Package 'outliers.ts.oga'

January 30, 2026

Type Package

Title Efficient Outlier Detection for Large Time Series Databases

Version 1.1.2

Maintainer Pedro Galeano <pedro.galeano@uc3m.es>

Description Programs for detecting and cleaning outliers in single time series and in time series from homogeneous and heterogeneous databases using an Orthogonal Greedy Algorithm (OGA) for saturated linear regression models. The programs implement the procedures presented in the paper entitled ``Efficient Outlier Detection for Large Time Series Databases" by Pedro Galeano, Daniel Peña and Ruey S. Tsay (2026), working paper, Universidad Carlos III de Madrid. Version 1.1.2 fixes one bug.

License GPL-3

Encoding UTF-8

Depends R (>= 4.3.0)

Imports caret (>= 6.0-94), forecast (>= 8.22.0), future (>= 1.67.0), future.apply (>= 1.20.0), gsarima (>= 0.1-5), parallelly (>= 1.37.1), robust (>= 0.7-4)

Suggests knitr, rmarkdown

NeedsCompilation no

Author Pedro Galeano [aut, cre] (ORCID: <https://orcid.org/0000-0003-2577-2747>), Daniel Peña [aut] (ORCID: <https://orcid.org/0000-0002-9137-1557>), Ruey S. Tsay [aut] (ORCID: <https://orcid.org/0000-0002-4949-4035>)

LazyData true

RoxygenNote 7.3.3

Repository CRAN

Date/Publication 2026-01-30 14:30:02 UTC

Contents

`db_het_oga` *Detecting and cleaning outliers in a heterogeneous time series database with OGA*

Description

Detects and cleans Additive Outliers (AOs) and Level Shifts (LSs) in time series that form a heterogeneous database, i.e. the series may have different definitions, sample sizes and/or frequencies. The function runs in parallel on the computer cores.

Usage

```
db_het_oga(Y)
```

Arguments

`Y`	The database, a `list` of `p` `ts` objects with possibly different lengths and/or frequencies. It is assumed that each time series has its frequency defined in its `ts` object.

Details

The function applies the `single_oga` function to each of the time series that make up the database to detect outlier effects and clean the series of such effects. This process is run in parallel on the computer cores, which saves a lot of computational cost. The function provides a `list` of `ts` objects with the original series cleaned from the effect of the AOs and LSs, in addition to the location, size and t-statistic corresponding to each of them.

Value

`n_AOs`	A `vector` with the number of AOs detected in each series of the database.
`n_LSs`	A `vector` with the number of LSs detected in each series of the database.
`AOs`	A `list` with the AOs detected in each series of the database.
`LSs`	A `list` with the LSs detected in each series of the database.
`Y_clean`	The cleaned database, a `list` of `p` cleaned time series.
`result`	A message indicating when the procedure has worked correctly or the problem encountered if the procedure stops.

Note

The computational cost depends on the size of the database and the level of contamination of the series. Note that the function may take several minutes if the database contains hundred of series with thousands of observations.

Author(s)

Pedro Galeano.

References

Galeano, P., Peña, D. and Tsay, R. S. (2025). Efficient outlier detection for large time series databases. Working paper, Universidad Carlos III de Madrid.

See Also

single_oga; db_hom_oga.

Examples

```
# Load FRED_MD dataset
data("FRED_MD")

# Define frequency s, the same for all series
s <- 12

# Define a list with the first 10 time series with frequency s
X <- FRED_MD[,1:10]
Y <- vector(mode='list',length=ncol(X))
for (k in 1:ncol(X)){Y[[k]] <- ts(X[,k],frequency=s)}

# Apply the function to Y
out_db_het_oga <- db_het_oga(Y)
```

---

db_hom_oga *Detecting and cleaning outliers in a homogeneous time series database with OGA*

Description

Detects and cleans Additive Outliers (AOs) and Level Shifts (LSs) in time series that form a homogeneous database, i.e. all series are defined similarly, have the same length and the same frequency. The function runs in parallel on the computer cores.

Usage

```
db_hom_oga(Y,s=NULL)
```

Arguments

Y	The database, a matrix of size Txp, where T is the time series length and p is the number of series.
s	Optional, the time series frequency, i.e., the number of observations per unit of time (s=1 for non-seasonal, s=4 for quarterly, s=7 for weekly, s=12 for monthly, s=24 for daily, s=52 for yearly, or s=60 for hourly). If the value of s is not given, the value s=1 is taken.

