Package ‘esvis’

May 1, 2020

Type Package

Title Visualization and Estimation of Effect Sizes

Version 0.3.1

Description A variety of methods are provided to estimate and visualize distributional differences in terms of effect sizes. Particular emphasis is upon evaluating differences between two or more distributions across the entire scale, rather than at a single point (e.g., differences in means). For example, Probability-Probability (PP) plots display the difference between two or more distributions, matched by their empirical CDFs (see Ho and Reardon, 2012; <doi:10.3102/1076998611411918>), allowing for examinations of where on the scale distributional differences are largest or smallest. The area under the PP curve (AUC) is an effect-size metric, corresponding to the probability that a randomly selected observation from the x-axis distribution will have a higher value than a randomly selected observation from the y-axis distribution. Binned effect size plots are also available, in which the distributions are split into bins (set by the user) and separate effect sizes (Cohen's d) are produced for each bin - again providing a means to evaluate the consistency (or lack thereof) of the difference between two or more distributions at different points on the scale. Evaluation of empirical CDFs is also provided, with built-in arguments for providing annotations to help evaluate distributional differences at specific points (e.g., semi-transparent shading). All function take a consistent argument structure. Calculation of specific effect sizes is also possible. The following effect sizes are estimable: (a) Cohen's d, (b) Hedges' g, (c) percentage above a cut, (d) transformed (normalized) percentage above a cut, (e) area under the PP curve, and (f) the V statistic (see Ho, 2009; <doi:10.3102/1076998609332755>), which essentially transforms the area under the curve to standard deviation units. By default, effect sizes are calculated for all possible pairwise comparisons, but a reference group (distribution) can be specified.

Depends R (>= 3.1)

Imports sfsmisc, ggplot2, magrittr, dplyr, rlang, tidyr (>= 1.0.0), purrr, Hmisc, tibble

URL https://github.com/datalorax/esvis

BugReports https://github.com/datalorax/esvis/issues

License MIT + file LICENSE

LazyData true

RoxygenNote 7.0.2

Suggests testthat, viridisLite

NeedsCompilation no

Author Daniel Anderson [aut, cre]

Maintainer Daniel Anderson <daniela@uoregon.edu>

Repository CRAN

Date/Publication 2020-04-30 23:20:02 UTC

R topics documented:

---

auc — *Compute the Area Under the pp_plot Curve Calculates the area under the pp curve. The area under the curve is also a useful effect-size like statistic, representing the probability that a randomly selected individual from the x distribution will have a higher value than a randomly selected individual from the y distribution.*

Description

Compute the Area Under the pp_plot Curve Calculates the area under the pp curve. The area under the curve is also a useful effect-size like statistic, representing the probability that a randomly selected individual from the x distribution will have a higher value than a randomly selected individual from the y distribution.

Usage

```
auc(data, formula, ref_group = NULL, rename = TRUE)
```

Arguments

`data` The data frame used for estimation - ideally structured in a tidy format.

`formula` A formula of the type `out ~ group` where `out` is the outcome variable and `group` is the grouping variable. Note this variable can include any arbitrary number of groups. Additional variables can be included with `+` to produce separate estimates by the secondary or tertiary variables of interest (e.g., `out ~ group + characteristic1 + characteristic2`).

`ref_group` Optional. A character vector or forumla listing the reference group levels for each variable on the right hand side of the formula, supplied in the same order as the formula. Note that if using the formula version, levels that are numbers, or include hyphens, spaces, etc., should be wrapped in back ticks (e.g., ``ref_group = ~ Active + `Non-FRL` ``, or ``ref_group = ~`8` ``). When in doubt, it is safest to use the back ticks, as they will not interfere with anything if they are not needed. See examples below for more details.

`rename` Used primarily for internal purposes. Should the column names be renamed to reference the focal and reference groups? Defaults to `TRUE`.

Value

By default the area under the curve for all possible pairings of the grouping factor are returned.

