Package 'msltrend'

October 13, 2022

Type Package

Title Improved Techniques to Estimate Trend, Velocity and Acceleration
from Sea Level Records

Version 1.0

Depends R (>= 3.2.2)

Date 2016-01-11

Description Analysis of annual average ocean water level time series
from long (minimum length 80 years) individual records, providing improved
estimates of trend (mean sea level) and associated real-time velocities and
accelerations. Improved trend estimates are based on Singular Spectrum Analysis
methods. Various gap-filling options are included to accommodate incomplete time
series records. The package also contains a forecasting module to consider the
implication of user defined quantum of sea level rise between the end of the
available historical record and the year 2100. A wide range of screen and pdf
plotting options are available in the package.

Author Phil J Watson <philwatson.slr@gmail.com>

Maintainer Phil J Watson <philwatson.slr@gmail.com>

License GPL (>= 3)

LazyData TRUE

Imports changepoint (>= 2.1.1), forecast (>= 6.2), plyr (>= 1.8.3),
Rssa (>= 0.13-1), tseries (>= 0.10-34), zoo (>= 1.7-12)

Repository CRAN

NeedsCompilation no

RoxygenNote 5.0.1

Date/Publication 2016-01-12 08:46:00

R topics documented:

1

Balt   *Ocean water level data for Baltimore, USA*

Description

Annual average ocean water level data from Permanent Service for Mean Sea Level (UK).

Usage

```
data(Balt)
```

Format

Time series data file with the first column the year and the second column the corresponding annual average ocean water level (mm). File contains 112 records spanning the period from 1904 to 2014 with a single missing value in 1990.

Details

The raw (*.csv) form of this data set is used extensively in the examples throughout this manual.

Source

[Permanent Service for Mean Sea Level (2015)](#)

References

Holgate, S.J., Matthews, A., Woodworth, P.L., Rickards, L.J., Tamisiea, M.E., Bradshaw, E., Foden, P.R., Gordon, K.M., Jevrejeva, S. and Pugh, J., 2013. New data systems and products at the Permanent Service for Mean Sea Level. *Journal of Coastal Research*, 29(3), pp. 493-504.

See Also

[msl.trend](#), [msl.forecast](#), [msl.plot](#), [msl.pdf](#), [summary](#).

Examples

```
data(Balt)
plot(Balt, type = "l", xlab = "Year", ylab = "Annual Average Mean Sea Level (mm)",
main = 'BALTIMORE, USA')
str(Balt) # check structure of data file
```

---

msl.forecast   *Projected sea level rise integrated with historical record.*

Description

Projected sea level rise integrated with historical record.

Usage

```
msl.forecast(object, slr = 800, plot = TRUE)
```

Arguments

object   of class "msl.trend" (see `msl.trend` and `s`).

slr   numeric, enables a user defined amount of projected sea level rise in millimetres.
                The user range is [200 to 1500] where 800 is the default setting.

plot   logical, if 'TRUE' then the original time series is plotted to the screen along
                with the trend component, the result of gap filling (where necessary) and the
                added quantum of sea level rise selected. 95% confidence intervals have also
                been applied. Default = TRUE.

Details

This routine adds a user specified quantum of sea level rise from the end of the deconstructed
historical record to the year 2100. All internal parameters captured in the `msl.trend` object are
passed directly to `msl.forecast`.

Value

An object of class "msl.forecast" is returned with the following elements:

$Station.Name: the name of the data record.

$Summary: a summary data frame of relevant attributes relating to the trend and the inputted
annual average data set extended to 2100 with projected sea level rise, including:

- $Year: input data;
- $MSL: input data;
- $Trend: mean sea level trend;
- $TrendSD: standard deviation of the determined mean sea level trend;
- $Vel: velocity (or first derivative) of mean sea level trend (mm/year);
- $VelSD: standard deviation of the velocity of the mean sea level trend;
- $Acc: acceleration (or second derivative) of mean sea level trend (mm/year/year);
- $AccSD: standard deviation of the acceleration of the mean sea level trend; and
- $FilledTS: gap-filled time series (where necessary).

$Velocity: outputs the peak velocity and the year in which it occurs.

$Acceleration: outputs the peak acceleration and the year in which it occurs.

$Historical.Record: outputs details of the start, end and length of the input data set.

