

Package ‘sweep’

August 28, 2025

Title Tidy Tools for Forecasting

Version 0.2.6

Description Tidies up the forecasting modeling and prediction work flow, extends the 'broom' package with 'sw_tidy', 'sw_glance', 'sw_augment', and 'sw_tidy_decomp' functions for various forecasting models, and enables converting 'forecast' objects to ``tidy'' data frames with 'sw_sweep'.

URL <https://business-science.github.io/sweep/>,
<https://github.com/business-science/sweep>

BugReports <https://github.com/business-science/sweep/issues>

License GPL (>= 3)

Encoding UTF-8

LazyData true

Depends R (>= 3.3.0)

Imports broom (>= 0.5.6), dplyr (>= 1.0.0), forecast (>= 8.0), rlang,
tibble (>= 1.2), timetk (>= 2.1.0)

Suggests fracdiff, ggplot2, knitr, lubridate, purrr, readr, rmarkdown,
scales, stringr, testthat (>= 2.0.0), tidyr (>= 1.0.0),
tidyquant, zoo

RoxygenNote 7.3.2

VignetteBuilder knitr

Config/testthat/edition 2

NeedsCompilation no

Author Matt Dancho [aut, cre],
Davis Vaughan [aut]

Maintainer Matt Dancho <mdancho@business-science.io>

Repository CRAN

Date/Publication 2025-08-28 19:00:02 UTC

Contents

add_index	2
arima_string	3
bats_string	3
bike_sales	3
sw_augment	4
sw_augment.default	5
sw_augment_columns	6
sw_glance	6
sw_glance.default	7
sw_sweep	7
sw_tidy	9
sw_tidy.default	10
sw_tidy_decomp	10
tbats_string	11
tidiers_arima	11
tidiers_bats	13
tidiers_decomposed_ts	15
tidiers_ets	16
tidiers_HoltWinters	18
tidiers_nnetar	20
tidiers_stl	22
tidiers_StructTS	24
validate_index	26

Index	27
-------	----

add_index	<i>Adds a sequential index column to a data frame</i>
-----------	---

Description

Adds a sequential index column to a data frame

Usage

add_index(ret, rename_index)

Arguments

- ret An object of class tibble
- rename_index A variable indicating the index name to be used in the tibble returned

arima_string	<i>Print the ARIMA model parameters</i>
--------------	---

Description

Refer to forecast:::arima.string. [forecast arima.R](#)

Usage

```
arima_string(object, padding = FALSE)
```

Arguments

object	An object of class Arima
padding	Add padding to the name returned

bats_string	<i>Print the BATS model parameters</i>
-------------	--

Description

Refer to forecast:::makeText. [forecast bats.R](#)

Usage

```
bats_string(object)
```

Arguments

object	An object of class bats
--------	-------------------------

bike_sales	<i>Fictional sales data for bike shops purchasing Cannondale bikes</i>
------------	--

Description

A dataset containing the fictional bicycle orders spanning 2011 through 2015. Hypothetically, the bike_sales data are similar to sales data maintained in a business' sales data base. The unit price and model names come from data provided by model for the bicycle manufacturer, Cannondale (2016). The customers (bicycle shops) including name, location, etc and the orders including quantity purchased and order dates are fictional. The data is intended for implementing business analytics techniques (e.g. forecast, clustering, etc) to identify underlying trends.

Usage

```
bike_sales
```

Format

A data frame with 15644 rows and 17 variables:

order.date Date the order was placed
order.id A unique order identification number
order.line The sequential identification number for products on an order
quantity Number of units purchased
price The unit price of the bicycle
price.ext The extended price = price x quantity
customer.id A unique customer identification number
bikeshop.name The customer name
bikeshop.city The city that the bike shop is located
bikeshop.state The state that the bike shop is located
latitude The geographic latitude of the customer location
longitude The geographic longitude of the customer location
product.id A unique product identification number
model The model name of the bicycle
category.primary The main bicycle category, either "Mountain" or "Road"
category.secondary One of nine more specific bicycle categories
frame The bicycle frame material, either "Carbon" or "Aluminum"

Source

The 2016 bicycle model names and prices originated from <https://www.cannondale.com/en-us>

```
sw_augment
```

```
Augment data according to a tidied model
```

Description

Given an R statistical model or other non-tidy object, add columns to the original dataset such as predictions, residuals and cluster assignments.