Examples

```
# Calculate AUC for all pairwise comparisons
auc(star, reading ~ condition)

# Report only relative to regular-sized classrooms
auc(star,
    reading ~ condition,
    ref_group = "reg")

# Report by ELL and FRL groups for each season, compare to non-ELL students
# who were not eligible for free or reduced price lunch in the fall (using
# the formula interface for reference group referencing).
## Not run:
auc(benchmarks,
      math ~ ell + frl + season,
      ref_group = ~`Non-ELL` + `Non-FRL` + Fall)

# Same thing but with character vector supplied, rather than a formula
auc(benchmarks,
      math ~ ell + frl + season,
      ref_group = c("Non-ELL", "Non-FRL", "Fall"))

## End(Not run)
```

benchmarks *Synthetic benchmark screening data*

Description

Across the country many schools engage in seasonal benchmark screenings to monitor to progress of their students. These are relatively brief assessments administered to "check-in" on students' progress throughout the year. This dataset was simulated from a real dataset from one large school district using the terrific synthpop R package. Overall characteristics of the synthetic data are remarkably similar to the real data.

Usage

```
benchmarks
```

Format

A data frame with 10240 rows and 9 columns.

sid Integer. Student identifier.

cohort Integer. Identifies the cohort from which the student was sampled (1-3).

sped Character. Special Education status: "Non-Sped" or "Sped"

ethnicity Character. The race/ethnicity to which the student identified. Takes on one of seven values: "Am. Indian", "Asian", "Black", "Hispanic", "Native Am.", "Two or More", and "White"

frl Character. Student's eligibility for free or reduced price lunch. Takes on the values "FRL" and "Non-FRL".

ell Character. Students' English language learner status. Takes on one of values: "Active", "Monitor", and "Non-ELL". Students coded "Active" were actively receiving English language services at the time of testing. Students coded "Monitor" had previously received services, but not at the time of testing. Students coded "Non-ELL" did not receive services at any time.

season Character. The season during which the assessment was administered: "Fall", "Winter", or "Spring"

reading Integer. Reading scale score.

math Integer. Mathematics scale score.

binned_es *Calculate binned effect sizes*

Description

Calculate binned effect sizes

Usage

```
binned_es(
  data,
  formula,
  ref_group = NULL,
  qtile_groups = 3,
  es = "g",
  rename = TRUE
)
```

Arguments

`data`	The data frame used for estimation - ideally structured in a tidy format.
`formula`	A formula of the type `out ~ group` where `out` is the outcome variable and `group` is the grouping variable. Note this variable can include any arbitrary number of groups. Additional variables can be included with `+` to produce separate estimates by the secondary or tertiary variables of interest (e.g., `out ~ group + characteristic1 + characteristic2`).
`ref_group`	Optional. A character vector or forumla listing the reference group levels for each variable on the right hand side of the formula, supplied in the same order as the formula. Note that if using the formula version, levels that are numbers, or include hyphens, spaces, etc., should be wrapped in back ticks (e.g., `` ref_group = ~ Active + `Non-FRL` ``, or `` ref_group = ~`8` ``). When in doubt, it is safest to use the back ticks, as they will not interfere with anything if they are not needed. See examples below for more details.
`qtile_groups`	The number of quantile bins to split the data by and calculate effect sizes. Defaults to 3 bins (lower, middle, upper).
`es`	The effect size to calculate. Currently the only options are "d" or "g".
`rename`	Logical. Should the column names be relabeled according to the reference and focal groups. Defaults to `TRUE`.

Value

A data frame with the corresponding effect sizes.

binned_plot *Quantile-binned effect size plot*

Description

Plots the effect size between focal and reference groups by matched (binned) quantiles (i.e., the results from binned_es), with the matched quantiles plotted along the x-axis and the effect size plotted along the y-axis. The intent is to examine how (if) the magnitude of the effect size varies at different points of the distributions. The mean differences within each quantile bin are divided by the overall pooled standard deviation for the two groups being compared.

Usage

```
binned_plot(
  data,
  formula,
  ref_group = NULL,
  qtile_groups = 3,
  es = "g",
  lines = TRUE,
  points = TRUE,
  shade = TRUE,
  shade_alpha = 0.4,
  rects = TRUE,
  rect_fill = "gray20",
  rect_alpha = 0.35,
  refline = TRUE,
  refline_col = "gray40",
  refline_lty = "solid",
  refline_lwd = 1.1
)
```

Arguments

`data`	The data frame to be plotted
`formula`	A formula of the type `out ~ group` where `out` is the outcome variable and `group` is the grouping variable. Note this variable can include any arbitrary number of groups. Additional variables can be included with `+` to produce separate plots by the secondary or tertiary variable of interest (e.g., `out ~ group + characteristic1 + characteristic2`). No more than two additional characteristics can be supplied at this time.
`ref_group`	Optional character vector (of length 1) naming the reference group. Defaults to the group with the highest mean score.
`qtile_groups`	The number of quantile bins to split the data by and calculate effect sizes. Defaults to 3 bins (lower, middle, upper).

