Package 'fable.intermittent'

September 8, 2026

Title Forecasting Models for Intermittent Time Series

Version 0.3.0

Description Extends the 'fable' framework to support forecasting methods specifically designed for intermittent time series data, where demand occurs sporadically with many zero values. All methods produce probabilistic forecasts returned as 'distributional' objects. The returned forecasts can be used to evaluate accuracy, plot and print the results seamlessly with 'fable'. The methods include:
Harvey, Fernandes (1989) <doi:10.1080/07350015.1989.10509750>,
Willemain, Smart, Schwarz (2004) <doi:10.1016/S0169-2070(03)00013-X>,
Zhou, Viswanathan (2011) <doi:10.1016/j.ijpe.2010.09.021>,
Snyder, Ord, Beaumont (2012) <doi:10.1016/j.ijforecast.2011.03.009>,
Kolassa (2016) <doi:10.1016/j.ijforecast.2015.12.004>,
Hasni, Aguir, Babai, Jemai (2019) <doi:10.1080/00207543.2018.1424375>,
Damato, Azzimonti, Corani (2025) <doi:10.1016/j.ijforecast.2025.10.001>,
Sbrana (2025) <doi:10.1080/01605682.2025.2569661>,
Sbrana, Babai (2026) <doi:10.1016/j.ejor.2026.06.009>.

License LGPL (>= 3)

Encoding UTF-8

LazyData true

LinkingTo Rcpp, RcppArmadillo

Depends R (>= 4.1.0), fabletools (>= 0.6.0)

Imports tsibble (>= 1.0.0), distributional (>= 0.9.0), nloptr (>= 2.0.0), rlang (>= 1.0.0), generics (>= 0.1.0), tibble (>= 3.0.0), tweedieDistr (>= 0.2.0)

Suggests covr, dplyr, testthat (>= 3.3.0), ggtime, ggplot2, knitr, rmarkdown

VignetteBuilder knitr

URL https://github.com/StefanoDamato/fable.intermittent

BugReports https://github.com/StefanoDamato/fable.intermittent/issues

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author Stefano Damato [aut, cre, cph] (ORCID:
<https://orcid.org/0009-0001-0225-3736>),
Lorenzo Zambon [aut] (ORCID: <https://orcid.org/0000-0002-8939-993X>),
Dario Azzimonti [aut] (ORCID: <https://orcid.org/0000-0001-5080-3061>)

Maintainer Stefano Damato <stefanodamato128@gmail.com>

Repository CRAN

Date/Publication 2026-09-08 07:00:08 UTC

Contents

auto . . . 3
BETANBB . . . 4
EMPSD . . . 5
fitted.BETANBB . . . 6
fitted.EMPSD . . . 7
fitted.GAMPOISB . . . 8
fitted.HSPES . . . 8
fitted.MARWAL . . . 9
fitted.NEGBINES . . . 10
fitted.NNARMA . . . 10
fitted.PARAMSD . . . 11
fitted.TWEES . . . 12
fitted.VZ . . . 12
fitted.WSS . . . 13
forecast.BETANBB . . . 14
forecast.EMPSD . . . 14
forecast.GAMPOISB . . . 15
forecast.HSPES . . . 16
forecast.MARWAL . . . 17
forecast.NEGBINES . . . 17
forecast.NNARMA . . . 18
forecast.PARAMSD . . . 19
forecast.TWEES . . . 20
forecast.VZ . . . 20
forecast.WSS . . . 21
GAMPOISB . . . 22
generate.BETANBB . . . 23
generate.EMPSD . . . 24
generate.GAMPOISB . . . 25
generate.HSPES . . . 25
generate.MARWAL . . . 26
generate.NEGBINES . . . 27
generate.NNARMA . . . 28
generate.PARAMSD . . . 28
generate.TWEES . . . 29

generate.VZ 30
generate.WSS 31
HSPES 31
MARWAL 33
NEGBINES 34
NNARMA 35
PARAMSD 36
pasta 38
raf 39
residuals.BETANBB 39
residuals.EMPSD 40
residuals.GAMPOISB 41
residuals.HSPES 41
residuals.MARWAL 42
residuals.NEGBINES 43
residuals.NNARMA 43
residuals.PARAMSD 44
residuals.TWEES 45
residuals.VZ 45
residuals.WSS 46
tinyM5 47
TWEES 48
VZ 49
WSS 50

Index **52**

`auto` *Automotive Spare Parts Demand Dataset*

Description

A dataset of monthly demand for automotive spare parts. The data contains 3000 intermittent time series, each spanning 24 monthly periods from January 2010 to December 2011.

Usage

```
auto
```

Format

A tsibble with 72,000 rows and 3 variables:

series_id Character. Unique identifier for each time series.

index Date (yearmonth). The monthly time index.

value Numeric. The demand quantity for the given month.

Source

Turkmen, A. C., Januschowski, T., Wang, Y., & Cemgil, A. T. (2021). Forecasting intermittent and sparse time series: A unified probabilistic framework via deep renewal processes. *PLOS ONE*, 16(11), e0259764.

Available at https://github.com/canerturkmen/gluon-ts/tree/intermittent-datasets/datasets.

Examples

```
library(tsibble)
auto
```

BETANBB *Beta-Negative Binomial Bayesian Dynamic Model*

Description

Conjugate Bayesian dynamic model for count time series with a negative binomial observation distribution and a Beta prior on the success probability parameter. This extends the conjugate updating framework of Harvey & Fernandes (1989) to the Beta-Negative Binomial family. The Beta prior is updated at each time step using a discount factor `w`. Forecasts are available as samples simulating from the model forward in time.

