# Package: fable.binary (via r-universe)

November 22, 2024

**Title** Forecasting Binary Time Series

**Version** 0.1.0

**Description** Provides a collection of time series forecasting models suitable for binary time series. These models work within the 'fable' framework provided by the 'fabletools' package, which provides the tools to evaluate, visualise, and combine models in a workflow consistent with the tidyverse.

**License** GPL-3

**URL** <https://tidyverts.github.io/fable.binary>

**BugReports** <https://github.com/tidyverts/fable.binary/issues>

**Depends** R (>= 3.4.0), fabletools (>= 0.3.0)

**Imports** distributional, dplyr (>= 1.0.0), nnet, purrr, rlang (>= 0.4.6), stats, tibble, tidyr, tsibble (>= 0.9.0)

**Suggests** broom, testthat (>= 3.0.0)

**ByteCompile** true

**Encoding** UTF-8

**Language** en-GB

**Roxygen** list(markdown = TRUE, roclets=c('rd', 'collate', 'namespace'))

**RoxygenNote** 7.2.3

**LazyData** true

**Config/testthat/edition** 3

**Config/pak/sysreqs** libicu-dev

**Repository** https://robjhyndman.r-universe.dev

**RemoteUrl** https://github.com/tidyverts/fable.binary

**RemoteRef** HEAD

**RemoteSha** be338e93a81dfa12484f3639cfc516e76bf3d102

# Contents

---

BINNET                          *Neural Network Binary Time Series Forecasts*

---

### Description

Feed-forward neural networks with a single hidden layer and lagged inputs for forecasting univariate binary time series.

### Usage

```
BINNET(formula, n_nodes = NULL, n_networks = 20, scale_inputs = TRUE, ...)
```

### Arguments

| | |
|---|---|
| formula | Model specification (see "Specials" section). |
| n_nodes | Number of nodes in the hidden layer. Default is half of the number of external regressors plus 1. |
| n_networks | Number of networks to fit with different random starting weights. These are then averaged when producing forecasts. |
| scale_inputs | If TRUE, inputs are scaled by subtracting the column means and dividing by their respective standard deviations. |
| ... | Other arguments passed to \link[nnet]{nnet}. |

## Details

A feed-forward neural network is fitted with a single hidden layer containing size nodes.

Exogenous regressors are used as inputs. A total of repeats networks are fitted, each with random starting weights. These are then averaged when computing forecasts.

## Value

A model specification.

## Examples

```
melb_rain |>
  model(nn = BINNET(Wet ~ fourier(K = 1, period = "year")))
```

---

fitted.BINNET                *Extract fitted values from a fable model*

---

## Description

Extracts the fitted values.

## Usage

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

## Arguments

| object | Fitted model |
|---|---|
| ... | Other arguments ignored |

## Examples

```
melb_rain |>
  model(nn = BINNET(Wet ~ fourier(K = 1, period = "year"))) |>
  fitted()
```

---

`fitted.LOGISTIC` *Extract fitted values from a fable model*

---

#### Description

Extracts the fitted values.

#### Usage

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

#### Arguments

| | |
|---|---|
| object | Fitted model |
| ... | Other arguments ignored |

#### Examples

```
melb_rain |>
  model(logistic = LOGISTIC(Wet ~ fourier(K = 5, period = "year"))) |>
  fitted()
```

---

`forecast.BINNET` *Forecast a model from the fable package*

---

#### Description

Produces forecasts from a trained model.

#### Usage

```
## S3 method for class 'BINNET'
forecast(object, new_data, specials = NULL, simulate = TRUE, times = 5000, ...)
```

#### Arguments

| | |
|---|---|
| object | A model for which forecasts are required. |
| new_data | A tsibble containing the time points and exogenous regressors to produce forecasts for. |
| specials | (passed by [`fabletools::forecast.mdl_df()`](fabletools::forecast.mdl_df())). |
| simulate | If TRUE, then forecast distributions are computed using simulation from a Bernoulli model. |
| times | The number of sample paths to use in estimating the forecast distribution when simulate = TRUE. |
| ... | Other arguments passed to methods |

## Examples

```
melb_rain |>
  model(nn = BINNET(Wet ~ fourier(K = 1, period = "year"))) |>
  forecast(times = 10)
```

---

forecast.LOGISTIC                *Forecast a model from the fable package*

---

### Description

Produces forecasts from a trained model.

