Package 'macror'

May 3, 2016

Title Functions for Macroeconomic time series analysis in R

Version 0.00.005

Author Fernando Barbi [aut,cre]

Maintainer Fernando Barbi <fcbarbi@gmail.com>

Description Functions for handling ts (time series) and zoo objects to: create index from percentage variation, create dummy series, complete series and generate Eviews-formatted dates.

URL http://www.macror.org

Depends R (>= 3.2.4)

Imports stats, zoo, tseries, xtable

Suggests testthat

License GPL (>=2)

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Repository github

R topics documented:

macror-package *Macroeconomic analysis in R: Set of functions to work with time series* ts *and* zoo *objects.*

Description

Package macror speeds up tasks related to time series (ts) and zoo objects: dlog, urTable, genEVtime, tsVarToIndex, tsNumToIndex, tsComplete, tsDummy and tsMapOutliers.

dlog *Calculates the percentage variation of a time series.*

Description

Generates series of the percentage variation of the series data passed as parameter.

Usage

```
dlog(data, lag = 1)
```

Arguments

data	Time series or vector to be converted to a ts object.
lag	(Optional) Number of lags to differentiate with diff(). Defaults to 1.

Details

Data must be time indexed, either as ts or zoo. If observation(s) are negative or zero, the algorithm changes to the traditional percentage variation.

Value

Time series of the percentage variation with the first position(s) as NA.

See Also

tsVarToIndex

Examples

```
CPI <- ts( rnorm(10,mean=100) )
Inflation <- dlCPI <- dlog( CPI )  # choose your naming convention!

x <- ts( seq(-.3,.6,.1) )
dlx <- dlog( x ) # switch algorithm to avoid generating NAs
dl2x <- dlog( x, lag=2 )
```

genEVtime *Generates a vector of Eviews date formatted strings (eg.2008M12).*

Description

Function to generate an Eviews-formatted vector of strings with time references.

Usage

```
genEVtime(start, end = NULL, qobs = NULL)
```

Arguments

start	String with the starting date formatted as "2000M12" or "2002Q4".
end	String with the ending date, uses the same format as start. Optional if qobs specified. Defaults to NULL.
qobs	Number of observations. Optional if end specified. Defaults to NULL.

Details

Eviews formats dates of regularly spaced time series as YYYYPSS where YYYY is the year, P is the period (Month,Quarter) and SS is the subperiod (eg.1 to 12 for months)

Examples

```
genEVtime("1973Y1","1975Y1") # yeap, must explicit "Y1"
genEVtime("1947q3","1948Q02") # note the 0 before 2 is optional
genEVtime("1947M1",qobs=6) #
```

tsComplete *Completes a series with data from anothe series or by interpolating and/or repeating.*

Description

Function to complete a time-indexed series (either ts or zoo) by combining it with another series to avoid NAs. The function has a third argument "fill" to chose whether to leave NA, interpolate or repeate the last value when both vectors are missing.

Usage

```
tsComplete(x, y, fill = "fill")
```

Arguments

x	main data series (either ts or zoo) to be completed for missing observations.
y	secondary series (either ts or zoo) to use only when missing in the main series.
fill	method to treat NA's after combining x and y, can be "na","fill","interpolate" or "repeat"

Details

Supports both ts and zoo. Converts to zoo if one of the parameters is zoo.

Value

Returns either a ts (default) or zoo object.

Examples

```
a  <- c(NA, 1, 2,NA,NA, 5,NA)
b  <- c(   NA,12,NA,13,NA,15,16)
a2 <- ts(a,start=c(2000,1),freq=4)
b2 <- ts(b,start=c(2000,2),freq=4)
b3 <- zoo::as.zoo(b2)
aa  <- tsComplete(a2)
ab1 <- tsComplete(a2,b2)
ab2 <- tsComplete(a2,b2,fill="na")
ab3 <- tsComplete(a2,b3)
ab4 <- tsComplete(a2,b3,fill="fill")
```

---

tsDummy *Generates a ts object for a dummy variable.*

Description

Generates a ts object filled with 0's, with 1's in a period or specific dates.

