Package ‘tempdisagg’

August 8, 2023

Version 1.1.1

Date 2023-08-08

Title Methods for Temporal Disaggregation and Interpolation of Time Series

URL https://journal.r-project.org/archive/2013-2/sax-steiner.pdf

BugReports https://github.com/christophsax/tempdisagg

Suggests tsbox, testthat, knitr, rmarkdown, covr

Description Temporal disaggregation methods are used to disaggregate and interpolate a low frequency time series to a higher frequency series, where either the sum, the mean, the first or the last value of the resulting high frequency series is consistent with the low frequency series. Temporal disaggregation can be performed with or without one or more high frequency indicator series. Contains the methods of Chow-Lin, Santos-Silva-Cardoso, Fernandez, Litterman, Denton and Denton-Cholette, summarized in Sax and Steiner (2013) <doi:10.32614/RJ-2013-028>. Supports most R time series classes.

License GPL-3

LazyLoad yes

RoxygenNote 7.2.3

Encoding UTF-8

VignetteBuilder knitr

NeedsCompilation no

Author Christoph Sax [aut, cre] (<https://orcid.org/0000-0002-7192-7044>), Peter Steiner [aut], Tommaso Di Fonzo [ctb]

Maintainer Christoph Sax <christoph.sax@gmail.com>

Repository CRAN

Date/Publication 2023-08-08 15:20:02 UTC

R topics documented:

tempdisagg-package . . . 2
exports.m . . . 3
gdp.q . . . 3
plot.td . . . 4
predict.td . . . 4
spi.d . . . 5
summary.td . . . 6
ta . . . 7
td . . . 8

Index **14**

`tempdisagg-package` *Methods for Temporal Disaggregation and Interpolation of Time Series*

Description

Temporal disaggregation methods are used to disaggregate or interpolate a low frequency time series to higher frequency series, where either the sum, the average, the first or the last value of the resulting high frequency series is consistent with the low frequency series. Temporal disaggregation can be performed with or without one or more high frequency indicator series.

A good way to start is the introductory vignette:

`vignette("intro", "tempdisagg")`

Our article on temporal disaggregation of time series (doi:10.32614/RJ2013028) in the R-Journal describes the package and the theory of temporal disaggregation in more detail.

Author(s)

Christoph Sax <christoph.sax@gmail.com>, Peter Steiner

See Also

`td()` for more information on usage.

exports.m — *Trade and Sales of Chemical and Pharmaceutical Industry*

Description

This data set contains the monthly and quarterly imports and exports of the chemical and pharmaceutical industry in Switzerland (in in millions of Swiss Francs) as well as their quarterly and annual sales (Index).

Format

Each time series is an object of class `"ts"`. The number of observations depends on the frequency.

Source

Import and Export Data are from the Swiss Federal Customs Administration. Sales Data are from the Swiss Federal Statistical Office.

gdp.q — *Gross Domestic Product*

Description

Qarterly real GDP, not seasonally adjusted, in millions of Swiss Francs (2010 prices).

Format

A `data.frame`.

Source

State Secretariat for Economic Affairs (SECO).

Examples

```
## Not run:
# recreate the series with latest data
library(tsbox)
library(dplyr)
library(dataseries)
library(imputeTS)
dataseries::ds("ch_seco_gdp.nsa.real.gdp") %>%
  ts_default() %>%
  ts_span(start = 2005)

## End(Not run)
```

plot.td — *Residual Plot for Temporal Disaggregation*

Description

`plot` method for class `"td"`. Plot the fitted and actual low frequency series, and residuals.

Usage

```
## S3 method for class 'td'
plot(x, ...)
```

Arguments

`x`	an object of class `"td"`, usually, a result of a call to `td()`.
`...`	further arguments passed to or from other methods.

Value

returns a a two panel plot as its side effect, showing the fitted and actual low frequency series, and the residuals.

See Also

`td()` for the main function for temporal disaggregation.

