

Package: rjd3x11plus (via r-universe)

October 29, 2024

Type Package

Title Interface to 'JDemetra+ 3.x' time series analysis software

Version 2.1.1.9000

Description R Interface to 'JDemetra+ 3.x'
(<https://github.com/jdemetra>) time series analysis software.

Depends R (>= 4.1.0)

Imports rJava (>= 1.0-6), rjd3toolkit (>= 3.2.2), rjd3filters (>= 2.1.0)

Remotes github::rjdverse/rjd3toolkit, github::rjdverse/rjd3filters

SystemRequirements Java (>= 17)

License EUPL

URL <https://github.com/rjdverse/rjd3x11plus>,
<https://rjdverse.github.io/rjd3x11plus>

LazyData TRUE

Suggests knitr, rmarkdown

RoxygenNote 7.3.1

Roxygen list(markdown = TRUE)

BugReports <https://github.com/rjdverse/rjd3x11plus/issues>

Encoding UTF-8

Collate 'ic_r.R' 'jd3_x11.R' 'macurves.R' 'utility.R'
'x11correction.R' 'x11plus_trend.R' 'zzz.R'

Repository <https://aqlt.r-universe.dev>

RemoteUrl <https://github.com/rjdverse/rjd3x11plus>

RemoteRef HEAD

RemoteSha adb73b26d39db0becf20c20962a0450782097c50

Contents

henderson	2
ic_ratio	3
macurves	3
select_trend_filter	4
x11plus	5
x11plus_trend	6
x11_extreme_values_corr	7
Index	8

henderson	<i>Apply Henderson linear filter</i>
-----------	--------------------------------------

Description

Apply Henderson linear filter

Usage

```
henderson(x, length, musgrave = TRUE, ic = 4.5)
```

Arguments

x	input time-series.
length	length of the Henderson filter.
musgrave	Boolean indicating if Musgrave asymmetric filters should be used.
ic	ic ratio: irregular/trend-cycle.

Value

A numeric array corresponding to the the trend

Examples

```
q<-x11plus(rjd3toolkit::ABS$X0.2.09.10.M, 12)
henderson(q$decomposition$sa, 13)
```

ic_ratio	<i>Compute IC-Ratio</i>
----------	-------------------------

Description

Compute IC-Ratio

Usage

```
ic_ratio(x, sc, mul = FALSE)
```

Arguments

x	input time series.
sc	trend-cycle component.
mul	boolean indicating if the decomposition is multiplicative or additive.

Examples

```
q<-x11plus(rjd3toolkit::retail$AllOtherGenMerchandiseStores, 12)
x <- q$decomposition$sa
sc <- henderson(x, length = 13, musgrave = FALSE)
ic_ratio(x, sc)
```

macurves	<i>Get Macurves Filters</i>
----------	-----------------------------

Description

Get Macurves Filters

Usage

```
macurves(seas_filter = c("S3X3", "S3X1", "S3X5", "S3X9", "S3X15"), period = 12)
```

Arguments

seas_filter	the filter to extract.
period	period of the filter.

Examples

```
macurves("S3X3")
```

select_trend_filter *X-11 Selection of Trend Filter*

Description

X-11 Selection of Trend Filter

Usage

```
select_trend_filter(x, ...)

## Default S3 method:
select_trend_filter(x, ..., freq)

## S3 method for class 'ts'
select_trend_filter(x, ..., length = 13)
```

Arguments

x	I/C ratio <code>ic_ratio()</code> or a time series
...	further arguments passed to or from other methods.
freq	frequency of the time series used to compute the I/C ratio.
length	length of the Henderson filter used to compute the I/C ratio.

Details

The following procedure is used in X-11 to select the length of the trend filter:

1. Computes the I/C ratio, *icr* with an Henderson filter of length 13.
2. The length depends on the value of *icr*:
 - if $icr < 1$ then the selected length is 9 for monthly data and 5 otherwise;
 - if $1 \leq icr < 3.5$ then the selected length is $freq + 1$ where *freq* is the frequency of data (12 for monthly data, 4 for quarterly data...).
 - if $icr \geq 3.5$ then the selected length is 23 for monthly data and 7 otherwise.

Examples

```
# example code
s<-rjd3toolkit::retail$AllOtherGenMerchandiseStores
q<-x11plus(s, 12)
x <- ts(data=q$decomposition$sa, start=start(s), frequency=frequency(s))
sc <- henderson(x, length = 13, musgrave = FALSE)
icr <- ic_ratio(x, sc)
select_trend_filter(icr, freq = 12)
# Because Henderson filter is used, this is equivalent to:
select_trend_filter(x)
```

x11plus	<i>Perform an X-11 like decomposition with any (non integer) periodicity.</i>
---------	---

Description

Perform an X-11 like decomposition with any (non integer) periodicity.