Usage

```
BETANBB(formula, ...)

## S3 method for class 'BETANBB'
report(object, ...)
```

Arguments

`formula`	Model specification.
`...`	Not used.
`object`	A fitted model object.

Value

A model specification.

References

Harvey, A. C., & Fernandes, C. (1989). Time series models for count or qualitative observations. *Journal of Business & Economic Statistics*, 7(4), 407–417. doi:10.1080/07350015.1989.10509750.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)

fc_ts <- ts |>
  model(BETANBB(value)) |>
  forecast(h = "7 days")

fc_ts |> print()


if (requireNamespace("ggtime", quietly = TRUE)) {
  library(ggtime)
  fc_ts |> autoplot(ts)
}
```

EMPSD *Empirical Distribution Resampling*

Description

Naive non-parametric baseline for intermittent demand forecasting. The predictive distribution at every horizon is simply the empirical distribution of the observed values: forecasts are produced by resampling with replacement from the historical series. Point forecasts are the sample mean.

Usage

```
EMPSD(formula, hot_start = FALSE, ...)

## S3 method for class 'EMPSD'
report(object, ...)
```

Arguments

`formula`	Model specification.
`hot_start`	Logical. If `TRUE`, leading zeros are removed from the time series before fitting.
`...`	Not used.
`object`	A fitted model object.

Value

A model specification.

References

Hasni, M., Aguir, M. S., Babai, M. Z., & Jemai, Z. (2019). Spare parts demand forecasting: a review on bootstrapping methods. *International Journal of Production Research*, 57(15–16), 4791–4804. doi:10.1080/00207543.2018.1424375.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)

fc_ts <- ts |>
  model(EMPSD(value)) |>
  forecast(h = "7 days")

fc_ts |> print()


if (requireNamespace("ggtime", quietly = TRUE)) {
  library(ggtime)
  fc_ts |> autoplot(ts)
}
```

`fitted.BETANBB` *Extract fitted values from a BETANBB model*

Description

Extract fitted values from a BETANBB model

Usage

```
## S3 method for class 'BETANBB'
fitted(object, ...)
```

Arguments

`object`	A model for which fitted values are required.
`...`	Not used.

Value

A numeric vector of fitted values.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, BETANBB(value))
fitted(fit)
```

`fitted.EMPSD` *Extract fitted values from an EMPSD model*

Description

Extract fitted values from an EMPSD model

Usage

```
## S3 method for class 'EMPSD'
fitted(object, ...)
```

Arguments

`object`	A model for which fitted values are required.
`...`	Not used.

Value

A numeric vector of fitted values.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, EMPSD(value))
fitted(fit)
```

fitted.GAMPOISB *Extract fitted values from a GAMPOISB model*

Description

Extract fitted values from a GAMPOISB model

Usage

```
## S3 method for class 'GAMPOISB'
fitted(object, ...)
```

Arguments

object	A model for which fitted values are required.
...	Not used.

Value

A numeric vector of fitted values.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, GAMPOISB(value))
fitted(fit)
```

fitted.HSPES *Extract fitted values from a HSPES model*

Description

Extract fitted values from a HSPES model

Usage

```
## S3 method for class 'HSPES'
fitted(object, ...)
```

Arguments

object	A model for which fitted values are required.
...	Not used.

Value

A numeric vector of fitted values.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, HSPES(value))
fitted(fit)
```

`fitted.MARWAL` *Extract fitted values from a MARWAL model*

Description

Extract fitted values from a MARWAL model

Usage

```
## S3 method for class 'MARWAL'
fitted(object, ...)
```

Arguments

`object`	A model for which fitted values are required.
`...`	Not used.

Value

A numeric vector of fitted values.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, MARWAL(value))
fitted(fit)
```

fitted.NEGBINES *Extract fitted values from a NEGBINES model*

Description

Extract fitted values from a NEGBINES model

Usage

```
## S3 method for class 'NEGBINES'
fitted(object, ...)
```

Arguments

object	A model for which fitted values are required.
...	Not used.

Value

A numeric vector of fitted values.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, NEGBINES(value))
fitted(fit)
```

fitted.NNARMA *Extract fitted values from a NNARMA model*

Description

Extract fitted values from a NNARMA model

Usage

```
## S3 method for class 'NNARMA'
fitted(object, ...)
```

Arguments

object	A model for which fitted values are required.
...	Not used.

Value

A numeric vector of fitted values.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, NNARMA(value))
fitted(fit)
```

`fitted.PARAMSD` *Extract fitted values from a PARAMSD model*

Description

Extract fitted values from a PARAMSD model

Usage

```
## S3 method for class 'PARAMSD'
fitted(object, ...)
```

Arguments

`object`	A model for which fitted values are required.
`...`	Not used.

Value

A numeric vector of fitted values.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, PARAMSD(value))
fitted(fit)
```

fitted.TWEES *Extract fitted values from a TWEES model*

Description

Extract fitted values from a TWEES model

Usage

```
## S3 method for class 'TWEES'
fitted(object, ...)
```

Arguments

object	A model for which fitted values are required.
...	Not used.

Value

A numeric vector of fitted values.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, TWEES(value))
fitted(fit)
```

fitted.VZ *Extract fitted values from a VZ model*

Description

Extract fitted values from a VZ model

Usage

```
## S3 method for class 'VZ'
fitted(object, ...)
```

Arguments

object	A model for which fitted values are required.
...	Not used.

Value

A numeric vector of fitted values.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, VZ(value))
fitted(fit)
```

`fitted.WSS` *Extract fitted values from a WSS model*

Description

Extract fitted values from a WSS model

Usage

```
## S3 method for class 'WSS'
fitted(object, ...)
```

Arguments

`object`	A model for which fitted values are required.
`...`	Not used.

