

# Package ‘broom.helpers’

September 18, 2020

**Title** Helpers for Model Coefficients Tibbles

**Version** 1.0.0

**Description** Provides suite of functions to work with regression model 'broom::tidy()' tibbles. The suite includes functions to group regression model terms by variable, insert reference and header rows for categorical variables, add variable labels, and more.

**License** GPL-3

**URL** <https://larmarange.github.io/broom.helpers/>

**BugReports** <https://github.com/larmarange/broom.helpers/issues>

**Imports** broom, dplyr, labelled, lifecycle, purrr, stats, stringr, tibble, usethis

**Suggests** broom.mixed, covr, datasets, forcats, gam, gee, geepack, ggplot2, gsummary, knitr, lavaan, lme4, MASS, nnet, ordinal, rmarkdown, survey, survival, testthat

**VignetteBuilder** knitr

**RdMacros** lifecycle

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Joseph Larmarange [aut, cre] (<<https://orcid.org/0000-0001-7097-700X>>), Daniel D. Sjoberg [aut] (<<https://orcid.org/0000-0003-0862-2018>>)

**Maintainer** Joseph Larmarange <joseph@larmarange.net>

**Repository** CRAN

**Date/Publication** 2020-09-18 12:00:24 UTC

**R topics documented:**

|   |    |
|---|----|
| model_get_contrasts . . . . .                 | 2  |
| model_get_model_frame . . . . .               | 3  |
| model_get_model_matrix . . . . .              | 4  |
| model_get_xlevels . . . . .                   | 5  |
| model_identify_variables . . . . .            | 5  |
| model_list_contrasts . . . . .                | 7  |
| model_list_terms_levels . . . . .             | 8  |
| model_list_variables . . . . .                | 9  |
| tidy_add_contrasts . . . . .                  | 10 |
| tidy_add_estimate_to_reference_rows . . . . . | 11 |
| tidy_add_header_rows . . . . .                | 13 |
| tidy_add_reference_rows . . . . .             | 14 |
| tidy_add_term_labels . . . . .                | 16 |
| tidy_add_variable_labels . . . . .            | 17 |
| tidy_attach_model . . . . .                   | 18 |
| tidy_identify_variables . . . . .             | 19 |
| tidy_plus_plus . . . . .                      | 21 |
| tidy_remove_intercept . . . . .               | 23 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>25</b> |
|--------------|-----------|

---

|                     |  |
|---------------------|--|
| model_get_contrasts | <i>Get contrasts used in the model</i> |
|---------------------|--|

---

**Description**

Get contrasts used in the model

**Usage**

```
model_get_contrasts(model)
```

```
## Default S3 method:
model_get_contrasts(model)
```

**Arguments**

model            a model object

**See Also**

Other model\_helpers: [model\\_get\\_model\\_frame\(\)](#), [model\\_get\\_model\\_matrix\(\)](#), [model\\_get\\_xlevels\(\)](#), [model\\_identify\\_variables\(\)](#), [model\\_list\\_contrasts\(\)](#), [model\\_list\\_terms\\_levels\(\)](#), [model\\_list\\_variables\(\)](#)

## Examples

```
glm(
  am ~ mpg + factor(cyl),
  data = mtcars,
  family = binomial,
  contrasts = list(`factor(cyl)` = contr.sum)
) %>%
  model_get_contrasts()
```

---

model\_get\_model\_frame *Get the model frame of a model*

---

## Description

The structure of the object returned by `stats::model.frame()` could slightly differ for certain types of models. `model_get_model_frame()` will always return an object with the same data structure or NULL if it is not possible to compute model frame from model.

## Usage

```
model_get_model_frame(model)

## Default S3 method:
model_get_model_frame(model)

## S3 method for class 'coxph'
model_get_model_frame(model)

## S3 method for class 'survreg'
model_get_model_frame(model)
```

## Arguments

model            a model object

## See Also

[stats::model.frame\(\)](#)

Other model\_helpers: [model\\_get\\_contrasts\(\)](#), [model\\_get\\_model\\_matrix\(\)](#), [model\\_get\\_xlevels\(\)](#), [model\\_identify\\_variables\(\)](#), [model\\_list\\_contrasts\(\)](#), [model\\_list\\_terms\\_levels\(\)](#), [model\\_list\\_variables\(\)](#)

## Examples

```
lm(hp ~ mpg + factor(cyl), mtcars) %>%
  model_get_model_frame() %>%
  head()
```

model\_get\_model\_matrix

*Get the model matrix of a model*

---

## Description

The structure of the object returned by `stats::model.matrix()` could slightly differ for certain types of models. `model_get_model_matrix()` will always return an object with the same structure as `stats::model.matrix.default()`.