Usage

```
sw_augment(x, ...)
```

Arguments

x	model or other R object to convert to data frame
...	other arguments passed to methods

Details

sw_augment() is a wrapper for `broom::augment()`. The benefit of sw_augment is that it has methods for various time-series model classes such as HoltWinters, ets, Arima, etc.

For non-time series, sw_augment() defaults to `broom::augment()`. The only difference is that the return is a tibble.

Note that by convention the first argument is almost always data, which specifies the original data object. This is not part of the S3 signature, partly because it prevents rowwise_df_tidiers from taking a column name as the first argument.

See Also

`broom::augment()`

sw_augment.default	<i>Default augment method</i>
--------------------	-------------------------------

Description

By default, sw_augment() uses `broom::augment()` to convert its output.

Usage

```
## Default S3 method:  
sw_augment(x, ...)
```

Arguments

x	an object to be tidied
...	extra arguments passed to <code>broom::augment()</code>

Value

A tibble generated by `broom::augment()`

sw_augment_columns	<i>Augments data</i>
--------------------	----------------------

Description

Augments data

Usage

```
sw_augment_columns(ret, data, rename_index, timetk_idx = FALSE)
```

Arguments

ret	An object of class tibble
data	Any time series data that is to be augmented
rename_index	A variable indicating the index name to be used in the tibble returned
timetk_idx	Uses the timetk index (irregular time index) if present.

sw_glance	<i>Construct a single row summary "glance" of a model, fit, or other object</i>
-----------	---

Description

Construct a single row summary "glance" of a model, fit, or other object

Usage

```
sw_glance(x, ...)
```

Arguments

x	model or other R object to convert to single-row data frame
...	other arguments passed to methods

Details

sw_glance() is a wrapper for [broom::glance\(\)](#). The benefit of sw_glance is that it has methods for various time-series model classes such as HoltWinters, ets, Arima, etc. sw_glance methods always return either a one-row tibble or NULL. The single row includes summary statistics relevant to the model accuracy, which can be used to assess model fit and quality.

For non-time series, sw_glance() defaults to [broom::glance\(\)](#). The only difference is that the return is a tibble.

Value

single-row tibble with model summary information.

See Also

[broom::glance\(\)](#)

sw_glance.default	<i>Default glance method</i>
-------------------	------------------------------

Description

By default, `sw_glance()` uses [broom::glance\(\)](#) to convert its output.

Usage

```
## Default S3 method:  
sw_glance(x, ...)
```

Arguments

x	an object to be tidied
...	extra arguments passed to broom::glance()

Value

A tibble generated by [broom::glance\(\)](#)

sw_sweep	<i>Tidy forecast objects</i>
----------	------------------------------

Description

Tidy forecast objects

Usage

```
sw_sweep(x, fitted = FALSE, timetk_idx = FALSE, rename_index = "index", ...)
```

Arguments

<code>x</code>	A time-series forecast of class <code>forecast</code> .
<code>fitted</code>	Whether or not to return the fitted values (model values) in the results. <code>FALSE</code> by default.
<code>timetk_idx</code>	If <code>timetk</code> index (non-regularized index) is present, uses it to develop forecast. Otherwise uses default index.
<code>rename_index</code>	Enables the index column to be renamed.
<code>...</code>	Additional arguments passed to <code>timetk::tk_make_future_timeseries()</code>

Details

`sw_sweep` is designed to coerce forecast objects from the `forecast` package into tibble objects in a "tidy" format (long). The returned object contains both the actual values and the forecasted values including the point forecast and upper and lower confidence intervals.

The `timetk_idx` argument is used to modify the return format of the index.

- If `timetk_idx = FALSE`, a regularized time index is always constructed. This may be in the format of numeric values (e.g. 2010.000) or the higher order `yearmon` and `yearqtr` classes from the `zoo` package. A higher order class is attempted to be returned.
- If `timetk_idx = TRUE` and a `timetk` index is present, an irregular time index will be returned that combines the original time series (i.e. date or datetime) along with a computed future time series created using `timetk::tk_make_future_timeseries()` from the `timetk` package. The `...` can be used to pass additional arguments to `timetk::tk_make_future_timeseries()` such as `inspect_weekdays`, `skip_values`, etc that can be useful in tuning the future time series sequence.

