Package ‘blsplotGG’

January 30, 2024

Title Plots for Seasonal Adjustment Analysts using ggplot2

Version 1.0

Description

Generates several types of time series plots useful for seasonal adjustment analysis using ggplot2. 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,
blsplotR,
BrailleR,
dplyr,
ggfortify,
ggplot2,
ggtext,
gplots,
grDevices,
lubridate,
magrittr,
rlang,
sautilities,
scales,
seasonal,
stringr,
tidyr,
tidyverse,
tis,
tsbox

Depends R (>= 2.10)

R topics documented:

`add_outlier_lines` *add lines for outliers*

Description

add lines for outliers to a ggplot plot object

Usage

```
add_outlier_lines(
  this_p = NULL,
  this_seas = NULL,
  this_color = c("red", "blue", "orangered", "green", "steelblue", "blue"),
 this_line_type = c("dashed", "dotdash", "dashed", "twodash", "dotdash", "dotdash")
)
```

Arguments

`this_p` List object that contains the attributes for a ggplot of a time series. Required entry.

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

this_color	Character array of length 6; color used for different outliers, with the order being 'ao', 'ls', 'tc', 'so', 'rp', 'tls'. Default is c('red', 'blue', 'orangered', 'green', 'steelblue', 'blue').
this_line_type	Character array of length 6; Line type used for different outliers, with the order being 'ao', 'ls', 'tc', 'so', 'rp', 'tls'. Default is c('dashed', 'dotdash', 'dashed', 'twodash', 'dotdash', 'dotdash').

Value

ggplot object with lines for outliers added

Examples

```
air_seas <-
   seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11 = '')
air_df   <-
   data.frame(date = tsbox::ts_df(AirPassengers)$time,
              ori = as.double(seasonal::original(air_seas)),
              sa = as.double(seasonal::final(air_seas)),
              trend = as.double(seasonal::trend(air_seas)))
this_x11_p <- ggplot2::ggplot() +
  ggplot2::geom_line(mapping = ggplot2::aes(x=date, y = ori),
                     color = "grey", data = air_df) +
  ggplot2::geom_line(mapping = ggplot2::aes(x=date, y = sa),
                     color="steelblue", linetype="twodash", data = air_df) +
  ggplot2::geom_line(mapping = ggplot2::aes(x=date, y = trend),
                     color="darkred", linetype="twodash", data = air_df) +
  ggplot2::labs(
    title = "Airline Passenger X-11 Seasonal Adjustment",
    subtitle = NULL,
    x = "Time",
    y = "Airline Passengers")
this_x11_p_with_outlier_lines <-
    add_outlier_lines(this_x11_p, air_seas, this_line_type = rep("dotted", 6))
```

add_recession_shade *Add shading for NBER recession dates*

Description

add shading for US NBER recession dates on ggplot plot object

Usage

```
add_recession_shade(
  this_p = NULL,
  shade_color = "lightblue1",
  shade_alpha = 0.2
)
```

Arguments

`this_p`	List object that contains the attributes for a ggplot of a time series. Required entry.
`shade_color`	character scalar; shading for recession region. Default is `"pink"`.
`shade_alpha`	numeric scalar; controls the intensity of the shading. Default is `0.2`.

Value

ggplot object

Examples

```
air_seas <-
   seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11 = '')
air_df   <-
   data.frame(date = tsbox::ts_df(AirPassengers)$time,
              ori = as.double(seasonal::original(air_seas)),
              sa = as.double(seasonal::final(air_seas)),
              trend = as.double(seasonal::trend(air_seas)))
this_p <- ggplot2::ggplot(air_df) +
  ggplot2::geom_line(ggplot2::aes(x=date, y = ori), color = "grey") +
 ggplot2::geom_line(ggplot2::aes(x=date, y = sa), color="steelblue", linetype="twodash") +
 ggplot2::geom_line(ggplot2::aes(x=date, y = trend), color="darkred", linetype="twodash") +
  ggplot2::labs(
    title = "Airline Passenger X-11 Seasonal Adjustment",
    subtitle = NULL,
    x = "Time",
    y = "Airline Passengers")
this_p_with_recession_shading <-
    add_recession_shade(this_p, shade_color = "steelblue")
```

`color_blind_palette` *Color-blind friendly color palette*

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 pallate 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_spectrum_code *Convert spectrum code to visual peak code*

Description

Convert the three character code used by the `plot_spectrum` function so that it can be used with the `visual_sig_peaks` function from the `blsplot` package.