Examples

```
data(swisspharma)

mod2 <- td(sales.a ~ imports.q + exports.q)
plot(mod2)
```

predict.td — *Predict Method for Temporal Disaggregation*

Description

Compute the disaggregated or interpolated (and extrapolated) high frequency series of a temporal disaggregation.

Usage

```
## S3 method for class 'td'
predict(object, ...)
```

Arguments

`object`	an object of class `"td"`, usually, a result of a call to `td()`.
`...`	further arguments passed to or from other methods.

Value

`summary.td` returns a vector or a `"ts"` object, containing the disaggregated or interpolated high frequency series of a temporal disaggregation.

See Also

`td()` for the main function for temporal disaggregation.

Examples

```
data(swisspharma)

mod1 <- td(sales.a ~ imports.q + exports.q)
predict(mod1)
```

`spi.d` *SPI Swiss Performance Index*

Description

Daily values of stock market index.

Format

A `data.frame`.

Source

Swiss National Bank (SNB)

Examples

```
## Not run:
# recreate the series with latest data
library(tsbox)
library(dplyr)
library(dataseries)
library(imputeTS)
dataseries::ds("ch_snb_capchstocki.gdr") %>%
  ts_default() %>%
  ts_regular() %>%
  imputeTS::na_interpolation(option = "spline") %>%
  ts_span(start = 2005)

## End(Not run)
```

summary.td *Summary of a Temporal Disaggregation*

Description

summary method for class "td".

Usage

```
## S3 method for class 'td'
summary(object, ...)

## S3 method for class 'summary.td'
print(
  x,
  digits = max(3, getOption("digits") - 3),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

object	an object of class "td", usually, a result of a call to td().
...	further arguments passed to or from other methods.
x	an object of class "summary.td", usually, a result of a call to summary.td.
digits	the number of significant digits to use when printing.
signif.stars	logical. If TRUE, 'significance stars' are printed for each coefficient.

Value

summary.td returns a list containing the summary statistics included in object, and computes the following additional statistics:

n_l	number of low frequency observations
n	number of high frequency observations
ar_l	empirical auto-correlation of the low frequency series
coefficients	a named matrix containing coefficients, standard deviations, t-values and p-values

The print method prints the summary output in a similar way as the method for "lm".

See Also

td() for the main function for temporal disaggregation.

Examples

```
data(swisspharma)

mod1 <- td(sales.a ~ imports.q + exports.q)
summary(mod1)

mod2 <- td(sales.a ~ 0, to = "quarterly", method = "uniform")
summary(mod2)
```

`ta` *Temporal Aggregation of Time Series*

Description

Performs temporal aggregation of high to low frequency time series. Currently, `ta` only works with `ts` or `mts` time series objects.

Usage

```
ta(x, ...)

## S3 method for class 'ts'
ta(x, conversion = "sum", to = "annual", ...)
```

Arguments

`x`	a time series object of class `"ts"` or `"mts"`.
`...`	additional arguments, passed to the methods.
`conversion`	type of conversion: `"sum"`, `"average"`, `"first"` or `"last"`.
`to`	(low-frequency) destination frequency as a character string (`"annual"` or `"quarterly"`) or as a scalar (e.g. `1`, `2`, `4`).

Details

`ta` is used to aggregate a high frequency time series into a low frequency series, while the latter is either the sum, the average, the first or the last value of the high-frequency series. `ta` is the inverse function of `td()`. If applied to an output series of `td`, `ta` yields the original series.

Value

`ta` returns an object of class `"ts"` or `"mts"`, depending on the class of the input series.

See Also

`td()` for the main function for temporal disaggregation.

Examples

```
data(swisspharma)

sales.q.a <- ta(sales.q, conversion = "sum", to = "annual")
all.equal(sales.a, sales.q.a)
```

---

td *Temporal Disaggregation of Time Series*

Description

Perform temporal disaggregation or interpolation of low frequency to high frequency time series. `td` can be used with objects of class `"ts"`, with numeric vectors or with any ts-boxable time series object.

Usage