The index column name can be changed using the `rename_index` argument.

Value

Returns a tibble object.

See Also

`timetk::tk_make_future_timeseries()`

Examples

```
library(forecast)
library(dplyr)

# ETS forecasts
USAccDeaths %>%
  ets() %>%
  forecast(level = c(80, 95, 99)) %>%
  sw_sweep()
```

sw_tidy*Tidy the result of a time-series model into a summary tibble*

Description

Tidy the result of a time-series model into a summary tibble

Usage

```
sw_tidy(x, ...)
```

Arguments

x	An object to be converted into a tibble ("tidy" data.frame)
...	extra arguments

Details

sw_tidy() is a wrapper for `broom::tidy()`. The main benefit of sw_tidy() is that it has methods for various time-series model classes such as HoltWinters, ets, Arima, etc. sw_tidy() methods always returns a "tidy" tibble with model coefficient / parameters.

For non-time series, sw_tidy() defaults to `broom::tidy()`. The only difference is that the return is a tibble. The output of sw_tidy() is always a tibble with disposable row names. It is therefore suited for further manipulation by packages like dplyr and ggplot2.

Value

a tibble

See Also

`broom::tidy()`

Examples

```
library(dplyr)
library(forecast)

WWWusage %>%
  auto.arima() %>%
  sw_tidy(conf.int = TRUE)
```

sw_tidy.default	<i>Default tidying method</i>
-----------------	-------------------------------

Description

By default, `sw_tidy()` uses `broom::tidy()` to convert its output.

Usage

```
## Default S3 method:
sw_tidy(x, ...)
```

Arguments

<code>x</code>	an object to be tidied
<code>...</code>	extra arguments passed to <code>broom::tidy()</code>

Value

A tibble generated by `broom::tidy()`

sw_tidy_decomp	<i>Coerces decomposed time-series objects to tibble format.</i>
----------------	---

Description

Coerces decomposed time-series objects to tibble format.

Usage

```
sw_tidy_decomp(x, timetk_idx = FALSE, rename_index = "index", ...)
```

Arguments

<code>x</code>	A time-series object of class <code>stl</code> , <code>ets</code> , <code>decomposed.ts</code> , <code>HoltWinters</code> , <code>bats</code> or <code>tbats</code> .
<code>timetk_idx</code>	When TRUE, uses a timetk index (irregular, typically date or datetime) if present.
<code>rename_index</code>	Enables the index column to be renamed.
<code>...</code>	Not used.

Details

`sw_tidy_decomp` is designed to coerce time-series objects with decompositions to tibble objects.

A regularized time index is always constructed. If no time index is detected, a sequential index is returned as a default. The index column name can be changed using the `rename_index` argument.

Value

Returns a tibble object.

Examples

```
library(dplyr)
library(forecast)
library(sweep)

# Decompose ETS model
USAccDeaths %>%
  ets() %>%
  sw_tidy_decomp()

# Decompose STL object
USAccDeaths %>%
  stl(s.window = 'periodic') %>%
  sw_tidy_decomp()
```

tbats_string

Print the TBATS model parameters

Description

Refer to `forecast::makeTextTBATS`. `forecast bats.R`

Usage

```
tbats_string(object)
```

Arguments

object An object of class bats or tbats

tidiers_arima

Tidying methods for ARIMA modeling of time series

Description

These methods tidy the coefficients of ARIMA models of univariate time series.

Usage

```
## S3 method for class 'Arima'
sw_tidy(x, ...)

## S3 method for class 'Arima'
sw_glance(x, ...)

## S3 method for class 'Arima'
sw_augment(x, data = NULL, rename_index = "index", timetk_idx = FALSE, ...)

## S3 method for class 'stlm'
sw_tidy(x, ...)
```

Arguments

x	An object of class "Arima"
...	Additional parameters (not used)
data	Used with sw_augment only. NULL by default which simply returns augmented columns only. User can supply the original data, which returns the data + augmented columns.
rename_index	Used with sw_augment only. A string representing the name of the index generated.
timetk_idx	Used with sw_augment only. Uses a irregular timetk index if present.

