

Package ‘groupedstats’

December 2, 2018

Type Package

Title Grouped Statistical Analyses in a Tidy Way

Version 0.0.4

Maintainer Indrajeet Patil <patilindrajeet.science@gmail.com>

Description Collection of functions to run statistical tests across all levels of multiple grouping variables.

License GPL-3 | file LICENSE

URL <https://indrajeetpatil.github.io/groupedstats/>,
<https://github.com/IndrajeetPatil/groupedstats>

BugReports <https://github.com/IndrajeetPatil/groupedstats/issues>

Depends R (>= 3.5.0)

Imports broom (>= 0.5.0), broom.mixed (>= 0.2.3), crayon (>= 1.3.4),
dplyr (>= 0.7.8), glue (>= 1.3.0), lme4 (>= 1.1-19), magrittr
(>= 1.5), purrr (>= 0.2.5), rlang (>= 0.3.0.1), robust (>=
0.4-18), rstudioapi (>= 0.8), sjstats (>= 0.17.2), skimr (>=
1.0.3), stats, tibble (>= 1.4.2), tidyr (>= 0.8.2)

Suggests gapminder, covr (>= 3.0.1), ggplot2, knitr (>= 1.20),
rmarkdown (>= 1.9), spelling, testthat

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Language en-US

NeedsCompilation no

Author Indrajeet Patil [aut, cre] (<<https://orcid.org/0000-0003-1995-6531>>)

Repository CRAN

Date/Publication 2018-12-01 23:30:03 UTC

R topics documented:

groupedstats-package	2
grouped_aov	3
grouped_glm	4
grouped_glmer	5
grouped_lm	7
grouped_lmer	8
grouped_proptest	9
grouped_robustslr	10
grouped_slr	11
grouped_summary	12
grouped_ttest	13
grouped_wilcox	14
intent_morality	15
lm_effsize_ci	16
movies_long	17
movies_wide	18
set_cwd	19
signif_column	19
specify_decimal_p	20
Titanic_full	21
Index	22

groupedstats-package groupedstats

Description

Collection of functions to run statistical operations on multiple variables across multiple grouping variables in a dataframe.

Details

For more documentation, see [README](#) on GitHub. Vignette preparation in progress.

grouped_aov	<i>Function to run analysis of variance (aov) across multiple grouping variables.</i>
-------------	---

Description

Function to run analysis of variance (aov) across multiple grouping variables.

Usage

```
grouped_aov(data, grouping.vars, formula, effsize = "eta",  
            output = "tidy", nboot = 1000)
```

Arguments

data	Dataframe from which variables are to be taken.
grouping.vars	List of grouping variables.
formula	A formula specifying the model.
effsize	Character describing the effect size to be displayed: "eta" (default) or "omega".
output	A character describing what output is expected. Two possible options: "tidy" (default), which will return the results, or "tukey", which will return results from Tukey's Honest Significant Differences method for <i>post hoc</i> comparisons. The "glance" method to get model summary is currently not supported for this function.
nboot	Number of bootstrap samples for confidence intervals for partial eta-squared and omega-squared (Default: 500).

Value

A tibble dataframe with tidy results from anova. No model summaries available.

Author(s)

Indrajeet Patil

Examples

```
# uses dataset included in the groupedstats package  
library(groupedstats)  
  
groupedstats::grouped_aov(  
  formula = rating ~ belief * outcome * question,  
  data = intent_morality,  
  grouping.vars = item,  
  effsize = "eta"  
)
```

grouped_glm	<i>Function to run generalized linear model (glm) across multiple grouping variables.</i>
-------------	---

Description

Function to run generalized linear model (glm) across multiple grouping variables.