Usage

```
convert_spectrum_code(this_spectrum = "sp0")
```

Arguments

`this_spectrum`	Character string; three character code for the X-13 spectrum to be generated. Allowed entries are `"sp0"` (modified original series), `"sp1"` (modified X-11 seasonally adjusted series), `"sp2"` (modified X-11 irregular), `"s1s"` (modified SEATS seasonally adjusted series), `"s2s"` (modified SEATS irregular), `"is0"` (modified composite series), `"is1"` (modified indirect seasonally adjusted series), `"is2"` (modified indirect irregular), `spr` (model residuals), or `"ser"` (extended residuals). Default: `"sp0"`.

Value

Text for spectrum associated with code used in `plot_spectrum` function. If improper `this_spectrum` specified, function will return `NULL`.

Examples

```
this_code <- convert_spectrum_code("sp2")
```

employment_data_mts *US Unemployment Series, four main components in an mts object*

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` *US Employment Series, four main components in a list object*

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+

`extract_range_from_ggplot` *Extract range of data from ggplot object*

Description

Computes the range of all data plotted in given ggplot object

Usage

```
extract_range_from_ggplot(this_p = NULL)
```

Arguments

`this_p`	List object that contains the attributes for a ggplot of a time series. Required entry.

Value

Vector of length 2 with the range of the data used to generate a given ggplot object.

Examples

```
air_seas <-
   seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11 = '')
air_df   <-
   data.frame(date = tsbox::ts_df(AirPassengers)$time,
              ori = as.double(seasonal::original(air_seas)),
              sa = as.double(seasonal::final(air_seas)),
              trend = as.double(seasonal::trend(air_seas)))
air_p <- ggplot2::ggplot(air_df, ggplot2::aes(x=date)) +
  ggplot2::geom_line(ggplot2::aes(y = ori), color = "grey") +
  ggplot2::geom_line(ggplot2::aes(y = trend), color="darkred",
                     linetype="twodash") +
  ggplot2::labs(
    title = "Airline Passenger X-11 Trend Component",
    subtitle = NULL,
    x = "Time",
    y = "Airline Passengers")
air_short_p <- plot_date_span(air_p, "1-1-1956", "1-1-1961")
air_short_range <- extract_range_from_ggplot(air_short_p)
```

---

generate_alt_text *Generate alt text for ggplot graphs*

Description

Generates alt text for ggplot graph objects using BrailleR package and adding text suggested by Amy Casale in her article "Writing Alt Text for Data Visualization" https://medium.com/nightingale/writing-alt-text-for-data-visualization-2a218ef43f81

Usage

```
generate_alt_text(
  gg_object = NULL,
  chart_type = NULL,
  data_type = NULL,
  reason_text = NULL,
  short_alt = FALSE
)
```

Arguments

gg_object	ggplot object from which alt text will be generated. Required entry if short_alt = FALSE.
chart_type	character scalar telling what type of plot is used in gg_object. Required entry.
data_type	character scalar detailing what data is used in gg_object. Required entry.
reason_text	character scalar detailing the reason gg_object is plotted. Required entry.
short_alt	logical scalar if TRUE BrailleR text will not be appended to the alt text. Default is FALSE.

Value

generate alt text for plot produced by gg_object

Examples

```
air_seas <- seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11='')
air_df   <-
   data.frame(date = tsbox::ts_df(AirPassengers)$time,
              ori = as.double(seasonal::original(air_seas)),
              sa = as.double(seasonal::final(air_seas)),
              trend = as.double(seasonal::trend(air_seas)))
air_p <- ggplot2::ggplot(air_df, ggplot2::aes(x=date)) +
  ggplot2::geom_line(ggplot2::aes(y = ori), color = "grey") +
  ggplot2::geom_line(ggplot2::aes(y = sa),
                     color="steelblue", linetype="twodash") +
  ggplot2::geom_line(ggplot2::aes(y = trend),
                     color="darkred", linetype="dotdash") +
  ggplot2::ggtitle("Airline Passenger X-11 Seasonal Adjustment")
air_alt_text <-
   generate_alt_text(air_p,
                     "Time series plot",
                     "International Airline Passengers time series",
                     "compare seasonal adjustment and trend to original series")
```

gen_outlier_label *generate x-axis label for outliers*

Description

generate an x-axis label when adding lines for outliers to a ggplot plot object

Usage

```
gen_outlier_label(
  this_seas = NULL,
  this_color = c("red", "blue", "green", "brown", "grey", "yellow")
)
```

