

# Package ‘modeltime.resample’

May 9, 2026

**Title** Resampling Tools for Time Series Forecasting

**Version** 0.3.0

**Description** A 'modeltime' extension that implements forecast resampling tools that assess time-based model performance and stability for a single time series, panel data, and cross-sectional time series analysis.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**URL** <https://business-science.github.io/modeltime.resample/>,  
<https://github.com/business-science/modeltime.resample>

**BugReports** <https://github.com/business-science/modeltime.resample/issues>

**Depends** modeltime (>= 0.3.0), R (>= 3.5)

**Imports** tune (>= 2.0.0), rsample, workflows, recipes, yardstick, timetk (>= 2.5.0), tibble, dplyr (>= 1.0.0), tidyr, purrr, stringr, ggplot2 (>= 3.4.0), plotly, cli, magrittr, rlang (>= 0.1.2), progressr, tictoc, hardhat, withr

**Suggests** testthat, tidymodels, tidyquant, glmnet, lubridate, knitr, rmarkdown

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Matt Dancho [aut, cre],  
Business Science [cph]

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

**Repository** CRAN

**Date/Publication** 2025-09-03 16:50:02 UTC

## Contents

|   |   |
|---|---|
| <code>get_target_text_from_resamples</code> . . . . . | 2 |
| <code>m750_training_resamples_fitted</code> . . . . . | 3 |
| <code>modeltime_fit_resamples</code> . . . . .        | 4 |
| <code>modeltime_resample_accuracy</code> . . . . .    | 5 |
| <code>plot_modeltime_resamples</code> . . . . .       | 7 |
| <code>unnest_modeltime_resamples</code> . . . . .     | 9 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>11</b> |
|--------------|-----------|

---

`get_target_text_from_resamples`  
*Gets the target variable as text from unnested resamples*

---

### Description

An internal function used by `unnest_modeltime_resamples()`.

### Usage

```
get_target_text_from_resamples(data, column_before_target = ".row")
```

### Arguments

`data`                    Unnested resample results

`column_before_target`  
                           The text column located before the target variable. This is ".row".

### Examples

```
# The .resample_results column is deeply nested
m750_training_resamples_fitted

# Unnest and prepare the resample predictions for evaluation
unnest_modeltime_resamples(m750_training_resamples_fitted) %>%
  get_target_text_from_resamples()
```

---

`m750_training_resamples_fitted`

*Time Series Cross Validation Resample Predictions (Results) from the M750 Data (Training Set)*

---

**Description**

Time Series Cross Validation Resample Predictions (Results) from the M750 Data (Training Set)

**Usage**

```
m750_training_resamples_fitted
```

**Format**

A Modeltime Table that has been fitted to resamples with predictions in the `.resample_results` column

**Details**

```
m750_training_resamples_fitted <- m750_models %>%  
  modeltime_fit_resamples(  
    resamples = m750_training_resamples,  
    control   = control_resamples(verbose = T)  
  )
```

**See Also**

- [modeltime::m750\\_models](#)
- [modeltime::m750\\_training\\_resamples](#)

**Examples**

```
m750_training_resamples_fitted
```

---

 modeltime\_fit\_resamples

*Fits Models in a Modeltime Table to Resamples*


---

## Description

Resampled predictions are commonly used for:

1. Analyzing accuracy and stability of models
2. As inputs to Ensemble methods (refer to the `modeltime.ensemble` package)

## Usage

```
modeltime_fit_resamples(object, resamples, control = control_resamples())
```

## Arguments

|                        |   |
|------------------------|---|
| <code>object</code>    | A Modeltime Table   |
| <code>resamples</code> | An rset resample object. Used to generate sub-model predictions for the meta-learner. See <code>timetk::time_series_cv()</code> or <code>rsample::vfold_cv()</code> for making resamples. |
| <code>control</code>   | A <code>tune::control_resamples()</code> object to provide control over the resampling process.   |

## Details

The function is a wrapper for `tune::fit_resamples()` to iteratively train and predict models contained in a Modeltime Table on resample objects. One difference between `tune::fit_resamples()` and `modeltime_fit_resamples()` is that predictions are always returned (i.e. `control = tune::control_resamples(save_predictions = TRUE)`). This is needed for `ensemble_model_spec()`.

### Resampled Prediction Accuracy

Calculating Accuracy Metrics on models fit to resamples can help to understand the model performance and stability under different forecasting windows. See `modeltime_resample_accuracy()` for getting resampled prediction accuracy for each model.

### Ensembles

Fitting and Predicting Resamples is useful in creating Stacked Ensembles using `modeltime.ensemble::ensemble_model_spec()`. The sub-model cross-validation predictions are used as the input to the meta-learner model.