`es`	The effect size to plot. Defaults to `"g"`, in which case Hedge's g is plotted, which is better for small samples. At present, the only other option is `"d"` for Cohen's D.
`lines`	Logical. Should the PP Lines be plotted? Defaults to `TRUE`.
`points`	Logical. Should points be plotted for each `qtiles` be plotted? Defaults to `TRUE`.
`shade`	Logical. Should the standard errors around the effect size point estimates be displayed? Defaults to `TRUE`, with the uncertainty displayed with shading.
`shade_alpha`	Transparency level of the standard error shading. Defaults to 0.40.
`rects`	Logical. Should semi-transparent rectangles be plotted in the background to show the binning? Defaults to `TRUE`.
`rect_fill`	Color fill of rectangles to be plotted in the background, if `rects == TRUE`. Defaults to "gray20".
`rect_alpha`	Transparency level of the rectangles in the background when `rects == TRUE`. Defaults to 0.35.
`refline`	Logical. Defaults to `TRUE`. Should a diagonal reference line, representing the point of equal probabilities, be plotted?
`refline_col`	The color of the reference line. Defaults to `"gray40"`
`refline_lty`	Line type of the reference line. Defaults to `"solid"`.
`refline_lwd`	Line width of the reference line. Defaults to `1.1`.

Examples

```
# Binned Effect Size Plot: Defaults to Hedges' G
binned_plot(star, math ~ condition)

# Same plot, separated by sex
binned_plot(star, math ~ condition + sex)

# Same plot by sex and race
## Not run:
  pp_plot(star, math ~ condition + sex + race)

## End(Not run)
## Evaluate with simulated data: Plot is most interesting when variance
# in the distributions being compared differ.

library(tidyr)
library(ggplot2)

# simulate data with different variances
set.seed(100)
common_vars <- data.frame(low  = rnorm(1000, 10, 1),
                          high = rnorm(1000, 12, 1),
                          vars = "common")
diff_vars <- data.frame(low  = rnorm(1000, 10, 1),
                        high = rnorm(1000, 12, 2),
                        vars = "diff")
d <- rbind(common_vars, diff_vars)
```

```
# Plot distributions
d <- d %>%
gather(group, value, -vars)

ggplot(d, aes(value, color = group)) +
 geom_density() +
 facet_wrap(~vars)

# Note that the difference between the distributions depends on where you're
# evaluating from on the x-axis. The binned plot helps us visualize this.
# The below shows the binned plots when there is a common versus different
# variance

binned_plot(d, value ~ group + vars)
```

---

coh_d *Compute Cohen's* d

Description

This function calculates effect sizes in terms of Cohen's *d*, also called the uncorrected effect size. See hedg_g for the sample size corrected version. Also see Lakens (2013) for a discussion on different types of effect sizes and their interpretation. Note that missing data are removed from the calculations of the means and standard deviations.

Usage

```
coh_d(data, formula, ref_group = NULL, se = TRUE)
```

Arguments

`data`	The data frame used for estimation - ideally structured in a tidy format.
`formula`	A formula of the type `out ~ group` where `out` is the outcome variable and `group` is the grouping variable. Note this variable can include any arbitrary number of groups. Additional variables can be included with `+` to produce separate estimates by the secondary or tertiary variables of interest (e.g., `out ~ group + characteristic1 + characteristic2`).
`ref_group`	Optional. A character vector or forumla listing the reference group levels for each variable on the right hand side of the formula, supplied in the same order as the formula. Note that if using the formula version, levels that are numbers, or include hyphens, spaces, etc., should be wrapped in back ticks (e.g., ``ref_group = ~ Active + `Non-FRL` ``, or ``ref_group = ~`8` ``). When in doubt, it is safest to use the back ticks, as they will not interfere with anything if they are not needed. See examples below for more details.
`se`	Logical. Should the standard error of the effect size be estimated and returned in the resulting data frame? Defaults to `TRUE`.