Arguments

`this_seas`	seas object generated from a call of `seas` on a single time series. Required entry.
`this_color`	Character array of length 6; color used for different outliers, with the order being `'ao'`, `'ls'`, `'tc'`, `'so'`, `'rp'`, `'tls'`. Default is `c('red', 'blue', 'green', 'brown', 'grey', 'yellow')`.

Value

character string with description of outliers for x-axis label

Examples

```
air_seas <-
   seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11 = '')
air_df   <-
   data.frame(date = tsbox::ts_df(AirPassengers)$time,
              ori = as.double(seasonal::original(air_seas)),
              sa = as.double(seasonal::final(air_seas)),
              trend = as.double(seasonal::trend(air_seas)))
this_p <- ggplot2::ggplot(air_df) +
  ggplot2::geom_line(ggplot2::aes(x=date, y = ori), color = "grey") +
  ggplot2::geom_line(ggplot2::aes(x=date, y = sa), color="steelblue", linetype="twodash") +
  ggplot2::geom_line(ggplot2::aes(x=date, y = trend), color="darkred", linetype="twodash") +
  ggplot2::labs(
    title = "Airline Passenger X-11 Seasonal Adjustment",
    subtitle = NULL,
    y = "Airline Passengers")
this_p_with_outlier_lines <- add_outlier_lines(this_p, air_seas)
outlier_lines_label <- gen_outlier_label(air_seas)
this_p_with_outlier_lines <-
    this_p_with_outlier_lines + ggplot2::xlab(outlier_lines_label)
```

plot_cpgram_resid *Generate cumulative periodogram of the regARIMA residuals*

Description

Generates a plot of the cumulative periodogram of the regARIMA residuals

Usage

```
plot_cpgram_resid(
  this_seas = NULL,
  main_title = "Cumulative periodogram",
  this_palette = NULL
)
```

Arguments

`this_seas`	seas object generated from a call of `seas` on a single time series
`main_title`	Title for the plot. Default is character string `'Cumulative periodogram'`.
`this_palette`	Color used for lines in plot. Default is a color-blind friendly palette generated by the function `color_blind_palette(FALSE)`

Value

Generates a ggplot object of the Cumulative periodogram of the regARIMA residuals. Diagnostic information is included in subheader.

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_date_span *Plot a span of data*

Description

Adds a date span to an existing time series ggplot object

Usage

```
plot_date_span(
  this_p = NULL,
  this_start_span = NULL,
  this_end_span = NULL,
  this_date_breaks = "1 year",
  this_date_format = "%Y",
  reset_y_limit = FALSE
)
```

Arguments

`this_p`	List object that contains the attributes for a ggplot of a time series. Required entry.
`this_start_span`	Character scalar with the date of the start of the span to be plotted. Required entry.
`this_end_span`	Character scalar with the date of the end of the span to be plotted. Required entry.
`this_date_breaks`	Character scalar with the interval for tic marks on the x-axis. Default is `"1 year"`.
`this_date_format`	Character scalar with the format used for the date label on the x-axis.
`reset_y_limit`	Logical scalar that if TRUE, the range of the y-axis is reset. Default is `FALSE`.

Value

A list with ggplot attributes that produces a subplot of the submitted plot.