Value

A numeric vector of fitted values.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, WSS(value))
fitted(fit)
```

forecast.BETANBB *Forecast a BETANBB model*

Description

Produces forecast distributions from a fitted BETANBB model using simulation.

Usage

```
## S3 method for class 'BETANBB'
forecast(object, new_data, specials = NULL, times = 10000, ...)
```

Arguments

object	A model for which forecasts are required.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
times	The number of sample paths to use in estimating the forecast distribution.
...	Other arguments passed to methods

Value

A distribution vector of class dist_sample.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, BETANBB(value))
forecast(fit, h = "7 days")
```

forecast.EMPSD *Forecast an EMPSD model*

Description

Produces forecast distributions by repeating the empirical distribution estimated from the training data at each forecast horizon.

Usage

```
## S3 method for class 'EMPSD'
forecast(object, new_data, specials = NULL, ...)
```

Arguments

object	A model for which forecasts are required.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
...	Other arguments passed to methods

Value

A distribution vector of class dist_sample.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, EMPSD(value))
forecast(fit, h = "7 days")
```

---

forecast.GAMPOISB *Forecast a GAMPOISB model*

Description

Produces forecast distributions from a fitted GAMPOISB model using simulation.

Usage

```
## S3 method for class 'GAMPOISB'
forecast(object, new_data, specials = NULL, times = 10000, ...)
```

Arguments

object	A model for which forecasts are required.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
times	The number of sample paths to use in estimating the forecast distribution.
...	Other arguments passed to methods

Value

A distribution vector of forecasts: for h=1 the vector class is dist_negative_binomial; for h>1 the vector class is dist_sample.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, GAMPOISB(value))
forecast(fit, h = "7 days")
```

`forecast.HSPES` *Forecast a HSPES model*

Description

Produces forecast distributions from a fitted HSPES model using simulation.

Usage

```
## S3 method for class 'HSPES'
forecast(object, new_data, specials = NULL, times = 10000, ...)
```

Arguments

`object`	A model for which forecasts are required.
`new_data`	A tsibble containing future index values to forecast.
`specials`	Passed by `fabletools::forecast.mdl_df()`.
`times`	The number of sample paths to use in estimating the forecast distribution.
`...`	Other arguments passed to methods

Value

A distribution vector of forecasts: for h=1 the vector class is `dist_inflated` (hurdle-shifted Poisson); for h>1 the vector class is `dist_sample`.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, HSPES(value))
forecast(fit, h = "7 days")
```

forecast.MARWAL *Forecast a MARWAL model*

Description

Produces forecast distributions from a fitted MARWAL model.

Usage

```
## S3 method for class 'MARWAL'
forecast(object, new_data, specials = NULL, ...)
```

Arguments

object	A model for which forecasts are required.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
...	Other arguments passed to methods

Value

A distribution vector of class dist_normal_nonneg.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, MARWAL(value))
forecast(fit, h = "7 days")
```

forecast.NEGBINES *Forecast a NEGBINES model*

Description

Produces forecast distributions from a fitted NEGBINES model using simulation.

Usage

```
## S3 method for class 'NEGBINES'
forecast(object, new_data, specials = NULL, times = 10000, ...)
```

Arguments

object	A model for which forecasts are required.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
times	The number of sample paths to use in estimating the forecast distribution.
...	Other arguments passed to methods

Value

A distribution vector of forecasts: for h=1 the vector class is dist_negative_binomial; for h>1 the vector class is dist_sample.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, NEGBINES(value))
forecast(fit, h = "7 days")
```

---

forecast.NNARMA *Forecast a NNARMA model*

Description

Produces forecast distributions from a fitted NNARMA model.

Usage

```
## S3 method for class 'NNARMA'
forecast(object, new_data, specials = NULL, ...)
```

Arguments

object	A model for which forecasts are required.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
...	Other arguments passed to methods

Value

A distribution vector of class dist_normal_nonneg.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, NNARMA(value))
forecast(fit, h = "7 days")
```

`forecast.PARAMSD` *Forecast a PARAMSD model*

Description

Produces forecast distributions from a fitted PARAMSD model.

Usage

```
## S3 method for class 'PARAMSD'
forecast(object, new_data, specials = NULL, ...)
```

Arguments

`object`	A model for which forecasts are required.
`new_data`	A tsibble containing future index values to forecast.
`specials`	Passed by `fabletools::forecast.mdl_df()`.
`...`	Other arguments passed to methods

Value

A distribution vector. The class depends on the static distribution fitted by the `PARAMSD` method.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, PARAMSD(value))
forecast(fit, h = "7 days")
```

forecast.TWEES *Forecast a TWEES model*

Description

Produces forecast distributions from a fitted TWEES model using simulation.

Usage

```
## S3 method for class 'TWEES'
forecast(object, new_data, specials = NULL, times = 10000, ...)
```

Arguments

object	A model for which forecasts are required.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
times	The number of sample paths to use in estimating the forecast distribution.
...	Other arguments passed to methods

Value

A distribution vector of forecasts: for h=1 the vector class is dist_tweedie; for h>1 the vector class is dist_sample.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, TWEES(value))
forecast(fit, h = "7 days")
```

forecast.VZ *Forecast a VZ model*

Description

Produces forecast distributions from a fitted VZ model using simulation.