Value

By default the Cohen's *d* for all possible pairings of the grouping factor(s) are returned.

Examples

```
# Calculate Cohen's d for all pairwise comparisons
coh_d(star, reading ~ condition)

# Report only relative to regular-sized classrooms
coh_d(star,
      reading ~ condition,
   ref_group = "reg")

# Report by ELL and FRL groups for each season, compare to non-ELL students
# who were not eligible for free or reduced price lunch in the fall (using
# the formula interface for reference group referencing).

coh_d(benchmarks,
      math ~ ell + frl + season,
      ref_group = ~`Non-ELL` + `Non-FRL` + Fall)

# Same thing but with character vector supplied, rather than a formula
coh_d(benchmarks,
      math ~ ell + frl + season,
      ref_group = c("Non-ELL", "Non-FRL", "Fall"))
```

---

ecdf_plot *Empirical Cumulative Distribution Plot*

Description

This is a wrapper function for the stat_ecdf function and helps make it easy to directly compare distributions at specific locations along the scale.

Usage

```
ecdf_plot(
  data,
  formula,
  cuts = NULL,
  linewidth = 1.2,
  ref_line_cols = "gray40",
  ref_linetype = "solid",
  center = FALSE,
  ref_rect = TRUE,
  ref_rect_col = "gray40",
  ref_rect_alpha = 0.15
)
```

Arguments

data	A tidy data frame containing the data to be plotted.
formula	A formula of the type `out ~ group` where `out` is the outcome variable and `group` is the grouping variable. Note this variable can include any arbitrary number of groups. Additional variables can be included with `+` to produce separate plots by the secondary or tertiary varaible (e.g., `out ~ group + characteristic1 + characteristic2`). No more than two additional characteristics can be supplied at this time.
cuts	Optional numeric vector stating the location of reference line(s) and/or rectangle(s).
linewidth	Width of ECDF lines. Note that the color of the lines can be controlled through additional functions (e.g., `scale_color_brewer,scale_color_manual`).
ref_line_cols	Optional vector (or single value) of colors for `cuts` lines.
ref_linetype	Optional vector (or single value) of line types for `cuts` lines. Takes any of the arguments supplied by linetype.
center	Logical. Should the functions be centered prior to plotting? Defaults to `FALSE`. Note that if paneled/faceted plots are produced, the centering occurs by group.
ref_rect	Logical, defaults to `TRUE` when `cuts` takes any non-null value. Should semi-transparent rectangle(s) be plotted at the locations of `cuts`?
ref_rect_col	Color of the fill for the reference rectangles. Defaults to a dark gray.
ref_rect_alpha	Transparency of the fill for the reference rectangles. Defaults to 0.7.

Examples

```
ecdf_plot(benchmarks, math ~ ell,
          cuts = c(190, 205, 210),
          ref_line_cols = c("#D68EE3", "#9BE38E", "#144ECA"))

# Customize the plot with ggplot2 functions
library(ggplot2)
ecdf_plot(benchmarks, math ~ ell,
          cuts = c(190, 205, 210),
          ref_line_cols = c("#D68EE3", "#9BE38E", "#144ECA")) +
  theme_minimal() +
  theme(legend.position = "bottom")

ecdf_plot(seda, mean ~ grade) +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal()

# Use within the dplyr pipeline
library(dplyr)
benchmarks %>%
  mutate(season = factor(season,
                         levels = c("Fall", "Winter", "Spring"))) %>%
  ecdf_plot(math ~ ell + season + frl)
```

hedg_g — *Compute Hedges' g This function calculates effect sizes in terms of Hedges' g, also called the corrected (for sample size) effect size. See* coh_d *for the uncorrected version. Also see* R*hrefhttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC3840331/Lakens (2013) for a discussion on different types of effect sizes and their interpretation. Note that missing data are removed from the calculations of the means and standard deviations.*

Description

Compute Hedges' *g* This function calculates effect sizes in terms of Hedges' *g*, also called the corrected (for sample size) effect size. See coh_d for the uncorrected version. Also see Lakens (2013) for a discussion on different types of effect sizes and their interpretation. Note that missing data are removed from the calculations of the means and standard deviations.

