

Package: ggdemetra3 (via r-universe)

September 30, 2024

Type Package

Title 'ggplot2' Extension for Seasonal and Trading Day Adjustment with 'JDemetra+' 3.0

Version 0.0.2

Description Provides 'ggplot2' functions to return the results of seasonal and trading day adjustment made by the R interface to 'JDemetra+' 3.0.

Depends R (>= 3.1.2), ggplot2 (>= 2.0.0)

Imports ggrepel, gridExtra, rjd3toolkit, rjd3x13 (>= 3.2.3), rjd3tramoseats (>= 3.2.3), rjd3filters, MASS, reshape2, rjd3x11plus, rjd3highfreq, rJava

Suggests knitr, rmarkdown

Remotes github::rjdverse/rjd3toolkit, github::rjdverse/rjd3x13, github::rjdverse/rjd3tramoseats, github::rjdverse/rjd3x11plus, github::rjdverse/rjd3highfreq

SystemRequirements Java (>= 17)

License EUPL

URL <https://aqlt.github.io/ggdemetra/>,
<https://github.com/AQLT/ggdemetra3>

BugReports <https://github.com/AQLT/ggdemetra3/issues>

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

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

RemoteUrl <https://github.com/AQLT/ggdemetra3>

RemoteRef HEAD

RemoteSha 1cd2cc127d182eadcfcfc2a74b31b475e5f255ac

Contents

autoplot.JD3_X13_OUTPUT	2
components	3
geom_arima	3
geom_diagnostics	5
geom_outlier	7
geom_sa	9
ggplot_filters	11
ipi_c_eu	12
siratio	13
ts2df	14
usclaims	15
Index	16

autoplot.JD3_X13_OUTPUT
Plot 'rjdemetra3' model

Description

Plot 'rjdemetra3' model

Usage

```
## S3 method for class 'JD3_X13_OUTPUT'
autoplot(
  object,
  components = c("y", "sa", trend = "t", seasonal = "s", irregular = "i"),
  forecast = FALSE,
  ...
)
```

Arguments

object	a seasonal adjustment model.
components	components to print, can be "y" (input time series), "sa" (seasonal adjusted), "t" (trend-cycle), "y_cal" (calendar adjusted), "s" (seasonal), "i" (irregular), "cal" (calendar). The vector can be named to change the label.
forecast	boolean indicating if the forecast series should be printed.
...	unused arguments.

Examples

```
x = rjd3x13::x13(ipi_c_eu["FR"])
ggplot2::autoplot(x)
```

components	<i>Extract Component from 'RJDemetra' model</i>
------------	---

Description

Extract Component from 'RJDemetra' model

Usage

```
seasonal(x, forecast = FALSE, calendar = FALSE, ...)
```

```
trendcycle(x, forecast = FALSE, ...)
```

```
irregular(x, forecast = FALSE, corrected = FALSE, ...)
```

```
seasonaladj(x, forecast = FALSE, corrected = FALSE, ...)
```

```
calendaradj(x, forecast = FALSE, ...)
```

```
calendar(x, forecast = FALSE, ...)
```

```
raw(x, forecast = FALSE, backcast = FALSE, ...)
```

Arguments

x	a seasonal adjustment object.
forecast, backcast	boolean indicating if the forecast/backcast series should be returned.
calendar	if TRUE, seasonal() returns de seasonal and calendar components.
...	unused parameters.
corrected	if TRUE large extreme values are removed from the components.

geom_arima	<i>ARIMA model</i>
------------	--------------------

Description

Function to add directly to the plot the ARIMA model used in the pre-adjustment process of the seasonal adjustment.

Usage

```
geom_arima(
  mapping = NULL,
  data = NULL,
  stat = "arima",
  geom = c("text", "label"),
  position = "identity",
  ...,
  method = c("x13", "tramoseats"),
  spec = NULL,
  frequency = NULL,
  message = TRUE,
  x_arima = NULL,
  y_arima = NULL,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or aes_() . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	A data.frame that contains the data used for the seasonal adjustment.
stat	The statistical transformation to use on the data for this layer, as a string.
geom	character. The geometric to use to display the data: <code>GeomText</code> (<code>geom = "text"</code> , the default, see geom_text()) or <code>GeomLabel</code> (<code>geom = "label"</code> , see geom_label()).
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	Other arguments passed on to layer() . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> .
method	the method used for the seasonal adjustment. "x13" (by default) for the X-13ARIMA method and "tramoseats" for TRAMO-SEATS.
spec	the specification used for the seasonal adjustment. See rjd3x13::x13() or rjd3tramoseats::tramoseats() .
frequency	the frequency of the time series. By default (<code>frequency = NULL</code>), the frequency is computed automatically.
message	a boolean indicating if a message is printed with the frequency used.
x_arima, y_arima	position of the text of the ARIMA model. By default, the first position of the data is used.
show.legend	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them.

Details

With the parameter `geom = "text"`, the ARIMA model used in the pre-adjustment process of the seasonal adjustment are directly added to the plot. With `geom = "label"` a rectangle is drawn behind the ARIMA model, making it easier to read.

Examples