## Usage

```
model_get_model_matrix(model)

## Default S3 method:
model_get_model_matrix(model)

## S3 method for class 'multinom'
model_get_model_matrix(model)

## S3 method for class 'glm'
model_get_model_matrix(model)
```

## Arguments

model            a model object

## See Also

[stats::model.matrix\(\)](#)

Other model\_helpers: [model\\_get\\_contrasts\(\)](#), [model\\_get\\_model\\_frame\(\)](#), [model\\_get\\_xlevels\(\)](#), [model\\_identify\\_variables\(\)](#), [model\\_list\\_contrasts\(\)](#), [model\\_list\\_terms\\_levels\(\)](#), [model\\_list\\_variables\(\)](#)

## Examples

```
lm(hp ~ mpg + factor(cyl), mtcars) %>%
  model_get_model_matrix() %>%
  head()
```

---

|                   |                                      |
|-------------------|--------------------------------------|
| model_get_xlevels | <i>Get xlevels used in the model</i> |
|-------------------|--------------------------------------|

---

### Description

Get xlevels used in the model

### Usage

```
model_get_xlevels(model)

## Default S3 method:
model_get_xlevels(model)

## S3 method for class 'lmerMod'
model_get_xlevels(model)

## S3 method for class 'glmerMod'
model_get_xlevels(model)
```

### Arguments

model            a model object

### See Also

Other model\_helpers: [model\\_get\\_contrasts\(\)](#), [model\\_get\\_model\\_frame\(\)](#), [model\\_get\\_model\\_matrix\(\)](#), [model\\_identify\\_variables\(\)](#), [model\\_list\\_contrasts\(\)](#), [model\\_list\\_terms\\_levels\(\)](#), [model\\_list\\_variables\(\)](#)

### Examples

```
lm(hp ~ mpg + factor(cyl), mtcars) %>%
  model_get_xlevels()
```

---

|                          |  |
|--------------------------|--|
| model_identify_variables | <i>Identify for each coefficient of a model the corresponding variable</i> |
|--------------------------|--|

---

### Description

It will also identify interaction terms and intercept(s).

**Usage**

```

model_identify_variables(model)

## Default S3 method:
model_identify_variables(model)

## S3 method for class 'lavaan'
model_identify_variables(model)

```

**Arguments**

```

model          a model object

```

**Value**

A tibble with four columns:

- term: coefficients of the model
- variable: the corresponding variable
- var\_class: class of the variable (cf. `stats::.MFclass()`)
- var\_type: "continuous", "categorical", "intercept" or "interaction"

**See Also**

[tidy\\_identify\\_variables\(\)](#)

Other model\_helpers: [model\\_get\\_contrasts\(\)](#), [model\\_get\\_model\\_frame\(\)](#), [model\\_get\\_model\\_matrix\(\)](#), [model\\_get\\_xlevels\(\)](#), [model\\_list\\_contrasts\(\)](#), [model\\_list\\_terms\\_levels\(\)](#), [model\\_list\\_variables\(\)](#)

**Examples**

```

Titanic %>%
  dplyr::as_tibble() %>%
  dplyr::mutate(Survived = factor(Survived, c("No", "Yes"))) %>%
  glm(
    Survived ~ Class + Age * Sex,
    data = ., weights = .$n,
    family = binomial
  ) %>%
  model_identify_variables()

iris %>%
  lm(
    Sepal.Length ~ poly(Sepal.Width, 2) + Species,
    data = .,
    contrasts = list(Species = contr.sum)
  ) %>%
  model_identify_variables()

```

---

model\_list\_contrasts *List contrasts used by a model*

---

## Description

List contrasts used by a model

## Usage

```
model_list_contrasts(model)
```

```
## Default S3 method:  
model_list_contrasts(model)
```

## Arguments

model            a model object

## Value

A tibble with three columns:

- variable: variable name
- contrasts: type of contrasts
- reference: for variables with treatment, SAS or sum contrasts, position of the reference level

## See Also

Other `model_helpers`: [model\\_get\\_contrasts\(\)](#), [model\\_get\\_model\\_frame\(\)](#), [model\\_get\\_model\\_matrix\(\)](#), [model\\_get\\_xlevels\(\)](#), [model\\_identify\\_variables\(\)](#), [model\\_list\\_terms\\_levels\(\)](#), [model\\_list\\_variables\(\)](#)

## Examples

```
glm(  
  am ~ mpg + factor(cyl),  
  data = mtcars,  
  family = binomial,  
  contrasts = list(`factor(cyl)` = contr.sum)  
) %>%  
  model_list_contrasts()
```

---

`model_list_terms_levels`*List levels of categorical terms*

---

**Description**

Only for categorical variables with treatment, SAS or sum contrasts.