Value

sw_tidy() returns one row for each coefficient in the model, with five columns:

- term: The term in the nonlinear model being estimated and tested
- estimate: The estimated coefficient

sw_glance() returns one row with the columns

- model.desc: A description of the model including the three integer components (p, d, q) are the AR order, the degree of differencing, and the MA order.
- sigma: The square root of the estimated residual variance
- logLik: The data's log-likelihood under the model
- AIC: The Akaike Information Criterion
- BIC: The Bayesian Information Criterion
- ME: Mean error
- RMSE: Root mean squared error
- MAE: Mean absolute error
- MPE: Mean percentage error
- MAPE: Mean absolute percentage error
- MASE: Mean absolute scaled error

- ACF1: Autocorrelation of errors at lag 1

`sw_augment()` returns a tibble with the following time series attributes:

- `index`: An index is either attempted to be extracted from the model or a sequential index is created for plotting purposes
- `.actual`: The original time series
- `.fitted`: The fitted values from the model
- `.resid`: The residual values from the model

`sw_tidy()` returns the underlying ETS or ARIMA model's `sw_tidy()` one row for each coefficient in the model, with five columns:

- `term`: The term in the nonlinear model being estimated and tested
- `estimate`: The estimated coefficient

See Also

`stats::arima()`

Examples

```
library(dplyr)
library(forecast)

fit_arima <- WWWusage %>%
  auto.arima()

sw_tidy(fit_arima)
sw_glance(fit_arima)
sw_augment(fit_arima)
```

Description

Tidying methods for BATS and TBATS modeling of time series

Usage

```
## S3 method for class 'bats'
sw_tidy(x, ...)

## S3 method for class 'bats'
sw_glance(x, ...)
```

```
## S3 method for class 'bats'
sw_augment(x, data = NULL, rename_index = "index", timetk_idx = FALSE, ...)

## S3 method for class 'bats'
sw_tidy_decomp(x, timetk_idx = FALSE, rename_index = "index", ...)
```

Arguments

<code>x</code>	An object of class "bats" or "tbats"
<code>...</code>	Additional parameters (not used)
<code>data</code>	Used with <code>sw_augment</code> only. NULL by default which simply returns augmented columns only. User can supply the original data, which returns the data + augmented columns.
<code>rename_index</code>	Used with <code>sw_augment</code> only. A string representing the name of the index generated.
<code>timetk_idx</code>	Used with <code>sw_augment</code> and <code>sw_tidy_decomp</code> . When TRUE, uses a timetk index (irregular, typically date or datetime) if present.

Value

`sw_tidy()` returns one row for each model parameter, with two columns:

- `term`: The various parameters (lambda, alpha, gamma, etc)
- `estimate`: The estimated parameter value

`sw_glance()` returns one row with the columns

- `model.desc`: A description of the model including the three integer components (p, d, q) are the AR order, the degree of differencing, and the MA order.
- `sigma`: The square root of the estimated residual variance
- `logLik`: The data's log-likelihood under the model
- `AIC`: The Akaike Information Criterion
- `BIC`: The Bayesian Information Criterion (NA for bats / tbats)
- `ME`: Mean error
- `RMSE`: Root mean squared error
- `MAE`: Mean absolute error
- `MPE`: Mean percentage error
- `MAPE`: Mean absolute percentage error
- `MASE`: Mean absolute scaled error
- `ACF1`: Autocorrelation of errors at lag 1

`sw_augment()` returns a tibble with the following time series attributes:

- `index`: An index is either attempted to be extracted from the model or a sequential index is created for plotting purposes

- `.actual`: The original time series
- `.fitted`: The fitted values from the model
- `.resid`: The residual values from the model

`sw_tidy_decomp()` returns a tibble with the following time series attributes:

- `index`: An index is either attempted to be extracted from the model or a sequential index is created for plotting purposes
- `observed`: The original time series
- `level`: The level component
- `slope`: The slope component (Not always present)
- `season`: The seasonal component (Not always present)

See Also

[forecast::bats\(\)](#), [forecast::tbats\(\)](#)

Examples

```
library(dplyr)
library(forecast)

fit_bats <- WWWusage %>%
  bats()

sw_tidy(fit_bats)
sw_glance(fit_bats)
sw_augment(fit_bats)
```

`tidiers_decomposed_ts` *Tidying methods for decomposed time series*

Description

Tidying methods for decomposed time series

Usage

```
## S3 method for class 'decomposed.ts'
sw_tidy_decomp(x, timetk_idx = FALSE, rename_index = "index", ...)
```

Arguments

<code>x</code>	An object of class "decomposed.ts"
<code>timetk_idx</code>	Used with <code>sw_augment</code> and <code>sw_tidy_decomp</code> . When TRUE, uses a timetk index (irregular, typically date or datetime) if present.
<code>rename_index</code>	Used with <code>sw_augment</code> and <code>sw_tidy_decomp</code> . A string representing the name of the index generated.
<code>...</code>	Not used.