```
p_sa_ipi_fr <- ggplot(data = ipi_c_eu_df, mapping = aes(x = date, y = FR)) +
  geom_line() +
  labs(title = "Seasonal adjustment of the French industrial production index",
        x = "time", y = NULL) +
  geom_sa(color = "red", message = FALSE)

# To add the ARIMA model
p_sa_ipi_fr +
  geom_arima(geom = "label",
            x_arima = - Inf, y_arima = -Inf,
            vjust = -1, hjust = -0.1,
            message = FALSE)
```

 geom_diagnostics

Table of diagnostics

Description

Adds a table of diagnostics to the plot

Usage

```
geom_diagnostics(
  mapping = NULL,
  data = NULL,
  position = "identity",
  ...,
  method = c("x13", "tramoseats"),
  spec = NULL,
  frequency = NULL,
  message = TRUE,
  diagnostics = NULL,
  digits = 2,
  xmin = -Inf,
  xmax = Inf,
  ymin = -Inf,
  ymax = Inf,
  table_theme = ttheme_default(),
  inherit.aes = TRUE
)
```

Arguments

mapping	Set of aesthetic mappings created by <code>aes()</code> or <code>aes_()</code> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	A data frame that contains the data used for the seasonal adjustment.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	Other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> .
method	the method used for the seasonal adjustment. "x13" (by default) for the X-13ARIMA method and "tramoseats" for TRAMO-SEATS.
spec	the specification used for the seasonal adjustment. See <code>rjd3x13::x13()</code> or <code>rjd3tramoseats::tramoseats()</code> .
frequency	the frequency of the time series. By default (<code>frequency = NULL</code>), the frequency is computed automatically.
message	a boolean indicating if a message is printed with the frequency used.
diagnostics	vector of character containing the name of the diagnostics to plot. See <code>rjd3x13::userdefined_variables</code> or <code>rjd3tramoseats::userdefined_variables_tramoseats()</code> for the available parameters.
digits	integer indicating the number of decimal places to be used for numeric diagnostics. By default <code>digits = 2</code> .
xmin, xmax	x location (in data coordinates) giving horizontal location of raster.
ymin, ymax	y location (in data coordinates) giving vertical location of raster.
table_theme	list of theme parameters for the table of diagnostics (see <code>ttheme_default()</code>).
inherit.aes	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them.

Examples

```
p_sa_ipi_fr <- ggplot(data = ipi_c_eu_df, mapping = aes(x = date, y = FR)) +
  geom_line() +
  labs(title = "Seasonal adjustment of the French industrial production index",
        x = "time", y = NULL) +
  geom_sa(color = "red", message = FALSE)

# To add of diagnostics with result of the X-11 combined test and the p-values
# of the residual seasonality qs and f tests:
diagnostics <- c("diagnostics.seas-sa-combined", "diagnostics.seas-sa-qs", "diagnostics.seas-sa-f")
p_sa_ipi_fr +
  geom_diagnostics(diagnostics = diagnostics,
                  ymin = 58, ymax = 72, xmin = 2010,
                  table_theme = gridExtra::ttheme_default(base_size = 8),
                  message = FALSE)

# To customize the names of the diagnostics in the plot:
diagnostics <- c(`Combined test` = "diagnostics.seas-sa-combined",
                `Residual qs-test (p-value)` = "diagnostics.seas-sa-qs",
```

```

      `Residual f-test (p-value)` = "diagnostics.seas-sa-f")
p_sa_ipi_fr +
  geom_diagnostics(diagnostics = diagnostics,
                  ymin = 58, ymax = 72, xmin = 2010,
                  table_theme = gridExtra::ttheme_default(base_size = 8),
                  message = FALSE)

