

Package ‘blsplotR’

January 30, 2024

Title Plots for Seasonal Adjustment Analysts

Version 1.4.2

Description Generates several types of time series plots useful for seasonal adjustment analysis. These routines rely heavily on the seasonal package to extract series and components from the 'seasonal adjustments generated by the US Census Bureau's X-13ARIMA-SEATS software, and can be generated from a single seas object or a list of seas objects. Types of plots include line plots, ratio plots, forecast plots, forecast error diagnostic plots, spectral plots, seasonal factor plots, seasonal adjustment component plots. Users can add grid lines and shade recession regions in selected plots.

License MIT + file LICENSE

Encoding UTF-8

LazyData TRUE

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports assertive.data,
gplots,
grDevices,
plotrix,
R.devices,
sautilities,
seasonal,
stringr,
tis

Depends R (>= 2.10)

R topics documented:

cnv_color_codes	2
color_blind_palette	3
convert_date_to_tis	3
display_color	4
draw_recession	5
employment_data_mts	6
employment_list	6
flag_peak	7
from_rgb_to_hue	7

get_recession_dates	8
plot_all	9
plot_cpgram_resid	12
plot_double_spectrum	13
plot_fcst	14
plot_fcst_history	15
plot_fts	17
plot_matrix	18
plot_multiple	20
plot_ratio	22
plot_resid	23
plot_sa_list	25
plot_sa_list_split	27
plot_series	29
plot_sf	32
plot_sf_mean	33
plot_sf_series	36
plot_single_cell	38
plot_table	39
plot_year_over_year	41
reset_par	42
sample_shades	43
sort_hex_by_hue	43
visual_sig_peaks	44
wheel_invisible	44
xt_data_list	45
Index	46

cnv_color_codes	<i>Generates color names from hexadecimal input</i>
-----------------	---

Description

Generates vector of closest color names from hexadecimal color input. This uses the color.id function of plotrix to do the conversion.

Usage

```
cnv_color_codes(color_vec)
```

Arguments

color_vec	vector of color codes
-----------	-----------------------

Value

vector of color names closest to hexadecimal color input

Examples

```

Moonrise_Codes <-
  c("#F3DF6C", "#CEAB07", "#D5D5D3", "#24281A", "#798E87",
    "#C27D38", "#CCC591", "#29211F", "#85D4E3", "#F4B5BD",
    "#9C964A", "#CDC08C", "#FAD77B")
Moonrise_All <- cnv_color_codes(Moonrise_Codes)

```

color_blind_palette	<i>Color-blind friendly color palette</i>
---------------------	---

Description

Color palettes that can be used that can be distinguished by color-blind people (From Cookbook for R - Colors (ggplot2)).

Usage

```
color_blind_palette(with_grey = TRUE)
```

Arguments

with_grey	Logical scalar; whether color blind palette contains 'grey', otherwise the palette contains black. Default is TRUE.
-----------	---

Value

vector of hexadecimal color codes that form a color palette that can be distinguished by color-blind people.

Examples

```
this_color_blind <- color_blind_palette(FALSE)
```

convert_date_to_tis	<i>Convert ts dates to tis format</i>
---------------------	---------------------------------------

Description

Convert dates used for monthly (or quarterly) ts series to tis formats

Usage

```

convert_date_to_tis(
  this_date,
  this_freq = 12,
  is_start = TRUE,
  return_tis = FALSE
)

```

Arguments

this_date	numeric scalar or vector; ts date to be converted
this_freq	numeric scalar; frequency of ts time series. Default is 12 (monthly).
is_start	logical scalar; is date assumed to be the beginning of the month? Default is TRUE; if FALSE, date is assumed to be at the end of the month.
return_tis	logical scalar; If true, return as tis object; otherwise return as integer Default is FALSE.

Value

a tis index value that is the equivalent of the codets date

Examples

```
this_month <- as.numeric(substr(Sys.Date(),1,4)) +
  (as.numeric(substr(Sys.Date(),6,7)) - 1) / 12
end_this_month_tis <-
  convert_date_to_tis(this_month, this_freq = 12,
    is_start = FALSE, return_tis = TRUE)
start_this_month_tis <-
  convert_date_to_tis(this_month, this_freq = 12,
    is_start = TRUE, return_tis = TRUE)
```

display_color	<i>Color name for display</i>
---------------	-------------------------------

Description

Generates color names for display on plot labels and subheaders

Usage

```
display_color(this_color_code, strip_numbers = FALSE)
```

Arguments

this_color_code	character string of color code to be used in plot
strip_numbers	logical scalar that controls if numbers at the end of the text are stripped from the color name. Default is TRUE.

Value

character string of color name closest to hexadecimal color input (if used) stripped of numbers if strip_numbers = TRUE

Examples

```
this_color_blind <- color_blind_palette()
this_color_blind_text <- array(NA, dim = 8)
for (i in 1:8) {
  this_color_blind_text[i] <- display_color(this_color_blind[i])
}
```

draw_recession	<i>Draw NBER recessions</i>
----------------	-----------------------------

Description

Draws shaded areas in plots corresponding to NBER recessions

Usage

```
draw_recession(
  this_col_recess = NULL,
  this_density = 50,
  this_border = NA,
  this_add_recess_start = NULL,
  this_sub_recess = TRUE,
  this_sub_line = 2.5,
  this_sub_cex = 0.75,
  display_color_strip = FALSE
)
```

Arguments

<code>this_col_recess</code>	Character string; color used for shading recession periods. Default is 'lightgrey'.
<code>this_density</code>	the density of shading lines, in lines per inch. The default value is 50. A zero value of density means no shading lines whereas negative values (and NA) suppress shading (and so allow color filling).
<code>this_border</code>	Integer scalar; thickness of border around region. Default is NA, meaning the border is not generated.
<code>this_add_recess_start</code>	numeric scalar; Starting date for an additional recession period at the end of the series. Default is not to add recession dates.
<code>this_sub_recess</code>	Logical scalar; indicates if x-axis label for recession is produced for this plot. Default is x-axis label is produced
<code>this_sub_line</code>	Integer scalar; position of subtitle of plot. Default is 2.5.
<code>this_sub_cex</code>	Numeric scalar; scaling for subtitle of plot. Default is 0.75.
<code>display_color_strip</code>	Logical scalar; indicates if the display color will be stripped of trailing numbers. Default is FALSE.

Value

Shades recession dates in plots

Examples

```
air_seas <- seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11='')
plot_table(air_seas, 'd12', 'AirPassengers', do_grid = TRUE, draw_recess = FALSE,
  use_ratio = FALSE, add_sub_title = TRUE, this_col = 'forestgreen')
draw_recession(this_col_recess = 'lightblue', this_border = 1,
  this_sub_line = 2.25, this_sub_cex = 0.75)
```

employment_data_mts	<i>US Unemployment Series, four main components in an mts object</i>
---------------------	--

Description

#' An mts object of the four main components of US Employment expressed as time series objects that end in December, 2022

Usage

```
employment_data_mts
```

Format

An mts object with 4 time series elements in four columns:

n2000013 Employed Males 16-19

n2000014 Employed Females 16-19

n2000025 Employed Males 20+

n2000026 Employed Females 20+

employment_list	<i>US Employment Series, four main components in a list object</i>
-----------------	--

Description

#' A list object of the four main components of US Employment expressed as time series objects that end in December, 2022

Usage

```
employment_list
```

Format

A list object with 4 time series elements:

n2000013 Employed Males 16-19

n2000014 Employed Females 16-19

n2000025 Employed Males 20+

n2000026 Employed Females 20+

flag_peak

*Flag visual significant peaks in spectra***Description**

Determine positions of visual significant peaks in spectra

Usage

```
flag_peak(this_seas, spec_type, spec_freq_code, max_freq)
```

Arguments

`this_seas` seas object generated from a call of seas on a single time series

`spec_type` Character string; type of spectrum. Possible values are 'ori', 'irr', 'rsd', 'sa'.

`spec_freq_code` Character string; type of frequency being tested. Possible values are 's' or 't'.

`max_freq` Numeric string; maximum number of frequencies to test.

Value

If visually significant peaks found, a numeric vector of the position of the peak frequencies. If no peaks found, 0.

Examples

```
air_seas <- seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11='')
this_flagged_peak_seas <- flag_peak(air_seas, 'ori', 's', 5)
this_flagged_peak_td <- flag_peak(air_seas, 'ori', 't', 2)
```

from_rgb_to_hue

*Sort hexadecimal colors by hue***Description**

Sort a vector of hexadecimal colors by hue. Taken from <https://www.biolinio.com/from-rgb-to-hsl-colors-in-r>

Usage

```
from_rgb_to_hue(rgb = NULL)
```

Arguments

`rgb` Character vector containing a list of rgb color codes.

Value

A character vector with the color codes sorted by hue

Examples

```
this_wheel_hex <- wheel_invisible("blue", 12)
this_wheel_rgb <- sapply(this_wheel_hex, function(x){col2rgb(x)})
this_wheel_hue <- from_rgb_to_hue(this_wheel_rgb)
```

get_recession_dates	<i>Get NBER recession dates</i>
---------------------	---------------------------------

Description

Generate starting and ending dates for NBER recessions between two monthly (or quarterly) dates

Usage

```
get_recession_dates(
  start_recess = NULL,
  end_recess = NULL,
  add_recess_start = NULL,
  this_freq = 12
)
```

Arguments

start_recess	numeric scalar; Starting date for plot. Default is first recession starting date.
end_recess	numeric scalar; Ending date for plot. Default is last recession ending date.
add_recess_start	numeric scalar; Starting date for an additional recession period at the end of the series. Default is not to add recession dates.
this_freq	numeric scalar; frequency of ts time series. Default is 12 (monthly).