Usage

```
hedg_g(data, formula, ref_group = NULL, keep_d = TRUE)
```

Arguments

data	The data frame used for estimation - ideally structured in a tidy format.
formula	A formula of the type `out ~ group` where `out` is the outcome variable and `group` is the grouping variable. Note this variable can include any arbitrary number of groups. Additional variables can be included with `+` to produce separate estimates by the secondary or tertiary variables of interest (e.g., `out ~ group + characteristic1 + characteristic2`).
ref_group	Optional. A character vector or forumla listing the reference group levels for each variable on the right hand side of the formula, supplied in the same order as the formula. Note that if using the formula version, levels that are numbers, or include hyphens, spaces, etc., should be wrapped in back ticks (e.g., ``ref_group = ~ Active + `Non-FRL` ``, or ``ref_group = ~`8` ``). When in doubt, it is safest to use the back ticks, as they will not interfere with anything if they are not needed. See examples below for more details.
keep_d	Logical. Should Cohen's *d* be reported along with Hedge's g? Defaults to `TRUE`.

Value

By default the Hedges' *g* for all possible pairings of the grouping factor are returned as a tidy data frame.

Examples

```
# Calculate Hedges' g for all pairwise comparisons
hedg_g(star, reading ~ condition)
```

```
# Report only relative to regular-sized classrooms
hedg_g(star,
       reading ~ condition,
       ref_group = "reg")

# Report by ELL and FRL groups for each season, compare to non-ELL students
# who were not eligible for free or reduced price lunch in the fall (using
# the formula interface for reference group referencing).

hedg_g(benchmarks,
      math ~ ell + frl + season,
      ref_group = ~`Non-ELL` + `Non-FRL` + Fall)

# Same thing but with character vector supplied, rather than a formula
hedg_g(benchmarks,
      math ~ ell + frl + season,
      ref_group = c("Non-ELL", "Non-FRL", "Fall"))
```

---

pac *Compute the proportion above a specific cut location*

Description

Computes the proportion of the corresponding group, as specified by the `formula`, scoring above the specified `cuts`.

Usage

```
pac(data, formula, cuts, ref_group = NULL)
```

Arguments

`data`	The data frame used for estimation - ideally structured in a tidy format.
`formula`	A formula of the type `out ~ group` where `out` is the outcome variable and `group` is the grouping variable. Note this variable can include any arbitrary number of groups. Additional variables can be included with `+` to produce separate estimates by the secondary or tertiary variables of interest (e.g., `out ~ group + characteristic1 + characteristic2`).
`cuts`	Optional vector of cut scores. If supplied, the ECDF will be guaranteed to include these points. Otherwise, there could be gaps in the ECDF at those particular points (used in plotting the cut scores).
`ref_group`	Optional. A character vector or forumla listing the reference group levels for each variable on the right hand side of the formula, supplied in the same order as the formula. Note that if using the formula version, levels that are numbers, or include hyphens, spaces, etc., should be wrapped in back ticks (e.g., `ref_group = ~ Active + `Non-FRL``, or `ref_group = ~`8``). When in doubt, it is safest to use the back ticks, as they will not interfere with anything if they are not needed. See examples below for more details.

Value

Tidy data frame of the proportion above the cutoff for each (or selected) groups.

See Also

[esvis::pac_compare(), esvis::tpac(), esvis::tpac_diff()]

Examples

```
# Compute differences for all pairwise comparisons for each of three cuts
pac(star,
    reading ~ condition,
 cuts = c(450, 500, 550))

pac(star,
    reading ~ condition + freelunch + race,
 cuts = c(450, 500))

pac(star,
    reading ~ condition + freelunch + race,
 cuts = c(450, 500),
 ref_group = ~small + no + white)
```

---

pac_compare — *Compute the difference in the proportion above a specific cut location*

Description

Computes the difference in the proportion above the specified `cuts` for all possible pairwise comparisons of the groups specified by the `formula`.