# To add the table below the plot:

p_diag <- ggplot(data = ipi_c_eu_df, mapping = aes(x = date, y = FR)) +
  geom_diagnostics(diagnostics = diagnostics,
                  table_theme = gridExtra::ttheme_default(base_size = 8),
                  message = FALSE) +
  theme_void()

gridExtra::grid.arrange(p_sa_ipi_fr, p_diag,
                        nrow = 2, heights = c(4, 1))

```

 geom_outlier

Outliers texts

Description

Function to add directly to the plot the outliers used in the pre-adjustment process of the seasonal adjustment.

Usage

```

geom_outlier(
  mapping = NULL,
  data = NULL,
  stat = "outlier",
  geom = c("text", "label", "text_repel", "label_repel"),
  position = "identity",
  ...,
  method = c("x13", "tramoseats"),
  spec = NULL,
  frequency = NULL,
  message = TRUE,
  first_date = NULL,
  last_date = NULL,
  coefficients = FALSE,
  digits = 1,
  show.legend = NA,
  inherit.aes = TRUE
)

```

Arguments

mapping	Set of aesthetic mappings created by <code>aes()</code> or <code>aes_()</code> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	A data frame that contains the data used for the seasonal adjustment.
stat	The statistical transformation to use on the data for this layer, as a string.
geom	character. The geometric to use to display the data: <code>GeomText</code> (<code>geom = "text"</code> , the default, see <code>geom_text()</code>); <code>GeomLabel</code> (<code>geom = "label"</code> , see <code>geom_label()</code>); <code>GeomTextRepel</code> (<code>geom = "text_repel"</code> , the default, see <code>geom_text_repel()</code>); <code>GeomLabelRepel</code> (<code>geom = "label_repel"</code> , the default, see <code>geom_label_repel()</code>).
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	Other arguments passed on to <code>layer()</code> . They may be parameters of <code>geom_text()</code> (if <code>geom = "text"</code>), <code>geom_label()</code> (if <code>geom = "label"</code>), <code>geom_text_repel()</code> (if <code>geom = "text_repel"</code>) or <code>geom_label_repel()</code> (if <code>geom = "label_repel"</code>).
method	the method used for the seasonal adjustment. "x13" (by default) for the X-13ARIMA method and "tramoseats" for TRAMO-SEATS.
spec	the specification used for the seasonal adjustment. See <code>rjd3x13::x13()</code> or <code>rjd3tramoseats::tramoseats()</code> .
frequency	the frequency of the time series. By default (<code>frequency = NULL</code>), the frequency is computed automatically.
message	a boolean indicating if a message is printed with the frequency used.
first_date	A numeric specifying the first date from which the outliers are plotted. By default (<code>first_date = NULL</code>) the outliers are plotted from the beginning of the time series.
last_date	A numeric specifying the first date from which the outliers are plotted. By default (<code>first_date = NULL</code>) the outliers are plotted until the end of the time series.
coefficients	boolean indicating if the estimates coefficients are printed. By default <code>coefficients = FALSE</code> .
digits	integer indicating the number of decimal places to be used for numeric diagnostics. By default <code>digits = 1</code> .
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them.

Details

With the parameter `geom = "text"`, the outliers used in the pre-adjustment process of the seasonal adjustment are directly added to the plot. With `geom = "label"` a rectangle is drawn behind the names of the outliers, making them easier to read. The same with `geom = "text_repel"` or `geom = "label_repel"` but text labels are also repelled away from each other and away from the data points (see `geom_label_repel()`).

Examples

```

p_sa_ipi_fr <- ggplot(data = ipi_c_eu_df, mapping = aes(x = date, y = FR)) +
  geom_line() +
  labs(title = "Seasonal adjustment of the French industrial production index",
        x = "time", y = NULL) +
  geom_sa(color = "red", message = FALSE)

# To add the outliers:
p_sa_ipi_fr + geom_outlier(geom = "label",
                          message = FALSE)

# To have a more readable plot with outliers names that repeled away from each other
# and from the data points:
p_sa_ipi_fr +
  geom_outlier(geom = "label_repel",
              message = FALSE,
              vjust = 4,
              ylim = c(NA, 65), force = 10,
              arrow = arrow(length = unit(0.03, "npc"),
                            type = "closed", ends = "last"))

# To only plot the outliers from a specific date (2009):
p_sa_ipi_fr +
  geom_outlier(geom = "label_repel",
              message = FALSE,
              first_date = 2009,
              vjust = 4,
              ylim = c(NA, 65), force = 10,
              arrow = arrow(length = unit(0.03, "npc"),
                            type = "closed", ends = "last"))

```

geom_sa

Seasonal adjustment time series

Description

Performs a seasonal adjustment and plots a time series. Aids the eye in seeing patterns in the presence of overplotting. `geom_sa()` and `stat_sa()` are aliases: they both use the same arguments. Use `stat_sa()` if you want to display the results with a non-standard geom.

Usage