Value

Starting and ending dates for NBER recessions within a span of data

Examples

```
plot_limits <- c(1949, 1961)
thisRec <-
  get_recession_dates(start_recess = plot_limits[1], end_recess = plot_limits[2])
```


plot_all

*Generate all diagnostic plots this***Description**

Generates a series of diagnostic plots from a single seas object and store the results in a separate file

Usage

```
plot_all(
  this_seas = NULL,
  series_name = NULL,
  file_base = NULL,
  this_dir = NULL,
  this_start = NULL,
  split_plots = FALSE,
  plot_type = "pdf",
  this_grid = FALSE,
  this_draw_recess = FALSE,
  this_add_recess_start = NULL,
  this_recess_col = NULL,
  this_recess_sub = TRUE,
  this_otl = FALSE,
  this_si = FALSE,
  this_mean_line = TRUE,
  this_specturm_axis = TRUE,
  this_ratio = FALSE,
  this_add_identified_otl = FALSE,
  this_sub_title = FALSE,
  col_ori = "grey",
  col_sa = "forestgreen",
  col_one = "steelblue",
  col_factor = "forestgreen",
  col_fcst = c("grey", "forestgreen", "red"),
  col_otl = c("red", "blue", "forestgreen", "brown", "grey", "yellow"),
  col_sf = c("forestgreen", "darkblue", "grey"),
  col_spec = c("blue", "forestgreen", "grey", "brown", "red", "orange"),
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
  this_axis_cex = NULL,
  this_otl_cex = 0.5,
  main_title_cex = NULL,
  sub_title_cex = NULL,
  this_mar = c(4, 4, 4, 0.5),
  this_reset = FALSE
)
```

Arguments

this_seas seas object generated from a call of seas on a single time series

<code>series_name</code>	Character scalar; name of the time series used in <code>this_seas</code> .
<code>file_base</code>	Character scalar; base file name for the graphics file generated. Default base file name is 'BLSplot'.
<code>this_dir</code>	Character scalar; directory where the graphics file generated. Default is the current working directory.
<code>this_start</code>	Integer vector of length 2; Starting date for plot. Default is starting date for the time series.
<code>split_plots</code>	Logical scalar; indicates if plots will be split into different files. Default is combine the plots into one file.
<code>plot_type</code>	Character scalar; Type of graphics file generated. Default is 'pdf'.
<code>this_grid</code>	Logical scalar; indicates if certain plots will have grid lines. Default is no grid lines.
<code>this_draw_recess</code>	Logical scalar; indicates if certain plots will have shaded areas for NBER recession dates. Default is no recession shading.
<code>this_add_recess_start</code>	numeric scalar; Starting date for an additional recession period at the end of the series. Default is not to add recession dates.
<code>this_recess_col</code>	Character string; color used for shading of recession region. Default is 'lightgrey'.
<code>this_recess_sub</code>	Logical scalar; indicates if x-axis label for recession is produced for this plot. Default is x-axis label is produced
<code>this_otl</code>	Logical scalar; indicates if lines for identified outliers are included in series plots. Default is not including lines for identified outliers.
<code>this_si</code>	Logical scalar; indicates if seasonal factor plots will include SI ratios for X-11 seasonal adjustments. Default is not including SI ratios.
<code>this_mean_line</code>	Logical scalar; indicates if seasonal factor plots will include lines for seasonal means. Default includes lines for seasonal means.
<code>this_spectrum_axis</code>	Logical scalar; indicates if x-axis of spectral plot will be frequency by month rather than the actual frequencies. Default sets x-axis to frequency by month.
<code>this_ratio</code>	Logical scalar; indicates if plots of seasonal factors, irregular, and residuals are done as ratio plots. Default has these plots as time series line plots.
<code>this_add_identified_otl</code>	Logical scalar; indicates if outlier plots will include identified outliers. Default is not including identified outliers.
<code>this_sub_title</code>	Logical scalar; indicates if certain plots will include subtitles denoting what series are plotted. Default is not including subheaders.
<code>col_ori</code>	Character scalar; color used for the original series. Default is grey.
<code>col_sa</code>	Character scalar; color used for the seasonally adjusted series. Default is forestgreen.
<code>col_one</code>	Character scalar; color used for individual series. Default is blue.
<code>col_factor</code>	Character scalar; color used for factor plots. Default is forestgreen.
<code>col_fcst</code>	Array of character strings; color used for original series, forecast, and forecast bounds. Default is <code>c('grey', 'forestgreen', 'red')</code> .
<code>col_otl</code>	Character array of length 6; color used for different outliers, with the order being 'ao', 'ls', 'tc', 'so', 'rp', 'tls'. Default is <code>c('red', 'blue', 'forestgreen', 'brown', 'grey', 'yellow')</code> .

col_sf	Character array of length 3; color used for special seasonal plots, with the order being seasonal factors, SI ratio, seasonal mean. Default is c('forestgreen', 'darkblue', 'grey').
col_spec	Character array of length 6; color used in specturm plots, in the order of spectrum of ori, spectrum of SA, line for seasonal frequency, line for TD frequency, star for visually significant seasonal frequency, star for visually significant TD frequency. Default is c('blue', 'forestgreen', 'grey', 'brown', 'red', 'orange').
this_plot_cex	Numeric scalar; scaling for the plot itself. Default is 0.8.
this_lab_cex	Numeric scalar; scaling for plot labels. Default is the value of this_plot_cex.
this_axis_cex	Numeric scalar; scaling for plot axis. Default is the value of this_plot_cex.
this_otl_cex	Numeric scalar; sets the cex plotting parameter for the fts plot. Default sets cex = 0.5.
main_title_cex	Numeric scalar; scaling for main title of plot. Default is this_plot_cex + 0.1.
sub_title_cex	Numeric scalar; scaling for subtitle of plot. Default is this_plot_cex - 0.1.
this_mar	Numeric vector; set margins for the plot. Default is c(4,4,4,0,5).
this_reset	Logical scalar; if TRUE, the values of par are reset. Default is FALSE.

Value

Graphics file with number of diagnostic plots routinely used at BLS.

Examples

```
air_seas <-
  seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)',
    forecast.maxlead = 60, x11='',
    check.print = c("none", "+acf", "+acfplot", "+normalitytest"))

## Not run: blsplotR::plot_all(air_seas, series_name = 'AirPassengers',
  file_base = 'AirPass', this_dir = 'X:/seasonalAdj/graphs/',
  split_plots = TRUE, plot_type = 'png', this_grid = FALSE,
  this_draw_recess = FALSE,
  this_ratio = TRUE, this_add_identified_otl = TRUE,
  col_sa = 'darkred', col_one = 'steelblue')
## End(Not run)

## Not run: blsplotR::plot_all(air_seas, series_name = 'AirPassengers',
  file_base = 'AirPass', this_dir = 'X:/seasonalAdj/graphs/',
  this_grid = TRUE,
  this_ratio = TRUE, this_add_identified_otl = TRUE,
  col_factor = 'darkorange',
  col_fcst = c('steelblue', 'forestgreen', 'darkred'),
  col_sa = 'orange', col_one = 'violet')
## End(Not run)
```

plot_cpgram_resid	<i>Generate cumulative periodogram of the regARIMA residuals</i>
-------------------	--

Description

Generates a plot of the cumulative periodogram of the regARIMA residuals

Usage

```
plot_cpgram_resid(
  this_seas = NULL,
  main_title = "Cumulative periodogram",
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
  this_main_cex = 0.9,
  this_sub_cex = 0.65,
  this_axis_cex = NULL,
  this_reset = TRUE
)
```

Arguments

this_seas	seas object generated from a call of seas on a single time series
main_title	Character string; main title of plot. Default is 'Cumulative periodogram'.
this_plot_cex	Numeric scalar; scaling for the plot itself. Default is 0.8.
this_lab_cex	Numeric scalar; scaling for plot labels. Default is the value of this_plot_cex.
this_main_cex	Numeric scalar; scaling for main plot title. Default is the value of 0.9.
this_sub_cex	Numeric scalar; scaling for plot subtitle. Default is the value of 0.65.
this_axis_cex	Numeric scalar; scaling for plot axis. Default is the value of this_plot_cex.
this_reset	Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generates a plot of the Cumulative periodogram of the regARIMA residuals. Diagnostic information is included in subheaders.

Examples

```
air_seas <- seasonal::seas(AirPassengers, transform.function= 'log', arima.model = '(0 1 1)(0 1 1)')
plot_cpgram_resid(air_seas, main_title = 'Cumulative periodogram for Airline Passenger Residuals')
```

plot_double_spectrum	<i>Generate double spectrum plot of the original and seasonally adjusted series.</i>
----------------------	--

Description

Generate plot of spectrum of original series and seasonally adjusted series on same axis.

Usage

```
plot_double_spectrum(
  this_seas = NULL,
  xaxis_bls = TRUE,
  main_title = "AR Spectrum",
  series_name = NULL,
  this_col = c("blue", "green", "grey", "brown", "red", "orange"),
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
  this_main_cex = NULL,
  this_sub_cex = NULL,
  this_axis_cex = NULL,
  main_title_line = 3,
  sub_title_line = 1,
  display_color_strip = FALSE,
  this_reset = TRUE
)
```

Arguments

this_seas	seas object generated from a call of seas on a single time series
xaxis_bls	Logical scalar; indicates if x-axis of spectral plot will be frequency by month rather than the actual frequencies. Default sets x-axis to frequency by month.
main_title	Character string; main title of plot. Default is 'AR Spectrum'.
series_name	Character scalar; name of the time series used in m.
this_col	Character array of length 6; color used in spectrum plots, in the order of spectrum of ori, spectrum of SA, line for seasonal frequency, line for TD frequency, star for visually significant seasonal frequency, star for visually significant TD frequency. Default is c('blue', 'green', 'grey', 'brown', 'red', 'orange').
this_plot_cex	Numeric scalar; scaling for the plot itself. Default is 0.8.
this_lab_cex	Numeric scalar; scaling for plot labels. Default is the value of this_plot_cex.
this_main_cex	Numeric scalar; scaling for main plot title. Default is the value of this_plot_cex.
this_sub_cex	Numeric scalar; scaling for plot subtitle. Default is the value of this_plot_cex.
this_axis_cex	Numeric scalar; scaling for plot axis. Default is the value of this_plot_cex.
main_title_line	Integer scalar; position of main title of plot. Default is 3.00.
sub_title_line	Integer scalar; position of main title of plot. Default is 1.00.
display_color_strip	Logical scalar; indicates if the display color will be stripped of trailing numbers. Default is FALSE.
this_reset	Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generate plot of spectrum of original series and seasonally adjusted series on same axis.

