# Package ‘iNZightTS’

## January 17, 2024

**Type** Package  

**Title** Time Series for 'iNZight'  

**Version** 2.0.0  

**Depends** R (>= 3.2)  

**Imports** colorspace, dplyr (>= 1.1.0), forcats, ggplot2, ggtext, glue, graphics, grDevices, grid, patchwork, rlang, stats, tidyr, utils, tibble, fable, fabletools, feasts, evaluate, lubridate, stringr  

**Suggests** covr, testthat  

**Description**  
Provides a collection of functions for working with time series data, including functions for drawing, decomposing, and forecasting. Includes capabilities to compare multiple series and fit both additive and multiplicative models. Used by 'iNZight', a graphical user interface providing easy exploration and visualisation of data for students of statistics, available in both desktop and online versions. Holt (1957) [10.1016/j.ijforecast.2003.09.015], Winters (1960) [10.1287/mnsc.6.3.324], Cleveland, Cleveland, & Terpenning (1990) "STL: A Seasonal-Trend Decomposition Procedure Based on Loess".

**BugReports** https://github.com/iNZightVIT/iNZightTS/issues  

**Contact** inzight_support@stat.auckland.ac.nz  

**URL** https://inzight.nz  

**LazyData** true  

**License** GPL-3  

**Encoding** UTF-8  

**Language** en-GB  

**RoxygenNote** 7.2.3  

**NeedsCompilation** no  

**Author** Tom Elliott [aut, cre] (https://orcid.org/0000-0002-7815-6318), Zhaoming Su [aut], Junjie Zeng [ctb], Simon Potter [ctb],
The iNZightTS package provides some simple analysis tools for exploring time series data. It is used in the iNZight software.

Author(s)

Tom Elliott (previously: Marco Kuper, Simon Potter, and David Banks)

See Also

inzightts
decomp

Decompose a time series object

Description
Decomposes a time series represented by an inz_ts object into its seasonal, trend, and remainder components using the specified smoothing method.

Usage
decomp(
  x,
  var = NULL,
  sm_model = c("stl"),
  mult_fit = FALSE,
  model_range = NULL,
  ...
)

## S3 method for class 'inz_dcmp'
plot(
  x,
  recompose.progress = c(0, 0),
  recompose = any(recompose.progress > 0),
  ylab = NULL,
  title = NULL,
  colour = c("#1B9E46", "#45a8ff", "orangered"),
  ...
)

Arguments
x An 'inz_dcmp' object representing the decomposed time series.
var A character vector of length one, or NULL.
sm_model The smoothing method to be used. Currently on "stl" is available.
mult_fit If TRUE, a multiplicative model is used; otherwise, an additive model is used by default.
model_range The range of data to be decomposed by the model. It can be specified as dates or years. If part of model_range lies outside the range of the data, the exceeding proportion is ignored.
... Additional arguments (ignored).
recompose.progress A numeric vector of length 2, controlling the display of recomposition progress when 'recompose' is 'TRUE'. The first component shows the progress for the seasonal component (0 to 1), and the second component tracks the number of observations recomposed so far.
**ggplotable**

- **recompose**: Logical indicating whether the recomposition should be displayed or not.
- **ylab**: The label for the y-axis of the plot.
- **title**: The title for the plot.
- **colour**: A vector of three colors specifying the colors for the trend, seasonal, and residuals components, respectively.

**Value**

An `inz_dcmp` object, which is a sub-class of `dable`, representing the decomposed components of the time series.

**References**


**See Also**

dable

**Examples**

```r
ts <- inzightts(visitorsQ)
d <- decomp(ts)

## Not run:
plot(d)

## End(Not run)
```

---

**ggplotable**

Preliminary check for a plotly::ggplotly() call

**Description**

Check if a plot generated by iNZightTS can be passed to plotly::ggplotly().

**Usage**

ggplotable(x)

**Arguments**

- **x**: a ggplot object produced by iNZightTS

**Value**

- **logical**
inzightts

See Also

ggplotly

Examples

x <- inzightts(visitorsQ)
## Not run:
ggplotable(plot(x))
ggplotable(plot(x, names(x)[-1]))
## End(Not run)

inzightts

Coerce data to an inzightts (time-series) object

Description

The function inzightts creates temporal data frames for use in iNZight. Unlike ts objects, these are tsibble objects that enable temporal data wrangling, adapting to tidy data principles, which are both data- and model-oriented.

Usage

inzightts(x, ...)