**Usage**

```
model_list_terms_levels(model)
```

```
## Default S3 method:  
model_list_terms_levels(model)
```

**Arguments**

`model` a model object

**Value**

A tibble with four columns:

- `variable`: variable
- `term`: term name
- `level`: term level
- `reference`: logical indicating which term is the reference level

**See Also**

Other `model_helpers`: [model\\_get\\_contrasts\(\)](#), [model\\_get\\_model\\_frame\(\)](#), [model\\_get\\_model\\_matrix\(\)](#), [model\\_get\\_xlevels\(\)](#), [model\\_identify\\_variables\(\)](#), [model\\_list\\_contrasts\(\)](#), [model\\_list\\_variables\(\)](#)

**Examples**

```
glm(  
  am ~ mpg + factor(cyl),  
  data = mtcars,  
  family = binomial,  
  contrasts = list(`factor(cyl)` = contr.sum)  
) %>%  
  model_list_terms_levels()  
  
df <- Titanic %>%  
  dplyr::as_tibble() %>%  
  dplyr::mutate(Survived = factor(Survived, c("No", "Yes")))  
  
df %>%
```



```
glm(  
  Survived ~ Class + Age + Sex,  
  data = ., weights = .$n, family = binomial,  
  contrasts = list(Age = contr.sum, Class = "contr.helmert")  
) %>%  
model_list_terms_levels()
```

---

model\_list\_variables *List all the variables used in a model*

---

## Description

Including variables used only in an interaction.

## Usage

```
model_list_variables(model)  
  
## Default S3 method:  
model_list_variables(model)  
  
## S3 method for class 'lavaan'  
model_list_variables(model)
```

## Arguments

model            a model object

## Value

A tibble with three columns:

- variable: the corresponding variable
- var\_class: class of the variable (cf. [stats::MFclass\(\)](#))
- label\_attr: variable label defined in the original data frame with the label attribute (cf. [labelled::var\\_label\(\)](#))

## See Also

Other model\_helpers: [model\\_get\\_contrasts\(\)](#), [model\\_get\\_model\\_frame\(\)](#), [model\\_get\\_model\\_matrix\(\)](#), [model\\_get\\_xlevels\(\)](#), [model\\_identify\\_variables\(\)](#), [model\\_list\\_contrasts\(\)](#), [model\\_list\\_terms\\_levels\(\)](#)

**Examples**

```

Titanic %>%
  dplyr::as_tibble() %>%
  dplyr::mutate(Survived = factor(Survived, c("No", "Yes"))) %>%
  glm(
    Survived ~ Class + Age : Sex,
    data = ., weights = .$n,
    family = binomial
  ) %>%
  model_list_variables()

iris %>%
  lm(
    Sepal.Length ~ poly(Sepal.Width, 2) + Species,
    data = .,
    contrasts = list(Species = contr.sum)
  ) %>%
  model_list_variables()

if (requireNamespace("gtsummary")) {
  glm(
    response ~ poly(age, 3) + stage + grade * trt,
    na.omit(gtsummary::trial),
    family = binomial,
  ) %>%
  model_list_variables()
}

```

---

`tidy_add_contrasts`     *Add contrasts type for categorical variables*

---

**Description**

Add a contrasts column corresponding to the type of contrasts

**Usage**

```
tidy_add_contrasts(x, model = tidy_get_model(x))
```

**Arguments**

`x`                    a tidy tibble  
`model`                the corresponding model, if not attached to `x`

**Details**

If the variable column is not yet available in `x`, `tidy_identify_variables()` will be automatically applied.

**See Also**

Other tidy\_helpers: [tidy\\_add\\_estimate\\_to\\_reference\\_rows\(\)](#), [tidy\\_add\\_header\\_rows\(\)](#), [tidy\\_add\\_reference\\_rows\(\)](#), [tidy\\_add\\_term\\_labels\(\)](#), [tidy\\_add\\_variable\\_labels\(\)](#), [tidy\\_attach\\_model\(\)](#), [tidy\\_identify\\_variables\(\)](#), [tidy\\_plus\\_plus\(\)](#), [tidy\\_remove\\_intercept\(\)](#)

**Examples**

```
df <- Titanic %>%
  dplyr::as_tibble() %>%
  dplyr::mutate(Survived = factor(Survived, c("No", "Yes")))

df %>%
  glm(
    Survived ~ Class + Age + Sex,
    data = ., weights = .$n, family = binomial,
    contrasts = list(Age = contr.sum, Class = "contr.helmert")
  ) %>%
  tidy_and_attach() %>%
  tidy_add_contrasts()
```

---

`tidy_add_estimate_to_reference_rows`

*Add an estimate value to references rows for categorical variables*

---

**Description**

For categorical variables with a treatment contrast (`stats::contr.treatment()`) or a SAS contrast (`stats::contr.SAS()`) will add an estimate equal to 0 (or 1 if `exponentiate = TRUE`) to the reference row.

**Usage**

```
tidy_add_estimate_to_reference_rows(
  x,
  exponentiate = FALSE,
  model = tidy_get_model(x),
  quiet = FALSE
)
```

**Arguments**

|                           |   |
|---------------------------|---|
| <code>x</code>            | a tidy tibble   |
| <code>exponentiate</code> | logical indicating whether or not to exponentiate the coefficient estimates. It should be consistent with the original call to <code>broom::tidy()</code> |
| <code>model</code>        | the corresponding model, if not attached to <code>x</code>  |
| <code>quiet</code>        | logical argument whether <code>broom.helpers</code> should return an error when requested output cannot be generated. Default is <code>FALSE</code>       |

## Details

For categorical variables with a sum contrast (`stats::contr.sum()`), the estimate value of the reference row will be equal to the sum of all other coefficients multiplied by  $-1$  (eventually exponentiated if `exponentiate = TRUE`), and obtained with `stats::dummy.coef()`. For sum contrasts, the model coefficient corresponds to the difference of each level with the grand mean.