$Historical.Fillgaps: outputs the extent of missing data (years) in the original record and the gap filling method used (where necessary).

$Projected.SLR: details the amount of sea level rise applied between the end of the historical record and the year 2100.

$Bootstrapping.Iterations: outputs the number of iterations used to generate the respective standard deviations for error margins.

See Also

msl.trend, msl.plot, msl.pdf, summary, Balt, s, t.

Examples

```
# --------------------------------------------------------------------------
# Isolate trend from Baltimore record, filling gaps with spline interpolation,
# 500 iterations and adding 1000 mm of slr to 2100. Use raw 'Balt.csv' data file.
# Note: ordinarily user would call 'File.csv' direct from working directory
# using the following sample code:
# s <- msl.trend('Balt.csv', fillgaps = 3, iter = 500, 'BALTIMORE, USA')
# t <- msl.forecast(s, slr = 1000)
# --------------------------------------------------------------------------

data(s) # msl.trend object from above-mentioned example
data(t) # msl.forecast object from above-mentioned example
str(t) # check structure of msl.forecast object
msl.plot(s, type=2) # check screen output of gapfilling and trend estimate
msl.plot(t, type=2) # check screen output of adding 1000 mm of sea level rise
```

---

msl.pdf   *Pdf plotting options.*

Description

Pdf plotting options.

Usage

```
msl.pdf(x, file_name = " ", type = 1, ci = 1)
```

Arguments

x	object of class "msl.trend" (see `msl.trend` and `s`) or "msl.forecast" (see `msl.forecast` and `t`).
file_name	is a character string indicating the name of the pdf output file. If this field is left blank the output file will be automatically saved in the working directory under the default name "File1.pdf".
type	numeric, enables a user defined input to select the type of chart to be plotted. The default setting (type = 1) provides 3 charts in the same plot area with the time series in the top panel, instantaneous velocity in the middle panel and instantaneous acceleration in the bottom panel. The alternatives (2, 3 and 4) are single panel plots of time series, instantaneous velocity and instantaneous acceleration, respectively.
ci	numeric, enables a user defined input to select the type of confidence interval to be displayed on the plots. The default setting (ci = 1) corresponds to a 95% confidence interval whilst ci=2 provides a 99% confidence interval.

Details

This routine provides a range of pdf plotting options for both "msl.trend" (see `msl.trend`) and "msl.forecast" (see `msl.forecast`) objects. Three panel plots (type 1 or default) are formatted with width = 16.54 inches and height = 20 inches. Single panel plots (types 2, 3, 4) are formatted with width = 16.54 inches and height = 15 inches. All plots are designed to be proportionally correct when imported into documents and re-sized to the width of a standard A4 page. The same range of alternative screen plotting options are available via `msl.plot`.

See Also

`msl.trend`, `msl.forecast`, `msl.plot`, `Balt`, `s`, `t`.

Examples

```
# --------------------------------------------------------------------------
# Isolate trend from Baltimore record, filling gaps with spline interpolation,
# 500 iterations and adding 1000 mm of slr to 2100. Use raw 'Balt.csv' data file.
# Note: ordinarily user would call 'File.csv' direct from working directory
# using the following sample code:
# s <- msl.trend('Balt.csv', fillgaps = 3, iter = 500, 'BALTIMORE, USA')
# t <- msl.forecast(s, slr = 1000)
# --------------------------------------------------------------------------

data(s) # msl.trend object from above-mentioned example
data(t) # msl.forecast object from above-mentioned example

# default output, 3 panels, 95% confidence intervals.
msl.pdf(s)
# Check 'File1.pdf' in working directory

# pdf plot time series, 95% confidence intervals.
msl.pdf(s, file_name = 'Series.pdf', type = 2)
```

```
# Check 'Series.pdf' file in working directory

# pdf plot instantaneous velocity, 95% confidence intervals.
msl.pdf(s, file_name = 'Velocity.pdf', type = 3)
# Check 'Velocity.pdf' file in working directory

# pdf plot instantaneous acceleration, 99% confidence intervals.
msl.pdf(s, file_name = 'Acceleration.pdf', type = 4, ci = 2)
# Check 'Acceleration.pdf' file in working directory

# default output, 3 panels, 95% confidence intervals.
msl.pdf(t, file_name = 'Forecast.pdf')
# Check 'Forecast.pdf' file in working directory
```

---

msl.plot   *Screen plotting options.*

Description

Screen plotting options.