```
td(
  formula,
  conversion = "sum",
  to = "quarterly",
  method = "chow-lin-maxlog",
  truncated.rho = 0,
  fixed.rho = 0.5,
  criterion = "proportional",
  h = 1,
  start = NULL,
  end = NULL,
  ...
)
```

Arguments

`formula`	an object of class `"formula"`: a symbolic description of the the temporal disaggregation model. The details of model specification are given under 'Details'.
`conversion`	type of conversion: `"sum"`, `"mean"` (or: `"average"`), `"first"` or `"last"`.
`to`	high-frequency destination frequency as a character string (`"quarter"` (or `"quarterly"`), `"month"` (or `"monthly"`), `"day"`, `"hour"`, `"minute"`, `"second"`, or `"year"`) or as a scalar (e.g. `2`, `4`, `7`, `12`). Required if no right hand side indicator series is provided. The tsbox package must be installed to deal with frequencies other than monthly or quarterly. If the input series are numeric, `to` is a scalar indicating the frequency ratio.
`method`	method of temporal disaggregation: `"chow-lin-maxlog"`, `"chow-lin-minrss-ecotrim"`, `"chow-lin-minrss-quilis"`, `"chow-lin-fixed"`, `"dynamic-maxlog"` (experimental), `"dynamic-minrss"` (experimental), `"dynamic-fixed"` (experimental), `"fernandez"`, `"litterman-maxlog"`, `"litterman-minrss"`, `"litterman-fixed"`, `"denton-cholette"`, `"denton"`, `"fast"`, `"uniform"` or `"ols"`. See 'Details'.

`truncated.rho`	lower bound for the autoregressive parameter ρ. If set to `0` (default), no negative values are allowed. If set to `-1`, truncation is disabled.
`fixed.rho`	set a predefined autoregressive parameter ρ. Only works with the methods `"chow-lin-fixed"` and `"litterman-fixed"`.
`criterion`	minimzation criterion for Denton methods: `"proportional"` or `"additive"`. See 'Details'.
`h`	degree of differencing for Denton methods. See 'Details'.
`start`	(optional) start date. Similar to pre-processing the input series with `window()`.
`end`	(optional) end date. Similar to pre-processing the input series with `window()`.
`...`	additional arguments to be passed to the low level subfunctions.

Details

`td` is used to disaggregate or interpolate a low frequency to a higher frequency time series, while either the sum, the average, the first or the last value of the resulting high-frequency series is consistent with the low frequency series. Disaggregation can be performed with or without the help of one or more right hand side indicator series. It can deal with both with a regular disaggregation setting (e.g. quarters to months) but also with an irregular disaggregation setting (e.g. months to days), where it respects the the different lengths of the months.

If the high-frequency indicator(s) cover(s) a longer time span than the low-frequency series, an extrapolation or retropolation (Wei, 1994, p. 138) is performed, using the same model as for interpolation.

The selection of a temporal disaggregation model is similar to the selection of a linear regression model. Thus, `td` closely mirrors the working of the `lm()` function. The left hand side of the `formula()` denotes the low-frequency series, the right hand side the indicators. If no indicator is specified, the right hand side must be set equal to `1` (see examples). Unlike `lm`, `td` handles `ts()` and `mts` time-series objects, as a typical application involves the use of these objects. Alternatively, If used with basic vectors, the `to` argument specifies the ratio between the high and the low frequency series.

For the generalized least squares (GLS) methods `"chow-lin-maxlog"`, `"chow-lin-minrss-ecotrim"`, `"chow-lin-minrss-quilis"`, `"litterman-maxlog"` and `"litterman-minrss"`, an autoregressive parameter ρ is estimated. Default (and recommended) method is `chow-lin-maxlog`. With `truncated.rho = 0` (default), it produces good results for a wide range of applications.

There are two variants of the `chow-lin-minrss` approach that lead to different results: Ecotrim by Barcellan (2003) uses a correlation matrix instead of the variance covariance matrix (implemented in `"chow-lin-minrss-ecotrim"`), the Matlab library by Quilis (2009) multiplies the correlation matrix with $1/(1-\rho^2)$ (implemented in `"chow-lin-minrss-quilis"`).

