Package ‘iNZightTS’

May 21, 2021

Type Package

Title Time Series for ‘iNZight’

Version 1.5.8

Depends R (>= 3.2)

Imports colorspace, dplyr, forcats, ggplot2, ggtext, glue, graphics, grDevices, grid, gridExtra, magrittr, methods, patchwork, rlang, stats, tidyr, utils

Suggests covr, testthat


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

Contact inzight_support@stat.auckland.ac.nz

URL http://inzight.nz

LazyData true

License GPL-3

Encoding UTF-8

Language en-GB

RoxygenNote 7.1.1

NeedsCompilation no

Author Tom Elliott [aut, cre] (https://orcid.org/0000-0002-7815-6318), Junjie Zeng [aut], Simon Potter [aut], David Banks [aut], Marco Kuper [aut], Dongning Zhang [ctb]
Description

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

iNZight
**compareplot**

*Comparison plot - depreciated*

**Description**

Comparison plot - depreciated

**Usage**

```r
compareplot(x, ...)
```

**Arguments**

- `x` an `iNZightTS` object
- `...` additional arguments passed to `plot()`

**Value**

No return value, called for the side effect of drawing a plot.

---

**decompose**

*Decompose a time series object*

**Description**

Decompose a time series object

**Usage**

```r
decompose(
  obj,
  multiplicative = FALSE,
  t = 10,
  model.lim = NULL,
  data.name = NULL,
  ...
)
```

```r
## S3 method for class 'inzdecomp'
plot(
  x,
  recompose.progress = c(0, 0),
  recompose = any(recompose.progress > 0),
  ylab = x$currVar,
  xlab = "Date",
```
title = NULL,
xlim = c(NA, NA),
colour = c("#1B9E46", "#45a8ff", "orangered"),
...
)

Arguments

- **obj**: an iNZightTS object
- **multiplicative**: fit a multiplicative time series model?
- **t**: the smoothing parameter
- **model.lim**: limits for the time series model
- **data.name**: the name of the data
- **...**: additional arguments (ignored)
- **x**: an inzdecomp object (from decompose(ts))
- **recompose.progress**: if recompose is TRUE, this shows how much to show (for animation!). Length 2 numeric: the first is 0 for seasonal, and 1 for residual; second component is how many observations have been recomposed so far
- **recompose**: logical as to whether the recomposition is shown or not
- **ylab**: the label for the y axis
- **xlab**: the label for the x axis
- **title**: the title for the plot
- **xlim**: the x axis limits
- **colour**: vector of three colours for trend, seasonal, and residuals, respectively

Value

- an inzdecomp object (this is the original object with an additional decompVars component)
  
  Invisibly returns the original decomposition object. Mainly called to plot the decomposition.

Methods (by generic)

- plot: Plot a time series decomposition

References


Examples

t <- iNZightTS(visitorsQ)
decomp.ts <- decompose(t, data.name = "Visitors")
plot(decomp.ts)
decompositionplot

Plot a Time Series Decomposition

Description

Decomposes a time series into trend, seasonal and residual components using loess.

Usage

decompositionplot(...)

Arguments

... additional arguments, ignored

Details

If the frequency is greater than 1, the components are found using the stl function with s.window set to TRUE (effectively replacing smoothing by taking the mean). If the frequency is 1, the trend component is found directly by using loess and the residuals are the difference between trend and actual values. The trend, seasonal and residual components are plotted on the same scale allowing for easy visual analysis.

Value

The original iNZightTS object with an item decompVars appended, containing results from the decomposition.

References


See Also

stl, loess, iNZightTS
**forecastplot**  
*Forecast plot - DEPRECATED*

**Description**
Plot a raw time series together with it’s fitted curve and add forecasts and prediction intervals to the end.

**Usage**
```
forecastplot(x, ...)
```

**Arguments**
- `x` : iNZightTS object
- `...` : additional arguments passed on

**Details**
The predictions and prediction intervals are the result of models fitted by the Holt-Winters method. The amount of predicted observations is calculated by $2 \times freq$, where `freq` is the frequency of the time series object.