Usage

```
grouped_glm(data, grouping.vars, formula, family = stats::binomial(link =
  "logit"), quick = FALSE, exponentiate = FALSE, output = "tidy")
```

Arguments

data	Dataframe from which variables are to be taken.
grouping.vars	List of grouping variables.
formula	an object of class " formula " (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
family	a description of the error distribution and link function to be used in the model. For glm this can be a character string naming a family function, a family function or the result of a call to a family function. For glm.fit only the third option is supported. (See family for details of family functions.)
quick	Logical indicating if the only the term and estimate columns should be returned. Often useful to avoid time consuming covariance and standard error calculations. Defaults to FALSE.
exponentiate	Logical indicating whether or not to exponentiate the 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.
output	A character describing what output is expected. Two possible options: "tidy" (default), which will return the results, or "glance", which will return model summaries.

Value

A tibble dataframe with tidy results from linear model.

Author(s)

Indrajeet Patil

See Also

grouped_lm, grouped_glmer

Examples

```
# to get tidy output
groupedstats::grouped_glm(
  data = groupedstats::Titanic_full,
  formula = Survived ~ Sex,
  grouping.vars = Class,
  family = stats::binomial(link = "logit")
)

# to get glance output
groupedstats::grouped_glm(
  data = groupedstats::Titanic_full,
  formula = Survived ~ Sex,
  grouping.vars = Class,
  family = stats::binomial(link = "logit"),
  output = "glance"
)
```

grouped_glm	<i>Function to run generalized linear mixed-effects model (glmer) across multiple grouping variables.</i>
-------------	---

Description

Function to run generalized linear mixed-effects model (glmer) across multiple grouping variables.

Usage

```
grouped_glm(data, grouping.vars, formula, family = stats::binomial(link
  = "probit"), control = lme4::glmerControl(optimizer = "bobyqa",
  boundary.tol = 1e-07, calc.derivs = FALSE, use.last.params = FALSE,
  optCtrl = list(maxfun = 2e+09)), output = "tidy")
```

Arguments

data	Dataframe from which variables are to be taken.
grouping.vars	List of grouping variables.
formula	a two-sided linear formula object describing both the fixed-effects and random-effects part of the model, with the response on the left of a ~ operator and the terms, separated by + operators, on the right. Random-effects terms are distinguished by vertical bars (" ") separating expressions for design matrices from grouping factors.
family	a GLM family, see glm and family .
control	a list (of correct class, resulting from lmerControl() or glmerControl() respectively) containing control parameters, including the nonlinear optimizer to be used and parameters to be passed through to the nonlinear optimizer, see the *lmerControl documentation for details.

output A character describing what output is expected. Two possible options: "tidy" (default), which will return the results, or "glance", which will return model summaries.

Value

A tibble dataframe with tidy results from linear model or model summaries.

Author(s)

Indrajeet Patil

See Also

`grouped_lmer`

Examples

```
# commented out because the examples are time-consuming and the R CMD CHECK
# makes a NOTE (> 5s)

# categorical outcome; binomial family
groupedstats::grouped_glmer(
  formula = Survived ~ Age + (Age |
    Class),
  family = stats::binomial(link = "probit"),
  data = dplyr::sample_frac(groupedstats::Titanic_full, size = 0.3),
  grouping.vars = Sex
)

# continuous outcome; gaussian family
library(gapminder)

groupedstats::grouped_glmer(
  data = dplyr::sample_frac(gapminder, size = 0.3),
  formula = scale(lifeExp) ~ scale(gdpPercap) + (gdpPercap | continent),
  family = stats::gaussian(),
  control = lme4::lmerControl(
    optimizer = "bobyqa",
    restart_edge = TRUE,
    boundary.tol = 1e-7,
    calc.derivs = FALSE,
    optCtrl = list(maxfun = 2e9)
  ),
  grouping.vars = year,
  output = "tidy"
)
```

grouped_lm	<i>Function to run linear model (lm) across multiple grouping variables.</i>
------------	--

Description

Function to run linear model (lm) across multiple grouping variables.