Usage

```
x11plus(
  y,
  period,
  mul = TRUE,
  trend.horizon = 6,
  trend.degree = 2,
  trend.kernel = c("Henderson", "BiWeight", "TriWeight", "TriCube", "Uniform",
    "Triangular", "Epanechnikov", "Trapezoidal"),
  trend.asymmetric = c("CutAndNormalize", "Direct", "MMSRE"),
  seas.s0 = c("S3X3", "S3X1", "S3X5", "S3X9", "S3X15"),
  seas.s1 = c("S3X5", "S3X3", "S3X1", "S3X9", "S3X15"),
  extreme.lsig = 1.5,
  extreme.usig = 2.5
)
```

Arguments

<code>y</code>	input time-series.
<code>period</code>	Period of the seasonal component, any positive real number.
<code>mul</code>	Boolean indicating if the decomposition mode is multiplicative (TRUE).
<code>trend.horizon</code>	bandwidth of trend filters.
<code>trend.degree</code>	polynomial order in local trend model.
<code>trend.kernel</code>	kernel weights in objective function.
<code>trend.asymmetric</code>	truncation type for symmetric filter.
<code>seas.s0</code>	Seasonal filter for B5, C5, D5.
<code>seas.s1</code>	seasonal filter for B10, C10, D10.
<code>extreme.lsig</code>	lower boundary used for outlier correction in irregular.
<code>extreme.usig</code>	upper boundary used for outlier correction in irregular.

Value

An object of the class 'JD3_X11PLUS', containing the decomposition and the parameters

Examples

```
q<-x11plus(rjd3toolkit::ABS$X0.2.09.10.M, 12)
```

x11plus_trend

X-11 Decomposition With Custom Trend Filters

Description

Perform the X-11 decomposition using custom trend filter

Usage

```
x11plus_trend(
  x,
  period = frequency(x),
  trend.coefs,
  mul = TRUE,
  seas.s0 = c("S3X3", "S3X1", "S3X5", "S3X9", "S3X15"),
  seas.s1 = c("S3X5", "S3X3", "S3X1", "S3X9", "S3X15"),
  extreme.lsig = 1.5,
  extreme.usig = 2.5,
  userdefined = NULL
)
```

Arguments

x	input time-series.
period	period.
trend.coefs	coefficients of the filters used for the trend-cycle extraction from the real-time asymmetric filter to the symmetric filter. Can be a, object of class "list", "matrix", "lp_filter" or "rkhs_filter".
mul	boolean indicating if the decomposition mode is multiplicative.
seas.s0, seas.s1	seasonal filters.
extreme.lsig, extreme.usig	boundaries used for outlier correction in irregular.
userdefined	a vector containing the additional output variables.

Examples

```
q<-x11plus(rjd3toolkit::retail$AllOtherGenMerchandiseStores, 12)
x <- q$decomposition$a
decomposition_lp <- x11plus_trend(x, trend.coefs = rjd3filters::lp_filter())
decomposition_rkhs <- x11plus_trend(x, trend.coefs = rjd3filters::rkhs_filter())
plot(x)
lines(decomposition_lp$decomposition[, "t"], col = "red")
lines(decomposition_rkhs$decomposition[, "t"], col = "green")
```

x11_extreme_values_corr
'X11' Extreme Values Corrector

Description

'X11' Extreme Values Corrector

Usage

```
x11_extreme_values_corr(  
    x,  
    corrected_s,  
    period,  
    lsigma = 1.5,  
    usigma = 2.5,  
    mul = FALSE,  
    start = 0,  
    clean_extremities = TRUE  
)
```

Arguments

x	the analysed time series.
corrected_s	other time series if the series being corrected is different from x.
period	the period of the input time series if x is not a "ts" object.
lsigma, usigma	the lower and the upper sigma boundaries for the detection of extreme values.
mul	boolean indicating if the decomposition is multiplicative or additive.
start	position of the first "complete" considered period.
clean_extremities	boolean indicating if the extremities should be cleaned.

Details

The 'X11' Extreme Values Corrector is used to compute the tables b4, b4g, b9, b9g, b17, b20, c17 and c20.

Index

`henderson`, [2](#)

`ic_ratio`, [3](#)

`ic_ratio()`, [4](#)

`macurves`, [3](#)

`select_trend_filter`, [4](#)

`x11_extreme_values_corr`, [7](#)

`x11plus`, [5](#)

`x11plus_trend`, [6](#)