Usage

```
tsDummy(start, end, frequency = 12, period = NULL, dates = NULL)
```

Arguments

start	Same as the start option in a ts object.
end	Same as the end option in a ts object.
frequency	Same as the frequency option in a ts object.
period	(optional) List with two elements: the starting and ending dates for the 1's formated as for ex. c(2008,12).
dates	(optional) List of discontinuous dates formated as for ex. c(2008,12).

Details

Note that more than one period can be specified.

Value

ts object with 1's and 0's.

See Also

tsComplete

Examples

```
lehman <- tsDummy( start=c(2000,1), end=c(2015,12), freq=12, period=list( c(2008,9), c(2008,12) ))

oilshocks <- tsDummy( start=c(1970,1), end=c(2015,12), freq=12,
                                  dates=list( c(1973,10), c(1979,12), c(1990,10), c(2008,6) )  )
```

---

tsMapOutliers *Maps outliers to a dummy series.*

Description

Creates a ts or zoo object with 1's in outlier positions and 0's elsewhere.

Usage

```
tsMapOutliers(x, range = c(0.01, 0.99))
```

Arguments

x	The time series data may not be stationary.
range	Quantile range of acceptable values, defaults to range=c(0.01,0.99).

Details

uses tsDummy() in identified outliers.

Value

ots a dummy ts object with 1 when the outlier is detected and 0 everywhere else.

Examples

```
x <- ts( c(8,rnorm(10),-15), start=c(2000,1),freq=12 )
xo1 <- tsMapOutliers( x )
xo2 <- tsMapOutliers( x, c(.05,.95) )
```

---

tsNumToIndex *Converts a numeric series to an index using the first observation as reference.*

Description

Converts a numeric series to an index using the first observation as reference.

Usage

```
tsNumToIndex(x, base = 100, reference = 1)
```

Arguments

x	Time-indexed data to convert, must be a ts or zoo object.
base	(Optional) Default is 100.
reference	(Optional) Observation to be used as proportions are calculated.

Details

If using a monetary series be sure to deflate it before indexing.

Value

Time series (ts or zoo) with the index.

See Also

tsVarToIndex

Examples

```
# Production averages 1254 units per month
production <- ts( rnorm(30*12,mean=1254,sd=425),start=c(1980,1),freq=12)
ip  <- tsNumToIndex( production ) # industrial production index
ip2 <- tsNumToIndex( production, reference = c(1992,6) ) # rebase for June 1992
```

---

tsVarToIndex *Converts a percentage variation series into a base 100 index series.*

Description

Compounds a percentage variation series into a 100 based index corresponding to the reference observation.

Usage

```
tsVarToIndex(x, base = 100, reference = 1)
```

Arguments

x	Time series to be converted to index.
base	(optional) Default is 100.
reference	(optional) Index of the observation to be used as reference. Default is 1 (first).

Details

The index is based in 100. Reference is by default the first observation.

See Also

dlog, tsNumToIndex

Examples

```
# Monthly inflation is around 2%
Inflation <- ts( rnorm(13,mean=2,sd=.5), start=c(2007,12), freq=12 )
CPI_dec07 <- tsVarToIndex( Inflation )
```

---

urTable *Generates a table with Unit Roor test results*

Description

Returns a datatable and generates a .csv or .tex file with Unit Root test results.

Usage

```
urTable(df, tests = c("adf", "pp", "kpss"), order = 1, file = NULL,
  format = "csv")
```

Arguments

df	Dataframe with series to test
tests	List of tests to be performed, by default tests are c("adf","pp","kpss").
order	Maximum order of integration to test, usually noted as I(?). It can be c(0,1,2).
file	(Optional) Filename to be generated according to format.
format	(Optional) Format of the output table, can be c("csv","latex"). Defaults to "txt".

Details

Supports ADF, PP and KPSS unit root tests.

Value

A dataframe with the test results and a file (optional).

Examples

```
require(tseries)
data(USeconomic,package="tseries")
data <- data.frame( USeconomic )

dft1 <- urTable( data, tests=c("adf","pp","kpss"), file="US.csv" )

dft2 <- urTable( data, tests=c("pp","kpss"), order=2, file="US.tex", format="latex" )

# Even series with missing observations can be tested
bad <- data.frame( var1=rnorm(100), var2=cumsum(rnorm(100)) )
bad[seq(95,100),"var1"] <- bad[1,"var2"] <- bad[2,"var2"] <- NA
dft3 <- urTable( bad )
```

Index