## S3 method for class 'character'
inzightts(x, stringsAsFactors = TRUE, as.is = TRUE, ...)

## S3 method for class 'data.frame'
inzightts(
  x,
  var = NULL,
  index = NULL,
  key = NULL,
  start = NULL,
  end = NULL,
  freq = NULL,
  ...
)

## S3 method for class 'ts'
inzightts(x, var_name = NULL, pivot_longer = FALSE, ...)

## S3 method for class 'tbl_ts'
inzightts(x, ...)

Arguments

x  A data.frame, ts, tibble, or path.
...
Additional arguments to be passed to or from methods.
stringsAsFactors
See read.csv
as.is
See read.csv
var
The column number or name in data representing the observations used in the actual time series.
index
The column number or name in data containing the time variable.
key
The variable(s) that uniquely determine time indices.
start
The time of the first observation. It can be a single number or a vector of two integers representing a natural time unit and a (1-based) number of samples into the time unit.
end
The time of the last observation, specified in the same way as start.
freq
The number of observations per unit of time.
var_name
The new name for the variable column of the univariate time series, applicable only if x is not an mts object.
pivot_longer
Logical; set to TRUE to transform data to a "longer" form, otherwise keep the current form. Applicable only if x is an mts object.

Details

If a ts object is used to create the inzightts object, all the domain information is extracted from that object.
The index parameter should be a character, Date, yearweek, yearmonth, or yearquarter object.
If index is a character, the function recognizes the following time variable formats without case sensitivity:

• "(Y)yyyy": annually data, e.g., "(Y)1991"
• "(Y)yyyyMmm": monthly data, e.g., "(Y)1991M01"
• "(Y)yyyyQqq": quarterly data, e.g., "(Y)1991Q01"
• "(Y)yyyyWww": weekly data with yearly seasonality, e.g., "(Y)1991W01"
• "(Y)yyyyDdd": daily data with yearly seasonality, e.g., "(Y)1991D01"
• "WwwDdd": daily data with weekly seasonality, e.g., "W01D01"
• "DddHhh": hourly data with daily seasonality, e.g., "D01H01"

The length of digits of each time unit could be flexible, and spaces between the time unit are allowed.
In case data is a data.frame or path to a .csv file, and start is omitted, the starting date and the freq are extracted from the column that includes the time information. This column is either named "Time" or is the first column. If end is omitted, all of the data will be used for the time-series.
Value

An `inzightts` object, a sub-class of tsibble, which includes the index variable, temporal variable, and, if applicable, relevant keys.

See Also

`tsibble`, `as_tsibble` and `new_tsibble`

Examples

```r
# create from a ts object
z <- inzightts(UKgas)
## Not run:
plot(z)
## End(Not run)

# create from a data.frame
x <- inzightts(
  data.frame(Return = rnorm(100), Time = 1900:1999),
  var = "Return"
)
# or specify a time column
x <- inzightts(
  data.frame(Return = rnorm(100), Year = 1900:1999),
  var = "Return", index = "Year"
)
# create from a data.frame with modified time frame
y <- inzightts(
  data.frame(Return = rnorm(100)),
  start = c(1990, 1), end = c(1993, 5), freq = 12, var = 1
)
## Not run:
plot(y)
## End(Not run)
```

log_if

Apply logarithmic transformation

Description

Log-transforms the input x if `mult_fit` is TRUE; otherwise, returns the original input x unchanged.

Usage

`log_if(x, mult_fit)`
Arguments

- **x**: A numeric vector to be transformed.
- **mult_fit**: Logical; set to TRUE to apply logarithmic transformation, and FALSE to keep the original input.

Value

A numeric vector after applying the logarithmic transformation (if mult_fit = TRUE); otherwise, it returns the original input.

See Also

- `new_transformation`

Examples

```r
x <- runif(1e4, 1, 100)
all.equal(log_if(x, TRUE), log(x))
all.equal(log_if(x, FALSE), x)
```

---

**plot.inz_ts**

*Draw a simple time series plot*

**Description**

Draws a plot of a given `inzightts` (`inz_ts`) object with the trend superimposed.