Examples

```
air_seas <- seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11='')
plot_double_spectrum(air_seas, series_name = 'AirPassengers',
  this_col = c('blue', 'green', 'darkblue', 'darkgreen', 'red', 'orange'))
```

plot_fcst	<i>Forecast plot</i>
-----------	----------------------

Description

Generates regARIMA forecasts with confidence bounds

Usage

```
plot_fcst(
  this_seas = NULL,
  main_title = "ARIMA forecasts",
  do_grid = FALSE,
  do_sub = TRUE,
  length_ori = 2,
  this_col = c("darkgrey", "blue", "darkgreen"),
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
  this_main_cex = 1,
  this_sub_cex = 0.7,
  this_axis_cex = NULL,
  main_title_line = 2.25,
  this_reset = TRUE
)
```

Arguments

this_seas	seas object generated from a call of seas on a single time series
main_title	Character string; main title of plot. Default is 'ARIMA Residuals'.
do_grid	Logical scalar; indicates if certain plots will have grid lines. Default is no grid lines.
do_sub	Logical scalar; indicates if subtitle is generated. Default is to generate the subtitle.
length_ori	Integer scalar; number of years of the original series to show with forecasts. Default is 2 years.
this_col	Array of character strings; color used for original series, forecast, and forecast bounds. Default is c("darkgrey", "blue", "darkgreen").
this_plot_cex	Numeric scalar; scaling for the plot itself. Default is 0.8.
this_lab_cex	Numeric scalar; scaling for plot labels. Default is the value of this_plot_cex.
this_main_cex	Numeric scalar; scaling for main plot title. Default is the value of 1.0.

this_sub_cex Numeric scalar; scaling for plot subtitle. Default is the value of 0.7.
 this_axis_cex Numeric scalar; scaling for plot axis. Default is the value of this_plot_cex.
 main_title_line
 Integer scalar; position of main title of plot. Default is 2.25.
 this_reset Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generates a plot of the regARIMA forecasts with confidence bounds.

Examples

```
air_seas <- seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', forecast.maxlead = 60)
plot_fcst(air_seas, main_title = 'Forecasts for Airline Passengers', do_grid = TRUE)
```

plot_fcst_history	<i>Generate forecast history plot</i>
-------------------	---------------------------------------

Description

Generate forecast history plot, which compares the sum of squared forecast errors for two models.

Usage

```
plot_fcst_history(
  seas_md11 = NULL,
  seas_md12 = NULL,
  start_hist = NULL,
  main_title = "Differences in the Sum of Squared Forecast Errors",
  name_md11 = "Model 1",
  name_md12 = "Model 2",
  this_col = c("blue", "darkgreen"),
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
  this_main_cex = NULL,
  this_sub_cex = NULL,
  this_axis_cex = NULL,
  this_mar = c(5.1, 2.1, 5.1, 0.5),
  main_title_line = 2.5,
  sub_title_line = 1,
  display_color_strip = FALSE,
  this_reset = TRUE
)
```

Arguments

seas_md11	seas object generated from a call of seas on a single time series for the first model
seas_md12	seas object generated from a call of seas on a single time series for the second model

start_hist	integer scalar; starting date for the history analysis. Could be an array of length 2; will be converted to a scalar
main_title	Character string; main title of plot. Default is 'Differences in the Sum of Squared Forecast Errors'.
name_md1	Character string; Description of first model for use in the subtitle. Default is 'Model 1'.
name_md2	Character string; Description of second model for use in the subtitle. Default is 'Model 2'.
this_col	Character array of length 2; color used for each forecast lag. Default is c('blue', 'darkgreen').
this_plot_cex	Numeric scalar; scaling for the plot itself. Default is 0.8.
this_lab_cex	Numeric scalar; scaling for plot labels. Default is the value of this_plot_cex.
this_main_cex	Numeric scalar; scaling for main plot title. Default is the value of this_plot_cex + 0.1.
this_sub_cex	Numeric scalar; scaling for plot subtitle. Default is the value of this_plot_cex.
this_axis_cex	Numeric scalar; scaling for plot axis. Default is the value of this_plot_cex.
this_mar	Numeric vector; set margins for the plot. Default is c(5.1, 2.1, 5.1, 0.5).
main_title_line	Integer scalar; position of main title of plot. Default is 2.25.
sub_title_line	Integer scalar; position of main title of plot. Default is 1.
display_color_strip	Logical scalar; indicates if the display color will be stripped of trailing numbers. Default is FALSE.
this_reset	Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generate forecast history plot. Can be more than one series. If series not specified, print out error message and return NULL.

Examples

```
air_seas_md1 <-
  seasonal::seas(AirPassengers, x11='', slidingspans = '', transform.function = 'log',
    arima.model = '(0 1 1)(0 1 1)', regression.aictest = NULL, outlier = NULL,
    forecast.maxlead = 36, check.print = c('pacf', 'pacfplot'),
    history.fstep = c(1, 12), history.estimates = 'fcst',
    history.save = 'fcsterrors')
air_seas_md2 <-
  seasonal::seas(AirPassengers, x11='', slidingspans = '', transform.function = 'log',
    arima.model = '(0 1 1)(0 1 1)', regression.variables = c("td"),
    forecast.maxlead = 36, check.print = c('pacf', 'pacfplot'),
    history.fstep = c(1, 12), history.estimates = 'fcst',
    history.save = 'fcsterrors')
plot_fcst_history(air_seas_md1, air_seas_md2, start_hist = 1957.0,
  main_title = 'Differences in the Sum of Squared Forecast Errors for Airline Passengers',
  name_md1 = 'Airline model', name_md2 = 'Airline model + regressors')
```


plot_fts

*Final t-statistics for the outlier identification procedure plot***Description**

Generates a plot of the final t-statistics for the outlier identification procedure.

Usage

```
plot_fts(
  this_seas = NULL,
  fts,
  this_cex = 0.5,
  start_plot = NULL,
  main_title = "Outlier T-Values",
  add_identified_otl = FALSE,
  col_otl = c("red", "blue", "darkgreen"),
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
  this_main_cex = NULL,
  this_sub_cex = NULL,
  this_axis_cex = NULL,
  this_mar = c(4.1, 2.1, 4.1, 0.5),
  main_title_line = 2.25,
  sub_title_line = 1,
  this_reset = TRUE
)
```

Arguments

this_seas	seas object generated from a call of seas on a single time series
fts	time series matrix containing final outlier t-statistics for all types of outlier specified by the user.
this_cex	Numeric scalar; sets the cex plotting parameter. Default sets cex = 0.5.
start_plot	Integer vector of length 2; Starting date for plot. Default is starting date for the time series.
main_title	Character string; main title of plot. Default is 'Outlier T-Values'.
add_identified_otl	Logical scalar; indicates if outlier plots will include identified outliers. Default is not including identified outliers.
col_otl	Character array of length 3; color used for different outliers, with the order being 'ao', 'ls', 'tc'. Default is c('red', 'blue', 'darkgreen').
this_plot_cex	Numeric scalar; scaling for the plot itself. Default is 0.8.
this_lab_cex	Numeric scalar; scaling for plot labels. Default is the value of this_plot_cex.
this_main_cex	Numeric scalar; scaling for main plot title. Default is the value of this_plot_cex + 0.1.
this_sub_cex	Numeric scalar; scaling for plot subtitle. Default is the value of this_plot_cex.
this_axis_cex	Numeric scalar; scaling for plot axis. Default is the value of this_plot_cex.

this_mar Numeric vector; set margins for the plot. Default is c(4.1, 2.1, 4.1, 0.5).
 main_title_line Integer scalar; position of main title of plot. Default is 2.25.
 sub_title_line Integer scalar; position of main title of plot. Default is 1.
 this_reset Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generates a plot of the final t-statistics from the automatic outlier identification procedure.

Examples

```

air_seas_outlier <-
  seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', outlier.types = 'all')
air_fts_good <- seasonal::series(air_seas_outlier, "fts")
plot_fts(air_seas_outlier, air_fts_good,
  main_title = 'Outlier T-Values for Airline Passengers')

```

plot_matrix	<i>Plot time series matrix</i>
-------------	--------------------------------

Description

Generate plot of a matrix of user-specified time series.

Usage

```

plot_matrix(
  this_matrix = NULL,
  main_title = NULL,
  main_title_line = 1.5,
  main_title_cex = NULL,
  y_label = NULL,
  start_plot = NULL,
  do_grid = FALSE,
  draw_recess = FALSE,
  recess_start = NULL,
  recess_col = NULL,
  recess_sub = TRUE,
  recess_sub_line = 2.25,
  recess_sub_cex = NULL,
  this_col = c("grey", "blue", "green", "brown", "red", "yellow"),
  this_line_type = rep(1, 6),
  add_legend = FALSE,
  this_legend_position = "topleft",
  this_legend_title = "Series",
  this_legend_inset = 0,
  this_legend_entry = paste("srs", 1:6, sep = ""),
  this_legend_cex = 0.8,
  this_plot_cex = 0.8,
  this_lab_cex = NULL,

```

```

    this_axis_cex = NULL,
    this_mar = c(4.1, 4.1, 4.1, 0.5),
    this_reset = TRUE
)

```

Arguments

<code>this_matrix</code>	Numeric matrix; columns of time series object to be plotted.
<code>main_title</code>	Character string; main title of plot. Default is column name.
<code>main_title_line</code>	Integer scalar; position of main title of plot. Default is 1.5.
<code>main_title_cex</code>	Numeric scalar; scaling for main title of plot. Default is <code>this_plot_cex + 0.1</code> .
<code>y_label</code>	Character string; y-axis label for plot, if specified.
<code>start_plot</code>	Integer vector of length 2; Starting date for plot. Default is starting date for the time series.
<code>do_grid</code>	Logical scalar; indicates if certain plots will have grid lines. Default is no grid lines.
<code>draw_recess</code>	Logical scalar; indicates if certain plots will have shaded areas for NBER recession dates. Default is no recession shading.
<code>recess_start</code>	numeric matrix; Rows of dates for additional recession starting and ending dates. Default is not to add recession dates.
<code>recess_col</code>	Character string; color used for shading of recession region. Default is 'lightgrey'.
<code>recess_sub</code>	Logical scalar; indicates if x-axis label for recession is produced for this plot. Default is x-axis label is produced
<code>recess_sub_line</code>	Integer scalar; position of recession text of plot. Default is 2.25.
<code>recess_sub_cex</code>	Numeric scalar; scaling for recessoion text of plot. Default is the value of <code>this_plot_cex</code> .
<code>this_col</code>	Character array of length 6; color used for series in the order specified by the user. Default is <code>c('grey', 'blue', 'green', 'brown', 'red', 'yellow')</code> .
<code>this_line_type</code>	Integer vector; indicates line type of each plot produced. Default is 1:6
<code>add_legend</code>	Logical scalar; indicates if legend is produced for this plot. Default is legend not produced
<code>this_legend_position</code>	Character string; indicates position of legend. Default is 'topleft'.
<code>this_legend_title</code>	Character string; indicates title of legend. Default is 'Series'.
<code>this_legend_inset</code>	Integer scalar; indicates inset for legend. Default is 0.
<code>this_legend_entry</code>	Character array; entries for the lengend. Default is 'Srs1'.
<code>this_legend_cex</code>	Numeric scalar; scaling for legend. Default is 0.8.
<code>this_plot_cex</code>	Numeric scalar; scaling for the plot itself. Default is 0.8.
<code>this_lab_cex</code>	Numeric scalar; scaling for plot labels. Default is the value of <code>this_plot_cex</code> .
<code>this_axis_cex</code>	Numeric scalar; scaling for plot axis. Default is the value of <code>this_plot_cex</code> .
<code>this_mar</code>	Numeric vector; set margins for the plot. Default is <code>c(4.1, 2.1, 4.1, 0.5)</code> .
<code>this_reset</code>	Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generate plot of user-specified series. If matrix not specified, print out error message and return NULL.