For other variables, no change will be made.

If the `reference_row` column is not yet available in `x`, `tidy_add_reference_rows()` will be automatically applied.

## See Also

Other `tidy_helpers`: `tidy_add_contrasts()`, `tidy_add_header_rows()`, `tidy_add_reference_rows()`, `tidy_add_term_labels()`, `tidy_add_variable_labels()`, `tidy_attach_model()`, `tidy_identify_variables()`, `tidy_plus_plus()`, `tidy_remove_intercept()`

## Examples

```
df <- Titanic %>%
  dplyr::as_tibble() %>%
  dplyr::mutate(dplyr::across(where(is.character), factor))

df %>%
  glm(
    Survived ~ Class + Age + Sex,
    data = ., weights = .$n, family = binomial,
    contrasts = list(Age = contr.sum, Class = "contr.SAS")
  ) %>%
  tidy_and_attach(exponentiate = TRUE) %>%
  tidy_add_reference_rows() %>%
  tidy_add_estimate_to_reference_rows(exponentiate = TRUE)

if (requireNamespace("gtsummary")) {
  glm(
    response ~ stage + grade * trt,
    gtsummary::trial,
    family = binomial,
    contrasts = list(
      stage = contr.treatment(4, base = 3),
      grade = contr.treatment(3, base = 2),
      trt = contr.treatment(2, base = 2)
    )
  ) %>%
  tidy_and_attach() %>%
  tidy_add_reference_rows() %>%
  tidy_add_estimate_to_reference_rows()
}
```

---

tidy\_add\_header\_rows *Add header rows variables with several terms*

---

### Description

For variables with several terms (usually categorical variables but could also be the case of continuous variables with polynomial terms or splines), `tidy_add_header_rows()` will add an additional row per variable, where `label` will be equal to `var_label`. These additional rows could be identified with `header_row` column.

### Usage

```
tidy_add_header_rows(  
  x,  
  show_single_row = NULL,  
  model = tidy_get_model(x),  
  quiet = FALSE,  
  strict = FALSE  
)
```

### Arguments

|                              |   |
|------------------------------|---|
| <code>x</code>               | a tidy tibble   |
| <code>show_single_row</code> | a vector indicating the names of binary variables that should be displayed on a single row                                |
| <code>model</code>           | the corresponding model, if not attached to <code>x</code>  |
| <code>quiet</code>           | logical argument whether broom.helpers should return an error when requested output cannot be generated. Default is FALSE |
| <code>strict</code>          | logical argument whether broom.helpers should return an error when requested output cannot be generated. Default is FALSE |

### Details

The `show_single_row` argument allows to specify a list of dichotomous variables that should be displayed on a single row instead of two rows.

The added `header_row` column will be equal to:

- TRUE for an header row;
- FALSE for a normal row of a variable with an header row;
- NA for variables without an header row.

If the `label` column is not yet available in `x`, `tidy_add_term_labels()` will be automatically applied.

**See Also**

Other tidy\_helpers: [tidy\\_add\\_contrasts\(\)](#), [tidy\\_add\\_estimate\\_to\\_reference\\_rows\(\)](#), [tidy\\_add\\_reference\\_rows\(\)](#), [tidy\\_add\\_term\\_labels\(\)](#), [tidy\\_add\\_variable\\_labels\(\)](#), [tidy\\_attach\\_model\(\)](#), [tidy\\_identify\\_variables\(\)](#), [tidy\\_plus\\_plus\(\)](#), [tidy\\_remove\\_intercept\(\)](#)

**Examples**

```
df <- Titanic %>%
  dplyr::as_tibble() %>%
  dplyr::mutate(Survived = factor(Survived, c("No", "Yes")))

df %>%
  glm(
    Survived ~ Class + Age + Sex,
    data = ., weights = .$n, family = binomial,
    contrasts = list(Age = contr.sum, Class = "contr.SAS")
  ) %>%
  tidy_and_attach() %>%
  tidy_add_variable_labels(labels = list(Class = "Custom label for Class")) %>%
  tidy_add_reference_rows() %>%
  tidy_add_header_rows()

if (requireNamespace("gtsummary")) {
  glm(
    response ~ stage + grade * trt,
    gtsummary::trial,
    family = binomial,
    contrasts = list(
      stage = contr.treatment(4, base = 3),
      grade = contr.treatment(3, base = 2),
      trt = contr.treatment(2, base = 2)
    )
  ) %>%
  tidy_and_attach() %>%
  tidy_add_reference_rows() %>%
  tidy_add_header_rows()
}
```

---

tidy\_add\_reference\_rows

*Add references rows for categorical variables*

---

**Description**

For categorical variables with a treatment contrast ([stats::contr.treatment\(\)](#)), a SAS contrast ([stats::contr.SAS\(\)](#)) or a sum contrast ([stats::contr.sum\(\)](#)), add a reference row.