**Usage**

```r
## S3 method for class 'inz_ts'
plot(
  x,
  var = NULL,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  xlim = NULL,
  aspect = NULL,
  compare = TRUE,
  pal = NULL,
  smoother = TRUE,
  sm_model = "stl",
  t = 0,
  mult_fit = FALSE,
  emphasise = NULL,
  non_emph_opacity = 0.2,
```

---
show_iso_obs = TRUE,
iso_obs_size = 1,
seasonal_adjustment = FALSE,
... )

Arguments

x   An inzightts (inz_ts) object representing the time series.
var A character vector specifying the variable(s) to be plotted, or set to NULL.
xlab A title for the x-axis of the plot.
ylab A title for the y-axis of the plot.
title A title for the graph.
xlim Axis limits, specified as dates or years.
aspect The aspect ratio of the plot; it will be about aspect times wider than it is high.
compare Logical; set to TRUE to plot the key levels in a single plot.
pal (Only if a categorical variable is passed to var): The colour palette for the categorical plot. The palette vector should be in the same order per the rows of tsibble::key_data(x).
smoother Logical; if TRUE, the smoother will be drawn.
sm_model The smoothing method to be used.
t The smoothing parameter (between 0 and 100).
mult_fit Logical; set to TRUE for a multiplicative model, or FALSE for the default additive model.
emphasise Integer vector to specify the key level(s) to focus in the plot. The integer maps to the specific key level(s) corresponding to the ith row of tsibble::key_data(x).
non_emph_opacity Numeric. If (0, 1], this argument determines the opacity of the series other than the focused one(s) (to highlight the focused series). If non_emph_opacity = 0, the plot draws the focused series in its own scales.
show_iso_obs Logical; set to TRUE to plot isolated observations between time series gaps (if any).
iso_obs_size Numeric; scaling the size of isolated observations, if show_iso_obs = TRUE and they exist.
seasonal_adjustment Logical; set to TRUE to show the seasonally adjusted time series (i.e., removed the estimated seasonal effects as determined by STL decomposition; see decomp()).
...

Value

A time series plot (constructed with ggplot2) is returned, which can be added to if desired.
predict.inz.ts

Forecast future observations

Description

Generates future predictions of the time series from an inzightts object. The output object includes predicted means, prediction intervals, raw data, and fitted values.

Usage

## S3 method for class 'inz_ts'
predict(
  object,
  var = NULL,
  h = 8,
  mult_fit = FALSE,
  pred_model = "auto",
  confint_width = 0.95,
  model_range = NULL,
  ...
)

## S3 method for class 'inz_frct'
plot(x, t_range = NULL, xlab = NULL, ylab = NULL, title = NULL, ...)

Arguments

object An inzightts object representing the time series.

var A character vector specifying the variable(s) to forecast, or set to NULL to forecast all variables.
The forecast horizon, either the number of observations to predict, or a character string specifying the time interval to predict (e.g., "2 years").

**mult_fit** Logical; set to TRUE for a multiplicative model, or FALSE for the default additive model.

**pred_model** The name of a fable model function or "auto".

**confint_width** A decimal representing the width of the prediction interval.

**model_range** The range of data to be used for fitting forecasts, specified as dates or years.

**x** An inz_frct object containing the forecasts.

**t_range** The range of data to be plotted, specified as dates or years.

**xlab** A title for the x-axis of the plot.

**ylab** A title for the y-axis of the plot.

**title** A title for the graph.

**Value**

An inz_frct object containing the forecasts.

**See Also**

fable-package

**Examples**

```r
t <- inzightts(visitorsQ, var = c(2, 4))
## The following two examples are equivalent
pred <- predict(t, names(t)[-1], h = "2 years")
pred <- predict(t, names(t)[-1], h = 8)

## Not run:
plot(pred)

## End(Not run)
```

**Description**

A dataset containing sea ice measurements from 1990 to 2011.

**Usage**

seaice
seasonplot

Format

A data frame with 265 rows and 3 variables:

- **Time**: The time variable
- **Arctic**: Sea ice measurement for the Arctic
- **Antarctica**: Sea ice measurement for Antarctica

seasonplot  
*Plot seasonal subseries from a time series*

Description

This function plots the seasonal components of a time series together with the estimated seasonal effects of that series.

Usage

```r
seasonplot(x, ...)
```

Arguments

- **x**: An ‘inzightts’ (‘inz_ts’) object representing the time series.
- **...**: Further arguments to be passed onto specific methods and the ‘gg_season’ function.

Details

The resulting window will contain two plots. On the left, every seasonal subseries of the time series is plotted. On the right will be the average seasonal effect of the series.

Value

A ‘patchwork’ object of seasonal plots.