Value

`sw_tidy_decomp()` returns a tibble with the following time series attributes:

- `index`: An index is either attempted to be extracted from the model or a sequential index is created for plotting purposes
- `season`: The seasonal component
- `trend`: The trend component
- `random`: The error component
- `seasadj`: observed - season

See Also

[decompose\(\)](#)

Examples

```
library(dplyr)
library(forecast)

fit_decomposed <- USAccDeaths %>%
  decompose()

sw_tidy_decomp(fit_decomposed)
```

tidiers_ets

Tidying methods for ETS (Error, Trend, Seasonal) exponential smoothing modeling of time series

Description

Tidying methods for ETS (Error, Trend, Seasonal) exponential smoothing modeling of time series

Usage

```
## S3 method for class 'ets'
sw_tidy(x, ...)

## S3 method for class 'ets'
sw_glance(x, ...)

## S3 method for class 'ets'
sw_augment(x, data = NULL, timetk_idx = FALSE, rename_index = "index", ...)

## S3 method for class 'ets'
sw_tidy_decomp(x, timetk_idx = FALSE, rename_index = "index", ...)
```

Arguments

x	An object of class "ets"
...	Not used.
data	Used with <code>sw_augment</code> only. NULL by default which simply returns augmented columns only. User can supply the original data, which returns the data + augmented columns.
timetk_idx	Used with <code>sw_augment</code> and <code>sw_tidy_decomp</code> . When TRUE, uses a timetk index (irregular, typically date or datetime) if present.
rename_index	Used with <code>sw_augment</code> and <code>sw_tidy_decomp</code> . A string representing the name of the index generated.

Value

`sw_tidy()` returns one row for each model parameter, with two columns:

- term: The smoothing parameters (alpha, gamma) and the initial states (l, s0 through s10)
- estimate: The estimated parameter value

`sw_glance()` returns one row with the columns

- model.desc: A description of the model including the three integer components (p, d, q) are the AR order, the degree of differencing, and the MA order.
- sigma: The square root of the estimated residual variance
- logLik: The data's log-likelihood under the model
- AIC: The Akaike Information Criterion
- BIC: The Bayesian Information Criterion
- ME: Mean error
- RMSE: Root mean squared error
- MAE: Mean absolute error
- MPE: Mean percentage error
- MAPE: Mean absolute percentage error

- MASE: Mean absolute scaled error
- ACF1: Autocorrelation of errors at lag 1

`sw_augment()` returns a tibble with the following time series attributes:

- `index`: An index is either attempted to be extracted from the model or a sequential index is created for plotting purposes
- `.actual`: The original time series
- `.fitted`: The fitted values from the model
- `.resid`: The residual values from the model

`sw_tidy_decomp()` returns a tibble with the following time series attributes:

- `index`: An index is either attempted to be extracted from the model or a sequential index is created for plotting purposes
- `observed`: The original time series
- `level`: The level component
- `slope`: The slope component (Not always present)
- `season`: The seasonal component (Not always present)

See Also

[forecast::ets\(\)](#)

Examples

```
library(dplyr)
library(forecast)

fit_ets <- WWWusage %>%
  ets()

sw_tidy(fit_ets)
sw_glance(fit_ets)
sw_augment(fit_ets)
sw_tidy_decomp(fit_ets)
```

`tidiers_HoltWinters` *Tidying methods for HoltWinters modeling of time series*

Description

These methods tidy HoltWinters models of univariate time series.