Usage

```
msl.plot(x, type = 1, ci = 1)
```

Arguments

x	object of class "msl.trend" (see msl.trend and s) or "msl.forecast" (see msl.forecast and t).
type	numeric, enables a user defined input to select the type of chart to be plotted. The default setting (type = 1) provides 3 charts in the same plot area with the time series in the top panel, instantaneous velocity in the middle panel and instantaneous acceleration in the bottom panel. The alternatives (2, 3 and 4) are single panel plots of time series, instantaneous velocity and instantaneous acceleration, respectively.
ci	numeric, enables a user defined input to select the type of confidence interval to be displayed on the plots. The default setting (ci = 1) corresponds to a 95% confidence interval whilst ci=2 provides a 99% confidence interval.

Details

This routine provides a range of screen plotting options for both "msl.trend" (see msl.trend) and "msl.forecast" (see msl.forecast) objects. The same range of alternative pdf plotting options are available via msl.pdf.

See Also

msl.trend, msl.forecast, msl.pdf, Balt, s, t

Examples

```
# -------------------------------------------------------------------------
# Isolate trend from Baltimore record, filling gaps with spline interpolation,
# 500 iterations and adding 1000 mm of slr to 2100. Use raw 'Balt.csv' data file.
# Note: ordinarily user would call 'File.csv' direct from working directory
# using the following sample code:
# s <- msl.trend('Balt.csv', fillgaps = 3, iter = 500, 'BALTIMORE, USA')
# t <- msl.forecast(s, slr = 1000)
# -------------------------------------------------------------------------

data(s) # msl.trend object from above-mentioned example
data(t) # msl.forecast object from above-mentioned example
msl.plot(s) # default screen plot output, 3 panels, 95% confidence intervals
msl.plot(s, type = 2) # plot time series, 95% confidence intervals
msl.plot(s, type = 3) # plot instantaneous velocity, 95% confidence intervals
msl.plot(s, type = 4, ci = 2) # plot acceleration, 99% confidence intervals
msl.plot(t) # default screen plot output, 3 panels, 95% confidence intervals
msl.plot(t, type = 2) # plot time series, 95% confidence intervals
msl.plot(t, type = 3) # plot instantaneous velocity, 95% confidence intervals
msl.plot(t, type = 4, ci = 2) # plot acceleration, 99% confidence intervals
```

---

msl.trend	*Isolate trend component from mean sea level records.*

Description

Isolate trend component from mean sea level records.

Usage

```
msl.trend(file, station_name = " ", fillgaps = 1, iter = 10000,
  plot = TRUE)
```

Arguments

file
: csv format input file **with no header row** of annual average water levels. This file must contain 2 columns with the first column the time period (in years) and the second column annual average ocean water levels (in millimetres). Missing data must be denoted by "NA". Missing data and maximum missing data gap are limited to 15% and 5%, respectively, of the data record. The minimum length data record processed by the package is 80 years.

: **Warning:** If input data files do not conform to these pre-conditions, the analysis will be terminated. It should be further noted that the existence of quasi 60 year oscillations in global mean sea level have been well recognised in the literature. Therefore, in order to be effective for climate change and sea level research, only input files with a minimum length exceeding 80 years have been considered in order that the package can identify and isloate such signals.

station_name   character string, providing the name of the data record.

> **Note:** This field can be left blank, however, it is retained for use in banner labelling of all plotting and pdf outputs.

fillgaps   numeric, provides 3 alternative gap filling procedures for missing data. The default procedure (fillgaps = 1) is based on iterative gap filling using Singular Spectrum Analysis (refer [igapfill](#)) . The alternatives (2 and 3) are based on linear interpolation and cubic spline interpolation, respectively (refer [na.approx](#)).

> **Note:** Gap filled portions of the time series are denoted in red on the default screen plot. This is done specifically to provide ready visual observation to discern if the selected gap filling method provides an appropriate estimate within the gaps in keeping with the remainder of the historical record. Depending on the nature of the record and extent of gaps, some trial and error between alternatives might be necessary to optimise gap filling.

iter   numeric, enables a user defined number of iterations for bootstrapping to determine error margins. The user range is [500 to 10000] where 10000 is the default setting.