See Also

- `gg_season`

Examples

```r
## Not run:
seasonplot(inzightts(visitorsQ))

## End(Not run)
```
subseries

Subseries Plots for inzights

Description

Time series subseries plot by seasonal period

Usage

subseries(
  x,
  var = NULL,
  show_mean = TRUE,
  xlab = NULL,
  ylab = NULL,
  title = NULL
)

Arguments

x | A time series object represented by an inz_ts or tbl_ts object.
var | A character vector specifying the variable(s) to be plotted, or set to NULL to plot all variables.
show_mean | Logical; set to FALSE to exclude the mean line from the plot.
xlab | A title for the x-axis of the plot.
ylab | A title for the y-axis of the plot.
title | A title for the graph.

Details

Plots seasonal subseries of a time series represented by an inz_ts or tbl_ts object. Each subseries represents one seasonal period.

Value

A ggplot object of the seasonal subseries plot.

See Also

gg_subseries
Examples

```r
t <- inzightts(visitorsQ)
## Not run:
subseries(t)
## End(Not run)
```

**summary.inz_frct**

**Summarise iNZightTS forecasts**

**Description**

Summary method for objects of class `inz_frct`.

**Usage**

```r
## S3 method for class 'inz_frct'
summary(object, var = NULL, ...)

## S3 method for class 'summary_inz_frct'
print(x, show_details = FALSE, ...)
```

**Arguments**

- `object`: An `inz_frct` object representing the forecasts.
- `var`: A character vector specifying the variable to summarize, or set to `NULL` to summarize all variables.
- `...`: Additional arguments (ignored).
- `x`: A `summary_inz_frct` object containing forecast summaries.
- `show_details`: Logical; set to `TRUE` to show model details only when `pred_model` is an "ARIMA" model.

**Value**

A `summary_inz_frct` object containing the first few forecast observations, the forecasting model used, and its details (such as call, coefficients, and goodness of fit statistics).

**See Also**

`predict.inz_ts`
Examples

```r
ts <- inzightts(visitorsQ, var = 2:5)
p <- predict(ts, "Japan")
s <- summary(p, "Japan")
s
print(s, show_details = TRUE)
```

---

**visitorsA2**

### Visitors (annual)

**Description**

A dataset containing annual visitor numbers for several countries.

**Usage**

```r
visitorsA2
```

**Format**

A data frame with 13 rows and 5 variables:

- **Time**  The time variable (year)
- **Australia**  Visitor counts for Australia
- **China..People.s.Republic.of**  Visitor counts for China
- **Japan**  Visitor counts for Japan
- **United.Kingdom**  Visitor counts for the UK

---

**visitorsM2**

### Visitors (monthly)

**Description**

A dataset containing monthly visitor numbers for several countries.

**Usage**

```r
visitorsM2
```
Format

A data frame with 164 rows and 5 variables:

- **Time**  The time variable (year/month)
- **Australia**  Visitor counts for Australia
- **China..People.s.Republic.of**  Visitor counts for China
- **Japan**  Visitor counts for Japan
- **United.Kingdom**  Visitor counts for the UK

---

**visitorsQ**  
**Visitors (quarterly)**

Description

A dataset containing quarterly visitor numbers for several countries.

Usage

**visitorsQ**

Format

A data frame with 54 rows and 5 variables:

- **Date**  The time variable (year/quarter)
- **Australia**  Visitor counts for Australia
- **China..People.s.Republic.of**  Visitor counts for China
- **Japan**  Visitor counts for Japan
- **United.Kingdom**  Visitor counts for the UK
Index

* **datasets**
  - seaice, 11
  - visitorsA2, 15
  - visitorsM2, 15
  - visitorsQ, 16

* **iNZight**
  - iNZightTS-package, 2

* **timeseries**
  - plot.inz_ts, 8
    - as_tsibble, 7
    - dable, 4
    - decomp, 3
    - gg_season, 12
    - gg_subseries, 13
    - ggplotable, 4
    - ggplotly, 5
    - inzightts, 2, 5
    - iNZightTS-package, 2

  - key_data, 10
  - log_if, 7
  - new_transformation, 8
  - new_tsibble, 7
  - plot.inz_dcmp(decomp), 3
  - plot.inz_frct(predict.inz_ts), 10
  - plot.inz_ts, 8
  - predict.inz_ts, 10, 14
  - print.summary_inz_frct
    - (summary.inz_frct), 14
  - read.csv, 6

  - seaice, 11
  - seasonplot, 12

  - subseries, 13
  - summary.inz_frct, 14
  - tsibble, 7
  - visitorsA2, 15
  - visitorsM2, 15
  - visitorsQ, 16