## Value

A Modeltime Table (`mdl_time_tbl`) object with a column containing resample results (`.resample_results`)

**Examples**

```

library(tidymodels)
library(modeltime)
library(timetk)
library(magrittr)

# Make resamples
resamples_tscv <- training(m750_splits) %>%
  time_series_cv(
    assess      = "2 years",
    initial     = "5 years",
    skip        = "2 years",
    # Normally we do more than one slice, but this speeds up the example
    slice_limit = 1
  )

# Fit and generate resample predictions
m750_models_resample <- m750_models %>%
  modeltime_fit_resamples(
    resamples = resamples_tscv,
    control   = control_resamples(verbose = TRUE)
  )

# A new data frame is created from the Modeltime Table
# with a column labeled, '.resample_results'
m750_models_resample

```

---

modeltime\_resample\_accuracy

*Calculate Accuracy Metrics from Modeltime Resamples*

---

**Description**

This is a wrapper for `yardstick` that simplifies time series regression accuracy metric calculations from a Modeltime Table that has been resampled and fitted using `modeltime_fit_resamples()`.

**Usage**

```

modeltime_resample_accuracy(
  object,
  summary_fns = mean,
  metric_set = default_forecast_accuracy_metric_set(),
  ...
)

```

**Arguments**

|             |  |
|-------------|--|
| object      | a Modeltime Table with a column '.resample_results' (the output of <code>modeltime_fit_resamples()</code> )  |
| summary_fns | One or more functions to analyze resamples. The default is <code>mean()</code> . Possible values are: <ul style="list-style-type: none"> <li>• NULL, to returns the resamples untransformed.</li> <li>• A function, e.g. <code>mean</code>.</li> <li>• A purrr-style lambda, e.g. <code>~ mean(.x, na.rm = TRUE)</code></li> <li>• A list of functions/lambda, e.g. <code>list(mean = mean, sd = sd)</code></li> </ul> |
| metric_set  | A <code>yardstick::metric_set()</code> that is used to summarize one or more forecast accuracy (regression) metrics.   |
| ...         | Additional arguments passed to the function calls in <code>summary_fns</code> .  |

**Details****# Default Accuracy Metrics**

The following accuracy metrics are included by default via `modeltime::default_forecast_accuracy_metric_set()`:

- MAE - Mean absolute error, `yardstick::mae()`
- MAPE - Mean absolute percentage error, `yardstick::mape()`
- MASE - Mean absolute scaled error, `yardstick::mase()`
- SMAPE - Symmetric mean absolute percentage error, `yardstick::smape()`
- RMSE - Root mean squared error, `yardstick::rmse()`
- RSQ - R-squared, `yardstick::rsq()`

**Summary Functions**

By default, `modeltime_resample_accuracy()` returns the *average* accuracy metrics for each re-sample prediction.

The user can change this default behavior using `summary_fns`. Simply pass one or more Summary Functions. Internally, the functions are passed to `dplyr::across(.fns)`, which applies the summary functions.

**Returning Unsummarized Results**

You can pass `summary_fns = NULL` to return unsummarized results by `.resample_id`.

**Professional Tables (Interactive & Static)**

Use `modeltime::table_modeltime_accuracy()` to format the results for reporting in `reactable` (interactive) or `gt` (static) formats, which are perfect for Shiny Apps (interactive) and PDF Reports (static).

**Examples**

```
library(modeltime)

# Mean (Default)
m750_training_resamples_fitted %>%
  modeltime_resample_accuracy() %>%
```

```

    table_modeltime_accuracy(.interactive = FALSE)

# Mean and Standard Deviation
m750_training_resamples_fitted %>%
  modeltime_resample_accuracy(
    summary_fns = list(mean = mean, sd = sd)
  ) %>%
  table_modeltime_accuracy(.interactive = FALSE)

# When summary_fns = NULL, returns the unsummarized resample results
m750_training_resamples_fitted %>%
  modeltime_resample_accuracy(
    summary_fns = NULL
  )

```

---

plot\_modeltime\_resamples

*Interactive Resampling Accuracy Plots*


---

## Description

A convenient plotting function for visualizing resampling accuracy by resample set for each model in a Modeltime Table.

## Usage

```

plot_modeltime_resamples(
  .data,
  .metric_set = default_forecast_accuracy_metric_set(),
  .summary_fn = mean,
  ...,
  .facet_ncol = NULL,
  .facet_scales = "free_x",
  .point_show = TRUE,
  .point_size = 1,
  .point_shape = 16,
  .point_alpha = 1,
  .summary_line_show = TRUE,
  .summary_line_size = 0.5,
  .summary_line_type = 1,
  .summary_line_alpha = 1,
  .x_intercept = NULL,
  .x_intercept_color = "red",
  .x_intercept_size = 0.5,
  .legend_show = TRUE,
  .legend_max_width = 40,
  .title = "Resample Accuracy Plot",

```

```

.x_lab = "",
.y_lab = "",
.color_lab = "Legend",
.interactive = TRUE
)