Usage

```
## S3 method for class 'VZ'
forecast(object, new_data, specials = NULL, times = 10000, ...)
```

Arguments

object	A model for which forecasts are required.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
times	The number of sample paths to use in estimating the forecast distribution.
...	Other arguments passed to methods

Value

A distribution vector of forecasts of class dist_sample.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, VZ(value))
forecast(fit, h = "7 days")
```

---

forecast.WSS *Forecast a WSS model*

Description

Produces forecast distributions from a fitted WSS model using simulation.

Usage

```
## S3 method for class 'WSS'
forecast(object, new_data, specials = NULL, times = 10000, ...)
```

Arguments

object	A model for which forecasts are required.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
times	The number of sample paths to use in estimating the forecast distribution.
...	Other arguments passed to methods

Value

A distribution vector of forecasts of class dist_sample.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, WSS(value))
forecast(fit, h = "7 days")
```

GAMPOISB *Gamma-Poisson Bayesian Dynamic Model*

Description

Conjugate Bayesian dynamic model for count time series with a Poisson observation distribution and a Gamma prior on the rate parameter, following Harvey & Fernandes (1989). The Gamma prior is updated at each time step using a discount factor `w` that controls how quickly past information decays. The first-step forecast follows a Negative Binomial distribution, and multi-step forecasts are obtained by simulating from the model forward in time.

Usage

```
GAMPOISB(formula, ...)

## S3 method for class 'GAMPOISB'
report(object, ...)
```

Arguments

`formula`	Model specification.
`...`	Not used.
`object`	A fitted model object.

Value

A model specification.

References

Harvey, A. C., & Fernandes, C. (1989). Time series models for count or qualitative observations. *Journal of Business & Economic Statistics*, 7(4), 407–417. doi:10.1080/07350015.1989.10509750.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)

fc_ts <- ts |>
  model(GAMPOISB(value)) |>
  forecast(h = "7 days")

fc_ts |> print()


if (requireNamespace("ggtime", quietly = TRUE)) {
  library(ggtime)
  fc_ts |> autoplot(ts)
}
```

`generate.BETANBB` *Generate sample paths from a BETANBB model*

Description

Generate sample paths from a BETANBB model

Usage

```
## S3 method for class 'BETANBB'
generate(x, new_data, specials = NULL, ...)
```

Arguments

`x`	A fitted `BETANBB` model object.
`new_data`	A tsibble containing future index values to forecast.
`specials`	Passed by `fabletools::forecast.mdl_df()`.
`...`	Other arguments passed to methods

Value

A vector of future paths from a dataset using a fitted model.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, BETANBB(value))
generate(fit, new_data = tsibble::new_data(ts, 7))
```

`generate.EMPSD` *Generate sample paths from an EMPSD model*

Description

Generate sample paths from an EMPSD model

Usage

```
## S3 method for class 'EMPSD'
generate(x, new_data, specials = NULL, ...)
```

Arguments

`x`	A fitted `EMPSD` model object.
`new_data`	A tsibble containing future index values to forecast.
`specials`	Passed by `fabletools::forecast.mdl_df()`.
`...`	Other arguments passed to methods

Value

A `new_data` tibble with a `.sim` column of simulated values.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, EMPSD(value))
generate(fit, h = 7)
```

generate.GAMPOISB *Generate sample paths from a GAMPOISB model*

Description

Generate sample paths from a GAMPOISB model

Usage

```
## S3 method for class 'GAMPOISB'
generate(x, new_data, specials = NULL, ...)
```

Arguments

x	A fitted GAMPOISB model object.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
...	Other arguments passed to methods

Value

A vector of future paths from a dataset using a fitted model.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, GAMPOISB(value))
generate(fit, new_data = tsibble::new_data(ts, 7))
```

generate.HSPES *Generate sample paths from a HSPES model*

Description

Generate sample paths from a HSPES model

Usage

```
## S3 method for class 'HSPES'
generate(x, new_data, specials = NULL, ...)
```

Arguments

x	A fitted HSPES model object.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
...	Other arguments passed to methods

Value

A vector of future paths from a dataset using a fitted model.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, HSPES(value))
generate(fit, new_data = tsibble::new_data(ts, 7))
```

---

generate.MARWAL *Generate sample paths from a MARWAL model*

Description

Generate sample paths from a MARWAL model

Usage

```
## S3 method for class 'MARWAL'
generate(x, new_data, specials = NULL, ...)
```

Arguments

x	A fitted MARWAL model object.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
...	Other arguments passed to methods

Value

A vector of future paths from a dataset using a fitted model.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, MARWAL(value))
generate(fit, new_data = tsibble::new_data(ts, 7))
```

`generate.NEGBINES` *Generate sample paths from a NEGBINES model*

Description

Generate sample paths from a NEGBINES model

Usage

```
## S3 method for class 'NEGBINES'
generate(x, new_data, specials = NULL, ...)
```

Arguments

`x`	A fitted `NEGBINES` model object.
`new_data`	A tsibble containing future index values to forecast.
`specials`	Passed by `fabletools::forecast.mdl_df()`.
`...`	Other arguments passed to methods

Value

A vector of future paths from a dataset using a fitted model.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, NEGBINES(value))
generate(fit, new_data = tsibble::new_data(ts, 7))
```

generate.NNARMA *Generate sample paths from a NNARMA model*

Description

Generate sample paths from a NNARMA model

Usage

```
## S3 method for class 'NNARMA'
generate(x, new_data, specials = NULL, ...)
```

Arguments

x	A fitted NNARMA model object.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
...	Other arguments passed to methods

Value

A vector of future paths from a dataset using a fitted model.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, NNARMA(value))
generate(fit, new_data = tsibble::new_data(ts, 7))
```

generate.PARAMSD *Generate sample paths from a PARAMSD model*

Description

Generate sample paths from a PARAMSD model

Usage

```
## S3 method for class 'PARAMSD'
generate(x, new_data, specials = NULL, ...)
```

Arguments

x	A fitted PARAMSD model object.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
...	Other arguments passed to methods

Value

A vector of future paths from a dataset using a fitted model.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, PARAMSD(value))
generate(fit, new_data = tsibble::new_data(ts, 7))
```

---

generate.TWEES *Generate sample paths from a TWEES model*

Description

Generate sample paths from a TWEES model

Usage