Usage

```
## S3 method for class 'HoltWinters'
sw_tidy(x, ...)

## S3 method for class 'HoltWinters'
sw_glance(x, ...)

## S3 method for class 'HoltWinters'
sw_augment(x, data = NULL, rename_index = "index", timetk_idx = FALSE, ...)

## S3 method for class 'HoltWinters'
sw_tidy_decomp(x, timetk_idx = FALSE, rename_index = "index", ...)
```

Arguments

x	An object of class "HoltWinters"
...	Additional parameters (not used)
data	Used with <code>sw_augment</code> only. NULL by default which simply returns augmented columns only. User can supply the original data, which returns the data + augmented columns.
rename_index	Used with <code>sw_augment</code> only. A string representing the name of the index generated.
timetk_idx	Used with <code>sw_augment</code> and <code>sw_tidy_decomp</code> . When TRUE, uses a timetk index (irregular, typically date or datetime) if present.

Value

`sw_tidy()` returns one row for each model parameter, with two columns:

- `term`: The various parameters (alpha, beta, gamma, and coefficients)
- `estimate`: The estimated parameter value

`sw_glance()` returns one row with the following columns:

- `model.desc`: A description of the model
- `sigma`: The square root of the estimated residual variance
- `logLik`: The data's log-likelihood under the model
- `AIC`: The Akaike Information Criterion
- `BIC`: The Bayesian Information Criterion (NA for bats / tbats)
- `ME`: Mean error
- `RMSE`: Root mean squared error
- `MAE`: Mean absolute error
- `MPE`: Mean percentage error
- `MAPE`: Mean absolute percentage error
- `MASE`: Mean absolute scaled error

- ACF1: Autocorrelation of errors at lag 1

`sw_augment()` returns a tibble with the following time series attributes:

- `index`: An index is either attempted to be extracted from the model or a sequential index is created for plotting purposes
- `.actual`: The original time series
- `.fitted`: The fitted values from the model
- `.resid`: The residual values from the model

`sw_tidy_decomp()` returns a tibble with the following time series attributes:

- `index`: An index is either attempted to be extracted from the model or a sequential index is created for plotting purposes
- `observed`: The original time series
- `season`: The seasonal component
- `trend`: The trend component
- `remainder`: `observed - (season + trend)`
- `seasadj`: `observed - season (or trend + remainder)`

See Also

[stats::HoltWinters\(\)](#)

Examples

```
library(dplyr)
library(forecast)

fit_hw <- USAccDeaths %>%
  stats::HoltWinters()

sw_tidy(fit_hw)
sw_glance(fit_hw)
sw_augment(fit_hw)
sw_tidy_decomp(fit_hw)
```

Description

These methods tidy the coefficients of NNETAR models of univariate time series.

Usage

```
## S3 method for class 'nnetar'
sw_tidy(x, ...)

## S3 method for class 'nnetar'
sw_glance(x, ...)

## S3 method for class 'nnetar'
sw_augment(x, data = NULL, timetk_idx = FALSE, rename_index = "index", ...)
```

Arguments

<code>x</code>	An object of class "nnetar"
<code>...</code>	Additional parameters (not used)
<code>data</code>	Used with <code>sw_augment</code> only. NULL by default which simply returns augmented columns only. User can supply the original data, which returns the data + augmented columns.
<code>timetk_idx</code>	Used with <code>sw_augment</code> only. Uses a irregular timetk index if present.
<code>rename_index</code>	Used with <code>sw_augment</code> only. A string representing the name of the index generated.

Value

`sw_tidy()` returns one row for each model parameter, with two columns:

- `term`: The smoothing parameters (alpha, gamma) and the initial states (l, s0 through s10)
- `estimate`: The estimated parameter value

`sw_glance()` returns one row with the columns

- `model.desc`: A description of the model including the three integer components (p, d, q) are the AR order, the degree of differencing, and the MA order.
- `sigma`: The square root of the estimated residual variance
- `logLik`: The data's log-likelihood under the model (NA)
- `AIC`: The Akaike Information Criterion (NA)
- `BIC`: The Bayesian Information Criterion (NA)
- `ME`: Mean error
- `RMSE`: Root mean squared error
- `MAE`: Mean absolute error
- `MPE`: Mean percentage error
- `MAPE`: Mean absolute percentage error
- `MASE`: Mean absolute scaled error
- `ACF1`: Autocorrelation of errors at lag 1

`sw_augment()` returns a tibble with the following time series attributes:

- `index`: An index is either attempted to be extracted from the model or a sequential index is created for plotting purposes
- `.actual`: The original time series
- `.fitted`: The fitted values from the model
- `.resid`: The residual values from the model