Usage

```
pac_compare(data, formula, cuts, ref_group = NULL)
```

Arguments

`data`	The data frame used for estimation - ideally structured in a tidy format.
`formula`	A formula of the type `out ~ group` where `out` is the outcome variable and `group` is the grouping variable. Note this variable can include any arbitrary number of groups. Additional variables can be included with `+` to produce separate estimates by the secondary or tertiary variables of interest (e.g., `out ~ group + characteristic1 + characteristic2`).
`cuts`	Optional vector of cut scores. If supplied, the ECDF will be guaranteed to include these points. Otherwise, there could be gaps in the ECDF at those particular points (used in plotting the cut scores).

ref_group — Optional. A character vector or forumla listing the reference group levels for each variable on the right hand side of the formula, supplied in the same order as the formula. Note that if using the formula version, levels that are numbers, or include hyphens, spaces, etc., should be wrapped in back ticks (e.g., `ref_group = ~ Active + `Non-FRL``, or `ref_group = ~`8``). When in doubt, it is safest to use the back ticks, as they will not interfere with anything if they are not needed. See examples below for more details.

Value

Tidy data frame of the proportion above the cutoff for each (or selected) groups.

See Also

[esvis::pac(), esvis::tpac(), esvis::tpac_diff()]

Examples

```
# Compute differences for all pairwise comparisons for each of three cuts
pac_compare(star,
    reading ~ condition,
 cuts = c(450, 500, 550))

pac_compare(star,
    reading ~ condition + freelunch + race,
 cuts = c(450, 500))

pac_compare(star,
    reading ~ condition + freelunch + race,
 cuts = c(450, 500),
 ref_group = ~small + no + white)
```

---

pp_plot — *Produces the paired probability plot for two groups*

Description

The paired probability plot maps the probability of obtaining a specific score for each of two groups. The area under the curve (auc) corresponds to the probability that a randomly selected observation from the x-axis group will have a higher score than a randomly selected observation from the y-axis group. This function extends the basic pp-plot by allowing multiple curves and faceting to facilitate a variety of comparisons. Note that because the plotting is built on top of ggplot2, additional customization can be made on top of the plots, as illustrated in the examples.

Usage

```
pp_plot(
  data,
  formula,
  ref_group = NULL,
  cuts = NULL,
  cut_labels = TRUE,
  cut_label_x = 0.02,
  cut_label_size = 3,
  lines = TRUE,
  linetype = "solid",
  linewidth = 1.1,
  shade = TRUE,
  shade_alpha = 0.2,
  refline = TRUE,
  refline_col = "gray40",
  refline_type = "dashed",
  refline_width = 1.1
)
```

Arguments

`data`	The data frame to be plotted
`formula`	A formula of the type `out ~ group` where `out` is the outcome variable and `group` is the grouping variable. Note this variable can include any arbitrary number of groups. Additional variables can be included with `+` to produce separate plots by the secondary or tertiary variable of interest (e.g., `out ~ group + characteristic1 + characteristic2`). No more than two additional characteristics can be supplied at this time.
`ref_group`	Optional character vector (of length 1) naming the reference group. Defaults to the group with the highest mean score.
`cuts`	Integer. Optional vector (or single number) of scores used to annotate the plot. If supplied, line segments will extend from the corresponding x and y axes and meet at the PP curve.
`cut_labels`	Logical. Should the reference lines corresponding to `cuts` be labeled? Defaults to `TRUE`.
`cut_label_x`	The x-axis location of the cut labels. Defaults to 0.02.
`cut_label_size`	The size of the cut labels. Defaults to 3.
`lines`	Logical. Should the PP Lines be plotted? Defaults to `TRUE`.
`linetype`	The linetype for the PP lines. Defaults to "solid".
`linewidth`	The width of the PP lines. Defaults to 1.1 (just marginally larger than the default ggplot2 lines).
`shade`	Logical. Should the area under the curve be shaded? Defaults to `TRUE`.
`shade_alpha`	Transparency of the shading. Defaults to 0.2.

`refline` Logical. Should a diagonal reference line be plotted, representing the value at which no difference is observed between the reference and focal distributions? Defaults to `TRUE`.

`refline_col` Color of the reference line. Defaults to a dark gray.

`refline_type` The linetype for the reference line. Defaults to "dashed".

`refline_width` The width of the reference line. Defaults to 1, or just slightly thinner than the PP lines.

Value

A ggplot2 object displaying the specified PP plot.

