

# Package: tidycat (via r-universe)

August 27, 2024

**Type** Package

**Title** Expand Tidy Output for Categorical Parameter Estimates

**Version** 0.1.2

**Maintainer** Guy J. Abel <g.j.abel@gmail.com>

**URL** <https://guyabel.github.io/tidycat/>

**BugReports** <https://github.com/guyabel/tidycat/issues/>

**Description** Create additional rows and columns on broom::tidy() output to allow for easier control on categorical parameter estimates.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.0

**Imports** magrittr, utils, tidyr, tibble, dplyr, stringr, stats, forcats

**Suggests** broom, ggplot2, ggforce, knitr, rmarkdown, spelling

**VignetteBuilder** knitr

**Roxygen** list(markdown = TRUE)

**Language** en-US

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

**RemoteUrl** <https://github.com/guyabel/tidycat>

**RemoteRef** HEAD

**RemoteSha** 42725e19964c01725c40abe59b747dc25ed93fa9

## Contents

factor_regex . . . . .	2
tidy_categorical . . . . .	2

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

---

factor_regex	<i>Generate Regular Expression to Detect Factors</i>
--------------	--

---

### Description

Primarily developed for use within [tidycat::tidy\\_categorical\(\)](#)

### Usage

```
factor_regex(m, at_start = TRUE)
```

### Arguments

m	A model object, created using a function such as <a href="#">stats::lm()</a>
at_start	Logical indicating whether or not to include ^ in the regular expression to begin search at start of string

### Value

A character string for use as a regular expression.

### Author(s)

Guy J. Abel

### Examples

```
m0 <- lm(formula = mpg ~ disp + as.factor(am)*as.factor(vs), data = mtcars)
factor_regex(m = m0)
```

---

tidy_categorical	<i>Expand broom::tidy() Outputs for Categorical Parameter Estimates</i>
------------------	---

---

### Description

Create additional columns in a tidy model output (such as [broom::tidy.lm\(\)](#)) to allow for easier control when plotting categorical parameter estimates.

**Usage**

```
tidy_categorical(
  d = NULL,
  m = NULL,
  include_reference = TRUE,
  reference_label = "Baseline Category",
  non_reference_label = paste0("Non-", reference_label),
  exponentiate = FALSE,
  n_level = FALSE
)
```

**Arguments**

**d** A data frame `tibble::tibble()` output from `broom::tidy.lm()`; with one row for each term in the regression, including column term

**m** A model object, created using a function such as `lm()`

**include\_reference** Logical indicating to include additional rows in output for reference categories, obtained from `dummy.coef()`. Defaults to TRUE

**reference\_label** Character string. When used will create an additional column in output with labels to indicate if terms correspond to reference categories.

**non\_reference\_label** Character string. When `reference_label` is used will be in output to indicate if terms not corresponding to reference categories.

**exponentiate** Logical indicating whether or not the results in `broom::tidy.lm()` are exponentiated. Defaults to FALSE.

**n\_level** Logical indicating whether or not to include a column `n_level` for the number of observations per category. Defaults to FALSE.

**Value**

Expanded `tibble::tibble()` from the version passed to `d` including additional columns:

**variable** The name of the variable that the regression term belongs to.

**level** The level of the categorical variable that the regression term belongs to. Will be an the term name for numeric variables.

**effect** The type of term (main or interaction)

**reference** The type of term (reference or non-reference) with label passed from `reference_label`. If `reference_label` is set NULL will not be created.

**n\_level** The the number of observations per category. If `n_level` is set NULL (default) will not be created.

In addition, extra rows will be added, if `include_reference` is set to FALSE for the reference categories, obtained from `dummy.coef()`

**Author(s)**

Guy J. Abel

**See Also**[broom::tidy.lm\(\)](#)**Examples**

```

# strip ordering in factors (currently ordered factor not supported)
library(dplyr)
library(broom)

m0 <- esoph %>%
  mutate_if(is.factor, ~factor(., ordered = FALSE)) %>%
  glm(cbind(ncases, ncontrols) ~ agegp + tobgp * alcgp, data = .,
      family = binomial())
# tidy
tidy(m0)

# add further columns to tidy output to help manage categorical variables
m0 %>%
  tidy() %>%
  tidy_categorical(m = m0, include_reference = FALSE)

# include reference categories and column to indicate the additional terms
m0 %>%
  tidy() %>%
  tidy_categorical(m = m0)

# coefficient plots
d0 <- m0 %>%
  tidy(conf.int = TRUE) %>%
  tidy_categorical(m = m0) %>%
  # drop the intercept term
  slice(-1)
d0

# typical coefficient plot
library(ggplot2)
library(tidyr)
ggplot(data = d0 %>% drop_na(),
      mapping = aes(x = term, y = estimate,
                    ymin = conf.low, ymax = conf.high)) +
  coord_flip() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_pointrange()

# enhanced coefficient plot using additional columns from tidy_categorical and ggforce::facet_row()
library(ggforce)
ggplot(data = d0,
      mapping = aes(x = level, colour = reference,

```

```
      y = estimate, ymin = conf.low, ymax = conf.high)) +  
facet_row(facets = vars(variable), scales = "free_x", space = "free") +  
geom_hline(yintercept = 0, linetype = "dashed") +  
geom_pointrange() +  
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

# Index

`broom::tidy.lm()`, 2–4

`dummy.coef()`, 3

`factor_regex`, 2

`lm()`, 3

`stats::lm()`, 2

`tibble::tibble()`, 3

`tidy_categorical`, 2

`tidycat::tidy_categorical()`, 2