpoint,bfm$magnitude))
# }
#
# ## apply on one pixel for testing
# ndvi <- timeser(modisraster[1],dates)
# bfm <- bfastmonitor(data = ndvi,start=c(2010,12),history = c("ROC"))
# bfm$magnitude
# plot(bfm)
# xbfastmonitor(modisraster[1],dates) ## helper function on one pixel
#
# ## apply the bfastmonitor function onto a raster brick
# timeofbreak <- calc(modisraster, fun=function(x){
#  res <- t(apply(x, 1, xbfastmonitor, dates))
#  return(res)
# })
# plot(timeofbreak) ## time of break and magnitude of change
```

---

bfastpp *Time Series Preprocessing for BFAST-Type Models*

Description

Time series preprocessing for subsequent regression modeling. Based on a (seasonal) time series, a data frame with the response, seasonal terms, a trend term, (seasonal) autoregressive terms, and covariates is computed. This can subsequently be employed in regression models.

Usage

```
bfastpp(data, order = 3,
  lag = NULL, slag = NULL, na.action = na.omit,
  stl = c("none", "trend", "seasonal", "both"))
```

Arguments

`data`	A time series of class `ts`, or another object that can be coerced to such. For seasonal components, a frequency greater than 1 is required.
`order`	numeric. Order of the harmonic term, defaulting to `3`.
`lag`	numeric. Orders of the autoregressive term, by default omitted.
`slag`	numeric. Orders of the seasonal autoregressive term, by default omitted.
`na.action`	function for handling `NA`s in the data (after all other preprocessing).
`stl`	character. Prior to all other preprocessing, STL (season-trend decomposition via LOESS smoothing) can be employed for trend-adjustment and/or season-adjustment. The `"trend"` or `"seasonal"` component or both from `stl` are removed from each column in `data`. By default (`"none"`), no STL adjustment is used.

Details

To facilitate (linear) regression models of time series data, `bfastpp` facilitates preprocessing and setting up regressor terms. It returns a `data.frame` containing the first column of the `data` as the `response` while further columns (if any) are used as covariates `xreg`. Additionally, a linear trend, seasonal dummies, harmonic seasonal terms, and (seasonal) autoregressive terms are provided.

Optionally, each column of `data` can be seasonally adjusted and/or trend-adjusted via STL (season-trend decomposition via LOESS smoothing) prior to preprocessing. The idea would be to capture season and/or trend nonparametrically prior to regression modelling.

Value

`bfastpp` returns a `"data.frame"` with the following variables (some of which may be matrices).

`time`	numeric vector of time stamps,
`response`	response vector (first column of `data`),
`trend`	linear time trend (running from 1 to number of observations),
`season`	factor indicating season period,
`harmon`	harmonic seasonal terms (of specified `order`),
`lag`	autoregressive terms (or orders `lag`, if any),
`slag`	seasonal autoregressive terms (or orders `slag`, if any),
`xreg`	covariate regressor (all columns of `data` except the first, if any).

Author(s)

Achim Zeileis

References

Verbesselt J, Zeileis A, Herold M (2011). Near Real-Time Disturbance Detection in Terrestrial Ecosystems Using Satellite Image Time Series: Drought Detection in Somalia. Working Paper 2011-18. Working Papers in Economics and Statistics, Research Platform Empirical and Experimental Economics, Universitaet Innsbruck. http://EconPapers.RePEc.org/RePEc:inn:wpaper:2011-18. Submitted to Remote Sensing and Environment.

See Also

bfastmonitor

Examples

```
## set up time series
ndvi <- as.ts(zoo(cbind(a = som$NDVI.a, b = som$NDVI.b), som$Time))
ndvi <- window(ndvi, start = c(2006, 1), end = c(2009, 23))

## parametric season-trend model
d1 <- bfastpp(ndvi, order = 2)
d1lm <- lm(response ~ trend + harmon, data = d1)
summary(d1lm)

## autoregressive model (after nonparametric season-trend adjustment)
d2 <- bfastpp(ndvi, stl = "both", lag = 1:2)
d2lm <- lm(response ~ lag, data = d2)
summary(d2lm)
```

---

bfastts — *Create a regular time series object by combining data and date information*

Description

Create a regular time series object by combining measurements (data) and time (dates) information.

Usage

```
bfastts(data, dates,
type = c("irregular", "16-day"))
```

Arguments

data	A data vector
dates	Optional input of dates for each measurement in the 'data' variable. In case the data is a irregular time series, a vector with 'dates' for each measurement can be supplied using this 'dates' variable. The irregular data will be linked with the dates vector to create daily regular time series with a frequency = 365. Extra days in leap years might cause problems. Please be carefull using this option as it is experimental. Feedback is welcome.

type ("irregular") indicates that the data is collected at irregular dates and as such will be converted to a daily time series. ("16-day")indicates that data is collected a a regular time interval (every 16-days e.g. like the MOD13Q1 data products)

Details

bfastts create a regular time series

Value

bfastts returns an object of class "ts", i.e., a list with components as follows.

zz a regular "ts" time series with a frequency equal to 365 or 23 i.e. 16-day time series.

Author(s)

Achim Zeileis, Jan Verbesselt

References

Verbesselt J, Zeileis A, Herold M (2011). Near Real-Time Disturbance Detection in Terrestrial Ecosystems Using Satellite Image Time Series: Drought Detection in Somalia. Working Paper 2011-18. Working Papers in Economics and Statistics, Research Platform Empirical and Experimental Economics, Universitaet Innsbruck. http://EconPapers.RePEc.org/RePEc:inn:wpaper:2011-18. Submitted to Remote Sensing and Environment.

See Also

monitor, mefp, breakpoints

Examples