```
## S3 method for class 'TWEES'
generate(x, new_data, specials = NULL, ...)
```

Arguments

x	A fitted TWEES model object.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
...	Other arguments passed to methods

Value

A vector of future paths from a dataset using a fitted model.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, TWEES(value))
generate(fit, new_data = tsibble::new_data(ts, 7))
```

`generate.VZ` *Generate sample paths from a VZ model*

Description

Generate sample paths from a VZ model

Usage

```
## S3 method for class 'VZ'
generate(x, new_data, specials = NULL, ...)
```

Arguments

`x`	A fitted `VZ` model object.
`new_data`	A tsibble containing future index values to forecast.
`specials`	Passed by `fabletools::forecast.mdl_df()`.
`...`	Other arguments passed to methods

Value

A vector of future paths from a dataset using a fitted model.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, VZ(value))
generate(fit, new_data = tsibble::new_data(ts, 7))
```

generate.WSS *Generate sample paths from a WSS model*

Description

Generate sample paths from a WSS model

Usage

```
## S3 method for class 'WSS'
generate(x, new_data, specials = NULL, ...)
```

Arguments

x	A fitted WSS model object.
new_data	A tsibble containing future index values to forecast.
specials	Passed by fabletools::forecast.mdl_df().
...	Other arguments passed to methods

Value

A vector of future paths from a dataset using a fitted model.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, WSS(value))
generate(fit, new_data = tsibble::new_data(ts, 7))
```

HSPES *Hurdle-Shifted Poisson Exponential Smoothing*

Description

Exponential smoothing state space model for intermittent demand with a hurdle-shifted Poisson observation distribution, following the framework of Snyder, Ord & Beaumont (2012). The model decomposes demand into two components: a Bernoulli occurrence process (probability of non-zero demand) and a shifted Poisson demand-size process. Both components are driven by independent (optionally damped) exponential smoothing state equations. The first-step forecast follows an hurdle-shifted Poisson distribution, multi-step forecasts are obtained by simulating from the model forward in time.

Usage

```
HSPES(formula, damped = TRUE, ...)

## S3 method for class 'HSPES'
report(object, ...)
```

Arguments

formula	Model specification.
damped	Logical. If TRUE (default), both the occurrence and demand smoothing components use a damping parameter.
...	Not used.
object	A fitted model object.

Value

A model specification.

References

Snyder, R. D., Ord, J. K., & Beaumont, A. (2012). Forecasting the intermittent demand for slow-moving inventories: A modelling approach. *International Journal of Forecasting*, 28(2), 485–496. doi:10.1016/j.ijforecast.2011.03.009.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)

fc_ts <- ts |>
  model(HSPES(value)) |>
  forecast(h = "7 days")

fc_ts |> print()


if (requireNamespace("ggtime", quietly = TRUE)) {
  library(ggtime)
  fc_ts |> autoplot(ts)
}
```

MARWAL *Markov Chain Model with Random Walk dynamic*

Description

A Random Walk model on the demand composed by a switching constant and an ARMA model, where the changes are controlled by a Markov Chain on the occurrence process. Parameters are estimated in closed-form on deseasonalized data, and forecasts are returned as Gaussian distributions for each time step.

Usage

```
MARWAL(formula, ...)

## S3 method for class 'MARWAL'
report(object, ...)
```

Arguments

formula	Model specification.
...	Not used.
object	A fitted model object.

Value

A model specification.

References

Sbrana, G. (2025). Markov Walk and Walmart sales prediction. *Journal of the Operational Research Society*, 1–12. doi:10.1080/01605682.2025.2569661.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)

fc_ts <- ts |>
  model(MARWAL(value)) |>
  forecast(h = "7 days")

fc_ts |> print()


if (requireNamespace("ggtime", quietly = TRUE)) {
```

```
  library(ggtime)
  fc_ts |> autoplot(ts)
}
```

NEGBINES *Negative Binomial Exponential Smoothing*

Description

Exponential smoothing state space model for intermittent demand with a negative binomial observation distribution, as proposed by Snyder, Ord & Beaumont (2012). The conditional mean of the negative binomial is governed by a (optionally damped) exponential smoothing process. The probability parameter is estimated to maximise the likelihood. The first-step forecast follows a Negative Binomial distribution, and multi-step forecasts are obtained by simulating from the model forward in time.

Usage

```
NEGBINES(formula, damped = TRUE, ...)

## S3 method for class 'NEGBINES'
report(object, ...)
```

Arguments

formula	Model specification.
damped	Logical. If TRUE (default), the exponential smoothing component uses a damping parameter.
...	Not used.
object	A fitted model object.

Value

A model specification.

References

Snyder, R. D., Ord, J. K., & Beaumont, A. (2012). Forecasting the intermittent demand for slow-moving inventories: A modelling approach. *International Journal of Forecasting*, 28(2), 485–496. doi:10.1016/j.ijforecast.2011.03.009.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)

fc_ts <- ts |>
  model(NEGBINES(value)) |>
  forecast(h = "7 days")

fc_ts |> print()


if (requireNamespace("ggtime", quietly = TRUE)) {
  library(ggtime)
  fc_ts |> autoplot(ts)
}
```

NNARMA *Non-negative ARMA model*

Description

A non-negative autoregressive moving average (NNARMA) model for intermittent demand, where the demand is driven by an ARMA(1, 1), constrained to remain non-negative. Forecasts are returned as Gaussian distributions for each time step, with the negative probability mass collapsed to zero.

Usage

```
NNARMA(formula, ...)