Examples

```
# PP plot examining differences by condition
pp_plot(star, math ~ condition)

# The sample size gets very small in the above within cells (e.g., wild
# changes within the "other" group in particular). Overall, the effect doesn't
# seem to change much by condition.

# Look at something a little more interesting
## Not run:
pp_plot(benchmarks, math ~ ell + season + frl)

## End(Not run)
# Add some cut scores
pp_plot(benchmarks, math ~ ell, cuts = c(190, 210, 215))

## Make another interesting plot. Use ggplot to customize
## Not run:
library(tidyr)
library(ggplot2)
benchmarks %>%
  gather(subject, score, reading, math) %>%
  pp_plot(score ~ ell + subject + season,
          ref_group = "Non-ELL") +
  scale_fill_brewer(name = "ELL Status", palette = "Pastel2") +
  scale_color_brewer(name = "ELL Status", palette = "Pastel2") +
  labs(title = "Differences among English Language Learning Groups",
       subtitle = "Note crossing of reference line") +
  theme_minimal()

## End(Not run)
```

seda — *Portion of the Stanford Educational Data Archive (SEDA).*

Description

The full SEDA dataset contains mean test scores on statewide testing data in reading and math for every school district in the United States. See a description of the data here. The data represented in this package represent a random sample of 10 cases in the full dataset. To access the full data, please visit the data archive in the above link.

Usage

```
seda
```

Format

A data frame with 32625 rows and 8 columns.

leaid Integer. Local education authority identifier.

leaname Character. Local education authority name.

stateabb Character. State abbreviation.

year Integer. Year the data were collected.

grade Integer. Grade level the data were collected.

subject Character. Whether the data were from reading or mathematics.

mean Double. Mean test score for the LEA in the corresponding subject/grade/year.

se Double. Standard error of the mean.

Source

Sean F. Reardon, Demetra Kalogrides, Andrew Ho, Ben Shear, Kenneth Shores, Erin Fahle. (2016). Stanford Education Data Archive. http://purl.stanford.edu/db586ns4974. For more information, please visit https://edopportunity.org.

star — *Data from the Tennessee class size experiment*

Description

These data come from the Ecdat package and represent a cross-section of data from Project STAR (Student/Teacher Achievement Ratio), where students were randomly assigned to classrooms.

Usage

```
star
```

Format

A data frame with 5748 rows and 9 columns.

sid Integer. Student identifier.

schid Integer. School identifier.

condition Character. Classroom type the student was enrolled in (randomly assigned to).

tch_experience Integer. Number of years of teaching experience for the teacher in the classroom in which the student was enrolled.

sex Character. Sex of student: "girl" or "boy".

freelunch Character. Eligibility of the student for free or reduced price lunch: "no" or "yes"

race Character. The identified race of the student: "white", "black", or "other"

math Integer. Math scale score.

reading Integer. Reading scale score.

`tpac` *Transformed proportion above the cut*

Description

This function transforms calls to pac into standard deviation units. Function assumes that each distribution is distributed normally with common variances. See Ho & Reardon, 2012

Usage

```
tpac(data, formula, cuts, ref_group = NULL)
```

Arguments

`data`	The data frame used for estimation - ideally structured in a tidy format.
`formula`	A formula of the type `out ~ group` where `out` is the outcome variable and `group` is the grouping variable. Note this variable can include any arbitrary number of groups. Additional variables can be included with `+` to produce separate estimates by the secondary or tertiary variables of interest (e.g., `out ~ group + characteristic1 + characteristic2`).
`cuts`	Optional vector of cut scores. If supplied, the ECDF will be guaranteed to include these points. Otherwise, there could be gaps in the ECDF at those particular points (used in plotting the cut scores).
`ref_group`	Optional. A character vector or forumla listing the reference group levels for each variable on the right hand side of the formula, supplied in the same order as the formula. Note that if using the formula version, levels that are numbers, or include hyphens, spaces, etc., should be wrapped in back ticks (e.g., ``ref_group = ~ Active + `Non-FRL` ``, or ``ref_group = ~`8` ``). When in doubt, it is safest to use the back ticks, as they will not interfere with anything if they are not needed. See examples below for more details.

Value

Tidy data frame of the proportion above the cutoff for each (or selected) groups.

