Package 'vctsfr'

December 7, 2023

Title Visualizing Collections of Time Series Forecasts

Version 0.1.1

Description A way of visualizing collections of time series and, optionally
their future values, forecasts for their future values and prediction
intervals for the forecasts. A web-based GUI can be used to display the
information in a collection of time series.

Maintainer Francisco Martinez <fmartin@ujaen.es>

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Suggests forecast, knitr, Mcomp, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports ggplot2, methods, shiny

URL https://github.com/franciscomartinezdelrio/vctsfr

BugReports https://github.com/franciscomartinezdelrio/vctsfr/issues

VignetteBuilder knitr

NeedsCompilation no

Author Maria Pilar Frias-Bustamante [aut]
(<https://orcid.org/0000-0001-6886-0953>),
Francisco Martinez [aut, cre, cph]
(<https://orcid.org/0000-0002-5206-1898>)

Repository CRAN

Date/Publication 2023-12-07 18:50:02 UTC

R topics documented:

1

---

check_time_series_collection

Check that a collection of time series is properly formatted

Description

This function checks that an object holding a collection of time series, their future values and their forecasts has the correct format. This kind of objects are used in function [plot_collection()](). A collection of time series should be a list compounded of objects of class ts_info, which are built using the [ts_info()]() function.

Usage

```
check_time_series_collection(collection)
```

Arguments

collection   a list representing a collection of time series as described in [plot_collection()]().

Value

a character string with value "OK" if the object is properly formatted. Otherwise, the character string indicates the first error found in the object's format.

Examples

```
c <- list(ts_info(USAccDeaths), ts_info(ldeaths))
check_time_series_collection(c)
```

GUI_collection   *Launches the web-based GUI for visualizing time series*

Description

Launches the web-based GUI for visualizing a collection of time series in a web browser.

Usage

```
GUI_collection(collection)
```

Arguments

collection   a list with the collection of time series. Each component of the list must have been built with the [ts_info()](#) function.

Details

The **vctsfr** package provides a Shiny-based GUI to visualize collections of time series and their forecasts. The main features of the GUI are:

- It allows you to easily navigate through the different series.

- You can select which forecasting methods are displayed.

- In the case you display a single forecasting method with associated prediction intervals, you can select the prediction interval to display.

- Forecasting accuracy measures are displayed.

Value

Nothing

Examples

```
# create a collection of two time series and visualize them
c <- list(ts_info(USAccDeaths), ts_info(ldeaths))
GUI_collection(c)
```

---

pi_info   *Create a prediction interval object*

Description

The object created represents a prediction interval for the forecast of the future values of a time series.

Usage

```
pi_info(level, lpi, upi)
```

Arguments

level   a number in the interval (0, 100) indicating the level of the prediction interval.

lpi   a time series of class ts or a vector. Lower limit of a prediction interval.

upi   a time series of class ts or a vector. Upper limit of a prediction interval.

Value

An object of class pi_info. It is a list containing all the information supplied to the function.

See Also

[prediction_info()](#) which uses this function to specify prediction intervals.

Examples

```
if (require("forecast")) {
  time_series <- ts(rnorm(40))
  f <- meanf(time_series, level = 95)
  info <- pi_info(95, f$lower, f$upper)
}
```

---

plot_collection	*Create a* ggplot *object associated with a time series belonging to a collection.*

Description

Apart from the time series, future values and forecasts for the future values form part of the ggplot object.

Usage

```
plot_collection(collection, number, methods = NULL, level = NULL, sdp = TRUE)
```

Arguments

collection	a list with the collection of time series. Each component of the list must have been built with the [ts_info()](#) function.
number	an integer. The number of the time series. It should be a value between 1 and length(collection).
methods	NULL (default) or a character vector indicating the names of the forecasting methods to be displayed.
level	NULL (default) or a number in the interval (0, 100) indicating the level of the prediction interval to be shown. This parameter in considered only when just one forecasting method is plotted and the forecasting method has a prediction interval with the specified level.
sdp	logical. Should data points be shown in the plot? (default value TRUE)

Details

The `collection` parameter must be a list. Each component of the list stores a time series and, optionally, its future values, forecasts for the future values and prediction intervals for the forecasts. Each component should have been created using the `ts_info()` function.