> **Warning:** Although the default setting provides a more accurate basis for estimating error margins, the degree of iterations slows the analysis and can take several minutes to run.

plot   logical, if 'TRUE' then the original time series is plotted to the screen along with the trend component and the result of gap filling (where necessary). 95% confidence intervals have also been applied. Default = TRUE.

Details

This is the key entry point to the package. This function deconstructs annual average time series data into a trend and associated real-time velocities and accelerations, filling necessary internal structures to facilitate all other functions in this package. The trend is isloated using Singular Spectrum Analysis, in particular, aggregating components whose low frequency band [0 to 0.01] exceed a threshold contribution of 75%. Associated velocities and accelerations are determined through the fitting of a cubic smoothing spline to the trend with 1 degree of freedom per every 8 years of record length. Refer Watson (2016a,b) for more detail.

Value

An object of class "msl.trend" is returned with the following elements:

$Station.Name: the name of the data record.

$Summary: a summary data frame of the relevant attributes relating to the trend and the inputted annual average data set, including:

- $Year: input data;
- $MSL: input data;
- $Trend: mean sea level trend;
- $TrendSD: standard deviation of the determined mean sea level trend;
- $Vel: velocity (or first derivative) of mean sea level trend (mm/year);
- $VelSD: standard deviation of the velocity of the mean sea level trend;

- $Acc: acceleration (or second derivative) of mean sea level trend (mm/year/year);
- $AccSD: standard deviation of the acceleration of the mean sea level trend;
- $Resids: time series of uncorrelated residuals; and
- $FilledTS: gap-filled time series (where necessary).

$Velocity: outputs of the peak velocity and the year in which it occurred.

$Acceleration: outputs of the peak acceleration and the year in which it occurred.

$Record.Length: outputs details of the start, end and length of the input data set.

$Fillgaps: outputs the extent of missing data (years) in the original record and the gap filling method used (where necessary).

$Bootstrapping.Iterations: outputs the number of iterations used to generate the respective standard deviations for error margins.

$Changepoints: outputs the number and time at which changepoints in the variance of the uncorrelated residuals occur (if any). Where changepoints are identified, block bootstrapping procedures are used with residuals quarantined between changepoints.

References

Watson, P.J., 2016a. Identifying the best performing time series analytics for sea-level research. In: *Time Series Analysis and Forecasting, Contributions to Statistics*, ISBN 978-3-319-28725-6, Springer International Publishing (in press).

Watson, P.J., 2016b. How to improve estimates of real-time acceleration in the mean sea level signal. In: Vila-Concejo, A., Bruce, E., Kennedy, D.M., and McCarroll, R.J. (eds.), Proceedings of the 14th International Coastal Symposium (Sydney, Australia). *Journal of Coastal Research*, Special Issue, No. 75. Coconut Creek (Florida), ISSN 0749-0208 (in press).

See Also

msl.forecast, msl.plot, msl.pdf, summary, Balt, s.

Examples

```
# -------------------------------------------------------------------------
# Isolate trend from Baltimore record, filling gaps with spline interpolation and
# 500 iterations. Use raw 'Balt.csv' data file. Note: ordinarily user would call
# 'File.csv' direct from working directory using the following sample code:
# s <- msl.trend('Balt.csv', fillgaps = 3, iter = 500, 'BALTIMORE, USA') # DONT RUN
# -------------------------------------------------------------------------

data(s) # msl.trend object from above-mentioned example
str(s) # check structure of msl.trend object
msl.plot(s, type=2) # check screen output of gapfilling and trend estimate
```

msltrend	*msltrend: A package providing improved techniques to estimate trend, velocity and acceleration from sea level records.*

Description

The 'msltrend' package provides improved estimates of trend (mean sea level) and associated real-time velocities and accelerations from long (minimum 80 years), individual, annual average ocean water level data records. Improved trend estimates are based on Singular Spectrum Analysis methods. Various gap-filling options are included to accommodate incomplete time series records. The package also contains a forecasting module to consider the implication of user defined quantum of sea level rise between the end of the available historical record and the year 2100. A wide range of screen and pdf plotting options are available within the package.

msltrend functions

The `msl.trend` function is the key entry point to the package deconstructing annual average time series data records into a trend and associated real-time velocities and accelerations, filling necessary internal structures which facilitate all functions in this package (Refer Watson 2016a,b for more detail).