```

geom_sa(
  mapping = NULL,
  data = NULL,
  stat = "sa",
  position = "identity",
  ...,

```

```

method = c("x13", "tramoseats", "x11-extended", "fractionalairline",
  "fractionalairlineestimation", "multiairline"),
spec = NULL,
frequency = NULL,
message = TRUE,
component = "sa",
show.legend = NA,
inherit.aes = TRUE
)

stat_sa(
  mapping = NULL,
  data = NULL,
  geom = "line",
  position = "identity",
  ...,
  method = c("x13", "tramoseats", "x11-extended", "fractionalairline",
    "fractionalairlineestimation", "multiairline"),
  spec = NULL,
  frequency = NULL,
  message = TRUE,
  component = "sa",
  show.legend = NA,
  inherit.aes = TRUE
)

```

Arguments

mapping	Set of aesthetic mappings created by <code>aes()</code> or <code>aes_()</code> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	A <code>data.frame</code> that contains the data used for the seasonal adjustment.
stat	The statistical transformation to use on the data for this layer, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	Other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>colour = "red"</code> or <code>size = 3</code> .
method	the method used for the seasonal adjustment. "x13" (by default) for the X-13ARIMA method and "tramoseats" for TRAMO-SEATS.
spec	the specification used for the seasonal adjustment. See <code>rjd3x13::x13()</code> or <code>rjd3tramoseats::tramoseats()</code> .
frequency	the frequency of the time series. By default (<code>frequency = NULL</code>), the frequency is computed automatically.
message	a boolean indicating if a message is printed with the frequency used.
component	a character equals to the component to plot. The result must be a time series. See <code>rjd3x13::userdefined_variables_x13()</code> or <code>rjd3tramoseats::userdefined_variables_tram</code>

	the available parameters. By default (component = 'sa') the seasonal adjusted component is plotted.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them.
geom	The geometric object to use to display the data

Examples

```
p_ipi_fr <- ggplot(data = ipi_c_eu_df, mapping = aes(x = date, y = FR)) +
  geom_line() +
  labs(title = "Seasonal adjustment of the French industrial production index",
        x = "time", y = NULL)

# To add the seasonal adjusted series:
p_ipi_fr +
  geom_sa(color = "red")

# To add the forecasts of the input data and the seasonal adjusted series:
p_sa <- p_ipi_fr +
  geom_sa(component = "y_f", linetype = 2, message = FALSE) +
  geom_sa(component = "sa", color = "red", message = FALSE) +
  geom_sa(component = "sa_f", color = "red", linetype = 2, message = FALSE)
p_sa
```

ggplot_filters

'ggplot2' functions for 'rjd3filters' plots

Description

'ggplot2' functions for 'rjd3filters' plots

Usage

```
ggplot_coef(x, zero_as_na = TRUE, ylab = "Coefficients", ...)

## S3 method for class 'finite_filters'
ggplot_coef(x, zero_as_na = TRUE, ylab = "Coefficients", q = 0, ...)

ggplot_gain(x, nxlab = 7, xlim = c(0, pi), ..., n = 101)

ggplot_phase(x, nxlab = 7, xlim = c(0, pi), normalized = FALSE, ..., n = 101)
```

Arguments

x, zero_as_na, q, ylab, nxlab, xlim, n, normalized, ...
 see function [rjd3filters::plot_coef](#).

`ipi_c_eu`*Industrial Production Indices in manufacturing in the European Union*

Description

A dataset containing on monthly industrial production indices in manufacturing in the European Union (from `sts_inpr_m` dataset of Eurostat). Data are based 100 in 2015 and are unadjusted, i.e. neither seasonally adjusted nor calendar adjusted.

Usage`ipi_c_eu``ipi_c_eu_df`**Format**

A monthly `ts` object from january 1990 to december 2017 with 34 variables for `ipi_c_eu` and a `data.frame` for `ipi_c_eu_df`.

An object of class `data.frame` with 360 rows and 35 columns.

Details

The dataset contains 34 time series corresponding to the following geographical area

BE	Belgium
BG	Bulgaria
CZ	Czechia
DK	Denmark
DE	Germany (until 1990 former territory of the FRG)
EE	Estonia
IE	Ireland
EL	Greece
ES	Spain
FR	France
HR	Croatia
IT	Italy
CY	Cyprus
LV	Latvia
LT	Lithuania
LU	Luxembourg
HU	Hungary
MT	Malta
NL	Netherlands
AT	Austria
PL	Poland
PT	Portugal

RO Romania
SI Slovenia
SK Slovakia
FI Finland
SE Sweden
UK United Kingdom
NO Norway
CH Switzerland
ME Montenegro
MK Former Yugoslav Republic of Macedonia, the
RS Serbia
TR Turkey
BA Bosnia and Herzegovina

Source

http://ec.europa.eu/eurostat/wdds/rest/data/v2.1/json/en/sts_inpr_m?nace_r2=C&precision=1&sinceTimePeriod=1980M01&unit=I15&s_adj=NSA

siratio	<i>SI-ratio</i>
---------	-----------------

Description

SI-ratio

Usage