Examples

```
BP_Region_Matrix <-
  cbind(blsplotR::xt_data_list$mwlu, blsplotR::xt_data_list$ne1u,
        blsplotR::xt_data_list$so1u, blsplotR::xt_data_list$we1u)
colnames(BP_Region_Matrix) <- names(blsplotR::xt_data_list)
blsplotR::plot_matrix(BP_Region_Matrix, y_label = 'Building Permits',
  do_grid = TRUE, main_title = "US Building Permits, 1 Family Units",
  draw_recess = TRUE, this_col = c("red", "steelblue", "forestgreen", "brown"),
  add_legend = TRUE, this_legend_title = NULL,
  this_legend_entry = names(blsplotR::xt_data_list),
  this_legend_cex = 0.75)
```

plot_multiple

Multiple plots on a single page

Description

Generates a page of plots for a time series, seasonal adjustment of the time series, and trend component. Plotting the trend is optional. The series name is used for the title.

Usage

```
plot_multiple(
  seas_obj_list = NULL,
  first_series = 1,
  last_series = NULL,
  this_row = NULL,
  this_col = NULL,
  plot_trend = FALSE,
  seas_obj_names = NULL,
  plot_start = NULL,
  plot_end = NULL,
  outer_title,
  group_title = NULL,
  col_vec = c("grey", "blue", "green"),
  do_grid = FALSE,
  draw_recess = FALSE,
  recess_start = NULL,
  recess_col = NULL,
  recess_sub = TRUE,
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
  this_main_cex = NULL,
  this_axis_cex = NULL,
  this_mar = c(3.1, 2.1, 3.1, 1.1),
  this_oma = c(0, 0, 2, 0),
```

```

    main_title_line = 1.25,
    this_reset = TRUE
)

```

Arguments

seas_obj_list	List of seas arguments generated by seas() of the seasonal package. This argument must be specified.
first_series	Integer scalar; index of first series to be plotted within the plotting frame. Default: 1.
last_series	Integer scalar; index of last series to be plotted within the plotting frame. Default: number of seas objects in seas_obj_list
this_row	Integer scalar; number of rows in multi-frame plot. This argument must be specified.
this_col	Integer scalar; number of columns in multi-frame plot. This argument must be specified.
plot_trend	Logical scalar; if TRUE, trend is included in plot, FALSE trend is not included. Default: FALSE
seas_obj_names	Vector of character strings; the names of the series being plotted. By Default, the names will be taken directly from the seasonal object list.
plot_start	Integer array of length 2; start date for series to be plotted. If not specified, starting date of series used.
plot_end	Integer array of length 2; end date for series to be plotted. If not specified, ending date of series used.
outer_title	Character string, outer title of set of plots. If not specified, outer title is 'Series (grey), SA (blue), Trend (green) plot' or 'Series (grey), SA (blue) plot' if trend isn't specified.
group_title	Character string with a group title of series, if specified.
col_vec	Character array of length 5; color vector for lines in the plots. Default is c('grey', 'blue', 'green') for original series, SA, Trend
do_grid	Logical scalar; indicates if certain plots will have grid lines. Default is no grid lines.
draw_recess	Logical scalar; indicates if certain plots will have shaded areas for NBER recession dates. Default is no recession shading.
recess_start	numeric matrix; Rows of dates for additional recession starting and ending dates. Default is not to add recession dates.
recess_col	Character string; color used for shading of recession region. Default is 'lightgrey'.
recess_sub	Logical scalar; indicates if x-axis label for recession is produced for this plot. Default is x-axis label is produced
this_plot_cex	Numeric scalar; scaling for the plot itself. Default is 0.8.
this_lab_cex	Numeric scalar; scaling for plot labels. Default is the value of this_plot_cex.
this_main_cex	Numeric scalar; scaling for main plot title. Default is the value of this_plot_cex + 0.1.
this_axis_cex	Numeric scalar; scaling for plot axis. Default is the value of this_plot_cex.
this_mar	Numeric vector; set margins for the plot. Default is c(3.1, 2.1, 3.1, 1.1).
this_oma	Numeric vector; set margins for the outer plot. Default is c(0, 0, 2.0, 0).
main_title_line	Integer scalar; position of main title of plot. Default is 1.25.
this_reset	Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generate plots of seasonally adjusted series for every seas element in the list. Plots will be laid out in this_row rows and this_col columns. No values are returned.

Examples

```
EM_individual_seas <-
seasonal::seas(
  x11 = "", transform.function = "log",
  check.print = c("none", "+acf", "+acfplot", "+normalitytest"),
  regression.aictest = NULL,
  outlier.types = "all",
  arima.model = "(0 1 1)(0 1 1)",
  list = list(
    list(x = blsplotR::employment_list$n2000013),
    list(x = blsplotR::employment_list$n2000014),
    list(x = blsplotR::employment_list$n2000025),
    list(x = blsplotR::employment_list$n2000026)
  )
)

# Use Filter function to grab seas objects
EM_individual_seas_only <-
  Filter(function(x) inherits(x, "seas"), EM_individual_seas)

names(EM_individual_seas_only) <- names(blsplotR::employment_list)
blsplotR::plot_multiple(EM_individual_seas_only,
  first_series = 1, last_series = 4, this_row = 2, this_col = 2,
  plot_trend = FALSE, col_vec = c("grey", "steelblue"),
  seas_obj_names = names(EM_individual_seas_only),
  outer_title = 'Series (grey), SA (blue) plot',
  group_title='U. S. Employment Series',
  do_grid = TRUE, draw_recess = TRUE, recess_sub = FALSE)
```

plot_ratio

Ratio plot

Description

Generates a high-definition plot around a reference line other than zero.

Usage

```
plot_ratio(
  ratio_series = NULL,
  ratio_range = range(ratio_series),
  main_title = NULL,
  main_title_line = 2,
  main_title_cex = NULL,
  ratio_mean = 1,
  ratio_color = NULL,
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
```

```

    this_axis_cex = NULL,
    this_mar = c(4, 4, 4, 0.5),
    this_reset = TRUE,
    plot_series = TRUE
  )

```

Arguments

ratio_series	Time series of ratios/factors for which you want to generate a high definition plot
ratio_range	Range of values you wish the plot to be plotted over. Default is range of the series.
main_title	Title for the plot. Default is character string 'Ratio Plot'.
main_title_line	Integer scalar; position of main title of plot. Default is 2.
main_title_cex	Numeric scalar; scaling for main title of plot. Default is this_plot_cex + 0.1.
ratio_mean	Assumed mean value for the ratio. Default is 1.0
ratio_color	Color used for lines in ratio plot. Default is 'black'.
this_plot_cex	Numeric scalar; scaling for the plot itself. Default is 0.8.
this_lab_cex	Numeric scalar; scaling for plot labels. Default is the value of this_plot_cex.
this_axis_cex	Numeric scalar; scaling for plot axis. Default is the value of this_plot_cex.
this_mar	Numeric vector; set margins for the plot. Default is c(4,4,4,0.5).
this_reset	Logical scalar; if TRUE, the values of par are reset. Default is TRUE.
plot_series	Logical scalar. if TRUE, function will generate a plot of the series first of type='n'. If FALSE, the ratio will be plotted on the current defined plot. Default is TRUE.

Value

Generates a high definition plot of rations centered on one, by default.

Examples

```

air_seas <- seasonal::seas(AirPassengers, transform.function= 'log', arima.model = '(0 1 1)(0 1 1)')
air_sf <- seasonal::series(air_seas, 's10')
plot_ratio(air_sf, main_title = 'SEATS seasonal for Airline Passenger', ratio_color = 'darkblue')

```

plot_resid

Residual plot

Description

Generates a plot of the regARIMA residuals with diagnostic information

Usage

```
plot_resid(
  this_seas = NULL,
  main_title = "ARIMA Residuals",
  main_title_line = 3,
  main_title_cex = NULL,
  series_name = NULL,
  do_grid = TRUE,
  draw_recess = FALSE,
  recess_start = NULL,
  recess_col = NULL,
  recess_sub = TRUE,
  use_ratio = FALSE,
  this_col = "green",
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
  this_axis_cex = NULL,
  this_sub_cex = 0.5,
  this_mar = c(5.1, 4, 5.1, 0.5),
  this_reset = TRUE
)
```

Arguments

<code>this_seas</code>	seas object generated from a call of seas on a single time series
<code>main_title</code>	Character string; main title of plot. Default is 'ARIMA Residuals'.
<code>main_title_line</code>	Integer scalar; position of main title of plot. Default is 3.
<code>main_title_cex</code>	Numeric scalar; scaling for main title of plot. Default is <code>this_plot_cex + 0.1</code> .
<code>series_name</code>	Character scalar; name of the time series used in m.
<code>do_grid</code>	Logical scalar; indicates if certain plots will have grid lines. Default is grid lines plotted.
<code>draw_recess</code>	Logical scalar; indicates if certain plots will have shaded areas for NBER recession dates. Default is recession shading not plotted.
<code>recess_start</code>	numeric matrix; Rows of dates for additional recession starting and ending dates. Default is not to add recession dates.
<code>recess_col</code>	Character string; color used for shading of recession region. Default is 'lightgrey'.
<code>recess_sub</code>	Logical scalar; indicates if x-axis label for recession is produced for this plot. Default is x-axis label is produced
<code>use_ratio</code>	Logical scalar; indicates if plots of seasonal factors, irregular, and residuals are done as ratio plots. Default has these plots as time series line plots.
<code>this_col</code>	Character string; color used for residuals. Default is 'green'.
<code>this_plot_cex</code>	Numeric scalar; scaling for the plot itself. Default is 0.8.
<code>this_lab_cex</code>	Numeric scalar; scaling for plot labels. Default is the value of <code>this_plot_cex</code> .
<code>this_axis_cex</code>	Numeric scalar; scaling for plot axis. Default is the value of <code>this_plot_cex</code> .
<code>this_sub_cex</code>	Numeric scalar; scaling for plot labels. Default is the value of 0.5.
<code>this_mar</code>	Numeric vector; set margins for the plot. Default is <code>c(5.1, 4, 5.1, 0.5)</code> .
<code>this_reset</code>	Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generates a plot of the regARIMA residuals with diagnostic information in the sub-headers.