Usage

```
grouped_lm(data, grouping.vars, formula, output = "tidy")
```

Arguments

data	Dataframe from which variables are to be taken.
grouping.vars	List of grouping variables.
formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
output	A character describing what output is expected. Two possible options: "tidy" (default), which will return the results, or "glance", which will return model summaries.

Value

A tibble dataframe with tidy results from linear model.

Author(s)

Indrajeet Patil

See Also

grouped_slr

Examples

```
# loading needed libraries
library(ggplot2)

# getting tidy output of results
grouped_lm(
  data = mtcars,
  grouping.vars = cyl,
  formula = mpg ~ am * wt,
  output = "tidy"
)
```

```
# getting model summaries
# diamonds dataset from ggplot2
grouped_lm(
  data = diamonds,
  grouping.vars = c(cut, color),
  formula = price ~ carat * clarity,
  output = "glance"
)
```

grouped_lmer	<i>Function to run linear mixed-effects model (lmer) across multiple grouping variables.</i>
--------------	--

Description

Function to run linear mixed-effects model (lmer) across multiple grouping variables.

Usage

```
grouped_lmer(data, grouping.vars, formula, REML = TRUE,
  control = lme4::lmerControl(optimizer = "bobyqa", restart_edge = TRUE,
  boundary.tol = 1e-07, calc.derivs = FALSE, use.last.params = FALSE,
  optCtrl = list(maxfun = 2e+09)), p.kr = FALSE, output = "tidy")
```

Arguments

data	Dataframe from which variables are to be taken.
grouping.vars	List of grouping variables.
formula	a two-sided linear formula object describing both the fixed-effects and random-effects part of the model, with the response on the left of a ~ operator and the terms, separated by + operators, on the right. Random-effects terms are distinguished by vertical bars () separating expressions for design matrices from grouping factors. Two vertical bars () can be used to specify multiple uncorrelated random effects for the same grouping variable. (Because of the way it is implemented, the -syntax <i>works only for design matrices containing numeric (continuous) predictors</i> ; to fit models with independent categorical effects, see dummy or the <code>lmer_alt</code> function from the <code>afex</code> package.)
REML	logical scalar - Should the estimates be chosen to optimize the REML criterion (as opposed to the log-likelihood)?
control	a list (of correct class, resulting from <code>lmerControl()</code> or <code>glmerControl()</code> respectively) containing control parameters, including the nonlinear optimizer to be used and parameters to be passed through to the nonlinear optimizer, see the <code>*lmerControl</code> documentation for details.
p.kr	Logical, if TRUE, the computation of p-values is based on conditional F-tests with Kenward-Roger approximation for the df (see 'Details').
output	A character describing what output is expected. Two possible options: "tidy" (default), which will return the results, or "glance", which will return model summaries.

Value

A tibble dataframe with tidy results from linear model or model summaries.

Author(s)

Indrajeet Patil

Examples

```
# loading libraries containing data
library(ggplot2)
library(gapminder)

# getting tidy output of results
# let's use only 50% data to speed it up
groupedstats::grouped_lmer(
  data = dplyr::sample_frac(gapminder, size = 0.5),
  formula = scale(lifeExp) ~ scale(gdpPercap) + (gdpPercap | continent),
  grouping.vars = year,
  output = "tidy"
)

# getting model summaries
# let's use only 50% data to speed it up
grouped_lmer(
  data = ggplot2::diamonds,
  formula = scale(price) ~ scale(carat) + (carat | color),
  REML = FALSE,
  grouping.vars = c(cut, clarity),
  output = "glance"
)
```

grouped_proptest *Function to run proportion test on grouped data.*

Description

Function to run proportion test on grouped data.

Usage

```
grouped_proptest(data, grouping.vars, measure)
```

Arguments

data	Dataframe from which variables are to be drawn.
grouping.vars	List of grouping variables
measure	A variable for which proportion test needs to be carried out for each combination of levels of factors entered in grouping.vars.

Value

Dataframe with percentages and statistical details from a proportion test.