**Usage**

```
tidy_add_reference_rows(x, model = tidy_get_model(x), quiet = FALSE)
```

**Arguments**

|       |   |
|-------|---|
| x     | a tidy tibble   |
| model | the corresponding model, if not attached to x   |
| quiet | logical argument whether broom.helpers should return an error when requested output cannot be generated. Default is FALSE |

**Details**

The added reference\_row column will be equal to:

- TRUE for a reference row;
- FALSE for a normal row of a variable with a reference row;
- NA for variables without a reference row.

If the contrasts column is not yet available in x, [tidy\\_add\\_contrasts\(\)](#) will be automatically applied.

`tidy_add_reference_rows()` will not populate the label of the reference term. It is therefore better to apply [tidy\\_add\\_term\\_labels\(\)](#) after `tidy_add_reference_rows()` rather than before.

**See Also**

Other tidy\_helpers: [tidy\\_add\\_contrasts\(\)](#), [tidy\\_add\\_estimate\\_to\\_reference\\_rows\(\)](#), [tidy\\_add\\_header\\_rows\(\)](#), [tidy\\_add\\_term\\_labels\(\)](#), [tidy\\_add\\_variable\\_labels\(\)](#), [tidy\\_attach\\_model\(\)](#), [tidy\\_identify\\_variables\(\)](#), [tidy\\_plus\\_plus\(\)](#), [tidy\\_remove\\_intercept\(\)](#)

**Examples**

```
df <- Titanic %>%
  dplyr::as_tibble() %>%
  dplyr::mutate(Survived = factor(Survived, c("No", "Yes")))

df %>%
  glm(
    Survived ~ Class + Age + Sex,
    data = ., weights = .$n, family = binomial,
    contrasts = list(Age = contr.sum, Class = "contr.SAS")
  ) %>%
  tidy_and_attach() %>%
  tidy_add_reference_rows()

if (requireNamespace("gtsummary")) {
  glm(
    response ~ stage + grade * trt,
    gtsummary::trial,
    family = binomial,
    contrasts = list(
      stage = contr.treatment(4, base = 3),
      grade = contr.treatment(3, base = 2),
      trt = contr.treatment(2, base = 2)
    )
  )
}
```

```

) %>%
  tidy_and_attach() %>%
  tidy_add_reference_rows()
}

```

---

`tidy_add_term_labels` *Add term labels*

---

## Description

Will add term labels in a label column, based on:

1. labels provided in labels argument if provided;
2. factor levels for categorical variables coded with treatment, SAS or sum contrasts;
3. variable labels when there is only one term per variable;
4. term name otherwise.

## Usage

```

tidy_add_term_labels(
  x,
  labels = NULL,
  interaction_sep = " * ",
  model = tidy_get_model(x),
  quiet = FALSE,
  strict = FALSE
)

```

## Arguments

|                              |   |
|------------------------------|---|
| <code>x</code>               | a tidy tibble   |
| <code>labels</code>          | an optional named list or named vector of custom term labels  |
| <code>interaction_sep</code> | separator for interaction terms   |
| <code>model</code>           | the corresponding model, if not attached to <code>x</code>  |
| <code>quiet</code>           | logical argument whether broom.helpers should return an error when requested output cannot be generated. Default is FALSE |
| <code>strict</code>          | logical argument whether broom.helpers should return an error when requested output cannot be generated. Default is FALSE |

## Details

If the `variable_label` column is not yet available in `x`, `tidy_add_variable_labels()` will be automatically applied. If the `contrasts` column is not yet available in `x`, `tidy_add_contrasts()` will be automatically applied.

It is possible to pass a custom label for any term in `labels`, including interaction terms.



**See Also**

Other tidy\_helpers: [tidy\\_add\\_contrasts\(\)](#), [tidy\\_add\\_estimate\\_to\\_reference\\_rows\(\)](#), [tidy\\_add\\_header\\_rows\(\)](#), [tidy\\_add\\_reference\\_rows\(\)](#), [tidy\\_add\\_variable\\_labels\(\)](#), [tidy\\_attach\\_model\(\)](#), [tidy\\_identify\\_variables\(\)](#), [tidy\\_plus\\_plus\(\)](#), [tidy\\_remove\\_intercept\(\)](#)

**Examples**

```
df <- Titanic %>%
  dplyr::as_tibble() %>%
  dplyr::mutate(Survived = factor(Survived, c("No", "Yes"))) %>%
  labelled::set_variable_labels(
    Class = "Passenger's class",
    Sex = "Sex"
  )

df %>%
  glm(Survived ~ Class * Age * Sex, data = ., weights = .$n, family = binomial) %>%
  tidy_and_attach() %>%
  tidy_add_term_labels()
```

---

tidy\_add\_variable\_labels

*Add variable labels*

---

**Description**

Will add variable labels in a `var_label` column, based on:

1. labels provided in labels argument if provided;
2. variable labels defined in the original data frame with the label attribute (cf. [labelled::var\\_label\(\)](#));
3. variable name otherwise.