```
#if(require("raster")) {
# data <- modisraster[1]
# ndvi <- bfastts(data, dates, type = c("16-day"))
# plot(ndvi)
#}
```

`create16dayts` *A helper function to create time series*

Description

Time series creation

Usage

```
create16dayts(data, dates)
```

Arguments

`data`	A vector
`dates`	A vector ....

Author(s)

Achim Zeileis, Jan Verbesselt

See Also

bfastmonitor

Examples

```
## set up a 16-day time series
#ndvi <- create16dayts(modisraster[1],dates)
#plot(ndvi)
```

`dates` *A vector with date information (a Datum type) to be linked with each NDVI layer within the modis raster brick (modisraster data set)*

Description

`dates` is an object of class "Date" and contains the "Date" information to create a 16-day time series object.

Source

Verbesselt, J., R. Hyndman, G. Newnham, and D. Culvenor (2012). Near real-time disturbance detection using satellite image time series. *Remote Sensing of Environment*. http://eeecon.uibk.ac.at/wopec2/repec/inn/wpaper/2011-18.pdf.

Examples

```
## select modis raster brick data for one pixel out of the raster brick
## and link it with the dates information.

# ndvi <- bfastts(modisraster[1], dates, type = c("16-day"))
# plot(ndvi)
```

---

harvest *16-day NDVI time series for a Pinus radiata plantation.*

Description

A univariate time series object of class "ts". Frequency is set to 23 – the approximate number of observations per year.

Usage

```
data(harvest)
```

Source

Verbesselt, J., R. Hyndman, G. Newnham, and D. Culvenor (2009). Detecting trend and seasonal changes in satellite image time series. *Remote Sensing of Environment*. http://dx.doi.org/10.1016/j.rse.2009.08.014. Or see http://robjhyndman.com/papers/bfast1.

Examples

```
plot(harvest,ylab='NDVI')
# References
citation("bfast")
```

---

modisraster *A raster brick of 16-day satellite image NDVI time series for a small subset in south eastern Somalia.*

Description

A raster brick containing 16-day NDVI satellite images (MOD13C1 product). See for description of the data set itself https://lpdaac.usgs.gov/products/modis_products_table.

Source

Verbesselt, J., R. Hyndman, G. Newnham, and D. Culvenor (2012). Near real-time disturbance detection using satellite image time series. *Remote Sensing of Environment*. http://eeecon.uibk.ac.at/wopec2/repec/inn/wpaper/2011-18.pdf.

Examples

```
## example of spatial and temporal raster plotting functionality
#plot(modisraster,1) # plot the first image layer
#plot(modisraster[1]) # plot for the first pixel the full time series

## select modis data for one pixel out of the raster brick
## and link it with the dates information.
#ndvi <- bfastts(modisraster[1],dates)
#plot(ndvi)

# References
citation("bfast")
```

`plot.bfast` *Methods for objects of class "bfast".*

Description

Plot methods for objects of class "bfast".

Usage

```
## S3 method for class 'bfast'
plot(x, type = c("components", "all", "data", "seasonal",
"trend", "noise"), sim = NULL, largest=FALSE, main, ANOVA = FALSE, ...)
```

Arguments

`x`	`bfast` object
`type`	Indicates the type of plot. See details.
`sim`	Optional `stl` object containing the original components used when simulating `x`.
`largest`	If TRUE, show the largest jump in the trend component.
`ANOVA`	if TRUE Derive Slope and Significance values for each identified trend segment
`main`	an overall title for the plot.
`...`	further arguments passed to the `plot` function.

Details

This function creates various plots to demonstrate the results of a bfast decomposition. The type of plot shown depends on the value of `type`.

- componentsShows the final estimated components with breakpoints.
- allPlots the estimated components and breakpoints from all iterations.

- dataJust plots the original time series data.
- seasonalShows the trend component including breakpoints.
- trendShows the trend component including breakpoints.
- noisePlots the noise component along with its acf and pacf.

If `sim` is not `NULL`, the components used in simulation are also shown on each graph.

Author(s)

Jan Verbesselt, Rob Hyndman and Rogier De Jong

Examples

```
## See \code{\link[bfast]{bfast}} for examples.
# References
citation("bfast")
```

`simts` *Simulated seasonal 16-day NDVI time series*

Description

`simts` is an object of class "stl" and consists of seasonal, trend (equal to 0) and noise components. The simulated noise is typical for remotely sensed satellite data.

Usage

```
data(simts)
```

Source

Verbesselt, J., R. Hyndman, G. Newnham, and D. Culvenor (2009). Detecting trend and seasonal changes in satellite image time series. *Remote Sensing of Environment*. `http://dx.doi.org/10.1016/j.rse.2009.08.014`. Or see `http://robjhyndman.com/papers/bfast1`.

Examples

```
plot(simts)
# References
citation("bfast")
```

som *Two 16-day NDVI time series from the south of Somalia*

Description

som is a dataframe containing time and two NDVI time series to illlustrate how the monitoring approach works.

Usage

```
data(som)
```

Source

Needs to be added.

Examples

```
## first define the data as a regular time series (i.e. ts object)
NDVI <- as.ts(zoo(som$NDVI.b,som$Time))
plot(NDVI)
```

Index