## S3 method for class 'NNARMA'
report(object, ...)
```

Arguments

formula	Model specification.
...	Not used.
object	A fitted model object.

Value

A model specification.

References

Sbrana, G., & Babai, M. Z. (2026). Non-negative autoregressive moving average models for intermittent demand: Forecast accuracy and inventory implications. *European Journal of Operational Research*. doi:10.1016/j.ejor.2026.06.009.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)

fc_ts <- ts |>
  model(NNARMA(value)) |>
  forecast(h = "7 days")

fc_ts |> print()


if (requireNamespace("ggtime", quietly = TRUE)) {
  library(ggtime)
  fc_ts |> autoplot(ts)
}
```

PARAMSD *Static Count Distribution Model*

Description

Static (IID) count distribution model for intermittent demand, following Kolassa (2016). The method fits several candidate distributions — Poisson, hurdle-shifted Poisson, negative binomial, and hurdle-shifted negative binomial — plus a Tweedie, to the observed series and selects the best by AIC.

Usage

```
PARAMSD(
  formula,
  distr = c("auto", "pois", "hsp", "nbinom", "hsnb", "tweedie"),
  hot_start = FALSE,
  criterion = c("aic", "bic"),
  ...
)

## S3 method for class 'PARAMSD'
report(object, ...)
```

Arguments

formula	Model specification.
distr	Distribution choice: one of "auto", "pois", "hsp", "nbinom", "hsnb", or "tweedie".
hot_start	Logical. If TRUE, leading zeros are removed from the time series before fitting.
criterion	Information criterion to use for model selection when distr = "auto". One of "aic" or "bic".
...	Not used.
object	A fitted model object.

Details

A Tweedie is continuous on the positive half-line with an atom at zero, so its log-likelihood is a density rather than a probability mass and cannot be ranked against the count candidates by AIC/BIC. The two entry points therefore use different forms of it, and are reported distinctly:

- distr = "auto" ranks a *discretised* Tweedie, obtained by rounding to the non-negative integers, with probability mass function P(Y = 0) = F(0.5) and P(Y = k) = F(k + 0.5) − F(k − 0.5). This puts it on the same scale as the four count distributions. When it wins, the model is reported as PARAMSD(tweedie_discrete).
- distr = "tweedie" fits the *continuous* Tweedie, for intermittent series that are not counts. It is reported as PARAMSD(tweedie), and is the only choice for which generate.PARAMSD() returns non-integer sample paths.

Value

A model specification.

References

Kolassa, S. (2016). Evaluating predictive count data distributions in retail sales forecasting. *International Journal of Forecasting*, 32(3), 788–803. doi:10.1016/j.ijforecast.2015.12.004.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)

fc_ts <- ts |>
  model(PARAMSD(value)) |>
  forecast(h = "7 days")

fc_ts |> print()
```

```
if (requireNamespace("ggtime", quietly = TRUE)) {
  library(ggtime)
  fc_ts |> autoplot(ts)
}
```

`pasta` *Pasta Sales Dataset*

Description

Daily sales and promotional data for pasta products from an Italian grocery store, spanning five years from January 2014 to December 2018. The dataset covers 118 Stock Keeping Units (SKUs) organised across four brands (B1–B4), making it a standard benchmark for hierarchical and intermittent demand forecasting. Each SKU-day combination records the quantity sold and a binary promotional indicator. Dates on which the store was closed (public holidays) are included with a quantity sold of zero.

Usage

```
pasta
```

Format

A tsibble with 215,350 rows and 5 variables:

index Date. Daily time index.

brand Character. Brand identifier (B1, B2, B3, B4), with 42, 45, 21, and 10 SKUs respectively.

product Character. SKU number within the brand.

value Numeric. Daily quantity sold.

promotion Integer. Binary promotional indicator (1 = on promotion, 0 = not on promotion).

Source

Mancuso, P., Piccialli, V., & Sudoso, A. M. (2021). A machine learning approach for forecasting hierarchical time series. *Expert Systems with Applications*, 182, 115102. doi:10.1016/j.eswa.2021.115102

Dataset available at https://data.mendeley.com/datasets/njdkntcpc9/1 under a Creative Commons Attribution 4.0 International licence (CC BY 4.0).

Examples

```
library(tsibble)
pasta
```

raf *RAF Spare Parts Demand Dataset*

Description

A dataset of monthly demand for spare parts from the Royal Air Force (RAF). The data contains 5000 intermittent time series, each spanning 84 monthly periods from January 1996 to December 2002. This is a widely used benchmark dataset for intermittent demand forecasting.

Usage

```
raf
```

Format

A tsibble with 420,000 rows and 3 variables:

series_id Character. Unique identifier for each time series.

index Date (yearmonth). The monthly time index.

value Numeric. The demand quantity for the given month.

Source

Syntetos, A. A., & Boylan, J. E. (2005). The accuracy of intermittent demand estimates. *International Journal of Forecasting*, 21(2), 303–314.

Available at https://github.com/canerturkmen/gluon-ts/tree/intermittent-datasets/datasets.

Examples

```
library(tsibble)
raf
```

residuals.BETANBB *Extract residuals from a BETANBB model*

Description

Extract residuals from a BETANBB model

Usage

```
## S3 method for class 'BETANBB'
residuals(object, ...)
```

Arguments

object	A model for which residuals are required.
...	Not used.

Value

A numeric vector of residuals.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, BETANBB(value))
residuals(fit)
```

---

residuals.EMPSD *Extract residuals from an EMPSD model*

Description

Extract residuals from an EMPSD model

Usage

```
## S3 method for class 'EMPSD'
residuals(object, ...)
```

Arguments

object	A model for which residuals are required.
...	Not used.