Author(s)

Indrajeet Patil

Examples

```
groupedstats::grouped_proptest(  
  data = mtcars,  
  grouping.vars = cyl,  
  measure = am  
)
```

grouped_robustslr	<i>Function to run robust simple linear regression (slr) on multiple variables across multiple grouping variables.</i>
-------------------	--

Description

Function to run robust simple linear regression (slr) on multiple variables across multiple grouping variables.

Usage

```
grouped_robustslr(data, dep.vars, indep.vars, grouping.vars)
```

Arguments

data	Dataframe from which variables are to be taken.
dep.vars	List criterion or dependent variables for regression (y in $y \sim x$).
indep.vars	List predictor or independent variables for regression (x in $y \sim x$).
grouping.vars	List of grouping variables.

Value

A tibble dataframe with tidy results from robust linear regression analyses. The estimates are standardized, i.e. the lm model used is $\text{scale}(y) \sim \text{scale}(x)$, and not $y \sim x$.

Author(s)

Indrajeet Patil

Examples

```
# in case of just one grouping variable
groupedstats::grouped_robustslr(
  data = iris,
  dep.vars = c(Sepal.Length, Petal.Length),
  indep.vars = c(Sepal.Width, Petal.Width),
  grouping.vars = Species
)
```

grouped_slr	<i>Function to run simple linear regression (slr) on multiple variables across multiple grouping variables.</i>
-------------	---

Description

Function to run simple linear regression (slr) on multiple variables across multiple grouping variables.

Usage

```
grouped_slr(data, dep.vars, indep.vars, grouping.vars)
```

Arguments

data	Dataframe from which variables are to be taken.
dep.vars	List criterion or dependent variables for simple linear model (y in $y \sim x$).
indep.vars	List predictor or independent variables for simple linear model (x in $y \sim x$).
grouping.vars	List of grouping variables.

Value

A tibble dataframe with tidy results from simple linear regression analyses. The estimates are standardized, i.e. the lm model used is $\text{scale}(y) \sim \text{scale}(x)$, and not $y \sim x$.

Author(s)

Indrajeet Patil

See Also

grouped_lm

Examples

```
# in case of just one grouping variable
groupedstats::grouped_slr(
  data = iris,
  dep.vars = c(Sepal.Length, Petal.Length),
  indep.vars = c(Sepal.Width, Petal.Width),
  grouping.vars = Species
)
```

grouped_summary	<i>Function to get descriptive statistics for multiple variables for all grouping variable levels</i>
-----------------	---

Description

Function to get descriptive statistics for multiple variables for all grouping variable levels

Usage

```
grouped_summary(data, grouping.vars, measures = NULL,
  measures.type = "numeric", topcount.long = FALSE)
```

Arguments

data	Dataframe from which variables need to be taken.
grouping.vars	A list of grouping variables.
measures	List variables for which summary needs to be computed. If not specified, all variables of type specified in the argument <code>measures.type</code> will be used to calculate summaries. Don't explicitly set <code>measures.type = NULL</code> in function call, which will produce an error because the function will try to find a column in a dataframe named "NULL".
measures.type	A character indicating whether summary for <i>numeric</i> ("numeric") or <i>factor/character</i> ("factor") variables is expected (Default: <code>measures.type = "numeric"</code>). This function can't be used for both numeric and variables simultaneously.
topcount.long	If <code>measures.type = factor</code> , you can get the top counts in long format for plotting purposes. (Default: <code>topcount.long = FALSE</code>).

Value

Dataframe with descriptive statistics for numeric variables (n, mean, sd, median, min, max)

Author(s)

Indrajeet Patil

Examples

```
# another possibility
groupedstats::grouped_summary(
  data = datasets::iris,
  grouping.vars = Species,
  measures = Sepal.Length:Petal.Width,
  measures.type = "numeric"
)

# if you have just one variable per argument, you need not use `c()`
groupedstats::grouped_summary(
  data = datasets::ToothGrowth,
  grouping.vars = supp,
  measures = len,
  measures.type = "numeric"
)
```

grouped_ttest	<i>Function to run t-test on multiple variables across multiple grouping variables.</i>
---------------	---

Description

Function to run t-test on multiple variables across multiple grouping variables.

