

Package ‘tidysmd’

May 8, 2026

Title Tidy Standardized Mean Differences

Version 0.2.1

Description Tidy standardized mean differences ('SMDs'). 'tidysmd' uses the 'smd' package to calculate standardized mean differences for variables in a data frame, returning the results in a tidy format.

License MIT + file LICENSE

URL <https://github.com/r-causal/tidysmd>,
<https://r-causal.github.io/tidysmd/>

BugReports <https://github.com/r-causal/tidysmd/issues>

Depends R (>= 3.5)

Imports dplyr, purrr, rlang, smd, stats, tidyr, tidyselect, utils,
vctrs

Suggests covr, ggplot2, MatchIt, propensity, spelling, testthat (>= 3.0.0), vdiffr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Malcolm Barrett [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-0299-5825>>)

Maintainer Malcolm Barrett <malcolmbarrett@gmail.com>

Repository CRAN

Date/Publication 2026-03-04 06:30:26 UTC

Contents

bind_matches	2
geom_love	2
nhfs_weights	4
tidy_smd	5

Index	7
--------------	----------

bind_matches	<i>Bind Match Indicator Columns to a Data Frame</i>
--------------	---

Description

Given a data frame `.df`, the function `bind_matches` creates binary indicator variables for each match returned by the `MatchIt` library and binds the resulting columns to `.df`. In other words, the result is the original data frame plus a column for however many matches you want to bind.

Usage

```
bind_matches(.df, ...)
```

Arguments

<code>.df</code>	A data frame.
<code>...</code>	<code>matchit</code> objects returned by the <code>MatchIt</code> package. They can be named or unnamed.

Value

`.df` with addition columns for every element of `...`

geom_love	<i>Create a Love plot</i>
-----------	---------------------------

Description

`geom_love()` and `love_plot()` are helper functions to create Love plots in `ggplot2`. Love plots are a diagnostic approach to assessing balance before and after weighting. Many researchers use 0.1 on the absolute SMD scale to evaluate if a variable is well-balanced between groups, although this is just a rule of thumb. `geom_love()` is a simple wrapper around `ggplot2::geom_point()`, `ggplot2::geom_line()`, and `ggplot2::geom_vline()`. It also adds default aesthetics via `ggplot2::aes()`. `love_plot()` is a quick plotting function that further wraps `geom_love()`. For more complex Love plots, we recommend using `ggplot2` directly.

Usage

```
geom_love(
  data = NULL,
  linewidth = 0.8,
  line_size = NULL,
  point_size = 1.85,
  vline_xintercept = 0.1,
  vline_color = "grey70",
  vlinewidth = 0.6,
  vline_size = NULL
)

love_plot(
  .df,
  linewidth = 0.8,
  line_size = NULL,
  point_size = 1.85,
  vline_xintercept = 0.1,
  vline_color = "grey70",
  vlinewidth = 0.6,
  vline_size = NULL
)
```

Arguments

data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code> . A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).
linewidth	The line size, passed to <code>ggplot2::geom_line()</code> .
line_size	Deprecated. Please use linewidth.
point_size	The point size, passed to <code>ggplot2::geom_point()</code> .
vline_xintercept	The X intercept, passed to <code>ggplot2::geom_vline()</code> .
vline_color	The vertical line color, passed to <code>ggplot2::geom_vline()</code> .
vlinewidth	The vertical line size, passed to <code>ggplot2::geom_vline()</code> .
vline_size	Deprecated. Please use vlinewidth.
.df	a data frame produced by <code>tidy_smd()</code>

Value

a list of geoms or a ggplot

Examples

```
plot_df <- tidy_smd(
  nhefs_weights,
  race:active,
  .group = qsmk,
  .wts = starts_with("w_")
)

love_plot(plot_df)

# or use ggplot2 directly
library(ggplot2)
ggplot(
  plot_df,
  aes(
    x = abs(smd),
    y = variable,
    group = method,
    color = method,
    fill = method
  )
) +
  geom_love()
```

nhefs_weights

NHEFS with various propensity score weights

Description

A dataset containing various propensity score weights for `causaldata::nhefs_complete`.