Value

A numeric vector of residuals.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, EMPSD(value))
residuals(fit)
```

residuals.GAMPOISB *Extract residuals from a GAMPOISB model*

Description

Extract residuals from a GAMPOISB model

Usage

```
## S3 method for class 'GAMPOISB'
residuals(object, ...)
```

Arguments

object	A model for which residuals are required.
...	Not used.

Value

A numeric vector of residuals.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, GAMPOISB(value))
residuals(fit)
```

residuals.HSPES *Extract residuals from a HSPES model*

Description

Extract residuals from a HSPES model

Usage

```
## S3 method for class 'HSPES'
residuals(object, ...)
```

Arguments

object	A model for which residuals are required.
...	Not used.

Value

A numeric vector of residuals.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, HSPES(value))
residuals(fit)
```

---

residuals.MARWAL *Extract residuals from a MARWAL model*

Description

Extract residuals from a MARWAL model

Usage

```
## S3 method for class 'MARWAL'
residuals(object, ...)
```

Arguments

object	A model for which residuals are required.
...	Not used.

Value

A numeric vector of residuals.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, MARWAL(value))
residuals(fit)
```

residuals.NEGBINES *Extract residuals from a NEGBINES model*

Description

Extract residuals from a NEGBINES model

Usage

```
## S3 method for class 'NEGBINES'
residuals(object, ...)
```

Arguments

object	A model for which residuals are required.
...	Not used.

Value

A numeric vector of residuals.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, NEGBINES(value))
residuals(fit)
```

residuals.NNARMA *Extract residuals from a NNARMA model*

Description

Extract residuals from a NNARMA model

Usage

```
## S3 method for class 'NNARMA'
residuals(object, ...)
```

Arguments

object	A model for which residuals are required.
...	Not used.

Value

A numeric vector of residuals.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, NNARMA(value))
residuals(fit)
```

`residuals.PARAMSD` *Extract residuals from a PARAMSD model*

Description

Extract residuals from a PARAMSD model

Usage

```
## S3 method for class 'PARAMSD'
residuals(object, ...)
```

Arguments

`object`	A model for which residuals are required.
`...`	Not used.

Value

A numeric vector of residuals.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, PARAMSD(value))
residuals(fit)
```

residuals.TWEES *Extract residuals from a TWEES model*

Description

Extract residuals from a TWEES model

Usage

```
## S3 method for class 'TWEES'
residuals(object, ...)
```

Arguments

object	A model for which residuals are required.
...	Not used.

Value

A numeric vector of residuals.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, TWEES(value))
residuals(fit)
```

residuals.VZ *Extract residuals from a VZ model*

Description

Extract residuals from a VZ model

Usage

```
## S3 method for class 'VZ'
residuals(object, ...)
```

Arguments

object	A model for which residuals are required.
...	Not used.

Value

A numeric vector of residuals.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, VZ(value))
residuals(fit)
```

---

residuals.WSS *Extract residuals from a WSS model*

Description

Extract residuals from a WSS model

Usage

```
## S3 method for class 'WSS'
residuals(object, ...)
```

Arguments

object	A model for which residuals are required.
...	Not used.

Value

A numeric vector of residuals.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)
fit <- model(ts, WSS(value))
residuals(fit)
```

`tinyM5` *Tiny M5 Dataset*

Description

A subset of the M5 Walmart Challenge dataset, covering daily unit sales for 28 products across 10 stores (280 series), spanning 1913 days from January 2011 to April 2016. In addition to demand, the dataset carries the product hierarchy, calendar events, SNAP eligibility, and weekly selling price, so it can also be used to test models with exogenous regressors. The dataset is taken from the **m5** package, now removed from CRAN.

Usage

```
tinyM5
```

Format

A tsibble with 535,640 rows and 18 variables:

item_id Character. The id of the product.

dept_id Character. The id of the department the product belongs to.

cat_id Character. The id of the category the product belongs to.

store_id Character. The id of the store where the product is sold.

state_id Character. The state where the store is located.

value Integer. The number of units sold.

date Date. Daily time index.

wm_yr_wk Integer. The id of the week the date belongs to.

weekday Character. The name of the day of the week.

wday Integer. The id of the weekday, starting from Saturday.

month Integer. The month of the date.

year Integer. The year of the date.

event_name_1 Character. If the date includes an event, the name of this event.

event_type_1 Character. If the date includes an event, the type of this event.

event_name_2 Character. If the date includes a second event, the name of this event.

event_type_2 Character. If the date includes a second event, the type of this event.

snap Integer. Binary indicator (0 or 1) of whether SNAP purchases are allowed in the store's state on the given date.

sell_price Numeric. The price of the product for the given week/store, averaged across seven days. `NA` if the product was not sold during that week.

Source

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4), 1346–1364. doi:10.1016/j.ijforecast.2021.11.013

Extracted with the **m5** package (https://cran.r-project.org/package=m5), now removed from CRAN.

Use of the data is subject to the M5 Forecasting - Accuracy competition rules: https://www.kaggle.com/c/m5-forecasting-accuracy/rules.

Examples

```
library(tsibble)
tinyM5
```

TWEES *Tweedie Exponential Smoothing*

Description

Exponential smoothing state space model for intermittent demand with a Tweedie observation distribution. The conditional mean of the Tweedie is governed by a (optionally damped) exponential smoothing process, and the dispersion parameter is derived by a second (optionally damped) exponential smoothing process on the occurrence binary time series. The Tweedie family naturally models both zeros and large spikes via its compound Poisson-Gamma nature. The power parameter is optimised to maximise the likelihood. The first-step forecast follows a Tweedie distribution, and multi-step forecasts are obtained by simulating from the model forward in time.