Usage

```
grouped_ttest(data, dep.vars, indep.vars, grouping.vars, paired = FALSE,
  var.equal = FALSE)
```

Arguments

data	Dataframe from which variables are to be taken.
dep.vars	List dependent variables for a t-test (y in $y \sim x$).
indep.vars	List independent variables for a t-test (x in $y \sim x$).
grouping.vars	List of grouping variables.
paired	A logical indicating whether you want a paired t-test (Default: paired = FALSE; independent t-test, i.e.).
var.equal	A logical variable indicating whether to treat the two variances as being equal. If TRUE, then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used (Default: var.equal = FALSE; Welch's t-test, i.e.).

Value

A tibble dataframe with tidy results from t-test analyses.

Author(s)

Indrajeet Patil

Examples

```
groupedstats::grouped_ttest(
  data = dplyr::filter(.data = ggplot2::diamonds, color == "E" | color == "J"),
  dep.vars = c(carat, price, depth),
  indep.vars = color,
  grouping.vars = clarity,
  paired = FALSE,
  var.equal = FALSE
)
```

grouped_wilcox	<i>Function to run two-sample Wilcoxon tests on multiple variables across multiple grouping variables.</i>
----------------	--

Description

Function to run two-sample Wilcoxon tests on multiple variables across multiple grouping variables.

Usage

```
grouped_wilcox(data, dep.vars, indep.vars, grouping.vars, paired = FALSE,
  correct = TRUE)
```

Arguments

data	Dataframe from which variables are to be taken.
dep.vars	List dependent variables for a two-sample Wilcoxon tests (y in $y \sim x$).
indep.vars	List independent variables for a two-sample Wilcoxon tests (x in $y \sim x$).
grouping.vars	List of grouping variables.
paired	A logical indicating whether you want a paired two-sample Wilcoxon tests (Default: paired = FALSE).
correct	A logical indicating whether to apply continuity correction in the normal approximation for the p-value (Default: correct = TRUE).

Value

A tibble dataframe with tidy results from two-sample Wilcoxon tests analyses.

Author(s)

Indrajeet Patil

Examples

```
# only with one grouping variable
groupedstats::grouped_wilcox(
  data = dplyr::filter(.data = ggplot2::diamonds, color == "E" | color == "J"),
  dep.vars = depth:table,
  indep.vars = color,
  grouping.vars = clarity,
  paired = FALSE
)
```

intent_morality	<i>Moral judgments about third-party moral behavior.</i>
-----------------	--

Description

Moral judgments about third-party moral behavior.

Usage

```
intent_morality
```

Format

A data frame with 4016 rows and 8 variables

- id. Participant id.
- gender. Participant's gender.
- item. Which story/vignette participants read for a given condition.
- harm. What kind of harm was involved in the item.
- belief. What kind of belief the actor had (neutral or negative/harmful).
- outcome. What kind of outcome the actor caused (neutral or negative/harmful).
- condition. Type of harm, composed of belief and outcome.
- question. Type of moral judgment asked (wrongness or punishment).
- rating. Moral judgment rating on a scale of 1 to 7.

Details

This dataset contains data from a recent study about how people judge behavior of others when they unintentionally or intentionally cause harm to others.

Participants responded to four different vignettes that contains four different types of conditions-

- accidental harm. neutral belief, harmful/negative outcome
- intentional harm. harmful/negative belief, harmful/negative outcome
- attempted harm. harmful/negative belief, neutral outcome

- neutral harm. neutral belief, neutral outcome

Additionally, participants saw one of the four variants for each of the four items. Each of the item had a different type of harm.