**Usage**

```
tidy_add_variable_labels(
  x,
  labels = NULL,
  interaction_sep = " * ",
  model = tidy_get_model(x),
  quiet = FALSE,
  strict = FALSE
)
```

**Arguments**

|                 |   |
|-----------------|---|
| x               | a tidy tibble   |
| labels          | an optional named list or named vector of custom variable labels  |
| interaction_sep | separator for interaction terms   |
| model           | the corresponding model, if not attached to x   |
| quiet           | logical argument whether broom.helpers should return an error when requested output cannot be generated. Default is FALSE |
| strict          | logical argument whether broom.helpers should return an error when requested output cannot be generated. Default is FALSE |

**Details**

If the variable column is not yet available in x, `tidy_identify_variables()` will be automatically applied.

It is possible to pass a custom label for an interaction term in labels (see examples).

**See Also**

Other tidy\_helpers: `tidy_add_contrasts()`, `tidy_add_estimate_to_reference_rows()`, `tidy_add_header_rows()`, `tidy_add_reference_rows()`, `tidy_add_term_labels()`, `tidy_attach_model()`, `tidy_identify_variables()`, `tidy_plus_plus()`, `tidy_remove_intercept()`

**Examples**

```
df <- Titanic %>%
  dplyr::as_tibble() %>%
  dplyr::mutate(Survived = factor(Survived, c("No", "Yes"))) %>%
  labelled::set_variable_labels(
    Class = "Passenger's class",
    Sex = "Sex"
  )

df %>%
  glm(Survived ~ Class * Age * Sex, data = ., weights = .$n, family = binomial) %>%
  tidy_and_attach() %>%
  tidy_add_variable_labels(
    labels = list(Sex = "Gender", "Class:Age" = "Custom label")
  )
```

---

|                   |   |
|-------------------|---|
| tidy_attach_model | <i>Attach a full model to the tibble of model terms</i> |
|-------------------|---|

---

**Description**

To facilitate the use of broom helpers with pipe, it is recommended to attach the original model as an attribute to the tibble of model terms generated by `broom::tidy()`.

**Usage**

```
tidy_attach_model(x, model)

tidy_and_attach(model, tidy_fun = broom::tidy, ...)

tidy_get_model(x)

tidy_detach_model(x)
```

**Arguments**

|          |  |
|----------|--|
| x        | a tibble of model terms                    |
| model    | a model to be attached/tidied              |
| tidy_fun | option to specify a custom tidier function |
| ...      | other arguments passed to tidy_fun()       |

**Details**

tidy\_attach\_model() attach the model to a tibble already generated while tidy\_and\_attach() will apply broom::tidy() and attach the model.

Use tidy\_get\_model() to get the model attached to the tibble and tidy\_detach\_model() to remove the attribute containing the model.

**See Also**

Other tidy\_helpers: [tidy\\_add\\_contrasts\(\)](#), [tidy\\_add\\_estimate\\_to\\_reference\\_rows\(\)](#), [tidy\\_add\\_header\\_rows\(\)](#), [tidy\\_add\\_reference\\_rows\(\)](#), [tidy\\_add\\_term\\_labels\(\)](#), [tidy\\_add\\_variable\\_labels\(\)](#), [tidy\\_identify\\_variables\(\)](#), [tidy\\_plus\\_plus\(\)](#), [tidy\\_remove\\_intercept\(\)](#)

**Examples**

```
mod <- lm(Sepal.Length ~ Sepal.Width + Species, data = iris)
tt <- mod %>%
  tidy_and_attach(conf.int = TRUE)
tt
tidy_get_model(tt)
```

---

tidy\_identify\_variables

*Identify the variable corresponding to each model coefficient*

---

**Description**

tidy\_identify\_variables() will add to the tidy tibble three additional columns: variable, var\_class and var\_type.

**Usage**

```
tidy_identify_variables(
  x,
  model = tidy_get_model(x),
  quiet = FALSE,
  strict = FALSE
)
```

**Arguments**

|                     |   |
|---------------------|---|
| <code>x</code>      | a tidy tibble   |
| <code>model</code>  | the corresponding model, if not attached to <code>x</code>  |
| <code>quiet</code>  | logical argument whether broom.helpers should return an error when requested output cannot be generated. Default is FALSE |
| <code>strict</code> | logical argument whether broom.helpers should return an error when requested output cannot be generated. Default is FALSE |

**Details**

It will also identify interaction terms and intercept(s). `var_type` could be "continuous", "categorical", "intercept" or "interaction". Will be equal to "unknown" in the rare cases where `tidy_identify_variables()` will fail to identify the list of variables.

