Package ‘modeltests’

January 13, 2021

Type Package
Title Testing Infrastructure for Broom Model Generics
Version 0.1.4
Description Provides a number of testthat tests that can be used to verify that tidy(), glance() and augment() methods meet consistent specifications. This allows methods for the same generic to be spread across multiple packages, since all of those packages can make the same guarantees to users about returned objects.
License MIT + file LICENSE
URL https://github.com/alexpghayes/modeltests
BugReports https://github.com/alexpghayes/modeltests/issues
Depends R (>= 3.1)
Imports dplyr (>= 0.7.6), generics, purrr (>= 0.2.5), testthat (>= 2.0.0), tibble (>= 3.0.0)
Suggests covr
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
NeedsCompilation no
Author Alex Hayes [aut, cre] (<https://orcid.org/0000-0002-4985-5160>), Simon Couch [aut]
Maintainer Alex Hayes <alexpghayes@gmail.com>
Repository CRAN
Date/Publication 2021-01-13 18:30:05 UTC

R topics documented:

acceptable_augment_colnames ........................................... 2
argument_glossary ...................................................... 3
acceptable_augment_colnames

Determine acceptable names for augment output

Description

Given a data frame (or tibble), and a model object, makes a character vector of acceptable column names for augment output. This includes:

- Any column names of the passed dataset
- Any syntactically correct column names generated by calling `stats::model.frame()` on the object in question.

Usage

```r
acceptable_augment_colnames(object, passed_data)
```

Arguments

- `object` A model object.
- `passed_data` The dataset used to create the model object.

Value

A vector of colnames that are acceptable in augment output.
**argument_glossary**

### Allowed argument names in tidiers

**Description**

Allowed argument names in tidiers

**Usage**

```
argument_glossary
```

**Format**

A tibble with 3 variables:

- **method**  One of "glance", "augment" or "tidy".
- **argument** Character name of allowed argument name.
- **description** Character description of argument use.

**Examples**

```
argument_glossary
```

---

**check_arguments**

### Check that tidying methods use allowed argument names

**Description**

Call this function to perform tests. If a tests fails, an informative error will be thrown. Otherwise silent.

Tests when `strict = FALSE`:

- None

Tests when `strict = TRUE`:

- `tidy_method` has a `conf.int` argument if it has a `conf.level` argument.
- `tidy_method` has a `conf.level` argument if it has a `conf.int` argument.
- `conf.int` defaults to FALSE when present.
- `conf.level` defaults to ‘0.95” when present.
- `exponentiate` defaults to FALSE when present.
- All arguments to `tidy_method` are listed in the **argument_glossary**.
check_augment_function

Usage

check_arguments(tidy_method, strict = TRUE)

Arguments

tidy_method  A tidying method. For example: glance.Arima.
strict  Logical indicating whether the strict version of tests should be used. Defaults to TRUE.

Value

An invisible NULL. This function should be called for side effects, not return values.

See Also

testthat, testthat::expect_true()

check_augment_function

Check an augment method

Description

Call this function to perform tests. If a tests fails, an informative error will be thrown. Otherwise silent.

Test when strict = FALSE:

• aug(model, data = data) passes check_tibble()
• aug(model, newdata = newdata) passes check_tibble()

Additional tests when strict = TRUE:

• aug(model, data = data) passes check_augment_data_specification().
• aug(model, newdata = newdata) passes check_augment_data_specification().
• aug(model, newdata = newdata) passes check_augment_data_specification() with add_missing = TRUE.
• If aug has a newdata argument, the newdata argument takes precedence over a data argument, i.e. calls check_augment_newdata_precedence().
• aug either gives an informative error or produces a reasonable tibble, i.e. calls check_augment_no_data().

Note that it doesn’t make sense to test that aug(model, data = data) passes check_augment_data_specification() with add_missing = TRUE. This is because the user is already guaranteeing that data is the original dataset used to create model.

Usage

check_augment_function(aug, model, data = NULL, newdata = NULL, strict = TRUE)
check_augment_newdata_precedence

Arguments

<table>
<thead>
<tr>
<th>Aug</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aug</td>
<td>An augment method. For example, augment.betareg.</td>
</tr>
<tr>
<td>model</td>
<td>A fit model object to call the augment method on.</td>
</tr>
<tr>
<td>data</td>
<td>A data frame or tibble to use when testing aug.</td>
</tr>
<tr>
<td>newdata</td>
<td>A dataset to use to check the newdata behavior, ideally distinct for the dataset used to check the data behavior.</td>
</tr>
<tr>
<td>strict</td>
<td>Logical indicating whether the strict version of tests should be used. Defaults to TRUE.</td>
</tr>
</tbody>
</table>

Value

An invisible NULL. This function should be called for side effects, not return values.

Description

Call this function to perform tests. If a tests fails, an informative error will be thrown. Otherwise silent.