Source

<https://www.nature.com/articles/s41598-017-05299-9>

Examples

```
dim(intent_morality)
head(intent_morality)
```

lm_effsize_ci	<i>Confidence intervals for partial eta-squared and omega-squared for linear models.</i>
---------------	--

Description

This function will convert a linear model object to a dataframe containing statistical details for all effects along with partial eta-squared effect size and its confidence interval.

Usage

```
lm_effsize_ci(object, effsize = "eta", partial = TRUE,
              conf.level = 0.95, nboot = 500)
```

Arguments

object	The linear model object (can be of class <code>lm</code> , <code>aov</code> , <code>anova</code> , or <code>aovlist</code>).
effsize	Character describing the effect size to be displayed: "eta" (default) or "omega".
partial	Logical that decides if partial eta-squared or omega-squared are returned (Default: TRUE). If FALSE, eta-squared or omega-squared will be returned. Valid only for objects of class <code>lm</code> , <code>aov</code> , <code>anova</code> , or <code>aovlist</code> .
conf.level	Numeric specifying Level of confidence for the confidence interval (Default: 0.95).
nboot	Number of bootstrap samples for confidence intervals for partial eta-squared and omega-squared (Default: 500).

Value

A dataframe with results from `stats::lm()` with partial eta-squared, omega-squared, and bootstrapped confidence interval for the same.

Author(s)

Indrajeet Patil

movies_long	<i>Movie information and user ratings from IMDB.com (long format).</i>
-------------	--

Description

Movie information and user ratings from IMDB.com (long format).

Usage

```
movies_long
```

Format

A data frame with 2433 rows and 8 variables

- title. Title of the movie.
- year. Year of release.
- budget. Total budget (if known) in US dollars
- length. Length in minutes.
- rating. Average IMDB user rating.
- votes. Number of IMDB users who rated this movie.
- mpaa. MPAA rating.
- genre. Different genres of movies (action, animation, comedy, drama, documentary, romance, short).

Details

Modified dataset from ggplot2movies package.

The internet movie database, <http://imdb.com/>, is a website devoted to collecting movie data supplied by studios and fans. It claims to be the biggest movie database on the web and is run by amazon. More about information imdb.com can be found online, http://imdb.com/help/show_leaf?about, including information about the data collection process, http://imdb.com/help/show_leaf?infosource.

Movies were selected for inclusion if they had a known length and had been rated by at least one imdb user.

Source

<https://CRAN.R-project.org/package=ggplot2movies>

Examples

```
dim(movies_long)
head(movies_long)
```

`movies_wide`*Movie information and user ratings from IMDB.com (wide format).*

Description

Movie information and user ratings from IMDB.com (wide format).

Usage

```
movies_wide
```

Format

A data frame with 1813 rows and 14 variables

- `title`. Title of the movie.
- `year`. Year of release.
- `budget`. Total budget (if known) in US dollars
- `length`. Length in minutes.
- `rating`. Average IMDB user rating.
- `votes`. Number of IMDB users who rated this movie.
- `mpaa`. MPAA rating.
- `action`, `animation`, `comedy`, `drama`, `documentary`, `romance`, `short`. Binary variables representing if movie was classified as belonging to that genre.

Details

Modified dataset from `ggplot2movies` package.

The internet movie database, <http://imdb.com/>, is a website devoted to collecting movie data supplied by studios and fans. It claims to be the biggest movie database on the web and is run by amazon. More about information `imdb.com` can be found online, http://imdb.com/help/show_leaf?about, including information about the data collection process, http://imdb.com/help/show_leaf?infosource.

Movies were selected for inclusion if they had a known length and had been rated by at least one `imdb` user.

Source

<https://CRAN.R-project.org/package=ggplot2movies>

Examples

```
dim(movies_wide)
head(movies_wide)
```

`set_cwd`*Setting Working Directory in RStudio to where the R Script is.*

Description

This function will change the current working directory to whichever directory the R script you are currently working on is located. This preempts the trouble of setting the working directory manually.

Usage

```
set_cwd()
```

Value

Path to changed working directory.