### Usage

```
## S3 method for class 'LOGISTIC'
forecast(
  object,
  new_data,
  specials = NULL,
  simulate = FALSE,
  times = 5000,
  ...
)
```

### Arguments

| | |
|---|---|
| object | A model for which forecasts are required. |
| new_data | A tsibble containing the time points and exogenous regressors to produce forecasts for. |
| specials | (passed by [fabletools::forecast.mdl_df()](#)). |
| simulate | If TRUE, then forecast distributions are computed using simulation from a Bernoulli model. |
| times | The number of sample paths to use in estimating the forecast distribution when simulate = TRUE. |
| ... | Other arguments passed to methods |

### Examples

```
melb_rain |>
  model(logistic = LOGISTIC(Wet ~ fourier(K = 5, period = "year"))) |>
  forecast(h = "2 years")
```

---

generate.BINNET          *Generate new data from a fable model*

---

### Description

Simulates future paths from a dataset using a fitted model.

### Usage

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

### Arguments

| | |
|---|---|
| x | An object. |
| new_data | A tsibble containing the time points and exogenous regressors to produce forecasts for. |
| specials | (passed by [fabletools::generate.mdl_df()](fabletools::generate.mdl_df())). |
| ... | Other arguments passed to methods |

### Examples

```
melb_rain |>
  model(nn = BINNET(Wet ~ fourier(K = 1, period = "year"))) |>
  generate()
```

---

generate.LOGISTIC         *Generate new data from a fable model*

---

### Description

Simulates future paths from a dataset using a fitted model.

### Usage

```
## S3 method for class 'LOGISTIC'
generate(x, new_data, specials, ...)
```

### Arguments

| | |
|---|---|
| x | An object. |
| new_data | A tsibble containing the time points and exogenous regressors to produce forecasts for. |
| specials | (passed by [fabletools::generate.mdl_df()](fabletools::generate.mdl_df())). |
| ... | Other arguments passed to methods |

## Examples

```
melb_rain |>
  model(logistic = LOGISTIC(Wet ~ fourier(K = 5, period = "year"))) |>
  generate()
```

---

glance.BINNET    *Glance a BINNET model*

---

## Description

Construct a single row summary of the BINNET model. Contains the variance of residuals (`sigma2`).

## Usage

```
## S3 method for class 'BINNET'
glance(x, ...)
```

## Arguments

x                 model or other R object to convert to single-row data frame

...               Other arguments ignored

## Value

A one row tibble summarising the model's fit.

## Examples

```
melb_rain |>
  model(nn = BINNET(Wet ~ fourier(K = 1, period = "year"))) |>
  glance()
```

---

glance.LOGISTIC    *Glance a LOGISTIC*

---

## Description

Construct a single row summary of the LOGISTIC model.

## Usage

```
## S3 method for class 'LOGISTIC'
glance(x, ...)
```

## Arguments

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

## Details

Contains the R squared (`r_squared`), variance of residuals (`sigma2`), the log-likelihood (`log_lik`), and information criterion (`AIC`, `AICc`, `BIC`).

## Value

A one row tibble summarising the model's fit.

## Examples

```
melb_rain |>
  model(logistic = LOGISTIC(Wet ~ fourier(K = 5, period = "year"))) |>
  glance()
```

---

LOGISTIC *Fit a linear model with time series components*

---

## Description

The model formula will be handled using stats::model.matrix(), and so the the same approach to include interactions in stats::lm() applies when specifying the formula. In addition to stats::lm(), it is possible to include common_xregs in the model formula, such as trend(), season(), and fourier().

## Usage

```
LOGISTIC(formula)
```

## Arguments

| | |
|---|---|
| formula | Model specification. |

## Value

A model specification.

## Specials

**xreg:** Exogenous regressors can be included in a LOGISTIC model without explicitly using the xreg() special. Common exogenous regressor specials as specified in common_xregs can also be used. These regressors are handled using stats::model.frame(), and so interactions and other functionality behaves similarly to stats::lm().

```
xreg(...)
```

    ...    Bare expressions for the exogenous regressors (such as log(x))

## See Also

stats::lm(), stats::model.matrix() Forecasting: Principles and Practices, Time series regression models (chapter 6)

## Examples

```
melb_rain |>
  model(logistic = LOGISTIC(Wet ~ fourier(K = 5, period = "year")))
```

---

| melb_rain | *Daily rainfall in Melbourne, Australia* |

---

## Description

A dataset containing daily rainfall amounts from 1 January 2000 to 31 December 2011.

