

# Package: broomstick (via r-universe)

September 30, 2024

**Type** Package

**Title** Convert Decision Tree Objects into Tidy Data Frames

**Version** 0.1.2.9200

**Description** Convert Decision Tree objects into tidy data frames, by using the framework laid out by the package broom, this means that decision tree output can be easily reshaped, processed, and combined with tools like 'dplyr', 'tidyr' and 'ggplot2'. Like the package broom, broomstick provides three S3 generics: tidy, to summarise decision tree specific features - tidy returns the variable importance table; augment adds columns to the original data such as predictions and residuals; and glance, which provides a one-row summary of model-level statistics.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** dplyr, tibble, magrittr, purrr, tidyr, generics, rlang, broom

**RoxygenNote** 7.1.0

**Suggests** gbm, testthat, randomForest, MASS, rpart, covr

**Roxygen** list(markdown = TRUE)

**Repository** <https://njtierney.r-universe.dev>

**RemoteUrl** <https://github.com/njtierney/broomstick>

**RemoteRef** HEAD

**RemoteSha** 5215feed37c47d31f3a6a422e1e54525c66f3ce1

## Contents

augment.randomForest . . . . .	2
augment.rpart . . . . .	3
broomstick . . . . .	4
tidy.gbm . . . . .	4
tidy.rpart . . . . .	5

---

augment.randomForest *Tidying methods for a randomForest model*

---

### Description

These methods tidy the variable importance of a random forest model summary, augment the original data with information on the fitted values/classifications and error, and construct a one-row glance of the model's statistics.

### Usage

```
## S3 method for class 'randomForest'
augment(x, data = NULL, ...)

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

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

### Arguments

x	randomForest object
data	Model data for use by <code>augment.randomForest()</code> .
...	Additional arguments (ignored)

### Value

`augment.randomForest` returns the original data with additional columns:

.oob_times	The number of trees for which the given case was "out of bag". See <code>randomForest::randomForest()</code> for more details.
.fitted	The fitted value or class.

`augment` returns additional columns for classification and unsupervised trees:

.votes	For each case, the voting results, with one column per class.
.local_var_imp	The casewise variable importance, stored as data frames in a nested list-column, with one row per variable in the model. Only present if the model was created with <code>importance = TRUE</code>

`glance.randomForest` returns a data.frame with the following columns for regression trees:

mse	The average mean squared error across all trees.
rsq	The average pseudo-R-squared across all trees. See <code>randomForest::randomForest()</code> for more information.

For classification trees: one row per class, with the following columns:

```
precision
recall
accuracy
f_measure
```

All tidying methods return a `data.frame` without rownames. The structure depends on the method chosen.

`tidy.randomForest` returns one row for each model term, with the following columns:

```
term          The term in the randomForest model
MeanDecreaseAccuracy  A measure of variable importance. See randomForest::randomForest\(\) for
more information. Only present if the model was created with importance =
TRUE
MeanDecreaseGini      A measure of variable importance. See randomForest::randomForest\(\) for
more information.
MeanDecreaseAccuracy_sd  Standard deviation of MeanDecreaseAccuracy. See randomForest::randomForest\(\)
for more information. Only present if the model was created with importance
= TRUE
classwise_importance  Classwise variable importance for each term, stored as data frames in a nested
list-column, with one row per class. Only present if the model was created with
importance = TRUE
```

---

augment.rpart	<i>Augment your model object</i>
---------------	----------------------------------

---

## Description

Augment your model object

## Usage

```
## S3 method for class 'rpart'
augment(x, data = NULL, newdata = NULL, ...)
```

## Arguments

```
x          rpart model
data       data.frame from the model
newdata    new data to use for predictions, residuals, etc.
...        extra arguments to pass
```

**Value**

`augment.rpart` returns the original data with additional columns:

- `.fitted`: The fitted value or class.
- `.resid`: only given when the same data as was used for the model is provided.

**Examples**

```
library(rpart)
rpart_fit <- rpart(Sepal.Width ~ ., iris)
augment(rpart_fit)
```

---

broomstick

*Convert Decision Tree Analysis Objects into Tidy Data Frames*

---

**Description**

Convert decision tree analysis objects from R into tidy data frames, so that they can more easily be combined, reshaped and otherwise processed with tools like `dplyr`, `tidyr` and `ggplot2`. The package provides three S3 generics: `tidy`, which summarizes a model's statistical findings such as coefficients of a regression; `augment`, which adds columns to the original data such as predictions, residuals and cluster assignments; and `glance`, which provides a one-row summary of model-level statistics.

---

tidy.gbm

*tidy up the model summary of gbm*

---

**Description**

`tidy` returns a tibble of variable importance for the `rpart` package

**Usage**

```
## S3 method for class 'gbm'
tidy(x, n_trees = x$n.trees, scale = FALSE, sort = TRUE, normalise = TRUE, ...)
```

**Arguments**

<code>x</code>	A <code>gbm</code> model
<code>n_trees</code>	integer. (optional) Number of trees to use for computing relative importance. Default is the number of trees in <code>x\$n.trees</code> . If not provided, a guess is made using the heuristic: If a test set was used in fitting, the number of trees resulting in lowest test set error will be used; else, if cross-validation was performed, the number of trees resulting in lowest cross-validation error will be used; otherwise, all trees will be used.
<code>scale</code>	(optional) Should importance be scaled? Default is <code>FALSE</code>

sort (optional) Should results be sorted? Default is TRUE  
 normalise (optional) Should results be normalised to sum to 100? Default is TRUE  
 ... extra functions or arguments

**Value**

A tibble containing the importance score for each variable

**Examples**

```

# retrieve a tibble of the variable importance from an gbm model

library(gbm)
library(MASS)
fit_gbm <- gbm(calories ~., data = UScereal)

tidy(fit_gbm)

```

---

tidy.rpart

*tidy up the model summary of rpart*


---

**Description**

tidy returns a tibble of variable importance for the rpart package

**Usage**

```

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

```

**Arguments**

x An rpart model  
 ... extra functions or arguments

**Value**

A tibble containing the importance score for each variable

**Examples**

```

# retrieve a tibble of the variable importance from an rpart model

library(rpart)
fit_rpart <- rpart(Kyphosis ~ Age + Number + Start, data = kyphosis)

tidy(fit_rpart)

```

# Index

`augment.randomForest`, 2  
`augment.randomForest()`, 2  
`augment.rpart`, 3  
  
`broomstick`, 4  
`broomstick-package (broomstick)`, 4  
  
`glance.randomForest`  
    (`augment.randomForest`), 2  
  
`randomForest::randomForest()`, 2, 3  
`rf_tidiers (augment.randomForest)`, 2  
  
`tidy.gbm`, 4  
`tidy.randomForest`  
    (`augment.randomForest`), 2  
`tidy.rpart`, 5