Examples

```
air_seas <-
   seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11 = '')
air_df   <-
   data.frame(date = tsbox::ts_df(AirPassengers)$time,
              ori = as.double(seasonal::original(air_seas)),
              sa = as.double(seasonal::final(air_seas)),
              trend = as.double(seasonal::trend(air_seas)))
air_p <- ggplot2::ggplot(air_df, ggplot2::aes(x=date)) +
  ggplot2::geom_line(ggplot2::aes(y = ori), color = "grey") +
  ggplot2::geom_line(ggplot2::aes(y = trend),
                     color="darkred",
                   linetype="twodash") +
  ggplot2::labs(
    title = "Airline Passenger X-11 Trend Component",
    subtitle = NULL,
    x = "Time",
    y = "Airline Passengers")
air_short_p <-
   plot_date_span(air_p, "1-1-1956", "1-1-1962", reset_y_limit = TRUE)
```

plot_double_spectrum *Generate double spectrum plot of the original and seasonally adjusted series.*

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_color = c("darkblue", "darkgreen"),
  this_median_color = c("blue", "green"),
  this_freq_color = c("steelblue", "forestgreen"),
  this_peak_color = c("violet", "brown")
)
```

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 this_seas. Used as the label of the Y-axis if specified.
this_color	Character vector of length 2. Colors used for original and seasonally adjusted spectrum in plot. Defaults are c('darkblue', 'darkgreen').
this_median_color	Character vector of length 2. Colors used for medians of the original and seasonally adjusted spectrum, respectively. Defaults are c('blue', 'green').
this_freq_color	Character vector of length 2. Colors used for seasonal and trading day frequencies, respectively. Defaults are c('steelblue', 'forestgreen').
this_peak_color	Character vector of length 2. Colors used for peaks at seasonal and trading day frequencies, respectively. Defaults are c('violet', 'brown').

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_color = c('steelblue', 'forestgreen'),
                     this_median_color = c("blue", "green"),
                     this_freq_color = c('darkblue', 'darkgreen'),
                     this_peak_color = c("red", "orange"))
```

plot_fcst *Forecast plot*

Description

Generates regARIMA forecasts with confidence bounds

Usage

```
plot_fcst(
  this_seas = NULL,
  main_title = "ARIMA forecasts",
  do_sub = TRUE,
  this_x_label = "Time",
  this_y_label = " ",
  length_ori = 2,
  do_grid = FALSE,
  this_palette = c("darkgrey", "blue", "darkgreen", "darkgreen"),
  this_guide_legend = "Forecast"
)
```

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_sub` Logical scalar; indicates if subtitle is generated. Default is to generate the subtitle.

`this_x_label` Label for X-axis. Default is `"Time"`

`this_y_label` Label for Y-axis. Default is `" "`

`length_ori` Integer scalar; number of years of the original series to show with forecasts. Default is 2 years.

`do_grid` Logical scalar; indicates if certain plots will have grid lines. Default is no grid lines.

`this_palette` Array of character strings; color used for original series, forecast, and upper and lower forecast bounds. Default is `c("darkgrey", "blue", "darkgreen", "darkgreen")`.

`this_guide_legend` Title for legend. Default is `"Forecast"`

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)
air_fcst_p <- plot_fcst(air_seas, main_title = 'Forecasts for Airline Passengers', do_grid = TRUE)
```

---

plot_fcst_history *Generate forecast history plot*

Description

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

Usage

```
plot_fcst_history(
  seas_mdl1 = NULL,
  seas_mdl2 = NULL,
  main_title = "Differences in the Sum of Squared Forecast Errors",
  name_mdl1 = "Model 1",
  name_mdl2 = "Model 2",
  do_grid = FALSE,
  this_x_label = "Time",
  this_y_label = " ",
  this_palette = c("blue", "darkgreen"),
  this_guide_legend = "Fcst History"
)
```

Arguments

`seas_mdl1`	seas object generated from a call of `seas` on a single time series for the first model
`seas_mdl2`	seas object generated from a call of `seas` on a single time series for the second model
`main_title`	Character string; main title of plot. Default is `'Differences in the Sum of Squared Forecast Errors'`.
`name_mdl1`	Character string; Description of first model for use in the subtitle. Default is `'Model 1'`.
`name_mdl2`	Character string; Description of second model for use in the subtitle. Default is `'Model 2'`.
`do_grid`	Logical scalar; indicates if certain plots will have grid lines. Default is no grid lines.
`this_x_label`	Label for X-axis. Default is `"Time"`
`this_y_label`	Label for Y-axis. Default is `" "`
`this_palette`	Character array of length 2; color used for each forecast lag. Default is `c('blue', 'darkgreen')`.
`this_guide_legend`	Title for legend. Default is `"Fcst History"`

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_mdl <-
    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_mdl2 <-
    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_mdl, air_seas_mdl2,
  main_title = 'Differences in the Sum of Squared Forecast Errors for Airline Passengers',
   name_mdl1 = 'Airline model', name_mdl2 = 'Airline model + regressors')
```