## Usage

```
melb_rain
```

## Format

A tsibble data frame with 4322 rows and 3 variables.

## Source

Australian Bureau of Meteorology. http://www.bom.gov.au/climate/data/

---

refit.BINNET                 *Refit a BINNET model*

---

### Description

Applies a fitted BINNET model to a new dataset.

### Usage

```
## S3 method for class 'BINNET'
refit(object, new_data, specials = NULL, reestimate = FALSE, ...)
```

### Arguments

| | |
|---|---|
| object | A model for which forecasts are required. |
| new_data | A tsibble containing the time points and exogenous regressors to produce forecasts for. |
| specials | (passed by [fabletools::forecast.mdl_df()](fabletools::forecast.mdl_df())). |
| reestimate | If TRUE, the networks will be initialized with random starting weights to suit the new data. If FALSE, for every network the best individual set of weights found in the pre-estimation process is used as the starting weight vector. |
| ... | Other arguments passed to methods |

### Value

A refitted model.

---

refit.LOGISTIC               *Refit a* LOGISTIC

---

### Description

Applies a fitted LOGISTIC to a new dataset.

### Usage

```
## S3 method for class 'LOGISTIC'
refit(object, new_data, specials = NULL, reestimate = FALSE, ...)
```

## Arguments

| | |
|---|---|
| `object` | A fitted model object. |
| `new_data` | A tsibble containing the time points and exogenous regressors for which a refit is required. |
| `specials` | A list of special functions used in the model, (passed by `fabletools::forecast.mdl_df`). |
| `reestimate` | If TRUE, the networks will be initialized with random starting weights to suit the new data. If FALSE, for every network the best individual set of weights found in the pre-estimation process is used as the starting weight vector. |
| `...` | Other arguments passed to methods |

---

| `residuals.BINNET` | *Extract residuals from a fable model.* |
|---|---|

---

## Description

Extracts the residuals.

## Usage

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

## Arguments

| | |
|---|---|
| `object` | Fitted model |
| `...` | Other arguments ignored |

## Examples

```
melb_rain |>
  model(nn = BINNET(Wet ~ fourier(K = 1, period = "year"))) |>
  residuals()
```

---

| `residuals.LOGISTIC` | *Extract residuals from a fable model.* |
|---|---|

---

## Description

Extracts the residuals.

## Usage

```
## S3 method for class 'LOGISTIC'
residuals(
  object,
 type = c("deviance", "innovation", "pearson", "working", "response", "partial"),
  ...
)
```

## Arguments

| | |
|---|---|
| object | Fitted model |
| type | the type of residuals which should be returned. alternatives are: "deviance" (default), "pearson", "working", "response", and "partial". Can be abbreviated. |
| ... | Other arguments ignored |

## Examples

```
melb_rain |>
  model(logistic = LOGISTIC(Wet ~ fourier(K = 5, period = "year"))) |>
  residuals(type = "deviance")
```

---

tidy.BINNET                *Tidy a fable model*

---

## Description

Returns the coefficients from the model in a `tibble` format.

## Usage

```
## S3 method for class 'BINNET'
tidy(x, ...)
```

## Arguments

| | |
|---|---|
| x | An object to be converted into a tidy [tibble::tibble()](). |
| ... | Additional arguments to tidying method. |

## Examples

```
melb_rain |>
  model(nn = BINNET(Wet ~ fourier(K = 1, period = "year"))) |>
  tidy()
```

---

tidy.LOGISTIC *Tidy a fable model*

---

## Description

Returns the coefficients from the model in a `tibble` format.

## Usage

```
## S3 method for class 'LOGISTIC'
tidy(x, ...)
```

## Arguments

| | |
|---|---|
| x | An object to be converted into a tidy `tibble::tibble()`. |
| ... | Additional arguments to tidying method. |

## Examples

```
melb_rain |>
  model(logistic = LOGISTIC(Wet ~ fourier(K = 5, period = "year"))) |>
  tidy()
```

# Index