The `msl.forecast` function enables a user defined quantum of sea level rise to be added from the end of the deconstructed historical record to the year 2100. Similalrly, this function estimates real-time velocities and accelerations from the start of the available historical record to the year 2100.

References

Watson, P.J., 2016a. Identifying the best performing time series analytics for sea-level research. In: *Time Series Analysis and Forecasting, Contributions to Statistics*, ISBN 978-3-319-28725-6, Springer International Publishing (in press).

Watson, P.J., 2016b. How to improve estimates of real-time acceleration in the mean sea level signal. In: Vila-Concejo, A., Bruce, E., Kennedy, D.M., and McCarroll, R.J. (eds.), Proceedings of the 14th International Coastal Symposium (Sydney, Australia). *Journal of Coastal Research*, Special Issue, No. 75. Coconut Creek (Florida), ISSN 0749-0208 (in press).

s	*sample 'msl.trend' object*

Description

Output of call to `msl.trend` used extensively in examples throughout this Manual.

Usage

```
data(s)
```

Format

msl.trend object

Details

This `msl.trend` object is used extensively in the examples throughout this manual in order to call the object direct rather than producing the same via original code which can be computationally expensive. This object results from a decomposition of the Baltimore record, filling gaps with spline interpolation and using 500 iterations to generate error margins via bootstrapping.

Note: Ordinarily the user would call 'File.csv' direct from working directory, creating the 'msl.trend' object using the following sample code:

s <- msl.trend('Balt.csv', fillgaps = 3, iter = 500, 'BALTIMORE, USA') # DON'T RUN

See Also

`msl.trend`, `msl.forecast`, `msl.plot`, `msl.pdf`, `summary`, `Balt`.

Examples

```
data(s)
str(s) # check structure of object
```

---

summary   *Summary outputs of decomposed time series.*

Description

Summary outputs of decomposed time series.

Usage

```
summary(object)
```

Arguments

object   of class "msl.trend" (see `msl.trend`) or "msl.forecast" (see `msl.forecast`).

Details

This routine provides a screen summary of the respective outputs from a `msl.trend` or `msl.forecast` object. The summary produced is identical to str( ) for an object of class "msl.trend" (see `msl.trend`) or "msl.forecast" (see `msl.forecast`).

See Also

`msl.trend`, `msl.forecast`, `Balt`, `s`, `t`.

Examples

```
# ----------------------------------------------------------------------
# Isolate trend from Baltimore record, filling gaps with spline interpolation,
# 500 iterations and adding 1000 mm of slr to 2100. Use raw 'Balt.csv' data file.
# Note: ordinarily user would call 'File.csv' direct from working directory
# using the following sample code:
# s <- msl.trend('Balt.csv', fillgaps = 3, iter = 500, 'BALTIMORE, USA')
# t <- msl.forecast(s, slr = 1000)
# ----------------------------------------------------------------------

data(s) # msl.trend object from above-mentioned example
data(t) # msl.forecast object from above-mentioned example
summary(s) # summary for object of class 'msl.trend' object
summary(t) # summary for object of class 'msl.forecast' object
```

---

t   *sample 'msl.forecast' object*

Description

Output of call to msl.forecast used extensively in examples throughout this Manual.

Usage

```
data(t)
```

Format

msl.forecast object

Details

This msl.forecast object is used extensively in the examples throughout this manual in order to call the object direct rather than producing the same via original code which can be computationally expensive. This object results from a decomposition of the Baltimore record, filling gaps with spline interpolation and using 500 iterations to generate error margins via bootstrapping (see s). This 'msl.trend' object is then parsed to msl.forecast with the addition of 1000 millimetres of sea level rise between the end of the historical record and 2100.

Note: Ordinarily the user would call 'File.csv' direct from working directory, creating the 'msl.trend' object first, then creating the above-mentioned msl.forecast object using the following sample code:

s <- msl.trend('Balt.csv', fillgaps = 3, iter = 500, 'BALTIMORE, USA') # DON'T RUN

t <- msl.forecast(s, slr = 1000) # DON'T RUN

See Also

msl.trend, msl.forecast, msl.plot, msl.pdf, summary, Balt, s.

Examples

```
data(t)
str(t) # check structure of object
```

Index