Usage

```
TWEES(formula, damped = TRUE, scaling = TRUE, ...)

## S3 method for class 'TWEES'
report(object, ...)
```

Arguments

formula	Model specification.
damped	Logical. If TRUE (default), the exponential smoothing component uses a damping parameter.
scaling	Logical. If TRUE (default), the time series is divided by its maximum value before fitting and predictions are back-transformed. This improves numerical stability.
...	Not used.
object	A fitted model object.

Value

A model specification.

References

Damato, S., Azzimonti, D., & Corani, G. (2025). Forecasting intermittent time series with Gaussian Processes and Tweedie likelihood. *International Journal of Forecasting* (in press). doi:10.1016/j.ijforecast.2025.10.001.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)

fc_ts <- ts |>
  model(TWEES(value)) |>
  forecast(h = "7 days")

fc_ts |> print()


if (requireNamespace("ggtime", quietly = TRUE)) {
  library(ggtime)
  fc_ts |> autoplot(ts)
}
```

VZ *Viswanathan–Zhou Bootstrap Method*

Description

Non-parametric bootstrap forecasting method for intermittent demand proposed by Zhou & Viswanathan (2011). The time series is decomposed into demand sizes and inter-demand intervals via Croston's decomposition. Forecasts are generated by independently resampling from the empirical distributions of both components, preserving the marginal structure of the original series.

Usage

```
VZ(formula, ...)

## S3 method for class 'VZ'
report(object, ...)
```

Arguments

formula	Model specification.
...	Not used.
object	A fitted model object.

Value

A model specification.

References

Zhou, Chenxi, Viswanathan, S., (2011). Comparison of a new bootstrapping method with parametric approaches for safety stock determination in service parts inventory systems. *International Journal of Production Economics* 133(1) 481–485. doi:10.1016/j.ijpe.2010.09.021

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)

fc_ts <- ts |>
  model(VZ(value)) |>
  forecast(h = "7 days")

fc_ts |> print()


if (requireNamespace("ggtime", quietly = TRUE)) {
  library(ggtime)
  fc_ts |> autoplot(ts)
}
```

WSS *Willemain–Smart–Schwarz Bootstrap Method*

Description

Markov-chain bootstrap forecasting method for intermittent demand proposed by Willemain, Smart & Schwarz (2004). Occurrence of non-zero demand is modelled with a two-state Markov chain whose transition probabilities are estimated from the observed binary occurrence sequence. Demand sizes are resampled from past non-zero observations with Gaussian jittering to smooth the empirical distribution.

Usage

```
WSS(formula, ...)

## S3 method for class 'WSS'
report(object, ...)
```

Arguments

formula	Model specification.
...	Not used.
object	A fitted model object.

Value

A model specification.

References

Willemain, T. R., Smart, C. N., & Schwarz, H. F. (2004). A new approach to forecasting intermittent demand for service parts inventories. *International Journal of Forecasting*, 20(3), 375–387. doi:10.1016/S01692070(03)00013X.

Examples

```
ts <- tsibble::tsibble(
  time = as.Date("2026-01-01") + seq_len(40),
  value = rnbinom(40, size = 1, prob = 0.3),
  index = time
)

fc_ts <- ts |>
  model(WSS(value)) |>
  forecast(h = "7 days")

fc_ts |> print()


if (requireNamespace("ggtime", quietly = TRUE)) {
  library(ggtime)
  fc_ts |> autoplot(ts)
}
```

Index

∗ **datasets**
auto, 3
pasta, 38
raf, 39
tinyM5, 47

auto, 3

BETANBB, 4

EMPSD, 5

fabletools::forecast.mdl_df(), *14–21*, *23–31*
fitted.BETANBB, 6
fitted.EMPSD, 7
fitted.GAMPOISB, 8
fitted.HSPES, 8
fitted.MARWAL, 9
fitted.NEGBINES, 10
fitted.NNARMA, 10
fitted.PARAMSD, 11
fitted.TWEES, 12
fitted.VZ, 12
fitted.WSS, 13
forecast.BETANBB, 14
forecast.EMPSD, 14
forecast.GAMPOISB, 15
forecast.HSPES, 16
forecast.MARWAL, 17
forecast.NEGBINES, 17
forecast.NNARMA, 18
forecast.PARAMSD, 19
forecast.TWEES, 20
forecast.VZ, 20
forecast.WSS, 21

GAMPOISB, 22
generate.BETANBB, 23
generate.EMPSD, 24
generate.GAMPOISB, 25
generate.HSPES, 25
generate.MARWAL, 26
generate.NEGBINES, 27
generate.NNARMA, 28
generate.PARAMSD, 28
generate.PARAMSD(), *37*
generate.TWEES, 29
generate.VZ, 30
generate.WSS, 31

HSPES, 31

MARWAL, 33

NEGBINES, 34
NNARMA, 35

PARAMSD, 36
pasta, 38

raf, 39
report.BETANBB *(*BETANBB*)*, 4
report.EMPSD *(*EMPSD*)*, 5
report.GAMPOISB *(*GAMPOISB*)*, 22
report.HSPES *(*HSPES*)*, 31
report.MARWAL *(*MARWAL*)*, 33
report.NEGBINES *(*NEGBINES*)*, 34
report.NNARMA *(*NNARMA*)*, 35
report.PARAMSD *(*PARAMSD*)*, 36
report.TWEES *(*TWEES*)*, 48
report.VZ *(*VZ*)*, 49
report.WSS *(*WSS*)*, 50
residuals.BETANBB, 39
residuals.EMPSD, 40
residuals.GAMPOISB, 41
residuals.HSPES, 41
residuals.MARWAL, 42
residuals.NEGBINES, 43
residuals.NNARMA, 43
residuals.PARAMSD, 44
residuals.TWEES, 45

residuals.VZ, 45
residuals.WSS, 46

tinyM5, 47
TWEES, 48

VZ, 49

WSS, 50