Note

This function will work **only with RStudio IDE**. Reference: <https://eranraviv.com/r-tips-and-tricks-working-directory/>

`signif_column`*Creating a new character type column with significance labels*

Description

This function will add a new column to a dataframe containing p -values

Usage

```
signif_column(data = NULL, p, messages = FALSE)
```

Arguments

<code>data</code>	Data frame from which variables specified are preferentially to be taken.
<code>p</code>	The column containing p -values.
<code>messages</code>	Logical decides whether to produce notes (Default: TRUE).

Value

Returns the originally entered object (either a vector or a dataframe) in tibble format with an additional column corresponding to statistical significance.

Author(s)

Indrajeet Patil

Examples

```
# vector as input
groupedstats::signif_column(p = c(0.05, 0.1, 1, 0.00001, 0.001, 0.01))

# dataframe as input
# preparing a newdataframe
df <- cbind.data.frame(
  x = 1:5,
  y = 1,
  p.value = c(0.1, 0.5, 0.00001, 0.05, 0.01)
)

groupedstats::signif_column(data = df, p = p.value)

# numbers entered as characters are also tolerated
groupedstats::signif_column(p = c("1", "0.1", "0.0002", "0.03", "0.65"))
```

specify_decimal_p	<i>Custom function for getting specified number of decimal places in results for p-value</i>
-------------------	--

Description

Function to format an R object for pretty printing with a specified (k) number of decimal places. The function also allows really small p -values to be denoted as " $p < 0.001$ " rather than " $p = 0.000$ ". Note that if `p.value` is set to TRUE, the minimum value of k allowed is 3. If k is set to less than 3, the function will ignore entered k value and use $k = 3$ instead.

Usage

```
specify_decimal_p(x, k = 3, p.value = FALSE)
```

Arguments

<code>x</code>	A numeric variable.
<code>k</code>	Number of digits after decimal point (should be an integer) (Default: $k = 3$).
<code>p.value</code>	Decides whether the number is a p -value (Default: FALSE).

Value

Formatted numeric values.

Author(s)

Indrajeet Patil

Examples

```
groupedstats::specify_decimal_p(x = 0.00001, k = 2, p.value = TRUE)
groupedstats::specify_decimal_p(x = 0.008, k = 2, p.value = TRUE)
groupedstats::specify_decimal_p(x = 0.008, k = 3, p.value = FALSE)
```

Titanic_full	<i>Titanic dataset.</i>
--------------	-------------------------

Description

Titanic dataset.

Usage

```
Titanic_full
```

Format

A data frame with 2201 rows and 5 variables

- id. Dummy identity number for each person.
- Class. 1st, 2nd, 3rd, Crew.
- Sex. Male, Female.
- Age. Child, Adult.
- Survived. No, Yes.

Details

This data set provides information on the fate of passengers on the fatal maiden voyage of the ocean liner 'Titanic', summarized according to economic status (class), sex, age, and survival.

This is a modified dataset from datasets package.

Source

<https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/Titanic.html>

Examples

```
dim(Titanic_full)
head(Titanic_full)
```

Index

*Topic **datasets**

- intent_morality, 15
- movies_long, 17
- movies_wide, 18
- Titanic_full, 21

dummy, 8

family, 4, 5
formula, 4, 7

glm, 5
glmerControl, 5, 8
grouped_aov, 3
grouped_glm, 4
grouped_glmer, 5
grouped_lm, 7
grouped_lmer, 8
grouped_proptest, 9
grouped_robustslr, 10
grouped_simplelm (grouped_slr), 11
grouped_slr, 11
grouped_summary, 12
grouped_ttest, 13
grouped_wilcox, 14
groupedstats-package, 2

intent_morality, 15

lm_effsize_ci, 16
lmerControl, 5, 8

movies_long, 17
movies_wide, 18

set_cwd, 19
signif_column, 19
specify_decimal_p, 20

Titanic_full, 21