**See Also**

[model\\_identify\\_variables\(\)](#)

Other tidy\_helpers: [tidy\\_add\\_contrasts\(\)](#), [tidy\\_add\\_estimate\\_to\\_reference\\_rows\(\)](#), [tidy\\_add\\_header\\_rows\(\)](#), [tidy\\_add\\_reference\\_rows\(\)](#), [tidy\\_add\\_term\\_labels\(\)](#), [tidy\\_add\\_variable\\_labels\(\)](#), [tidy\\_attach\\_model\(\)](#), [tidy\\_plus\\_plus\(\)](#), [tidy\\_remove\\_intercept\(\)](#)

**Examples**

```
Titanic %>%
  dplyr::as_tibble() %>%
  dplyr::mutate(Survived = factor(Survived, c("No", "Yes"))) %>%
  glm(Survived ~ Class + Age * Sex, data = ., weights = .$n, family = binomial) %>%
  tidy_and_attach() %>%
  tidy_identify_variables()

lm(
  Sepal.Length ~ poly(Sepal.Width, 2) + Species,
  data = iris,
  contrasts = list(Species = contr.sum)
) %>%
  tidy_and_attach(conf.int = TRUE) %>%
  tidy_identify_variables()
```

---

`tidy_plus_plus`*Tidy a model and compute additional informations*

---

## Description

This function will apply sequentially:

- `tidy_and_attach()`
- `tidy_identify_variables()`
- `tidy_add_contrasts()`
- `tidy_add_reference_rows()`
- `tidy_add_estimate_to_reference_rows()`
- `tidy_add_variable_labels()`
- `tidy_add_term_labels()`
- `tidy_add_header_rows()`
- `tidy_remove_intercept()`
- `tidy_detach_model()`

## Usage

```
tidy_plus_plus(  
  model,  
  tidy_fun = broom::tidy,  
  conf.int = TRUE,  
  exponentiate = FALSE,  
  variable_labels = NULL,  
  term_labels = NULL,  
  add_reference_rows = TRUE,  
  add_estimate_to_reference_rows = TRUE,  
  add_header_rows = FALSE,  
  show_single_row = NULL,  
  intercept = FALSE,  
  keep_model = FALSE,  
  quiet = FALSE,  
  strict = FALSE,  
  ...  
)
```

## Arguments

|                       |  |
|-----------------------|--|
| <code>model</code>    | a model to be attached/tidied  |
| <code>tidy_fun</code> | option to specify a custom tidier function                                 |
| <code>conf.int</code> | should confidence intervals be computed? (see <code>broom::tidy()</code> ) |

|                                |  |
|--------------------------------|--|
| exponentiate                   | logical indicating whether or not to exponentiate the coefficient estimates. This is typical for logistic and multinomial regressions, but a bad idea if there is no log or logit link. Defaults to FALSE. |
| variable_labels                | a named list or a named vector of custom variable labels   |
| term_labels                    | a named list or a named vector of custom term labels   |
| add_reference_rows             | should reference rows be added?  |
| add_estimate_to_reference_rows | should an estimate value be added to reference rows?   |
| add_header_rows                | should header rows be added?   |
| show_single_row                | a vector indicating the names of binary variables that should be displayed on a single row, when add_header_rows is TRUE   |
| intercept                      | should the intercept(s) be included?   |
| keep_model                     | should the model be kept as an attribute of the final result?  |
| quiet                          | logical argument whether broom.helpers should return an error when requested output cannot be generated. Default is FALSE  |
| strict                         | logical argument whether broom.helpers should return an error when requested output cannot be generated. Default is FALSE  |
| ...                            | other arguments passed to tidy_fun()   |

### See Also

Other tidy\_helpers: [tidy\\_add\\_contrasts\(\)](#), [tidy\\_add\\_estimate\\_to\\_reference\\_rows\(\)](#), [tidy\\_add\\_header\\_rows\(\)](#), [tidy\\_add\\_reference\\_rows\(\)](#), [tidy\\_add\\_term\\_labels\(\)](#), [tidy\\_add\\_variable\\_labels\(\)](#), [tidy\\_attach\\_model\(\)](#), [tidy\\_identify\\_variables\(\)](#), [tidy\\_remove\\_intercept\(\)](#)

### Examples

```
ex1 <- lm(Sepal.Length ~ Sepal.Width + Species, data = iris) %>%
  tidy_plus_plus()
ex1
```

```
df <- Titanic %>%
  dplyr::as_tibble() %>%
  dplyr::mutate(
    Survived = factor(Survived, c("No", "Yes"))
  ) %>%
  labelled::set_variable_labels(
    Class = "Passenger's class",
    Sex = "Gender"
  )
```

```
ex2 <- glm(
  Survived ~ Class + Age * Sex,
  data = df, weights = df$n,
```

```

    family = binomial
  ) %>%
  tidy_plus_plus(exponentiate = TRUE)
ex2

if (requireNamespace("gtsummary")) {
  ex3 <- glm(
    response ~ poly(age, 3) + stage + grade * trt,
    na.omit(gtsummary::trial),
    family = binomial,
    contrasts = list(
      stage = contr.treatment(4, base = 3),
      grade = contr.sum
    )
  ) %>%
  tidy_plus_plus(
    exponentiate = TRUE,
    variable_labels = c(age = "Age (in years)"),
    add_header_rows = TRUE,
    show_single_row = "trt",
    term_labels = c("poly(age, 3)3" = "Cubic age"),
    keep_model = TRUE
  )
ex3
}

```

---

tidy\_remove\_intercept *Remove intercept(s)*

---

### Description

Will remove terms where `var_type == "intercept"`.