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 = "Ratio Plot",
  main_subtitle = NULL,
  this_x_label = "Time",
  this_y_label = "Ratio",
  do_grid = FALSE,
  ratio_mean = 1,
  ratio_color = "steelblue"
)
```

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_subtitle` Subtitle for the plot. Default is `NULL`.

`this_x_label` Label for X axis. Default is "Time".

`this_y_label` Label for Y axis. Default is "Ratio".

`do_grid` Logical scalar; indicates if certain plots will have grid lines. Default is no grid lines.

`ratio_mean` Assumed mean value for the ratio. Default is 1.0

`ratio_color` Color used for lines in ratio plot. Default is 'steelblue'.

Value

A list with ggplot attributes that produces a ratio plot.

Examples

```
ukgas_x11_seas <-
   seasonal::seas(UKgas, series.period = 4, arima.model = '(0 1 1)(0 1 1)',
                  transform.function = 'log', forecast.maxlead = 20,
                  x11.seasonalma = "s3x5",
                  check.print = c( 'pacf', 'pacfplot' ))
ukgas_x11_sf     <- seasonal::series(ukgas_x11_seas, "d10")
ukgas_sf_ratio   <- plot_ratio(ukgas_x11_sf, main_title = "UK Gas X-11 Seasonal Factors",
                               main_subtitle = "Seasonal Filter: 3x5")
```

plot_series   *Plot time series object.*

Description

Generate plot of user-specified time series (ts) object.

Usage

```
plot_series(
  this_series = NULL,
  this_series_name = NULL,
  main_title = NULL,
  sub_title = NULL,
  y_label = NULL,
  y_limit = NULL,
  x_label = "Time",
  start_plot = NULL,
  do_grid = FALSE,
  draw_recess = FALSE,
  recess_color = "lightgrey",
  recess_sub = TRUE,
  this_line_type = "solid",
  line_color = "grey"
)
```

Arguments

`this_series` Time series object; required entry.

`this_series_name`
Character string; name of time series. No default.

`main_title` Character string; main title of plot. A title will be generated if no title is specified.

`sub_title` Character string; subtitle of plot. There is no default subtitile.

`y_label` Character string; y-axis label for plot. If not specified, set to `this_series_name`, 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 series specified.

`x_label` Label for X axis. Default is `"Time"`.

`start_plot` Integer vector of length 2; Starting date for plot. Default is starting date for the time series.

`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_color` 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.

`this_line_type` Character string; indicates line type of each plot produced. Default is "solid".

`line_color` Character string; color used for series in the plot. Default is `'grey'`.

Value

Generate plot of user-specified 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='',
                 transform.function = "log")
air_seas_d11 <- seasonal::final(air_seas)
air_d11_p <- plot_series(air_seas_d11, this_series_name = 'AirPassengers',
         main_title = 'X-11 Seasonal Adjustment of Airline Passengers',
         sub_title = 'Box-Jenkins Airline series',
         do_grid = TRUE, draw_recess = TRUE, line_color = 'darkblue')
```

plot_spectrum *Generate spectrum plot of either the original, seasonally adjusted, irregular, or model residuals.*

Description

Generate plot of spectrum from X-13ARIMA-SEATS specified by the user.