**Value**
Called for the side effect of drawing a plot. The constructed `ggplot` object is returned invisibly.

---

**iNZightTS**  
*iNZightTS (Time-Series) Objects*

**Description**
The function `iNZightTS` is used to create time-series objects used in iNZight.

**Usage**
```
iNZightTS(
  data,
  start = 1,
  end,
  freq = 1,
  var = 2,
  time.col = grep("time", names(data), ignore.case = TRUE)[1],
  ...
)
```
iNZightTS

Arguments

data                     a data.frame containing time information and observation or a path to a .csv file with such information or a ts object
start                    the time of the first observation. Either a single number or a vector of two integers, which specify 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                      the column number or name for the observations used from data in the actual time series
time.col                 which column contains the time variable
...                      additional information passed to read.csv() and used when data is a path
ignore.case              logical, ignore the case?

Details
The function iNZgithTS is used to create time-series objects. Unlike ts objects, these are lists containing information about the time-series as well as the data and the time-series (ts object) itself.

If a ts object is used to create the iNZightTS object, all the domain information is extracted from that object.

The function recognises the following time variable formats without case sensitive:

• "(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 allowing space between the time unit

In case of data being a data.frame or path to a .csv file and start being omitted, the starting date and the freq is 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

a iNZightTS object. If multiple variables are requested, the iNZightMTS class is added to the result. The result object contains the original data as a time series object, as well as information on the series start, end, and frequency.
See Also

ts, print.iNZightTS,

Examples

# create from a ts object
z <- iNZightTS(UKgas)
plot(z)

# 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", time.col = "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)
plot(y)

multiseries

Compare multiple time series - DEPRECATED

Description

Compare multiple time series - DEPRECATED

Usage

multiseries(x, ...)

Arguments

x iNZightMTS object containing data
...
Further arguments to be passed to `plot()`

Value

No return value, called for the side effect of drawing a plot.
plot.iNZightMTS  

Plot multiple time series

Description

Plot a multiple time series object to compare several series

Usage

```r
## S3 method for class 'iNZightMTS'
plot(
  x,
  compare = TRUE,
  multiplicative = FALSE,
  ylab = "Value",
  xlab = "Date",
  title = "%var",
  t = 10,
  smoother = TRUE,
  aspect = 2,
  xlim = c(NA, NA),
  model.lim = NULL,
  ...
)
```

Arguments

- **x**: Multiple time series object
- **compare**: logical, if true, the series will be graphed in a single plot; otherwise graphed in individual rows
- **multiplicative**: logical, if TRUE multiplicative series will be used; otherwise additive
- **ylab**: y axis label
- **xlab**: x axis label
- **title**: the title for the plot
- **t**: smoothing parameter
- **smoother**: logical, if TRUE the smoother will be drawn
- **aspect**: aspect ratio (width:height) for the time series
- **xlim**: limits to control how much of series is shown
- **model.lim**: time limits to use for modelling
- **...**: additional arguments

Value

No return value, called for the side effect of drawing a plot.
Author(s)
Tom Elliott

Examples
```r
tm <- iNZightTS(visitorsQ, var = 2:5)
plot(tm)
plot(tm, compare = FALSE)
```

plot.iNZightTS

Draw a simple time series plot

Description
Draws a plot of a given iNZightTS object with the trend superimposed.

Usage
```r
## S3 method for class 'iNZightTS'
plot(
  x,
  multiplicative = FALSE,
  ylab = obj$currVar,
  xlab = "Date",
  title = "%var",
  animate = FALSE,
  t = 10,
  smoother = TRUE,
  aspect = 3,
  plot = TRUE,
  col = ifelse(forecast > 0, "#0e8c07", "red"),
  xlim = c(NA, NA),
  model.lim = NULL,
  seasonal.trend = FALSE,
  forecast = 0,
  ...
)
```

Arguments
- `x`: an iNZightTS object
- `multiplicative`: logical. If TRUE, a multiplicative model is used, otherwise an additive model is used by default.
- `ylab`: a title for the y axis
- `xlab`: a title for the x axis
plot.iNZightTS