Examples

```
air_seas <- seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)')
plot_resid(air_seas, main_title = 'ARIMA Residuals for Airline Passengers', use_ratio = TRUE,
           this_col='darkblue')
```

plot_sa_list	<i>Plots of seasonally adjusted series for seasonal objects in a list.</i>
--------------	--

Description

Generate plots of seasonally adjusted series for every element in the list. The series name is used for the title.

Usage

```
plot_sa_list(
  seas_obj_list = NULL,
  this_row = 2,
  this_col = 2,
  seas_obj_names = NULL,
  pdf_file = NULL,
  this_dir = NULL,
  plot_trend = TRUE,
  group_title = NULL,
  col_vector = c("grey", "blue", "darkgreen"),
  outer_title = NULL,
  plot_start = NULL,
  plot_end = NULL,
  do_grid = FALSE,
  draw_recess = FALSE,
  recess_start = NULL,
  recess_col = NULL,
  recess_sub = TRUE,
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
  this_main_cex = NULL,
  this_axis_cex = NULL,
  this_mar = c(3.1, 2.1, 3.1, 1.1),
  this_oma = c(0, 0, 2, 0),
  main_title_line = 1.25,
  this_reset = TRUE
)
```

Arguments

seas_obj_list	List object of seas arguments generated by seas() of the seasonal package.
this_row	Number of rows in multi-frame plot
this_col	Number of columns in multi-frame plot
seas_obj_names	Vector of character strings; the names of the series being plotted. By Default, the names will be taken directly from the seasonal object list.
pdf_file	File name of PDF file, if specified.
this_dir	Character scalar; directory where the graphics file generated. Default is the current working directory.
plot_trend	Logical scalar; if TRUE, trend is included in plot, FALSE trend is not included.
group_title	Character string with a group title of series, if specified.
col_vector	Character array of length 3; color vector for lines in the plots. Default is c('grey', 'blue', 'green') for original series, SA, Trend
outer_title	Character string; outer title of set of plots, if specified.
plot_start	Integer array of length 2; start date for series to be plotted. If not specified, starting date of series used.
plot_end	Integer array of length 2; end date for series to be plotted. If not specified, ending date of series used.
do_grid	Logical scalar; if TRUE, grid lines are included in plot, FALSE grid lines are not included.
draw_recess	Logical scalar; if TRUE, recession periods are included in plot, FALSE recession periods are not included.
recess_start	numeric matrix; Rows of dates for additional recession starting and ending dates. Default is not to add recession dates.
recess_col	Character string; color used for shading of recession region. Default is 'lightgrey'.
recess_sub	Logical scalar; indicates if x-axis label for recession is produced for this plot. Default is x-axis label is produced
this_plot_cex	Numeric scalar; scaling for the plot itself. Default is 0.8.
this_lab_cex	Numeric scalar; scaling for plot labels. Default is the value of this_plot_cex.
this_main_cex	Numeric scalar; scaling for main plot title. Default is the value of this_plot_cex + 0.1.
this_axis_cex	Numeric scalar; scaling for plot axis. Default is the value of this_plot_cex.
this_mar	Numeric vector; set margins for the plot. Default is c(3.1, 2.1, 3.1, 1.1).
this_oma	Numeric vector; set margins for the outer plot. Default is c(0, 0, 2.0, 0).
main_title_line	Integer scalar; position of main title of plot. Default is 2.25.
this_reset	Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generate plots of seasonally adjusted series for every seas element in the list. Plots will be laid out in this_row rows and this_col columns. No values are returned.

Examples

```
EM_individual_seas <-
  seasonal::seas(
    x11 = "", transform.function = "log",
    check.print = c("none", "+acf", "+acfpplot", "+normalitytest"),
    regression.aictest = NULL,
    outlier.types = "all",
    arima.model = "(0 1 1)(0 1 1)",
    list = list(
      list(x = blsplotR::employment_list$n2000013),
      list(x = blsplotR::employment_list$n2000014),
      list(x = blsplotR::employment_list$n2000025),
      list(x = blsplotR::employment_list$n2000026)
    )
  )

# Use Filter function to grab seas objects
EM_individual_seas_only <-
  Filter(function(x) inherits(x, "seas"), EM_individual_seas)

EM_names <- c("Male 16-19", "Female 16-19", "Male 20+", "Female 20+")
names(EM_individual_seas_only) <- names(blsplotR::employment_list)

blsplotR::plot_sa_list(EM_individual_seas_only,
  this_row = 4, this_col = 1,
  plot_trend=TRUE, seas_obj_names = EM_names,
  group_title='US Employment',
  col_vec = c("grey", "steelblue", "forestgreen"),
  plot_start=c(2015,1), this_main_cex = 0.75, this_axis_cex = 0.5,
  this_mar = c(2.5, 2.0, 2.0, 0.25), main_title_line = 0.5)
```

plot_sa_list_split	<i>Plots of seasonally adjusted series for seasonal objects in a list, split into individual graphics files.</i>
--------------------	--

Description

Generates a page of plots for a time series, seasonal adjustment of the time series, and trend component. Plotting the trend is optional.

Usage

```
plot_sa_list_split(
  seas_obj_list = NULL,
  this_row = 2,
  this_col = 2,
  seas_obj_names = NULL,
  file_name_base = "SAPlot",
  this_dir = NULL,
  plot_type = "png",
  plot_trend = TRUE,
  group_title = NULL,
  outer_title = NULL,
```

```

plot_start = NULL,
plot_end = NULL,
do_grid = FALSE,
do_recess = FALSE,
col_vector = c("grey", "blue", "darkgreen"),
this_plot_cex = 0.8,
this_lab_cex = NULL,
this_main_cex = NULL,
this_axis_cex = NULL,
this_mar = c(3.1, 2.1, 3.1, 1.1),
this_oma = c(0, 0, 2, 0),
main_title_line = 1.25,
this_reset = TRUE
)

```

Arguments

seas_obj_list	List of seas arguments generated by seas() of the seasonal package.
this_row	Number of rows in multi-frame plot
this_col	Number of columns in multi-frame plot
seas_obj_names	Vector of character strings; the names of the series being plotted. By Default, the names will be taken directly from the seasonal object list.
file_name_base	Character string that serves as the base for graphics file name, Default is 'SAPlot'
this_dir	Character scalar; directory where the graphics file generated. Default is the current working directory.
plot_type	Character string; type of graphics file - possible entries include 'pdf', 'png', 'eps'. Default is 'png'
plot_trend	Logical scalar; if TRUE, trend is included in plot, FALSE trend is not included.
group_title	Character string with a group title of series, if specified.
outer_title	Character string, outer title of set of plots. If not specified, outer title is 'Series (grey), SA (blue), Trend (green) plot' or 'Series (grey), SA (blue)' if trend isn't specified.
plot_start	Start date for series to be plotted, expressed as a vector of length two. If not specified, starting date of series used.
plot_end	End date for series to be plotted, expressed as a vector of length two. If not specified, ending date of series used.
do_grid	Logical scalar; indicates if certain plots will have grid lines. Default is no grid lines.
do_recess	Logical scalar; indicates if certain plots will have shaded areas for NBER recession dates. Default is no recession shading.
col_vector	Character vector for the color of lines in the plots. Default is c('grey', 'blue', 'green') for original series, SA, Trend
this_plot_cex	Numeric scalar; scaling for the plot itself. Default is 0.8.
this_lab_cex	Numeric scalar; scaling for plot labels. Default is the value of this_plot_cex.
this_main_cex	Numeric scalar; scaling for main plot title. Default is the value of this_plot_cex + 0.1.
this_axis_cex	Numeric scalar; scaling for plot axis. Default is the value of this_plot_cex.

this_mar	Numeric vector; set margins for the plot. Default is <code>c(3.1, 2.1, 3.1, 1.1)</code> .
this_oma	Numeric vector; set margins for the outer plot. Default is <code>c(0, 0, 2.0, 0)</code> .
main_title_line	Integer scalar; position of main title of plot. Default is 2.25.
this_reset	Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generate plots of seasonally adjusted series for every seas element in the list. Plots will be laid out in this_row rows and this_col columns, into individual pages. No values are returned.

Examples

```
EM_seas <-
  seasonal::seas(employment_list, slidingspans = "",
    transform.function = "log",
    arima.model = "(0 1 1)(0 1 1)",
    forecast.maxlead = 36,
    check.print = c( "pacf", "pacfplot" ))
## Not run: plot_sa_list_split(EM_seas, this_row = 2, this_col = 2,
  plot_trend = TRUE, this_dir = 'X:/seasonalAdj/graphs/',
  seas_obj_names = c("Male 16-19", "Female 16-19", "Male 20+", "Female 20+"),
  file_name_base='EM_sa_trend', plot_type = 'png',
  group_title='US Employment',
  plot_start=c(2010,1))
## End(Not run)
```

plot_series	<i>Plot individual series.</i>
-------------	--------------------------------

Description

Generate plot of user-specified series.

