

# Package ‘modelfactory’

January 31, 2024

**Title** Combine Statistical Models into a Tibble for Comparison

**Version** 1.0.0

**Description** Statisticians often want to compare the fit of different models on the same data set. However, this usually involves a lot of manual code to fish items out of `summary()` or plain model objects. 'modelfactory' offers the capability to pass multiple models in and get out metrics or coefficients for quick comparison with easy-to-remember syntax.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Suggests** testthat (>= 3.0.0), lme4

**Config/testthat/edition** 3

**Imports** dplyr, MASS, stats, tibble

**URL** <https://willtirone.github.io/modelfactory/>,  
<https://github.com/WillTirone/modelfactory>

**BugReports** <https://github.com/WillTirone/modelfactory/issues>

**NeedsCompilation** no

**Author** Will Tirone [aut, cre, cph]

**Maintainer** Will Tirone <will.tirone1@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-01-31 14:40:06 UTC

## R topics documented:

stack_coeff	2
stack_metrics	3

<b>Index</b>	<b>4</b>
--------------	----------

---

stack_coeff	<i>Stack coefficients, confidence intervals, and standard errors for n models.</i>
-------------	--

---

## Description

stack\_coeff() takes several lm or glm models, pulls out their coefficients, standard errors, and confidence intervals, and stacks everything into a [tibble\(\)](#) for easy comparison across models.

## Usage

```
stack_coeff(..., ci = 0.95)
```

## Arguments

...            lm or glm models to summarize and combine.  
 ci            width of confidence, default = 0.95.

## Value

A [tibble\(\)](#) with coefficients, confidence intervals, and standard errors.

## Examples

```
# multiple lm example -----
lm_1 = lm(mpg ~ cyl + disp + hp, data = mtcars)
lm_2 = lm(mpg ~ hp + drat + wt, data = mtcars)
lm_3 = lm(mpg ~ ., data = mtcars)
lm_combined = stack_coeff(lm_1, lm_2, lm_3)
lm_combined

# sometimes you might just want 1 model's summary -----
single_lm = stack_coeff(lm_1)
single_lm

# glm example -----
glm_1 = glm(vs ~ drat + hp, data = mtcars)
glm_2 = glm(vs ~ wt + qsec, data = mtcars)
glm_3 = glm(vs ~ ., data = mtcars)
glm_combined = stack_coeff(glm_1, glm_2, glm_3)
glm_combined
```

---

`stack_metrics`*Combine model metrics for n number of lm, glm, and lmer models*

---

## Description

`stack_metrics()` calculates basic model metrics like MSE for the models passed in, then stacks them in a dataframe for comparison. This supports `lm`, `glm`, and `lmer` models, and different metrics are calculated for each. This does not perform model selection based on a given criteria, but it makes the tedious task of, say, comparing R-squared across several models very easy.

## Usage

```
stack_metrics(...)
```

## Arguments

... `lm`, `glm`, or `lmer` models to summarize and combine.

## Value

A `tibble()` that includes a variety of evaluation metrics.

## Examples

```
# lm example -----
lm_1 = lm(mpg ~ cyl + disp + hp, data = mtcars)
lm_2 = lm(mpg ~ hp + drat + wt, data = mtcars)
lm_3 = lm(mpg ~ ., data = mtcars)
lm_combined = stack_metrics(lm_1, lm_2, lm_3)
lm_combined

# glm example -----
glm_1 = glm(vs ~ drat + hp, data = mtcars)
glm_2 = glm(vs ~ wt + qsec, data = mtcars)
glm_3 = glm(vs ~ ., data = mtcars)
glm_combined = stack_metrics(glm_1, glm_2, glm_3)
glm_combined

# lme4 example -----
lmer_1 = lme4::lmer(Sepal.Length ~ (1 | Species), data = iris)
lmer_2 = lme4::lmer(Sepal.Length ~ (1 | Species) + Petal.Length, data = iris)
lmer_combined = stack_metrics(lmer_1, lmer_2)
lmer_combined
```

# Index

`stack_coeff`, 2  
`stack_metrics`, 3  
`tibble()`, 2, 3