Usage

```
plot_spectrum(
  this_seas = NULL,
  this_spectrum = "sp0",
  xaxis_bls = TRUE,
  main_title = "AR Spectrum",
  sub_title = NULL,
  series_name = NULL,
  do_grid = FALSE,
  this_color = "darkblue",
  this_median_color = "blue",
  this_freq_color = c("steelblue", "forestgreen"),
  this_peak_color = c("violet", "brown")
)
```

Arguments

`this_seas`	`seas` object generated from a call of `seas` on a single time series
`this_spectrum`	Character string; three character code for the X-13 spectrum to be generated. Allowed entries are `"sp0"` (modified original series), `"sp1"` (modified X-11 seasonally adjusted series), `"sp2"` (modified X-11 irregular), `"s1s"` (modified SEATS seasonally adjusted series), `"s2s"` (modified SEATS irregular), `"is0"` (modified composite series), `"is1"` (modified indirect seasonally adjusted series), `"is2"` (modified indirect irregular), `spr` (model residuals), or `"ser"` (extended residuals). Default: `"sp0"`.
`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'`.
`sub_title`	Character scalar; Description of time series used in `this_seas`. Used as the subtitle of the plot if specified.
`series_name`	Character scalar; name of the time series used in `this_seas`. Used as the label of the Y-axis if specified.
`do_grid`	Logical scalar; indicates if certain plots will have grid lines. Default is no grid lines.
`this_color`	Character string. Colors used for spectrum in plot. Default is `'darkblue'`.
`this_median_color`	Character string. Colors used for medians of the spectrum. Default is `'blue'`.
`this_freq_color`	Character vector of length 2. Colors used for seasonal and trading day frequencies, respectively. Defaults are `c('steelblue', 'forestgreen')`.

`this_peak_color`
: Character vector of length 2. Colors used for peaks at seasonal and trading day frequencies, respectively. Defaults are `c('violet', 'brown')`.

Value

Generate plot of spectrum generated by X-13

Examples

```
air_seas <- seasonal::seas(AirPassengers, arima.model = '(0 1 1)(0 1 1)', x11='')
plot_spectrum(air_seas, this_spectrum = "sp2", series_name = 'AirPassengers',
                   sub_title = "Airline Passengers",
                   this_color = 'steelblue', this_median_color = "blue",
                   this_freq_color = c('darkblue', 'darkgreen'),
                   this_peak_color = c("red", "orange"))
```

`plot_table` *Plot table from X-13ARIMA-SEATS seasonal adjustment run.*

Description

Generate plot of user-specified series from a `seas` object generated by the `seasonal` package.

Usage

```
plot_table(
  this_seas = NULL,
  this_table = NULL,
  main_title = NULL,
  sub_title = NULL,
  y_label = NULL,
  y_limit = NULL,
  x_label = "Time",
  start_plot = NULL,
  do_grid = FALSE,
  draw_recess = FALSE,
  recess_color = "lightgrey",
  recess_sub = TRUE,
  add_outlier = FALSE,
  use_ratio = FALSE,
  ratio_mean = 1,
  this_line_type = "solid",
  line_color = "grey",
  outlier_color = c("red", "blue", "orangered", "green", "steelblue", "blue"),
 outlier_line_type = c("dashed", "dotdash", "dashed", "twodash", "dotdash", "dotdash")
)
```

Arguments

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

`this_table` 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.

`main_title` Character string; main title of plot. A title will be generated if no title is specified.

`sub_title` Character string; subtitle of plot. There is no default subtitile.

`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 series specified.

`x_label` Label for X axis. Default is `"Time"`.

`start_plot` Integer vector of length 2; Starting date for plot. Default is starting date for the time series.

`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_color` 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.

`add_outlier` Logical scalar; indicates if lines for identified outliers are included in series plots. Default is not including lines for identified outliers.

`use_ratio` 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.

`ratio_mean` Assumed mean value for the ratio. Default is `1.0`

`this_line_type` Character string; indicates line type of each plot produced. Default is "solid".

`line_color` Character string; color used for series in the plot. Default is `'grey'`.

`outlier_color` Character array of length 6; color used for different outliers, with the order being `'ao'`, `'ls'`, `'tc'`, `'so'`, `'rp'`, `'tls'`. Default is `c('red', 'blue', 'orangered', 'green', 'steelblue', 'blue')`.

`outlier_line_type`
Character array of length 6; Line type used for different outliers, with the order being `'ao'`, `'ls'`, `'tc'`, `'so'`, `'rp'`, `'tls'`. Default is `c('dashed', 'dotdash', 'dashed', 'twodash', 'dotdash', 'dotdash')`.