Usage

```
plot_series(
  this_series = NULL,
  main_title = NULL,
  main_title_line = 2.75,
  main_title_cex = 1.25,
  y_label = NULL,
  y_limit = NULL,
  start_plot = NULL,
  do_grid = FALSE,
  draw_recess = FALSE,
  recess_start = NULL,
  recess_col = NULL,
  recess_sub = TRUE,
  this_trans = TRUE,
  use_ratio = FALSE,
  this_col = "grey",
```

```

    this_line_type = 1,
    this_point_type = NULL,
    add_legend = FALSE,
    this_legend_position = "topleft",
    this_legend_title = "Series",
    this_legend_inset = 0,
    this_legend_entry = "Srs1",
    this_legend_cex = 0.8,
    this_legend_col = "grey",
    this_legend_lty = 1,
    this_plot_cex = 0.8,
    this_lab_cex = NULL,
    this_axis_cex = NULL,
    this_mar = c(4, 4, 4, 0.5),
    this_reset = TRUE
  )

```

Arguments

<code>this_series</code>	Numeric vector; time series object to be plotted.
<code>main_title</code>	Character string; main title of plot. Default is no title.
<code>main_title_line</code>	Integer scalar; position of main title of plot. Default is 2.75.
<code>main_title_cex</code>	Numeric scalar; scaling for main title of plot. Default is 1.25.
<code>y_label</code>	Character string; y-axis label for plot, if specified.
<code>y_limit</code>	Numeric vector of length 2; Range of values you wish the plot to be plotted over. Default is range of the seasonal factors.
<code>start_plot</code>	Integer vector of length 2; Starting date for plot. Default is starting date for the time series.
<code>do_grid</code>	Logical scalar; indicates if certain plots will have grid lines. Default is no grid lines.
<code>draw_recess</code>	Logical scalar; indicates if certain plots will have shaded areas for NBER recession dates. Default is no recession shading.
<code>recess_start</code>	numeric matrix; Rows of dates for additional recession starting and ending dates. Default is not to add recession dates.
<code>recess_col</code>	Character string; color used for shading of recession region. Default is 'lightgrey'.
<code>recess_sub</code>	Logical scalar; indicates if x-axis label for recession is produced for this plot. Default is x-axis label is produced
<code>this_trans</code>	Logical scalar; indicates if the adjustment was done with a log transform. Default is TRUE.
<code>use_ratio</code>	Logical scalar; indicates if plots of seasonal factors, irregular, and residuals are done as ratio plots. Default has these plots as time series line plots.
<code>this_col</code>	Character array of length 6; color used for series in the order specified by the user. Default is c('grey', 'blue', 'green', 'brown', 'red', 'yellow').
<code>this_line_type</code>	Integer array; line type used for series
<code>this_point_type</code>	Integer array; point type used for series. Default is no points plotted.

add_legend	Logical scalar; indicates if legend is produced for this plot. Default is legend not produced
this_legend_position	Character string; indicates position of legend. Default is 'topleft'.
this_legend_title	Character string; indicates title of legend. Default is 'Series'.
this_legend_inset	Integer scalar; indicates inset for legend. Default is 0.
this_legend_entry	Character array; entries for the legend. Default is 'Srs1'.
this_legend_cex	Numeric scalar; scaling for legend. Default is 0.8.
this_legend_col	Character string; color of lines in the legend. Default is 'grey'.
this_legend_lty	Numeric scalar; color of lines in the legend. Default is 1.
this_plot_cex	Numeric scalar; scaling for the plot itself. Default is 0.8.
this_lab_cex	Numeric scalar; scaling for plot labels. Default is the value of this_plot_cex.
this_axis_cex	Numeric scalar; scaling for plot axis. Default is the value of this_plot_cex.
this_mar	Numeric vector; set margins for the plot. Default is c(4,4,4,0,5).
this_reset	Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generate plot of user-specified series. Can be first in a series of plots, with other lines or points added after calling this routine. If series not specified, print out error message and return NULL.

Examples

```
air_seas <- seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)',
                          outlier.types = "all", x11 = "",
                          forecast.maxlead = 36)
plot_series(AirPassengers, y_label = 'Air Passengers', do_grid = TRUE,
            draw_recess = TRUE, this_col = 'black',
            start_plot = c(1958,1), this_point_type = 1,
            main_title = "X-11 Seasonal Adjustment for Airline Passengers",
            main_title_line = 1.5, main_title_cex = 0.9,
            add_legend = TRUE,
            this_legend_position = "topleft",
            this_legend_title = "Air Passengers", this_legend_inset = 0,
            this_legend_entry = c("Series", "SA", "Trend"),
            this_legend_col = c("black", "steelblue", "darkgreen"),
            this_legend_lty = 1:3,
            this_reset = FALSE)
lines(window(seasonal::final(air_seas), start=c(1958,1)), col = "blue", lty = 2)
lines(window(seasonal::trend(air_seas), start=c(1958,1)), col = "darkgreen", lty = 3)
reset_par()
```

plot_sf	<i>Seasonal factor (and the SI-ratios) plot grouped by month/quarter</i>
---------	--

Description

Generates a special plot of the seasonal factors (and the SI-ratios) grouped by month/quarter

Usage

```
plot_sf(
  this_seas = NULL,
  this_table = NULL,
  add_si = FALSE,
  main_title = "Seasonal Sub-Plots",
  y_label = NULL,
  y_limit = NULL,
  this_xlab = " ",
  this_col = c("darkgreen", "darkblue", "darkgrey"),
  add_mean_line = TRUE,
  add_legend = FALSE,
  this_legend_position = "topleft",
  this_legend_title = "SF Plot",
  this_legend_inset = 0,
  this_legend_cex = 0.8,
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
  this_main_cex = NULL,
  this_axis_cex = NULL,
  this_mar = c(4.1, 2.1, 4.1, 0.5),
  main_title_line = 2.25,
  this_reset = TRUE
)
```

Arguments

this_seas	seas object generated from a call of seas on a single time series
this_table	Table from the X-13ARIMA-SEATS output (such as e18 or s18) used in the plot; if NULL, the seasonal factor (either D11 for X-11 or S11 for SEATS) will be used.
add_si	Logical scalar; indicates if seasonal factor plots will include SI ratios for X-11 seasonal adjustments. Default is not including SI ratios.
main_title	Character string; main title of plot. Default is 'Seasonal Sub-Plots'.
y_label	Character string; y-axis label for plot, if specified.
y_limit	Numeric vector of length 2; Range of values you wish the plot to be plotted over. Default is range of the seasonal factors.
this_xlab	Character string; label for x-axis of plot. Default is a blank x-axis.
this_col	Character array of length 3; color used for seasonal factors, SI-ratios, and seasonal mean. Default is c("darkgreen", "darkblue", "darkgrey").

add_mean_line	Logical scalar; indicates if seasonal factor plots will include lines for seasonal means. Default includes lines for seasonal means.
add_legend	Logical scalar; indicates if legend is produced for this plot. Default is legend not produced
this_legend_position	Character string; indicates position of legend. Default is 'topleft'.
this_legend_title	Character string; indicates title of legend. Default is 'Series'.
this_legend_inset	Integer scalar; indicates inset for legend. Default is 0.
this_legend_cex	Numeric scalar; scaling for legend. Default is 0.8.
this_plot_cex	Numeric scalar; scaling for the plot itself. Default is 0.8.
this_lab_cex	Numeric scalar; scaling for plot labels. Default is the value of this_plot_cex.
this_main_cex	Numeric scalar; scaling for main plot title. Default is the value of this_plot_cex + 0.1.
this_axis_cex	Numeric scalar; scaling for plot axis. Default is the value of this_plot_cex.
this_mar	Numeric vector; set margins for the plot. Default is c(4.1, 2.1, 4.1, 0.5).
main_title_line	Integer scalar; position of main title of plot. Default is 2.25.
this_reset	Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generates a special plot of the seasonal factors (and the SI-ratios) grouped by month/quarter

Examples

```
air_seas <- seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11='')
plot_sf(air_seas, add_si = TRUE, main_title = 'Air Passengers Seasonal Sub-Plots',
        this_col = c('darkgreen', 'darkblue', 'grey'), add_legend = TRUE)
```

plot_sf_mean	<i>Seasonal factor mean plot</i>
--------------	----------------------------------

Description

Generates a plot of the means of the seasonal factors

Usage

```
plot_sf_mean(
  this_sf = NULL,
  this_period = NULL,
  this_col = "green",
  y_limit = range(this_sf),
  this_freq,
  this_trans = TRUE,
```

```

    this_title = "Mean of Seasonal Factors",
    forecast = 0,
    this_type = "Seasonal",
    add_line = FALSE,
    add_legend = FALSE,
    this_legend_position = "topleft",
    this_legend_title = "SF Means",
    this_legend_inset = 0,
    this_legend_entry = "Srs1",
    this_legend_col = "green",
    this_legend_lty = 1,
    this_legend_cex = 0.8,
    this_reset = FALSE
)

```

Arguments

<code>this_sf</code>	time series object of the seasonal factors from a seasonal adjustment
<code>this_period</code>	Integer vector; indicates the number of the month or quarter for each observation.
<code>this_col</code>	Character scalar; color used for factor plots. Default is green.
<code>y_limit</code>	Numeric vector of length 2; Range of values you wish the plot to be plotted over. Default is range of the seasonal factors.
<code>this_freq</code>	integer scalar; time series frequency.
<code>this_trans</code>	Logical scalar; indicates if the adjustment was done with a log transform. Default is TRUE.
<code>this_title</code>	Character string; main title of plot. Default is 'Mean of Seasonal Factors'.
<code>forecast</code>	Integer scalar; Number of forecasts appended to the seasonal factors. Default is 0.
<code>this_type</code>	Character string; type of factors plotted. Default is 'seasonal'.
<code>add_line</code>	Logical scalar; indicates if this line is being added to an existing plot. Default is FALSE.
<code>add_legend</code>	Logical scalar; indicates if legend is produced for this plot. Default is legend not produced
<code>this_legend_position</code>	Character string; indicates position of legend. Default is 'topleft'.
<code>this_legend_title</code>	Character string; indicates title of legend. Default is 'Series'.
<code>this_legend_inset</code>	Integer scalar; indicates inset for legend. Default is 0.
<code>this_legend_entry</code>	Character array; entries for the legend. Default is 'Srs1'
<code>this_legend_col</code>	Character array; line colors for legend. Default is 'blue'.
<code>this_legend_lty</code>	Integer array; line types for legend. Default is 1.
<code>this_legend_cex</code>	Numeric scalar; scaling for legend. Default is 0.8.
<code>this_reset</code>	Logical scalar; if TRUE, the values of par are reset. Default is FALSE.

Value

Generate plot of the means of seasonal factors by period, or add to existing plot. If seasonal factors not specified, print out error message and return NULL.