### Usage

```
tidy_remove_intercept(x, model = tidy_get_model(x))
```

### Arguments

|                    |  |
|--------------------|--|
| <code>x</code>     | a tidy tibble  |
| <code>model</code> | the corresponding model, if not attached to <code>x</code> |

### Details

If the variable column is not yet available in `x`, [tidy\\_identify\\_variables\(\)](#) will be automatically applied.

**See Also**

Other tidy\_helpers: [tidy\\_add\\_contrasts\(\)](#), [tidy\\_add\\_estimate\\_to\\_reference\\_rows\(\)](#), [tidy\\_add\\_header\\_rows\(\)](#), [tidy\\_add\\_reference\\_rows\(\)](#), [tidy\\_add\\_term\\_labels\(\)](#), [tidy\\_add\\_variable\\_labels\(\)](#), [tidy\\_attach\\_model\(\)](#), [tidy\\_identify\\_variables\(\)](#), [tidy\\_plus\\_plus\(\)](#)

**Examples**

```
Titanic %>%  
  dplyr::as_tibble() %>%  
  dplyr::mutate(Survived = factor(Survived)) %>%  
  glm(Survived ~ Class + Age + Sex, data = ., weights = .$n, family = binomial) %>%  
  tidy_and_attach() %>%  
  tidy_remove_intercept()
```



# Index

- \* **model\_helpers**
  - model\_get\_contrasts, 2
  - model\_get\_model\_frame, 3
  - model\_get\_model\_matrix, 4
  - model\_get\_xlevels, 5
  - model\_identify\_variables, 5
  - model\_list\_contrasts, 7
  - model\_list\_terms\_levels, 8
  - model\_list\_variables, 9
- \* **tidy\_helpers**
  - tidy\_add\_contrasts, 10
  - tidy\_add\_estimate\_to\_reference\_rows, 11
  - tidy\_add\_header\_rows, 13
  - tidy\_add\_reference\_rows, 14
  - tidy\_add\_term\_labels, 16
  - tidy\_add\_variable\_labels, 17
  - tidy\_attach\_model, 18
  - tidy\_identify\_variables, 19
  - tidy\_plus\_plus, 21
  - tidy\_remove\_intercept, 23
- broom::tidy(), 11, 21
- labelled::var\_label(), 9, 17
- model\_get\_contrasts, 2, 3–9
- model\_get\_model\_frame, 2, 3, 4–9
- model\_get\_model\_matrix, 2, 3, 4, 5–9
- model\_get\_xlevels, 2–4, 5, 6–9
- model\_identify\_variables, 2–5, 5, 7–9
- model\_identify\_variables(), 20
- model\_list\_contrasts, 2–6, 7, 8, 9
- model\_list\_terms\_levels, 2–7, 8, 9
- model\_list\_variables, 2–8, 9
- stats::MFclass(), 6, 9
- stats::contr.SAS(), 11, 14
- stats::contr.sum(), 12, 14
- stats::contr.treatment(), 11, 14
- stats::dummy.coef(), 12
- stats::model.frame(), 3
- stats::model.matrix(), 4
- stats::model.matrix.default(), 4
- tidy\_add\_contrasts, 10, 12, 14, 15, 17–20, 22, 24
- tidy\_add\_contrasts(), 15, 16, 21
- tidy\_add\_estimate\_to\_reference\_rows, 11, 11, 14, 15, 17–20, 22, 24
- tidy\_add\_estimate\_to\_reference\_rows(), 21
- tidy\_add\_header\_rows, 11, 12, 13, 15, 17–20, 22, 24
- tidy\_add\_header\_rows(), 21
- tidy\_add\_reference\_rows, 11, 12, 14, 14, 17–20, 22, 24
- tidy\_add\_reference\_rows(), 12, 21
- tidy\_add\_term\_labels, 11, 12, 14, 15, 16, 18–20, 22, 24
- tidy\_add\_term\_labels(), 13, 15, 21
- tidy\_add\_variable\_labels, 11, 12, 14, 15, 17, 17, 19, 20, 22, 24
- tidy\_add\_variable\_labels(), 16, 21
- tidy\_and\_attach(tidy\_attach\_model), 18
- tidy\_and\_attach(), 21
- tidy\_attach\_model, 11, 12, 14, 15, 17, 18, 18, 20, 22, 24
- tidy\_detach\_model(tidy\_attach\_model), 18
- tidy\_detach\_model(), 21
- tidy\_get\_model(tidy\_attach\_model), 18
- tidy\_identify\_variables, 11, 12, 14, 15, 17–19, 19, 22, 24
- tidy\_identify\_variables(), 6, 10, 18, 21, 23
- tidy\_plus\_plus, 11, 12, 14, 15, 17–20, 21, 24
- tidy\_remove\_intercept, 11, 12, 14, 15, 17–20, 22, 23
- tidy\_remove\_intercept(), 21