Usage

```
nhefs_weights
```

Format

A data frame with 1566 rows and 14 variables:

qsmk Quit smoking

race Race

age Age

education Education level

smokeintensity Smoking intensity

smokeyears Number of smoke-years

exercise Exercise level

active Daily activity level
wt71 Participant weight in 1971 (baseline)
w_ate ATE weight
w_att ATT weight
w_atc ATC weight
w_atm ATM weight
w_ato ATO weight

tidy_smd

Tidy Standardized Mean Differences

Description

tidy_smd() calculates the standardized mean difference (SMD) for variables in a dataset between groups. Optionally, you may also calculate weighted SMDs. tidy_smd() wraps `smd::smd()`, returning a tidy dataframe with the columns `variable`, `method`, and `smd`, as well as fourth column the contains the level of `.group` the SMD represents. You may also supply multiple weights to calculate multiple weighted SMDs, useful when comparing different types of weights. Additionally, the `.wts` argument supports matched datasets where the variable supplied to `.wts` is a binary variable indicating whether the row was included in the match. If you're using `MatchIt`, the helper function `bind_matches()` will bind these indicators to the original dataset, making it easier to compare across matching specifications.

Usage

```

tidy_smd(
  .df,
  .vars,
  .group,
  .wts = NULL,
  include_observed = TRUE,
  include_unweighted = NULL,
  na.rm = FALSE,
  gref = 1L,
  std.error = FALSE,
  make_dummy_vars = FALSE
)
  
```

Arguments

<code>.df</code>	A data frame
<code>.vars</code>	Variables for which to calculate SMD. Can be unquoted (x) or quoted ("x").
<code>.group</code>	Grouping variable. Can be unquoted (x) or quoted ("x").

<code>.wts</code>	Variables to use for weighting the SMD calculation. These can be, for instance, propensity score weights or a binary indicator signaling whether or not a participant was included in a matching algorithm. Can be unquoted (<code>x</code>) or quoted (" <code>x</code> ").
<code>include_observed</code>	Logical. If using <code>.wts</code> , also calculate the unweighted SMD?
<code>include_unweighted</code>	Deprecated. Please use <code>include_observed</code> .
<code>na.rm</code>	Remove NA values from <code>x</code> ? Defaults to FALSE.
<code>gref</code>	an integer indicating which level of <code>g</code> to use as the reference group. Defaults to 1.
<code>std.error</code>	Logical indicator for computing standard errors using <code>compute_smd_var</code> . Defaults to FALSE.
<code>make_dummy_vars</code>	Logical. Transform categorical variables to dummy variables using <code>model.matrix()</code> ? By default, <code>smd::smd</code> uses a summary value based on the Mahalanobis distance distance to approximate the SMD of categorical variables. An alternative approach is to transform categorical variables to a set of dummy variables.

Value

a tibble

Examples

```
tidy_smd(nhefs_weights, c(age, education, race), .group = qsmk)
tidy_smd(nhefs_weights, c(age, education), .group = qsmk, std.error = TRUE)

tidy_smd(
  nhefs_weights,
  c(age, race, education),
  .group = qsmk,
  .wts = c(w_ate, w_att, w_atm)
)
```

Index

* datasets

nhefs_weights, 4

bind_matches, 2

bind_matches(), 5

compute_smd_var, 6

fortify(), 3

geom_love, 2

ggplot(), 3

ggplot2::aes(), 2

ggplot2::geom_line(), 2, 3

ggplot2::geom_point(), 2, 3

ggplot2::geom_vline(), 2, 3

love_plot (geom_love), 2

nhefs_weights, 4

smd::smd, 6

tidy_smd, 5