Details

The function applies the single_oga function to each of the time series that make up the database to detect outlier effects and clean the series of such effects. This process is run in parallel on the computer cores, which saves a lot of computational cost. The function provides a matrix with the original series cleaned from the effect of the AOs and LSs, in addition to the location, size and t-statistic corresponding to each of them.

Value

n_AOs	A vector with the number of AOs detected in each series of the database.
n_LSs	A vector with the number of LSs detected in each series of the database.
AOs	A list with the AOs detected in each series of the database.
LSs	A list with the LSs detected in each series of the database.
Y_clean	The cleaned database, a matrix of size Txp.
result	A message indicating when the procedure has worked correctly or the problem encountered if the procedure stops.

Note

The computational cost depends on the size of the database and the level of contamination of the series. Note that the function may take several minutes if the database contains hundred of series with thousands of observations.

Author(s)

Pedro Galeano.

References

Galeano, P., Peña, D. and Tsay, R. S. (2025). Efficient outlier detection for large time series databases. Working paper, Universidad Carlos III de Madrid.

See Also

single_oga; db_het_oga.

Examples

```
# Load FRED_MD dataset
data("FRED_MD")

# Define frequency s
s <- 12

# Apply the procedure to the first 10 time series in FREDMDApril19
Y <- FRED_MD[,1:10]
out_db_hom_oga <- db_hom_oga(Y,s=s)
```

FRED_MD *Federal Reserve Bank at St Louis.*

Description

Data obtained from the Federal Research Bank after process to remove missing values.

Usage

```
data("FRED_MD")
```

Format

An object of class `"data.frame"`.

Source

https://www.stlouisfed.org/

---

single_oga *Detect and clean outlying effects in a single time series with OGA*

Description

Algorithm for detecting and cleaning additive outliers and level shifts in a single time series with an Orthogonal Greedy Algorithm (OGA).

Usage

```
single_oga(yt,s=NULL)
```

Arguments

yt	A numeric vector or a ts object.
s	Optional, the time series frequency, i.e., the number of observations per unit of time (s=1 for non-seasonal, s=4 for quarterly, s=7 for weekly, s=12 for monthly, s=24 for daily, s=52 for yearly, or s=60 for hourly). If yt is of format ts, the value of the frequency in yt is taken. If not and the value of s is not given, the value s=1 is also taken.

Details

The program detects and cleans a time series from the effect of Additive Outliers (AOs) and Level Shifts (LSs). For this purpose, the procedure proposed in the paper 'Efficient outlier detection in heterogeneous time series databases' by Galeano, Peña and Tsay (2024) is used. The procedure can be divided into three automatic steps. The initial step involves fitting a sufficiently high-order AR model to yt using robust regression to obtain an AR representation and a residual series. Then, an Orthogonal Greedy Algorithm (OGA) procedure is applied to the residual series to identify a set of potential AOs and LSs and to remove their effects from yt. The identified set of outlying effects is referred to as the first set of potential outliers. The second step is to identify and fit an ARIMA or SARIMA model, depending on whether seasonality is detected, to the outlier-adjusted series of the first step and to obtain a new residual series. The OGA procedure is then applied to this new residual series to identify a new set of potential AOs and LSs, if any. The detected outlying effects form the second set of potential outliers. The third step involves combining the potential outliers identified in the first and second steps to remove any redundancies so as to obtain a final set of potential AOs and LSs, and fitting an ARIMA (or SARIMA) model jointly with the final set of potential outliers. Then, any negligible outlying effects, if any, are removed. Finally, any detected AOs and LSs are removed from the observed time series yt to produce an outlier-free time series.

Value

yt_clean	A ts object with the cleaned time series after removing the effects of the outliers in the observed time series.
aos	A matrix with the Additive Outliers (AOs) detected including location, size and t-test. If NULL, no AOs have been found in the series.
lss	A matrix with the Level Shifts (LSs) detected including location, size and t-test. If NULL, no LSs have been found in the series.

Author(s)

Pedro Galeano.

References

Galeano, P., Peña, D. and Tsay, R. S. (2025). Efficient outlier detection for large time series databases. Working paper, Universidad Carlos III de Madrid.

See Also

db_hom_oga; db_het_oga.

Examples

```
## Load FRED_MD dataset
data("FRED_MD")
Y <- FRED_MD

## Define time series yt and frequency s
yt <- Y[,1]
s <- 12

## Apply the function to yt
out_single_oga <- single_oga(yt,s=s)
```

Index