```
siratio(x, ...)
```

```
siratioplot(  
  x,  
  labels = NULL,  
  add = FALSE,  
  box = TRUE,  
  col.s = "darkblue",  
  col.i = "gray",  
  col.mean = "red",  
  cex.i = 0.1,  
  lwd.s = par("lwd"),  
  lwd.mean = lwd.s,  
  main = "SI ratio",  
  xlab = NULL,  
  ylab = NULL,  
  xlim = NULL,  
  ylim = NULL,  
  ...
```

```

)

ggsiratioplot(
  x,
  labels = NULL,
  col.s = "darkblue",
  col.i = "gray",
  col.mean = "red",
  cex.i = 0.5,
  lwd.s = 1,
  lwd.mean = lwd.s,
  main = "SI ratio",
  xlab = NULL,
  ylab = NULL,
  ...
)

```

Arguments

x	input model or data.
...	unused parameters.
labels	labels.
add	boolean indicating whether a new plot should be drawn.
box	boolean indicating a box around the current plot should be drawn.
col.s, col.i, col.mean	colors of the different components.
cex.i, lwd.s, lwd.mean	graphical parameters.
main, xlab, ylab	title, X and Y axis label.
xlim, ylim	X and Y axis limits.

Examples

```

x <- rjd3x13::x13(ipi_c_eu[, "FR"])
siratioplot(x)
ggsiratioplot(x)

```

 ts2df

Convert 'ts' object to 'data.frame'

Description

Function to a ts or mts object to a data.frame that can be directly used in the plot functions.

Usage

```
ts2df(x)
```

Arguments

x a ts or mts object.

Value

a data.frame object.

Examples

```
# To get the ipi_c_eu_df object:  
ts2df(ipi_c_eu)
```

usclaims

US Claims

Description

Weekly US claims

Usage

```
usclaims
```

Format

An object of class data.frame with 2868 rows and 3 columns.

Index

* datasets

ipi_c_eu, [12](#)
usclaims, [15](#)

aes(), [4](#), [6](#), [8](#), [10](#)

aes_(), [4](#), [6](#), [8](#), [10](#)

autoplot.JD3_X13_OUTPUT, [2](#)

calendar (components), [3](#)

calendaradj (components), [3](#)

components, [3](#)

geom_arima, [3](#)

geom_diagnostics, [5](#)

geom_label(), [4](#), [8](#)

geom_label_repel(), [8](#)

geom_outlier, [7](#)

geom_sa, [9](#)

geom_text(), [4](#), [8](#)

geom_text_repel(), [8](#)

ggplot_coef (ggplot_filters), [11](#)

ggplot_filters, [11](#)

ggplot_gain (ggplot_filters), [11](#)

ggplot_phase (ggplot_filters), [11](#)

ggsiratioplot (siratio), [13](#)

ipi_c_eu, [12](#)

ipi_c_eu_df (ipi_c_eu), [12](#)

irregular (components), [3](#)

layer(), [4](#), [6](#), [8](#), [10](#)

raw (components), [3](#)

rjd3filters::plot_coef, [11](#)

rjd3tramoseats::tramoseats(), [4](#), [6](#), [8](#), [10](#)

rjd3tramoseats::userdefined_variables_tramoseats(),
[6](#), [10](#)

rjd3x13::userdefined_variables_x13(),
[6](#), [10](#)

rjd3x13::x13(), [4](#), [6](#), [8](#), [10](#)

seasonal (components), [3](#)

seasonaladj (components), [3](#)

siratio, [13](#)

siratioplot (siratio), [13](#)

stat_sa (geom_sa), [9](#)

trendcycle (components), [3](#)

ts2df, [14](#)

ttheme_default(), [6](#)

usclaims, [15](#)