Examples

```
EM_individual_seas <-
seasonal::seas(
  x11 = "", transform.function = "log",
  check.print = c("none", "+acf", "+acfplot", "+normalitytest"),
  regression.aictest = NULL,
  outlier.types = "all",
  arima.model = "(0 1 1)(0 1 1)",
  list = list(
    list(x = employment_list$n2000013),
    list(x = employment_list$n2000014),
    list(x = employment_list$n2000025),
    list(x = employment_list$n2000026)
  )
)

# Use Filter function to grab seas objects
EM_individual_seas_only <-
  Filter(function(x) inherits(x, "seas"), EM_individual_seas)

names(EM_individual_seas_only) <- names(employment_list)

EM_Comp_Sf <-
  cbind(seasonal::series(EM_individual_seas_only$n2000013, "d10"),
        seasonal::series(EM_individual_seas_only$n2000014, "d10"),
        seasonal::series(EM_individual_seas_only$n2000025, "d10"),
        seasonal::series(EM_individual_seas_only$n2000026, "d10"))
this_sf_limit <- range(EM_Comp_Sf)

blsplotR::plot_sf_mean(EM_Comp_Sf[,1], cycle(EM_Comp_Sf[,1]),
  this_col = 'steelblue',
  y_limit = this_sf_limit,
  this_freq = 12,
  forecast = 0,
  this_title = 'US Employment Seasonal Means',
  add_legend = TRUE,
  this_legend_position = "topleft",
  this_legend_title = "SF Means",
  this_legend_inset = 0,
  this_legend_entry = c("M 16-19", "F 16-19", "M 20+", "F 20+"),
  this_legend_col = c("steelblue", "red", "darkgreen", "purple"),
  this_legend_lty = rep(1,4),
  this_legend_cex = 0.6)

blsplotR::plot_sf_mean(EM_Comp_Sf[,2], cycle(EM_Comp_Sf[,2]),
  this_col = 'red',
  this_freq = 12,
  forecast = 0,
  add_line = TRUE)

blsplotR::plot_sf_mean(EM_Comp_Sf[,3], cycle(EM_Comp_Sf[,3]),
```

```

    this_col = 'darkgreen',
    this_freq = 12,
    forecast = 0,
    add_line = TRUE)

blsplotR::plot_sf_mean(EM_Comp_Sf[,4], cycle(EM_Comp_Sf[,4]),
  this_col = 'purple',
  this_freq = 12,
  forecast = 0,
  add_line = TRUE,
  this_reset = TRUE)

```

plot_sf_series	<i>Seasonal factor plot (for up to two sets of factors) grouped by month/quarter</i>
----------------	--

Description

Generates a special plot of the seasonal factors grouped by month/quarter. This can be done for up to two sets of seasonal factors.

Usage

```

plot_sf_series(
  this_sf = NULL,
  second_sf = NULL,
  y_limit = NULL,
  this_trans = TRUE,
  main_title = "Seasonal Sub-Plots",
  this_xlab = " ",
  this_col = c("darkgreen", "darkblue", "darkgrey"),
  first_year = NULL,
  add_mean_line = TRUE,
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
  this_main_cex = NULL,
  this_axis_cex = NULL,
  this_mar = c(4, 4, 4, 0.5),
  add_legend = FALSE,
  this_legend_position = "topleft",
  this_legend_text = c("SF", "SF Mean"),
  this_legend_title = "SF Plot",
  this_legend_inset = 0,
  this_legend_color = this_col,
  this_legend_cex = 0.8,
  main_title_line = 2.25,
  this_reset = TRUE
)

```

Arguments

<code>this_sf</code>	array of seasonal factors stored as a time series
<code>second_sf</code>	array of a second set of seasonal factors stored as a time series. If NULL, a second set of factors is not plotted.
<code>y_limit</code>	Numeric vector of length 2; Range of values you wish the plot to be plotted over. Default is range of the seasonal factors.
<code>this_trans</code>	Logical scalar; indicates if the adjustment was done with a log transform. Default is TRUE.
<code>main_title</code>	Character string; main title of plot. Default is 'Seasonal Sub-Plots'.
<code>this_xlab</code>	Character string; label for x-axis of plot. Default is a blank x-axis.
<code>this_col</code>	Character array of length 4; color used for seasonal factors, second set of seasonal factors, seasonal mean, and second seasonal mean. Default is <code>c("darkgreen", "darkblue", "darkgrey")</code> .
<code>first_year</code>	Integer scalar; First year used in plot. Default is start of the series.
<code>add_mean_line</code>	Logical scalar; indicates if seasonal factor plots will include lines for seasonal means. Default includes lines for seasonal means.
<code>this_plot_cex</code>	Numeric scalar; scaling for the plot itself. Default is 0.8.
<code>this_lab_cex</code>	Numeric scalar; scaling for plot labels. Default is the value of <code>this_plot_cex</code> .
<code>this_main_cex</code>	Numeric scalar; scaling for main plot title. Default is the value of <code>this_plot_cex</code> .
<code>this_axis_cex</code>	Numeric scalar; scaling for plot axis. Default is the value of <code>this_plot_cex</code> .
<code>this_mar</code>	Numeric vector; set margins for the plot. Default is <code>c(4,4,4,0,5)</code> .
<code>add_legend</code>	Logical scalar; indicates if legend is produced for this plot. Default is legend not produced
<code>this_legend_position</code>	Character string; indicates position of legend. Default is 'topleft'.
<code>this_legend_text</code>	Array of character strings; indicates text for each seasonal factor in plot. Default is <code>c("SF", "SF Mean")</code> .
<code>this_legend_title</code>	Character string; indicates title of legend. Default is 'Series'.
<code>this_legend_inset</code>	Integer scalar; indicates inset for legend. Default is 0.
<code>this_legend_color</code>	Array of character strings; indicates color for each seasonal factor in plot. Default is same as <code>this_col</code>
<code>this_legend_cex</code>	Numeric scalar; scaling for legend. Default is 0.8.
<code>main_title_line</code>	Integer scalar; position of main title of plot. Default is 2.25.
<code>this_reset</code>	Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generates a special plot of the seasonal factors (and the SI-ratios) grouped by month/quarter

Examples

```

air_seas      <- seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11='')
air_seats_seas <- seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)')
air_sf        <- seasonal::series(air_seas, "d10")
air_seats_sf  <- seasonal::series(air_seats_seas, "s10")
sf_range <- range(air_sf, air_seats_sf)
blsplotR::plot_sf_series(air_sf, air_seats_sf, y_limit = sf_range,
  add_mean_line = TRUE, add_legend = TRUE,
  main_title = 'Air Passengers Seasonal Sub-Plots',
  this_col = c('darkgreen', 'darkblue', 'lightgreen', 'lightblue'),
  this_legend_text = c("sf(x11)", "sf(seats)", 'mean(x11)', 'mean(seats)'),
  this_legend_color = c('darkgreen', 'darkblue', 'lightgreen', 'lightblue'),
  main_title_line = 1.25, this_main_cex = 0.95,
  this_plot_cex = 0.6, this_axis_cex = 0.75,
  this_mar = c(3,3,3,0.5), this_legend_cex = 0.75)
#' @import graphics

```

plot_single_cell	<i>Single time series plot.</i>
------------------	---------------------------------

Description

Generates a single plot of a time series, seasonal adjustment of the time series, and trend component. Plotting the trend is optional. The series name is used for the title.

Usage

```

plot_single_cell(
  this_series = NULL,
  this_sadj = NULL,
  this_trend = NULL,
  this_name = NULL,
  col_vector = c("grey", "blue", "darkgreen"),
  main_title_line = 1,
  this_main_cex = 0.8
)

```

Arguments

this_series	Original time series
this_sadj	Seasonal adjustment of this_series.
this_trend	Trend component estimated from this_series. Default is to not print the trend component.
this_name	Name of the original series. If specified, this is used as the title of the plot. Default is to generate the title based on what components are plotted.
col_vector	Character vector of length 3; colors used for the lines in the plot. First color is for the original series, second is for the SA series, third is for the trend. Default is col_vector=c('grey', 'blue', 'darkgreen').
main_title_line	Integer scalar; position of main title of plot. Default is 2.25.
this_main_cex	Numeric scalar; scaling for main plot title. Default is the value of this_plot_cex + 0.1.

Value

Produces plot of single plot of a time series, seasonal adjustment of the time series, and trend component. No values are returned.

Examples

```
air_seas <- seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11='')
air_final <- seasonal::final(air_seas)
air_trend <- seasonal::trend(air_seas)
plot_single_cell(AirPassengers, air_final, air_trend, 'Air Passengers')
```

plot_table	<i>Plot table from X-13ARIMA-SEATS seasonal adjustment.</i>
------------	---

Description

Generate plot of user-specified series.

Usage

```
plot_table(
  this_seas = NULL,
  this_table = NULL,
  main_title = NULL,
  main_title_line = 2,
  main_title_cex = NULL,
  y_label = NULL,
  y_limit = NULL,
  start_plot = NULL,
  do_grid = FALSE,
  draw_recess = FALSE,
  recess_start = NULL,
  recess_col = NULL,
  recess_sub = TRUE,
  add_otl = FALSE,
  use_ratio = FALSE,
  add_sub_title = FALSE,
  sub_title_line = 1,
  sub_title_cex = NULL,
  this_line_type = NULL,
  this_col = c("grey", "blue", "green", "brown", "red", "yellow"),
  otl_col = c("red", "blue", "green", "brown", "grey", "yellow"),
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
  this_axis_cex = NULL,
  this_mar = c(4, 4, 4, 0.5),
  this_reset = FALSE
)
```