Value

Generate plot of user-specified series from X-13 table. 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', transform.function = "log")
air_d11_p <- blsplotGG::plot_table(air_seas, 'd11', y_label = 'AirPassengers',
        main_title = 'X-11 Seasonal Adjustment of Airline Passengers',
         sub_title = 'Box-Jenkins Airline series',
         do_grid = TRUE, draw_recess = TRUE,
         use_ratio = FALSE, add_outlier = TRUE, line_color = 'darkblue')
air_d16_p <- blsplotGG::plot_table(air_seas, 'd16', y_label = 'AirPassengers',
         main_title = 'X-11 Seasonal Adjustment of Airline Passengers',
         do_grid = FALSE, draw_recess = TRUE,
         use_ratio = TRUE, add_outlier = TRUE, line_color = 'steelblue')
```

---

plot_x11_and_seats *Compare X-11 and SEATS seasonal adjustment*

Description

Generates a ggplot object with a time series plot that compares an X-11 and SEATS seasonal adjustment, optionally including the original series.

Usage

```
plot_x11_and_seats(
  this_x11 = NULL,
  this_seats = NULL,
  this_ori = NULL,
  main_title = "Compare X-11 and SEATS",
  main_subtitle = NULL,
  this_x_label = "Time",
  this_y_label = " ",
  do_grid = FALSE,
  this_palette = c("#999999", "#009E73", "#0072B2"),
  this_guide_legend = "Series"
)
```

Arguments

this_x11	Time series of the X-11 seasonal adjustment. Required entry.
this_seats	Time series of the SEATS seasonal adjustment. Required entry.
this_ori	Time series of the original series. Optional entry.
main_title	Title for the plot. Default is character string 'Comparison of X-11 and SEATS Seasonal Adjustments'.
main_subtitle	Subtitle for the plot. Optional entry.
this_x_label	Label for X-axis. Default is "Time"
this_y_label	Label for Y-axis. Default is " "
do_grid	Logical scalar; indicates if certain plots will have grid lines. Default is no grid lines.
this_palette	Color used for lines in plot. Default is a color-blind friendly palette - c("#999999", "#009E73", "#0072B2").
this_guide_legend	Title for legend. Default is "Series"

Value

A list with ggplot attributes that produces a plot comparing an X-11 and SEATS seasonal adjustment, trend, or factors.

Examples

```
ukgas_x11_seas   <-
   seasonal::seas(UKgas, series.period = 4, arima.model = '(0 1 1)(0 1 1)',
                  transform.function = 'log', forecast.maxlead = 20,
                  x11.seasonalma = "s3x5",
                  check.print = c( 'pacf', 'pacfplot' ))
ukgas_seats_seas <-
   seasonal::seas(UKgas, series.period = 4, arima.model = '(0 1 1)(0 1 1)',
                  transform.function = 'log', forecast.maxlead = 20,
                  check.print = c( 'pacf', 'pacfplot' ))
ukgas_x11_sa     <- seasonal::final(ukgas_x11_seas)
ukgas_seats_sa   <- seasonal::final(ukgas_seats_seas)
ukgas_seats_v_x11_p <-
    plot_x11_and_seats(this_x11 = ukgas_x11_sa, this_seats = ukgas_seats_sa,
                       main_title = "UK Gas Seasonal Adjustments",
                       main_subtitle = "X-11 - Blue, SEATS - Green",
                       this_palette = c("steelblue", "forestgreen"))
```

---

plot_x11_and_seats_facet

Compare X-11 and SEATS seasonal adjustments in a facet plot

Description

Generates a ggplot object with a time series facet plot that compares an X-11 and SEATS seasonal adjustment, optionally including the original series.

Usage

```
plot_x11_and_seats_facet(
  this_x11 = NULL,
  this_seats = NULL,
  this_ori = NULL,
  main_title = "Comnpare X-11 and SEATS",
  main_subtitle = NULL,
  this_x_label = "Time",
  this_y_label = " ",
  this_color = "steelblue"
)
```

Arguments

`this_x11`	Time series of the X-11 seasonal adjustment. Required entry.
`this_seats`	Time series of the SEATS seasonal adjustment. Required entry.
`this_ori`	Time series of the original series. Optional entry.
`main_title`	Title for the plot. Default is character string `'Comparison of X-11 and SEATS Seasonal Adjustments'`.

`main_subtitle`	Subtitle for the plot. Optional entry.
`this_x_label`	Label for X-axis. Default is `"Time"`
`this_y_label`	Label for Y-axis. Default is `" "`
`this_color`	Color used for lines in plot. Default is `"steelblue"`.

Value

A list with ggplot attributes that produces a facet plot comparing an X-11 and SEATS seasonal adjustment, trend, or factor.

Examples