See Also

[esvis::pac(), esvis::pac_diff(), esvis::tpac_compare()]

Examples

```
# Compute differences for all pairwise comparisons for each of three cuts
tpac(star,
    reading ~ condition,
 cut = c(450, 500, 550))

tpac(star,
    reading ~ condition + freelunch + race,
 cut = c(450, 500))

tpac(star,
    reading ~ condition + freelunch + race,
 cut = c(450, 500),
 ref_group = ~small + no + white)
```

---

tpac_compare — *Compare Transformed Proportion Above the Cut*

Description

This function compares all possible pairwise comparisons, as supplied by `formula`, in terms of the transformed proportion above the cut. This is an effect-size like measure of the differences between two groups as the cut point(s) in the distribution. See Ho & Reardon, 2012

Usage

```
tpac_compare(data, formula, cuts, ref_group = NULL)
```

Arguments

`data`	The data frame used for estimation - ideally structured in a tidy format.
`formula`	A formula of the type `out ~ group` where `out` is the outcome variable and `group` is the grouping variable. Note this variable can include any arbitrary number of groups. Additional variables can be included with `+` to produce separate estimates by the secondary or tertiary variables of interest (e.g., `out ~ group + characteristic1 + characteristic2`).
`cuts`	Optional vector of cut scores. If supplied, the ECDF will be guaranteed to include these points. Otherwise, there could be gaps in the ECDF at those particular points (used in plotting the cut scores).

`ref_group` Optional. A character vector or forumla listing the reference group levels for each variable on the right hand side of the formula, supplied in the same order as the formula. Note that if using the formula version, levels that are numbers, or include hyphens, spaces, etc., should be wrapped in back ticks (e.g., `ref_group = ~ Active + `Non-FRL``, or `ref_group = ~`8``). When in doubt, it is safest to use the back ticks, as they will not interfere with anything if they are not needed. See examples below for more details.

Value

Tidy data frame of the proportion above the cutoff for each (or selected) groups.

See Also

[esvis::pac(), esvis::pac_diff(), esvis::tpac()]

Examples

```
# Compute differences for all pairwise comparisons for each of three cuts
tpac_compare(star,
    reading ~ condition,
 cut = c(450, 500, 550))

tpac_compare(star,
    reading ~ condition + freelunch + race,
 cut = c(450, 500))

tpac_compare(star,
    reading ~ condition + freelunch + race,
 cut = c(450, 500),
 ref_group = ~small + no + white)
```

`v` *Calculate the V effect size statistic*

Description

This function calculates the effect size V, as discussed by Ho, 2009. The V statistic is a transformation of `auc`, interpreted as the average difference between the distributions in standard deviation units.

Usage

```
v(data, formula, ref_group = NULL)
```

Arguments

data — The data frame used for estimation - ideally structured in a tidy format.

formula — A formula of the type `out ~ group` where `out` is the outcome variable and `group` is the grouping variable. Note this variable can include any arbitrary number of groups. Additional variables can be included with `+` to produce separate estimates by the secondary or tertiary variables of interest (e.g., `out ~ group + characteristic1 + characteristic2`).

ref_group — Optional. A character vector or forumla listing the reference group levels for each variable on the right hand side of the formula, supplied in the same order as the formula. Note that if using the formula version, levels that are numbers, or include hyphens, spaces, etc., should be wrapped in back ticks (e.g., ``ref_group = ~ Active + `Non-FRL` ``, or ``ref_group = ~`8` ``). When in doubt, it is safest to use the back ticks, as they will not interfere with anything if they are not needed. See examples below for more details.

Value

By default the V statistic for all possible pairings of the grouping factor are returned as a tidy data frame. Alternatively, a vector can be returned, and/or only the V corresponding to a specific reference group can be returned.

Examples

```
# Calculate V for all pairwise comparisons
v(star, reading ~ condition)

# Report only relative to regular-sized classrooms
v(star,
    reading ~ condition,
    ref_group = "reg")

# Report by ELL and FRL groups for each season, compare to non-ELL students
# who were not eligible for free or reduced price lunch in the fall (using
# the formula interface for reference group referencing).

## Not run:
v(benchmarks,
      math ~ ell + frl + season,
      ref_group = ~`Non-ELL` + `Non-FRL` + Fall)

# Same thing but with character vector supplied, rather than a formula
v(benchmarks,
      math ~ ell + frl + season,
      ref_group = c("Non-ELL", "Non-FRL", "Fall"))

## End(Not run)
```

Index