Arguments

<code>this_seas</code>	seas object generated from a call of seas on a single time series
<code>this_table</code>	Character string; X-13ARIMA-SEATS table name or abbreviation. If not a valid table name, the function will print an error message and return a NULL.
<code>main_title</code>	Character string; main title of plot. Default is 'Cumulative periodogram'.
<code>main_title_line</code>	Integer scalar; position of main title of plot. Default is 2.75.
<code>main_title_cex</code>	Numeric scalar; scaling for main title of plot. Default is <code>this_plot_cex + 0.1</code> .
<code>y_label</code>	Character string; y-axis label for plot, if specified.
<code>y_limit</code>	Numeric vector of length 2; Range of values you wish the plot to be plotted over. Default is range of the series specified.
<code>start_plot</code>	Integer vector of length 2; Starting date for plot. Default is starting date for the time series.
<code>do_grid</code>	Logical scalar; indicates if certain plots will have grid lines. Default is no grid lines.
<code>draw_recess</code>	Logical scalar; indicates if certain plots will have shaded areas for NBER recession dates. Default is no recession shading.
<code>recess_start</code>	numeric matrix; Rows of dates for additional recession starting and ending dates. Default is not to add recession dates.
<code>recess_col</code>	Character string; color used for shading of recession region. Default is 'lightgrey'.
<code>recess_sub</code>	Logical scalar; indicates if x-axis label for recession is produced for this plot. Default is x-axis label is produced
<code>add_otl</code>	Logical scalar; indicates if lines for identified outliers are included in series plots. Default is not including lines for identified outliers.
<code>use_ratio</code>	Logical scalar; indicates if plots of seasonal factors, irregular, and residuals are done as ratio plots. Default has these plots as time series line plots.
<code>add_sub_title</code>	Logical scalar; indicates if plots will include subtitles denoting what series are plotted. Default is not including subheaders.
<code>sub_title_line</code>	Integer scalar; position of main title of plot. Default is 1.
<code>sub_title_cex</code>	Numeric scalar; scaling for main title of plot. Default is <code>this_plot_cex - 0.1</code> .
<code>this_line_type</code>	Integer vector; indicates line type of each plot produced. Default is <code>1:length(this_table)</code>
<code>this_col</code>	Character array of length 6; color used for series in the order specified by the user. Default is <code>c('grey', 'blue', 'green', 'brown', 'red', 'yellow')</code> .
<code>otl_col</code>	Character array of length 6; color used for different outliers, with the order being 'ao', 'ls', 'tc', 'so', 'rp', 'tls'. Default is <code>c('red', 'blue', 'green', 'brown', 'grey', 'yellow')</code> .
<code>this_plot_cex</code>	Numeric scalar; scaling for the plot itself. Default is 0.8.
<code>this_lab_cex</code>	Numeric scalar; scaling for plot labels. Default is the value of <code>this_plot_cex</code> .
<code>this_axis_cex</code>	Numeric scalar; scaling for plot axis. Default is the value of <code>this_plot_cex</code> .
<code>this_mar</code>	Numeric vector; set margins for the plot. Default is <code>c(4,4,4,0,5)</code> .
<code>this_reset</code>	Logical scalar; if TRUE, the values of par are reset. Default is FALSE.

Value

Generate plot of user-specified series. Can be more than one series. If series not specified, print out error message and return NULL.

Examples

```
air_seas <-
  seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11='',
    series.save = 'b1')
blsplotR::plot_table(air_seas, c('a1', 'b1', 'd11'),
  y_label = 'AirPassengers',
  main_title = 'Airline Passengers',
  main_title_cex = 1.0, add_sub_title = TRUE,
  do_grid = TRUE, draw_recess = TRUE, use_ratio = TRUE, add_otl = TRUE,
  this_col = c('grey', 'darkgreen', 'darkblue'))
```

plot_year_over_year	<i>Year over year plot of individual series.</i>
---------------------	--

Description

Generate plot of user-specified series with each year as a separate line.

Usage

```
plot_year_over_year(
  this_series = NULL,
  main_title = NULL,
  main_title_line = 1.75,
  main_title_cex = NULL,
  this_col = NULL,
  start_plot = NULL,
  this_legend = TRUE,
  this_legend_cex = 0.75,
  this_right_mar = 5.25,
  this_legend_inset = -0.15,
  this_plot_cex = 0.8,
  this_lab_cex = NULL,
  this_axis_cex = NULL,
  this_mar = c(4, 4, 4, 0.5),
  this_reset = TRUE
)
```

Arguments

this_series	Numeric vector; time series object to be plotted.
main_title	Character string; main title of plot. Default is no title.
main_title_line	Integer scalar; position of main title of plot. Default is 1.75.
main_title_cex	Numeric scalar; scaling for main title of plot. Default is this_plot_cex + 0.1.
this_col	Character array; color used for series in the order specified by the user. This array should be as long as the number of years plotted. If only one color is specified, <code>colortools::wheel(this_col)</code> is used to construct an array with enough colors. Default is <code>rainbow(ny)</code> , where <code>ny</code> is the number of years plotted.

start_plot	Integer vector of length 2; Starting date for plot. Default is starting date for the time series.
this_legend	Logical scalar; indicates if a legend is produced for the plot. Default is TRUE.
this_legend_cex	Numeric scalar; scaling for legend. Default is 0.75.
this_right_mar	Numeric scalar; value associated with the margin of the right y-axis specified in the mar entry of the graphics parameters (par). Default is 5.25.
this_legend_inset	Numeric scalar; value associated with the inset of the right legend. Default is -0.15.
this_plot_cex	Numeric scalar; scaling for the plot itself. Default is 0.8.
this_lab_cex	Numeric scalar; scaling for plot labels. Default is the value of this_plot_cex.
this_axis_cex	Numeric scalar; scaling for plot axis. Default is the value of this_plot_cex.
this_mar	Numeric vector; set margins for the plot. Default is c(4,4,4,0.5).
this_reset	Logical scalar; if TRUE, the values of par are reset. Default is TRUE.

Value

Generate year over year plot of user-specified series. If series not specified, print out error message and return NULL.

Examples

```
plot_year_over_year(AirPassengers,
                    "Airline Passenger Series (1949 - 1960)",
                    this_legend_inset = -0.175)
```

reset_par	<i>Reset par()</i>
-----------	--------------------

Description

Reset graphics parameters for plots; taken from stackoverflow post <https://stackoverflow.com/questions/9292563/reset-the-graphical-parameters-back-to-default-values-without-use-of-dev-off>

Usage

```
reset_par()
```

Value

returns default graphics parameters

Examples

```
par(mar=c(5.1, 3.1, 4.1, 1.1), mfrow=c(2,2))
xt_names <- names(xt_data_list)
for (i in 1:4) {
  plot(xt_data_list[[i]], main = xt_names[i], type="l")
}
reset_par()
```

sample_shades	<i>produce a specific number of shades of a given color</i>
---------------	---

Description

Sample from all possible shades of a user provided color. Adapted from <https://www.geeksforgeeks.org/create-distinct-color-palette-in-r/>

Usage

```
sample_shades(this_color = NULL, n_colors, this_index = NULL)
```

Arguments

this_color	An R color name or a color in hexadecimal notation
n_colors	Numeric scalar; number of colors to be generated. If set to NULL, return all shades.
this_index	Integer vector with index for values returned. Default is NULL, which generates a sample of colors matching this_color.

Value

A character vector with the color names for shades of color in hexadecimal notation.

Examples

```
this_shade <- sample_shades("blue", 12)
plot_year_over_year(AirPassengers, "Airline Passenger Series (1949 - 1960)",
  this_col = this_shade)
```

sort_hex_by_hue	<i>Sort hexadecimal colors by hue</i>
-----------------	---------------------------------------

Description

Sort a vector of hexadecimal colors by hue. Taken from <https://www.biolinio.com/sort-hex-colors-in-r/>

Usage

```
sort_hex_by_hue(hex = NULL)
```

Arguments

hex	Character vector containing a list of hexadecimal color codes.
-----	--

Value

A character vector with the color codes sorted by hue

Examples

```
this_wheel <- wheel_invisible("blue", 12)
this_sorted_wheel <- sort_hex_by_hue(this_wheel)
plot_year_over_year(AirPassengers, "Airline Passenger Series (1949 - 1960)",
  this_col = this_sorted_wheel)
```

visual_sig_peaks	<i>Flag visual significant peaks in spectra</i>
------------------	---

Description

Determine positions of visual significant peaks in spectra

Usage

```
visual_sig_peaks(this_seas, spec_type = "sa", spec_freq_code = "seas")
```

Arguments

this_seas	seas object generated from a call of seas on a single time series
spec_type	Character string; type of spectrum. Possible values are 'ori', 'irr', 'rsd', 'sa', 'comp', 'indsa', 'indirr', 'extrsd'. Default is 'sa'.
spec_freq_code	Character string; type of frequency being tested. Possible values are 'seas' or 'td'. Default is 'seas'.

Value

If visually significant peaks found, a numeric vector of the position of the peak frequencies. If no peaks found, 0.

Examples

```
air_seas <- seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11='')
vp_ori_seas <- visual_sig_peaks(air_seas, spec_type = 'ori')
vp_ori_td <- visual_sig_peaks(air_seas, spec_type = 'sa',
  spec_freq_code = 'td')
```

wheel_invisible	<i>Invisible version of color wheel without using colortools package</i>
-----------------	--

Description

Simulate values of the function wheel from the colortools package without using the colortools package, which is no longer in CRAN. sort_hex_by_hue function provided by BIOLINFO in <https://www.biolinfo.com/sort-hex-colors-in-r/> with additional code to sort from light to dark from <https://stackoverflow.com/questions/61193516/how-to-sort-colours-in-r>

Usage

```
wheel_invisible(this_color = NULL, n_colors = NULL)
```

Arguments

`this_color` An R color name or a color in hexadecimal notation
`n_colors` Numeric scalar; number of colors to be generated by wheel

Value

A character vector with the color names of the generated wheel in hexadecimal notation.

Examples

```
this_wheel <- wheel_invisible("blue", 12)
plot_year_over_year(AirPassengers, "Airline Passenger Series (1949 - 1960)",
                     this_col = this_wheel)
```

xt_data_list	<i>US Building Permits, One Family Units</i>
--------------	--

Description

#' A list object of US One family Building Permits for four regions expressed as time series objects that end in October, 2006

Usage

```
xt_data_list
```

Format

A list object with 4 time series elements:

mw1u Midwest one family building permits
ne1u Northeast one family building permits
so1u South one family building permits
we1u West one family building permits

Index

* datasets

- employment_data_mts, [6](#)
- employment_list, [6](#)
- xt_data_list, [45](#)

- cnv_color_codes, [2](#)
- color_blind_palette, [3](#)
- convert_date_to_tis, [3](#)

- display_color, [4](#)
- draw_recession, [5](#)

- employment_data_mts, [6](#)
- employment_list, [6](#)

- flag_peak, [7](#)
- from_rgb_to_hue, [7](#)

- get_recession_dates, [8](#)

- plot_all, [9](#)
- plot_cpgram_resid, [12](#)
- plot_double_spectrum, [13](#)
- plot_fcst, [14](#)
- plot_fcst_history, [15](#)
- plot_fts, [17](#)
- plot_matrix, [18](#)
- plot_multiple, [20](#)
- plot_ratio, [22](#)
- plot_resid, [23](#)
- plot_sa_list, [25](#)
- plot_sa_list_split, [27](#)
- plot_series, [29](#)
- plot_sf, [32](#)
- plot_sf_mean, [33](#)
- plot_sf_series, [36](#)
- plot_single_cell, [38](#)
- plot_table, [39](#)
- plot_year_over_year, [41](#)

- reset_par, [42](#)

- sample_shades, [43](#)
- sort_hex_by_hue, [43](#)

- visual_sig_peaks, [44](#)

- wheel_invisible, [44](#)

- xt_data_list, [45](#)