```
ukgas_x11_seas   <-
   seasonal::seas(UKgas, series.period = 4, arima.model = '(0 1 1)(0 1 1)',
                  transform.function = 'log', forecast.maxlead = 20,
                  x11.seasonalma = "s3x5",
                  check.print = c( 'pacf', 'pacfplot' ))
ukgas_seats_seas <-
   seasonal::seas(UKgas, series.period = 4, arima.model = '(0 1 1)(0 1 1)',
                  transform.function = 'log', forecast.maxlead = 20,
                  check.print = c( 'pacf', 'pacfplot' ))
ukgas_x11_sa     <- seasonal::final(ukgas_x11_seas)
ukgas_seats_sa   <- seasonal::final(ukgas_seats_seas)
ukgas_seats_v_x11_facet_p <-
    plot_x11_and_seats_facet(this_x11 = ukgas_x11_sa, this_seats = ukgas_seats_sa,
                       main_title = "UK Gas Seasonal Adjustments",
                       this_color = "forestgreen")
```

`seasonal_subplot` *Seasonal sub-plot*

Description

Generates a seasonal sub-plot from a ts object of seasonal (or combined adjustment) factors

Usage

```
seasonal_subplot(
  this_sf = NULL,
  this_sf_range = NULL,
  main_title = "Seasonal Subplot",
  main_subtitle = NULL,
  this_x_label = "Month",
  this_y_label = "",
  subplot_color = "steelblue"
)
```

Arguments

`this_sf` Time series of seasonal factors from X-11 or SEATS

`this_sf_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_subtitle` Subtitle for the plot. Default is `NULL`.

`this_x_label` Label for X axis. Default is `"Time"`.

`this_y_label` Label for Y axis. Default is `"Ratio"`.

`subplot_color` Color used for lines in ratio plot. Default is `'steelblue'`.

Value

A list with ggplot attributes that produces a ratio plot.

Examples

```
ukgas_x11_seas <-
   seasonal::seas(UKgas, series.period = 4, arima.model = '(0 1 1)(0 1 1)',
                  transform.function = 'log', forecast.maxlead = 20,
                  x11.seasonalma = "s3x5",
                  check.print = c( 'pacf', 'pacfplot' ))
ukgas_x11_sf     <- seasonal::series(ukgas_x11_seas, "d10")
p_ukgas_sf_sub   <-
    seasonal_subplot(ukgas_x11_sf,
                     main_title = "UK Gas X-11 Seasonal Factors",
                     main_subtitle = "Seasonal Filter: 3x5",
                     this_x_label = "Quarter")
```

`what_spectrum` *What spectrum is plotted*

Description

Return which spectrum plot is generated based on the three character code used by the `plot_spectrum` function.

Usage

```
what_spectrum(this_spectrum = "sp0", use_title_case = FALSE)
```

Arguments

`this_spectrum` Character string; three character code for the X-13 spectrum to be generated. Allowed entries are `"sp0"` (modified original series), `"sp1"` (modified X-11 seasonally adjusted series), `"sp2"` (modified X-11 irregular), `"s1s"` (modified SEATS seasonally adjusted series), `"s2s"` (modified SEATS irregular), `"is0"` (modified composite series), `"is1"` (modified indirect seasonally adjusted series), `"is2"` (modified indirect irregular), `spr` (model residuals), or `"ser"` (extended residuals). Default: `"sp0"`.

`use_title_case` Logical scalar; convert string to title case. Default is FALSE.

Value

Text for spectrum associated with code used in `plot_spectrum` function. If improper `this_spectrum` specified, function will return `NULL`.

Examples

```
sub_title <- what_spectrum("sp2", use_title_case = TRUE)
```

`xt_data_list` *US Building Permits, One Family Units*

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