

# Package ‘tidypredict’

January 10, 2019

**Version** 0.3.0

**Title** Run Predictions Inside the Database

**Description** It parses a fitted 'R' model object, and returns a formula in 'Tidy Eval' code that calculates the predictions. It works with several databases back-ends because it leverages 'dplyr' and 'dbplyr' for the final 'SQL' translation of the algorithm. It currently supports `lm()`, `glm()`, `randomForest()`, `ranger()` and `earth()` models.

**Depends** R (>= 3.1)

**Imports** dplyr(>= 0.7), rlang, purrr, knitr

**Suggests** dbplyr, testthat, randomForest, ranger, earth, rmarkdown, nycflights13, RSQLite, methods, DBI, covr

**License** GPL-3

**URL** <http://tidypredict.netlify.com/>

**BugReports** <https://github.com/edgararuiz/tidypredict/issues>

**RoxygenNote** 6.1.1

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Edgar Ruiz [aut, cre]

**Maintainer** Edgar Ruiz <edgar@rstudio.com>

**Repository** CRAN

**Date/Publication** 2019-01-10 07:00:03 UTC

## R topics documented:

acceptable_formula . . . . .	2
parse_model . . . . .	2
tidypredict_fit . . . . .	3
tidypredict_interval . . . . .	3
tidypredict_test . . . . .	4
tidypredict_to_column . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

acceptable_formula	<i>Checks that the formula can be parsed</i>
--------------------	--

---

**Description**

Uses an S3 method to check that a given formula can be parsed based on its class. It currently scans for contrasts that are not supported and in-line functions. (e.g: `lm(wt ~ as.factor(am))`). Since this function is meant for function interaction, as opposed to human interaction, a successful check is silent.

**Usage**

```
acceptable_formula(model)
```

**Arguments**

model	An R model object
-------	-------------------

**Examples**

```
model <- lm(mpg ~ wt, mtcars)
acceptable_formula(model)
```

---

parse_model	<i>Converts an R model object into a table</i>
-------------	--

---

**Description**

It parses a fitted R model's structure and extracts the components needed to create a dplyr formula for prediction. The function also creates a data frame using an specific format so that other functions in the future can also pass parsed tables to a given formula creating function.

**Usage**

```
parse_model(model)
```

**Arguments**

model	An R model object.
-------	--------------------

**Examples**

```
library(dplyr)
df <- mutate(mtcars, cyl = paste0("cyl", cyl))
model <- lm(mpg ~ wt + cyl * disp, offset = am, data = df)
parse_model(model)
```

---

tidypredict_fit	Returns a Tidy Eval formula to calculate fitted values
-----------------	--

---

### Description

It parses a model or uses an already parsed model to return a Tidy Eval formula that can then be used inside a dplyr command.

### Usage

```
tidypredict_fit(model)
```

### Arguments

model            An R model or a list with a parsed model.

### Examples

```
model <- lm(mpg ~ wt + cyl * disp, offset = am, data = mtcars)
tidypredict_fit(model)
```

---

tidypredict_interval	Returns a Tidy Eval formula to calculate prediction interval
----------------------	--

---

### Description

It parses a model or uses an already parsed model to return a Tidy Eval formula that can then be used inside a dplyr command.

### Usage

```
tidypredict_interval(model, interval = 0.95)
```

### Arguments

model            An R model or a list with a parsed model  
interval         The prediction interval, defaults to 0.95

### Details

The result still has to be added to the fit to obtain the upper bound, and subtracted from fit to obtain the lower bound.

**Examples**

```
model <- lm(mpg ~ wt + cyl * disp, offset = am, data = mtcars)
tidypredict_interval(model)
```

---

tidypredict_test	<i>Tests base predict function against tidypredict</i>
------------------	--

---

**Description**

Compares the results of predict() and tidypredict\_to\_column() functions.

**Usage**

```
tidypredict_test(model, df = model$model, threshold = 1e-12,
  include_intervals = FALSE, max_rows = NULL)
```

**Arguments**

model	An R model or a list with a parsed model. It currently supports lm(), glm() and randomForest() models.
df	A data frame that contains all of the needed fields to run the prediction. It defaults to the "model" data frame object inside the model object.
threshold	The number that a given result difference, between predict() and tidypredict_to_column() should not exceed. For continuous predictions, the default value is 0.000000000001 (1e-12), for categorical predictions, the default value is 0.
include_intervals	Switch to indicate if the prediction intervals should be included in the test. It defaults to FALSE.
max_rows	The number of rows in the object passed in the df argument. Highly recommended for large data sets.

**Examples**

```
model <- lm(mpg ~ wt + cyl * disp, offset = am, data = mtcars)
tidypredict_test(model)
```

---

tidypredict\_to\_column *Adds the prediction columns to a piped command set*

---

**Description**

Adds a new column with the results from `tidypredict_fit()` to a piped command set. If `add_interval` is set to `TRUE`, then it will add two additional columns, one for the lower and another for the upper prediction interval bounds.

**Usage**

```
tidypredict_to_column(df, model, add_interval = FALSE, interval = 0.95,  
  vars = c("fit", "upper", "lower"))
```

**Arguments**

<code>df</code>	A data.frame or tibble
<code>model</code>	An R model or a parsed model inside a data frame
<code>add_interval</code>	Switch that indicates if the prediction interval columns should be added. Defaults to <code>FALSE</code>
<code>interval</code>	The prediction interval, defaults to 0.95. It is ignored if <code>add_interval</code> is set to <code>FALSE</code>
<code>vars</code>	The name of the variables that this function will produce. It defaults to "fit", "upper", and "lower".

# Index

`acceptable_formula`, [2](#)

`parse_model`, [2](#)

`tidypredict_fit`, [3](#)

`tidypredict_interval`, [3](#)

`tidypredict_test`, [4](#)

`tidypredict_to_column`, [5](#)