See Also

`forecast::nnetar()`

Examples

```
library(dplyr)
library(forecast)

fit_nnetar <- lynx %>%
  nnetar()

sw_tidy(fit_nnetar)
sw_glance(fit_nnetar)
sw_augment(fit_nnetar)
```

tidiers_stl	<i>Tidying methods for STL (Seasonal, Trend, Level) decomposition of time series</i>
-------------	--

Description

Tidying methods for STL (Seasonal, Trend, Level) decomposition of time series

Usage

```
## S3 method for class 'stl'
sw_tidy(x, ...)

## S3 method for class 'stl'
sw_tidy_decomp(x, timetk_idx = FALSE, rename_index = "index", ...)

## S3 method for class 'stlm'
sw_tidy_decomp(x, timetk_idx = FALSE, rename_index = "index", ...)

## S3 method for class 'stlm'
sw_glance(x, ...)

## S3 method for class 'stlm'
sw_augment(x, data = NULL, rename_index = "index", timetk_idx = FALSE, ...)
```

Arguments

<code>x</code>	An object of class "stl"
<code>...</code>	Not used.
<code>timetk_idx</code>	Used with <code>sw_tidy_decomp</code> . When TRUE, uses a timetk index (irregular, typically date or datetime) if present.
<code>rename_index</code>	Used with <code>sw_tidy_decomp</code> . A string representing the name of the index generated.
<code>data</code>	Used with <code>sw_augment</code> only.

Value

`sw_tidy()` wraps `sw_tidy_decomp()`

`sw_tidy_decomp()` returns a tibble with the following time series attributes:

- `index`: An index is either attempted to be extracted from the model or a sequential index is created for plotting purposes
- `season`: The seasonal component
- `trend`: The trend component
- `remainder`: observed - (season + trend)
- `seasadj`: observed - season (or trend + remainder)

`sw_glance()` returns the underlying ETS or ARIMA model's `sw_glance()` results one row with the columns

- `model.desc`: A description of the model including the three integer components (p, d, q) are the AR order, the degree of differencing, and the MA order.
- `sigma`: The square root of the estimated residual variance
- `logLik`: The data's log-likelihood under the model
- `AIC`: The Akaike Information Criterion
- `BIC`: The Bayesian Information Criterion
- `ME`: Mean error
- `RMSE`: Root mean squared error
- `MAE`: Mean absolute error
- `MPE`: Mean percentage error
- `MAPE`: Mean absolute percentage error
- `MASE`: Mean absolute scaled error
- `ACF1`: Autocorrelation of errors at lag 1

`sw_augment()` returns a tibble with the following time series attributes:

- `index`: An index is either attempted to be extracted from the model or a sequential index is created for plotting purposes
- `.actual`: The original time series
- `.fitted`: The fitted values from the model
- `.resid`: The residual values from the model

See Also

```
stats::stl()
```

Examples

```
library(dplyr)
library(forecast)
library(sweep)

fit_stl <- USAccDeaths %>%
  stl(s.window = "periodic")

sw_tidy_decomp(fit_stl)
```

tidiers_StructTS	<i>Tidying methods for StructTS (Error, Trend, Seasonal) / exponential smoothing modeling of time series</i>
------------------	--

Description

These methods tidy the coefficients of StructTS models of univariate time series.

Usage

```
## S3 method for class 'StructTS'
sw_tidy(x, ...)

## S3 method for class 'StructTS'
sw_glance(x, ...)

## S3 method for class 'StructTS'
sw_augment(x, data = NULL, timetk_idx = FALSE, rename_index = "index", ...)
```

Arguments

x	An object of class "StructTS"
...	Additional parameters (not used)
data	Used with sw_augment only. NULL by default which simply returns augmented columns only. User can supply the original data, which returns the data + augmented columns.
timetk_idx	Used with sw_augment only. Uses a irregular timetk index if present.
rename_index	Used with sw_augment only. A string representing the name of the index generated.