- **title**: a title for the graph
- **animate**: logical, if true the graph is animated
- **t**: smoothing parameter
- **smoother**: logical, if TRUE the smoother will be drawn
- **aspect**: the aspect ratio of the plot; it will be about ASPECT times wider than it is high
- **plot**: logical, if FALSE, the graph isn’t drawn
- **col**: the colour of the smoothed trend line
- **xlim**: axis limits, specified as dates
- **model.lim**: limits of the series to use for modelling/forecast
- **seasonal.trend**: logical, if TRUE seasonal+trend curve added
- **forecast**: numeric, how many observations ahead to forecast (default is 0, no forecast)
- ... additional arguments (not used)

**Details**

If animate is set to TRUE, a scatterplot of all points in the time series will appear followed by slowly drawn lines connecting the points, simulating the drawing of a time series by hand.

**Value**

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

**Forecast**

The predictions and prediction intervals are the result of models fitted by the Holt-Winters method. The amount of predicted observations is specified by the value of ‘forecast’.

**References**


**Examples**

```r
t <- iNZightTS(visitorsQ)
plot(t)

# Forecast plot (8 quarterly forecasts):
plot(t, forecast = 8)
```
pred

**Get forecast prediction values**

**Description**
Get forecast prediction values

**Usage**
```
pred(x)
```

**Arguments**
- `x` the forecast object (a plot with predictions)

**Value**
a time series forecasts object

**print.iNZightTS**
*Print an iNZightTS object*

**Description**
Print method for iNZightTS (time series) objects.

**Usage**
```
## S3 method for class 'iNZightTS'
print(x, full = FALSE, ...)
```

**Arguments**
- `x` the iNZightTS object to be printed
- `full` whether to print all the underlying data
- `...` Unused arguments. Only here for consistency with the base S3 method.

**Details**
The `full` argument controls whether to print all the data from which the iNZightTS object has been created. The default is set to `FALSE` and only the `head()` of the data will be printed.

**Value**
No return value, called for side effect.
rawplot

See Also
print.iNZightTS

Examples
iNZightTS(UKgas)

---

rawplot  
*Time series plot - depreciated*

---

**Description**

Time series plot - depreciated

**Usage**

```
rawplot(...)  
```

**Arguments**

... arguments passed to `plot` method

**Value**

Called to draw a plot. Invisibly returns a `ggplot` object.

---

recompose  
*Recompose a decomposed time series*

---

**Description**

Recompose a time series object, with optional animation.

**Usage**

```
recompose(...)  
```

**Arguments**

... additional arguments, ignored

**Value**

the recomposed series

**Author(s)**

iNZight
**sealice**

*Sea Ice*

**Description**

A dataset containing sea ice measurements from 1990 to 2011.

**Usage**

`sealice`

**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**

`seasonplot(obj, ...)`

**Arguments**

- `obj`: an `iNZightTS` object
- `...`: Further arguments to be passed onto specific methods.

**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**

No return value, called for the side effect of drawing a plot.
visitorsA2

See Also

dNZightTS

Examples

ts <- iNZightTS(visitorsQ)
seasonplot(ts)

visitorsA2  Visitors (annual)

Description

A dataset containing annual visitor numbers for several countries.

Usage

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

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

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, 14
  visitorsA2, 15
  visitorsM2, 15
  visitorsQ, 16
* iNZight
  iNZightTS-package, 2
* timeseries
  plot.iNZightTS, 10

compareplot, 3
decompose, 3
decompositionplot, 5
forecastplot, 6
iNZightMTS (iNZightTS), 6
iNZightTS, 5, 6, 13, 15
iNZightTS-package, 2
loess, 5
multiseries, 8
plot.inzdecomp (decompose), 3
plot.iNZightMTS, 9
plot.iNZightTS, 10
pred, 12
print, 13
print.iNZightTS, 8, 12
rawplot, 13
recompose, 13
seaice, 14
seasonplot, 14
stl, 5
ts, 8
visitorsA2, 15
visitorsM2, 15
visitorsQ, 16