The methods `"dynamic-maxlog"`, `"dynamic-minrss"` and `"dynamic-fixed"` are dynamic extensions of Chow-Lin (Santos Silva and Cardoso, 2001). If the autoregressive parameter ρ is equal to 0, no truncation remainder is added.

The Denton methods `"denton"` and `"denton-cholette"` can be specified with one or without an indicator. The parameter `h` can be set equal to `0`, `1`, or `2`. Depending on the value, the `denton` procedure minimizes the sum of squares of the deviations between the levels (`0`), the first differences (`1`) or the second differences (`2`) of the indicator and the resulting series. Additionally, `criterion` can be set equal to `"proportional"` or `"additive"`, depending on whether the proportional or the absolute deviations should be considered for minimzation. "denton-cholette" removes the transient movement of the original "denton" method at the beginning of the resulting series. "fast" is a shortcut for "chow-lin-fixed" with fixed.rho = 0.99999. It returns approximately the same results as "denton-cholette" with h = 1, but is much faster.

"uniform" is a special case of the "denton" approach, with h equals 0 and criterion equals "additive". It distributes the residuals uniformly. If no indicator is used, this leads to a step-shaped series.

"ols" performs an ordinary least squares regression (OLS) and distributes the residuals uniformly. It is especially useful for comparing the estimators of GLS and OLS regressions.

Value

td returns an object of class "td".

The function predict() computes the interpolated high frequency series. If the high-frequency indicator series are longer than the low-frequency series, the resulting series will be extrapolated. The function coefficients extracts the coefficients. The function residuals extracts the low frequency residuals. The function summary() prints a summary of the estimation.

An object of class "td" is a list containing the following components:

values	disaggregated or interpolated (and extrapolated) high frequency series
fitted.values	low frequency fitted values of the regression; low frequency indicator for the Denton methods.
p	preliminary high frequency series
residuals	low-frequency residuals
rho	autoregressive parameter, ρ
truncated	logical, whether ρ has been truncated
coefficients	a named vector of coefficients
se	standard errors of the coefficients
s_2	ML-estimator of the variance of the high-frequency residuals
s_2_gls	GLS-estimator of the variance of the high-frequency residuals
tss	weighted (low frequency) total sum of squares
rss	weighted (low frequency) residual sum of squares
r.squared	R squared
adj.r.squared	adjusted R squared
logl	log-likelihood
aic	Akaike information criterion
bic	Schwarz information criterion
rank	number of right hand variables (including intercept)
df	degrees of freedom
method	method of temporal disaggregation
call	function call

name	name of the low frequency variable
fr	the ratio of high to low-frequency series
conversion	type of temporal conversion
actual	actual values of the low frequeny series
model	a matrix containing the indicators (and a constant if present)
criterion	minimization criterion in Denton methods
h	order of differencing in Denton methods

References

Chow, G. C., & Lin, A. L. (1971). Best linear unbiased interpolation, distribution, and extrapolation of time series by related series. *The review of Economics and Statistics*, 372-375.

Denton, F. T. (1971). Adjustment of monthly or quarterly series to annual totals: an approach based on quadratic minimization. *Journal of the American Statistical Association*, 66(333), 99-102.

Santos Silva, J. M. C. & Cardoso, F. N. (2001). The Chow-Lin method using dynamic models. *Economomic Modelling*, 18, 269-280.

Wei, W. W. S. (1994). Time series analysis. Addison-Wesley publ.

Sax, C. und Steiner, P. (2013). Temporal Disaggregation of Time Series. *The R Journal*, 5(2), 80-88. doi:10.32614/RJ2013028

See Also

`ta()` for temporal aggregation, the inverse function of `td`.

`summary()` is used to obtain and print a summary of the results.

`predict()` is used to extract the disaggregated or interpolated high frequency series.

`plot()` is used to plot the fitted and actual low frequency series, as well as the residuals.

Examples