In the example section you can see an example of a collection of time series. If the `collection` parameter is not specified correctly, a proper message is shown.

Value

The ggplot object representing the time series and its forecast.

See Also

`ts_info()` function to see how to build the components of the `collection` parameter.

Examples

```
# create a collection of two time series and plot both time series
c <- list(ts_info(USAccDeaths), ts_info(ldeaths))
plot_collection(c, number = 1)
plot_collection(c, number = 2, sdp = FALSE)

# create a collection of one time series with future values and forecasts
if (require(forecast)) {
  c <- vector(2, mode = "list")
  timeS <- window(USAccDeaths, end = c(1977, 12))
  f <- window(USAccDeaths, start = c(1978, 1))
  ets_fit   <- ets(timeS)
  ets_pred  <- forecast(ets_fit, h = length(f), level = 90)
  mean_pred <- meanf(timeS, h = length(f), level = 90)
  c[[1]] <- ts_info(timeS, future = f,
            prediction_info("ES", ets_pred$mean,
                              pi_info(90, ets_pred$lower, ets_pred$upper)),
            prediction_info("Mean", mean_pred$mean,
                              pi_info(90, mean_pred$lower, mean_pred$upper))
  )
  timeS <- ts(rnorm(30, sd = 3))
  f <- rnorm(5, sd = 3)
  rw <- rwf(timeS, h = length(f), level = 80)
  mean <- meanf(timeS, h = length(f), level = 90)
  c[[2]] <- ts_info(timeS, future = f,
            prediction_info("Random Walk", rw$mean,
                              pi_info(80, rw$lower, rw$upper)),
            prediction_info("Mean", mean$mean,
                              pi_info(90, mean$lower, mean$upper))
  )
  plot_collection(c, number = 1)
}
if (require("forecast"))
  plot_collection(c, number = 2)
if (require("forecast"))
```

```
  plot_collection(c, number = 2, methods = "Mean") # just plot a forecasting method
if (require("forecast"))
  plot_collection(c, number = 2, methods = "Random Walk", level = 80)
```

---

plot_predictions   *Creates a ggplot object with a time series and some forecasts*

Description

Create a ggplot object with a time series and, optionally, some future values of the time series and several forecast for those future values.

Usage

```
plot_predictions(ts, future = NULL, predictions = NULL, sdp = TRUE)
```

Arguments

ts	a time series of class ts.
future	NULL (default) or a time series of class ts or a vector. Future values of the time series.
predictions	NULL (default) or a named list containing the predictions for the future values. Each component of the list should contain a vector or an object of class ts representing a forecast, the name of the component should be the name of the forecasting method.
sdp	logical. Should data points be shown? (default value TRUE)

Details

If future or the forecasts in the prediction list are vectors then they are supposed to start after the last data of the time series.

Value

The ggplot object representing the time series and its forecast.

Examples

```
# plot a time series, its future values and two forecasts
ts <- window(USAccDeaths, end = c(1977, 12))
f <- window(USAccDeaths, start = c(1978, 1))
prediction1 <- rep(mean(ts), 12)
prediction2 <- as.vector(window(ts, start = c(1977, 1)))
p <- list(Mean = prediction1, Naive = prediction2)
plot_predictions(ts, future = f, predictions = p)
```

---

plot_ts   *Create a ggplot object with a time series and forecast*

Description

Create a `ggplot` object associated with a time series and, optionally, its future values, a forecast for its future values and a prediction interval of the forecast.

