Package ‘tidysmd’

May 26, 2023

**Title**  Tidy Standardized Mean Differences

**Version**  0.2.0

**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


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

**Depends**  R (>= 2.10)

**Imports**  dplyr, purrr, rlang, smd, stats, tidy, tidyselect, utils

**Suggests**  covr, ggplot2, MatchIt, spelling, testthat (>= 3.0.0), vdiff

**Config/testthat/edition**  3

**Encoding**  UTF-8

**Language**  en-US

**LazyData**  true

**RoxygenNote**  7.2.3

**NeedsCompilation**  no

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

**Maintainer**  Malcolm Barrett <malcolmbarrett@gmail.com>

**Repository**  CRAN

**Date/Publication**  2023-05-26 17:30:02 UTC

**R topics documented:**

`bind_matches` ......................................................... 2
`geom_love` ............................................................. 2
`nhefs_weights` ......................................................... 4
`tidy_smd` ............................................................... 5

**Index**
bind_matches  

**Bind Match Indicator Columns to a Data Frame**

**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**

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

**Arguments**

- `.df` A data frame.
- `...` matchit objects returned by the MatchIt package. They can be named or unnamed.

**Value**

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

---

geom_love  

**Create a Love plot**

**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**

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

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

linewidth  The line size, passed to ggplot2::geom_line().
line_size  Deprecated. Please use linewidth.
point_size  The point size, passed to ggplot2::geom_point().
vline_xintercept  The X intercept, passed to ggplot2::geom_vline().
vline_color  The vertical line color, passed to ggplot2::geom_vline().
vlinewidth  The vertical line size, passed to ggplot2::geom_vline().
vline_size  Deprecated. Please use vlinewidth.
.df  a data frame produced by tidy_smd()

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,
  )
)

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
- smokeyrs  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 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 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 an 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**

- **.df** A data frame
- **.vars** Variables for which to calculate SMD
- **.group** Grouping variable
- **.wts** 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.
- **include_observed** Logical. If using .wts, also calculate the unweighted SMD?
- **include_unweighted** Deprecated. Please use include_observed.
- **na.rm** Remove NA values from x? Defaults to FALSE.
- **gref** an integer indicating which level of g to use as the reference group. Defaults to 1.
std.error  Logical indicator for computing standard errors using `compute_smd_var`. Defaults to FALSE.

make_dummy_vars  Logical. Transform categorical variables to dummy variables using `model.matrix()`? By default, `smd::smd` 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
geom_love, 2
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