```
data(tempdisagg)

# one indicator, no intercept
mod1 <- td(sales.a ~ 0 + exports.q)
summary(mod1)  # summary statistics
plot(mod1)  # residual plot of regression
plot(predict(mod1))

# interpolated quarterly series

# temporally aggregated series is equal to the annual value
all.equal(window(
  ta(predict(mod1), conversion = "sum", to = "annual"),
  start = 1975), sales.a)

# several indicators, including an intercept
mod2 <- td(sales.a ~ imports.q + exports.q)
```

```
# no indicator (Denton-Cholette)
mod3 <- td(sales.a ~ 1, to = "quarterly", method = "denton-cholette")

# no indicator (uniform)
mod4 <- td(sales.a ~ 1, to = "quarterly", method = "uniform")

# Dynamic Chow-Lin (Santos Silva and Cardoso, 2001)
# (no truncation parameter added, because rho = 0)
mod5 <- td(sales.a ~ exports.q, method = "dynamic-maxlog")

# Example from Denton (1971), see references.
d.q <- ts(rep(c(50, 100, 150, 100), 5), frequency = 4)
d.a <- ts(c(500, 400, 300, 400, 500))

a1 <- predict(td(d.a ~ 0 + d.q, method = "denton",
                 criterion = "additive", h = 0))
a2 <- predict(td(d.a ~ 0 + d.q, method = "denton",
                 criterion = "additive", h = 1))
a3 <- predict(td(d.a ~ 0 + d.q, method = "denton",
                 criterion = "additive", h = 2))
a4 <- predict(td(d.a ~ 0 + d.q, method = "denton",
                 criterion = "additive", h = 3))

p1 <- predict(td(d.a ~ 0 + d.q, method = "denton",
                 criterion = "proportional", h = 0))
p2 <- predict(td(d.a ~ 0 + d.q, method = "denton",
                 criterion = "proportional", h = 1))
p3 <- predict(td(d.a ~ 0 + d.q, method = "denton",
                 criterion = "proportional", h = 2))
p4 <- predict(td(d.a ~ 0 + d.q, method = "denton",
                 criterion = "proportional", h = 3))

# Table in Denton (1971), page 101:
round(cbind(d.q, a1, a2, a3, a4, p1, p2, p3, p4))

## Not run:

# Using altvernative time series classes (see https://docs.ropensci.org/tsbox/)
library(tsbox)
sales.a.xts <- ts_xts(window(sales.a, start = 2000))
exports.q.xts <- ts_xts(window(exports.q, start = 2000))
mod1b <- td(sales.a.xts ~ 0 + exports.q.xts)
predict(mod1b)  # class 'xts'

# non-standard frequencies: decades to years
predict(td(ts_xts(uspop) ~ 1, "mean", to = "year", method = "fast"))

# quarter to daily (no indicator)
m.d.noind <- td(gdp.q ~ 1, to = "daily", method = "fast")
predict(m.d.noind)

# quarter to daily (one indicator)
```

```
m.d.stocks <- td(gdp.q ~ spi.d, method = "chow-lin-fixed", fixed.rho = 0.9)
predict(m.d.stocks)

## End(Not run)
```

Index

∗ **datasets**
    exports.m, 3
    gdp.q, 3
    spi.d, 5
∗ **models**
    plot.td, 4
    predict.td, 4
    summary.td, 6
    ta, 7
    td, 8
∗ **package**
    tempdisagg-package, 2
∗ **ts**
    plot.td, 4
    predict.td, 4
    summary.td, 6
    ta, 7
    td, 8

exports.m, 3
exports.q *(exports.m)*, 3

formula(), *9*

gdp.q, 3

imports.q *(exports.m)*, 3

lm(), *9*

plot(), *11*
plot.td, 4
predict(), *10, 11*
predict.td, 4
print.summary.td *(summary.td)*, 6

sales.a *(exports.m)*, 3
sales.q *(exports.m)*, 3
spi.d, 5
summary(), *10, 11*
summary.td, 6
ta, 7
ta(), *11*
td, 8
td(), *2*, *4–7*
tempdisagg *(tempdisagg-package)*, 2
tempdisagg-package, 2
ts(), *9*

window(), *9*