Value

`sw_tidy()` returns one row for each model parameter, with two columns:

- `term`: The model parameters
- `estimate`: The estimated parameter value

`sw_glance()` returns one row with the columns

- `model.desc`: A description of the model including the three integer components (p, d, q) are the AR order, the degree of differencing, and the MA order.
- `sigma`: The square root of the estimated residual variance
- `logLik`: The data's log-likelihood under the model
- `AIC`: The Akaike Information Criterion
- `BIC`: The Bayesian Information Criterion
- `ME`: Mean error
- `RMSE`: Root mean squared error
- `MAE`: Mean absolute error
- `MPE`: Mean percentage error
- `MAPE`: Mean absolute percentage error
- `MASE`: Mean absolute scaled error
- `ACF1`: Autocorrelation of errors at lag 1

`sw_augment()` returns a tibble with the following time series attributes:

- `index`: An index is either attempted to be extracted from the model or a sequential index is created for plotting purposes
- `.actual`: The original time series
- `.fitted`: The fitted values from the model
- `.resid`: The residual values from the model

See Also

[stats::StructTS\(\)](#)

Examples

```
library(dplyr)
library(forecast)

fit_StructTS <- WWWusage %>%
  StructTS()

sw_tidy(fit_StructTS)
sw_glance(fit_StructTS)
sw_augment(fit_StructTS)
```

validate_index	<i>Validates data frame has column named the same name as variable rename_index</i>
----------------	---

Description

Validates data frame has column named the same name as variable rename_index

Usage

```
validate_index(ret, rename_index)
```

Arguments

ret	An object of class tibble
rename_index	A variable indicating the index name to be used in the tibble returned

Index

- * **datasets**
 - bike_sales, 3
- add_index, 2
- arima_string, 3
- bats_string, 3
- bike_sales, 3
- broom::augment(), 5
- broom::glance(), 6, 7
- broom::tidy(), 9, 10
- decompose(), 16
- forecast::bats(), 15
- forecast::ets(), 18
- forecast::nnetar(), 22
- forecast::tbats(), 15
- stats::arima(), 13
- stats::HoltWinters(), 20
- stats::stl(), 24
- stats::StructTS(), 25
- sw_augment, 4
 - sw_augment.Arima(tidiers_arima), 11
 - sw_augment.bats(tidiers_bats), 13
 - sw_augment.default, 5
 - sw_augment.ets(tidiers_ets), 16
 - sw_augment.HoltWinters(tidiers_HoltWinters), 18
 - sw_augment.nnetar(tidiers_nnetar), 20
 - sw_augment.stlm(tidiers_stl), 22
 - sw_augment.StructTS(tidiers_StructTS), 24
- sw_augment_columns, 6
- sw_glance, 6
 - sw_glance.Arima(tidiers_arima), 11
 - sw_glance.bats(tidiers_bats), 13
 - sw_glance.default, 7
 - sw_glance.ets(tidiers_ets), 16
 - sw_glance.HoltWinters(tidiers_HoltWinters), 18
 - sw_glance.nnetar(tidiers_nnetar), 20
 - sw_glance.stlm(tidiers_stl), 22
 - sw_glance.StructTS(tidiers_StructTS), 24
- sw_sweep, 7
- sw_tidy, 9
 - sw_tidy.Arima(tidiers_arima), 11
 - sw_tidy.bats(tidiers_bats), 13
 - sw_tidy.default, 10
 - sw_tidy.ets(tidiers_ets), 16
 - sw_tidy.HoltWinters(tidiers_HoltWinters), 18
 - sw_tidy.nnetar(tidiers_nnetar), 20
 - sw_tidy.stl(tidiers_stl), 22
 - sw_tidy.stlm(tidiers_arima), 11
 - sw_tidy.StructTS(tidiers_StructTS), 24
 - sw_tidy_decomp, 10
 - sw_tidy_decomp.bats(tidiers_bats), 13
 - sw_tidy_decomp.decomposed.ts(tidiers_decomposed_ts), 15
 - sw_tidy_decomp.ets(tidiers_ets), 16
 - sw_tidy_decomp.HoltWinters(tidiers_HoltWinters), 18
 - sw_tidy_decomp.stl(tidiers_stl), 22
 - sw_tidy_decomp.stlm(tidiers_stl), 22
- tbats_string, 11
- tidiers_arima, 11
- tidiers_bats, 13
- tidiers_decomposed_ts, 15
- tidiers_ets, 16
- tidiers_HoltWinters, 18
- tidiers_nnetar, 20
- tidiers_stl, 22
- tidiers_StructTS, 24
- timetk::tk_make_future_timeseries(), 8
- validate_index, 26