Usage

```
plot_ts(
  ts,
  future = NULL,
  prediction = NULL,
  method = NULL,
  lpi = NULL,
  upi = NULL,
  level = NULL,
  sdp = TRUE
)
```

Arguments

ts	a time series of class `ts`.
future	NULL (default) or a time series of class `ts` or a vector. Future values of the time series.
prediction	NULL (default) or a time series of class `ts` or a vector. Forecast of the future values of the time series.
method	NULL (default) a character string with the name of the method used to forecast the future values of the time series. This name will appear in the legend.
lpi	NULL (default) or a time series of class `ts` or a vector. Lower limit of a prediction interval for the `prediction` parameter.
upi	NULL (default) or a time series of class `ts` or a vector. Upper limit of a prediction interval for the `prediction` parameter.
level	NULL (default) a number in the interval (0, 100) indicating the level of the prediction interval.
sdp	logical. Should data points be shown? (default value TRUE)

Details

If `future` or `prediction` are vectors then they are supposed to start after the last data of the time series.

Value

The `ggplot` object representing the time series and its forecast.

Examples

```
library(ggplot2)
plot_ts(USAccDeaths) # plot a time series

# plot a time series, not showing data points
plot_ts(USAccDeaths, sdp = FALSE)

# plot a time series, its future values and a prediction
ts <- window(USAccDeaths, end = c(1977, 12))
f <- window(USAccDeaths, start = c(1978, 1))
p <- ts(window(USAccDeaths, start = c(1976, 1), end = c(1976, 12)),
        start = c(1978, 1),
        frequency = 12
)
plot_ts(ts, future = f, prediction = p)

# plot a time series and a prediction
plot_ts(USAccDeaths, prediction = rep(mean(USAccDeaths), 12),
        method = "Mean")

# plot a time series, a prediction and a prediction interval
if (require(forecast)) {
  timeS <- window(USAccDeaths, end = c(1977, 12))
  f <- window(USAccDeaths, start = c(1978, 1))
  ets_fit <- ets(timeS)
  p <- forecast(ets_fit, h = length(f), level = 90)
  plot_ts(timeS, future = f, prediction = p$mean, method = "ES",
          lpi = p$lower, upi = p$upper, level = 90
  )
}
```

---

prediction_info	*Create an object with a prediction about the future values of a time series*

Description

The object created contains a forecast and, optionally, prediction intervals for the forecast.

Usage

```
prediction_info(name, forecast, ...)
```

Arguments

name	a character indicating the name of the method used to forecast.
forecast	a time series of class ts or a vector. It is a prediction for the future values of a time series.
...	prediction intervals for the forecast. These prediction intervals must have been built with the [pi_info()](#) function.

Value

an object of class `pred_info`. A list with the information supplied to the function.

See Also

[pi_info()](pi_info) for how to create prediction intervals.

Examples

```
if (require("forecast")) {
  time_series <- ts(rnorm(40))
  f <- meanf(time_series, level = 95)
  info <- prediction_info("mean", f$mean, pi_info(95, f$lower, f$upper))
}
```

---

ts_info   *Create an object with information about a time series*

Description

The information about the time series is compounded of the time series and, optionally, its future values and forecasts for those future values (and prediction intervals for those forecasts).

Usage

```
ts_info(historical, ..., future = NULL, name = NULL)
```

Arguments

historical	a time series of class `ts` with the historical values of the series.
...	forecasts for the future values of the time series. A forecast must have been built with the [prediction_info()](prediction_info) function. See the examples section.
future	NULL (default) or a time series of class `ts` or a vector. The future values of the time series (possibly to be forecast).
name	NULL (default) or a character string with information about the time series. Typically, its name.

Value

An object of class `ts_info`. It is a list containing all the information supplied to the function.

See Also

[prediction_info()](prediction_info) for how to create forecasts.

Examples

```
# only information about a time series
info <- ts_info(USAccDeaths)

# Information about a time series and its future values
info2 <- ts_info(ts(rnorm(50)), future = rnorm(10))

# Information about a time series, its future values and a forecast
if (require("forecast")) {
  t <- ts(rnorm(50))
  f <- rnorm(10)
  mf <- meanf(t, level = 95)
  info3 <- ts_info(t, future = f,
                   prediction_info("mean", mf$mean,
                                   pi_info(95, mf$lower, mf$upper)
                   )
  )
}
```

Index