Usage

check_augment_newdata_precedence(aug, model, data, strict = TRUE)

Arguments

<table>
<thead>
<tr>
<th>Aug</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>aug</td>
<td>An augment method. For example, augment.betareg.</td>
</tr>
<tr>
<td>model</td>
<td>A fit model object to call the augment method on.</td>
</tr>
<tr>
<td>data</td>
<td>A data frame or tibble to use when testing aug.</td>
</tr>
<tr>
<td>strict</td>
<td>Logical indicating whether the strict version of tests should be used. Defaults to TRUE.</td>
</tr>
</tbody>
</table>

Value

An invisible NULL. This function should be called for side effects, not return values.
check_augment_no_data  

Check an augment method when no data or newdata is passed

Description

Call this function to perform tests. If a tests fails, an informative error will be thrown. Otherwise silent.

Test when strict = FALSE:

• None

Additional tests when strict = TRUE:

• aug(model) either returns an informative error or produces output that passes check_tibble().
• If the output passes check_tibble, will issue warning when:
  – Augmented data is missing rows from original data.
  – Augmented data is missing columns from original data.
  – Original data has rownames but augmented data is missing .rownames column.

Usage

check_augment_no_data(aug, model, passed_data, strict = TRUE)

Arguments

aug  An augment method. For example, augment.betareg.
model  A fit model object to call the augment method on.
passed_data  The dataset that model was original fit on that aug should try to reconstruct when neither data nor newdata is specified.
strict  Logical indicating whether the strict version of tests should be used. Defaults to TRUE.

Value

An invisible NULL. This function should be called for side effects, not return values.
check_dims

Check that tibble has expected dimensions.

Description

Check that tibble has expected dimensions.

Usage

check_dims(data, expected_rows = NULL, expected_cols = NULL)

Arguments

data          A tibble or data frame.
expected_rows Expected number of rows of tibble.
expected_cols Expected number of columns of tibble.

Examples

check_dims(iris, expected_rows = 150)

check_glance_outputs

Check the output of a glance method

Description

Call this function to perform tests. If a tests fails, an informative error will be thrown. Otherwise silent.

Tests when strict = FALSE:

• Each item passed to ... passes check_tibble()
• Each item passed to ... has exactly 1 row.

Additional tests when strict = TRUE:

• Column names and order agree across all elements of ....

Usage

check_glance_outputs(..., strict = TRUE)
check_tibble

Arguments

Output returned from calls to (the same) glance method.

strict Logical indicating whether the strict version of tests should be used. Defaults to TRUE.

Value

An invisible NULL. This function should be called for side effects, not return values.

See Also

check_tibble()

Description

Call this function to perform tests. If a tests fails, an informative error will be thrown. Otherwise silent.

Tests when strict = FALSE:

• output is a tibble.

Additional tests when strict = TRUE:

• columns are listed in the column_glossary.

Usage

check_tibble(output, method, columns = colnames(output), strict = TRUE)

Arguments

output Object returned from tidy(), augment() or glance().

method One of "tidy", "augment" or "glance". Determines which set of column name checks are applied.

columns The names of the columns in the output data frame. Defaults to the column names of output. Useful when checking augment() when you only want to check the new columns in the data frame, as opposed to all columns.

strict Logical indicating whether the strict version of tests should be used. Defaults to TRUE.

Details

Do not call directly. Helper function used by check_tidy_output(), check_glance_outputs() and check_augment_function().
check_tidy_output

Value

An invisible NULL. This function should be called for side effects, not return values.

check_tidy_output  Check the output of a tidy method

Description

Call this function to perform tests. If a tests fails, an informative error will be thrown. Otherwise silent.
A thin wrapper around check_tibble().

Usage

check_tidy_output(td, strict = TRUE)

Arguments

td  Output from a tidy method.
strict  Logical indicating whether the strict version of tests should be used. Defaults to TRUE.

Value

An invisible NULL. This function should be called for side effects, not return values.

column_glossary  Allowed column names in tidied tibbles

Description

Allowed column names in tidied tibbles

Usage

column_glossary

Format

A tibble with 4 variables:

method  One of "glance", "augment" or "tidy".
column  Character name of allowed output column.
description  Character description of expected column contents.

Examples

column_glossary
has_rownames

Check whether or not a data-frame-like object has rownames

Description
Check whether or not a data-frame-like object has rownames

Usage
has_rownames(df)

Arguments
df A data frame

Value
Logical indicating if df has rownames. If df is a tibble, returns FALSE. If df is a data.frame, return FALSE if the rownames are simply row numbers. If the rownames are anything other than the return row numbers, returns TRUE.
Index

* datasets
  argument_glossary, 3
  column_glossary, 9

acceptable_augment_colnames, 2
argument_glossary, 3, 3
augment(), 8

check_arguments, 3
check_augment_data_specification(), 4
check_augment_function, 4
check_augment_function(), 8
check_augment_newdata_precedence, 5
check_augment_newdata_precedence(), 4
check_augment_no_data, 6
check_augment_no_data(), 4
check_dims, 7
check_glance_outputs, 7
check_glance_outputs(), 8
check_tibble, 8
check_tibble(), 4, 6–9
check_tidy_output, 9
check_tidy_output(), 8
column_glossary, 8, 9

glance, 8
glance(), 8

has_rownames, 10

stats::model.frame(), 2

testthat, 4
testthat::expect_true(), 4
tidy(), 8