```

## Arguments

|                                  |   |
|----------------------------------|---|
| <code>.data</code>               | A modeltime table that includes a column <code>.resample_results</code> containing the resample results. See <code>modeltime_fit_resamples()</code> for more information.                             |
| <code>.metric_set</code>         | A <code>yardstick::metric_set()</code> that is used to summarize one or more forecast accuracy (regression) metrics. See <code>modeltime::default_forecast_accuracy_metric_set()</code> for defaults. |
| <code>.summary_fn</code>         | A single summary function that is applied to aggregate the metrics across resample sets. Default: <code>mean</code> .   |
| <code>...</code>                 | Additional arguments passed to the <code>.summary_fn</code> .   |
| <code>.facet_ncol</code>         | Default: <code>NULL</code> . The number of facet columns.   |
| <code>.facet_scales</code>       | Default: <code>free_x</code> .  |
| <code>.point_show</code>         | Whether or not to show the individual points for each combination of models and metrics. Default: <code>TRUE</code> .   |
| <code>.point_size</code>         | Controls the point size. Default: <code>1</code> .  |
| <code>.point_shape</code>        | Controls the point shape. Default: <code>16</code> .  |
| <code>.point_alpha</code>        | Controls the opacity of the points. Default: <code>1</code> (full opacity).   |
| <code>.summary_line_show</code>  | Whether or not to show the summary lines. Default: <code>TRUE</code> .  |
| <code>.summary_line_size</code>  | Controls the summary line width. Default: <code>0.5</code> .  |
| <code>.summary_line_type</code>  | Controls the summary line type. Default: <code>1</code> .   |
| <code>.summary_line_alpha</code> | Controls the summary line opacity. Default: <code>1</code> (full opacity).  |
| <code>.x_intercept</code>        | Numeric. Adds an x-intercept at a location (e.g. <code>0</code> ). Default: <code>NULL</code> .   |
| <code>.x_intercept_color</code>  | Controls the x-intercept color. Default: <code>"red"</code> .   |
| <code>.x_intercept_size</code>   | Controls the x-intercept linewidth. Default: <code>0.5</code> .   |
| <code>.legend_show</code>        | Logical. Whether or not to show the legend. Can save space with long model descriptions.  |
| <code>.legend_max_width</code>   | Numeric. The width of truncation to apply to the legend text.   |
| <code>.title</code>              | Title for the plot  |
| <code>.x_lab</code>              | X-axis label for the plot   |
| <code>.y_lab</code>              | Y-axis label for the plot   |
| <code>.color_lab</code>          | Legend label if a <code>color_var</code> is used.   |
| <code>.interactive</code>        | Returns either a static ( <code>ggplot2</code> ) visualization or an interactive ( <code>plotly</code> ) visualization  |

## Details

See `modeltime::default_forecast_accuracy_metric_set()` for defaults.

## Examples

```
m750_training_resamples_fitted %>%  
  plot_modeltime_resamples(  
    .interactive = FALSE  
  )
```

---

unnest\_modeltime\_resamples

*Unnests the Results of Modeltime Fit Resamples*

---

## Description

An internal function used by `modeltime_resample_accuracy()`.

## Usage

```
unnest_modeltime_resamples(object)
```

## Arguments

`object` A Modeltime Table that has a column `'resample_results'`

## Details

The following data columns are unnested and prepared for evaluation:

- `.row_id` - A unique identifier to compare observations.
- `.resample_id` - A unique identifier given to the resample iteration.
- `.model_id` and `.model_desc` - Modeltime Model ID and Description
- `.pred` - The Resample Prediction Value
- `.row` - The actual row value from the original dataset
- *Actual Value Column* - The name changes to target variable name in dataset

## Value

Tibble with columns for `'row_id'`, `'resample_id'`, `'model_id'`, `'model_desc'`, `'pred'`, `'row'`, and actual value name from the data set

**Examples**

```
# The .resample_results column is deeply nested
m750_training_resamples_fitted

# Unnest and prepare the resample predictions for evaluation
unnest_modeltime_resamples(m750_training_resamples_fitted)
```

# Index

## \* datasets

m750\_training\_resamples\_fitted, 3

get\_target\_text\_from\_resamples, 2

m750\_training\_resamples\_fitted, 3

modeltime::default\_forecast\_accuracy\_metric\_set(),  
6, 8, 9

modeltime::m750\_models, 3

modeltime::m750\_training\_resamples, 3

modeltime::table\_modeltime\_accuracy(),  
6

modeltime\_fit\_resamples, 4

modeltime\_fit\_resamples(), 5, 6, 8

modeltime\_resample\_accuracy, 5

modeltime\_resample\_accuracy(), 4, 9

plot\_modeltime\_resamples, 7

rsample::vfold\_cv(), 4

timetk::time\_series\_cv(), 4

tune::control\_resamples(), 4

unnest\_modeltime\_resamples, 9

unnest\_modeltime\_resamples(), 2

yardstick::mae(), 6

yardstick::mape(), 6

yardstick::mase(), 6

yardstick::rmse(), 6

yardstick::rsq(), 6

yardstick::smape(), 6