Package 'visual.kaito'

September 12, 2026

Title Interactive 3D Visualizations for Group Comparisons

Version 0.1.0

Description Draws interactive, rotatable statistical visualizations in three dimensions, built on 'plotly'. Two families of plots are provided. Triaxial box plots (boxplot3d(), boxplot3d_interactive()) compare groups on three continuous variables at once, with Tukey, fixed-percentile, mean +/- SD, and letter-value box/whisker conventions, plus parametric and non-parametric significance testing (per-axis and joint 3D via MANOVA / PERMANOVA). Bivariate density plots (ttest_plot3d(), manova_plot3d()) compare two or more groups on two continuous variables as overlapping 3D density surfaces, reporting per-axis t-tests together with a joint Hotelling's T-squared test (two groups), or a one-way MANOVA omnibus test with Bonferroni, Tukey, Fisher's LSD, and Dunnett post-hoc comparisons (more than two groups). All plots include live, pre-computed controls (view, method, scale, transparency) so results can be explored interactively without re-running R code.

License MIT + file LICENSE

Encoding UTF-8

Config/roxygen2/version 8.1.0

URL https://github.com/gygpsicologos-eng/visual.kaito

BugReports https://github.com/gygpsicologos-eng/visual.kaito/issues

Suggests htmlwidgets, knitr, mirt, multcomp, plotly, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Armando González Sánchez [aut, cre] (ORCID: <https://orcid.org/0000-0002-6182-1000>)

Maintainer Armando González Sánchez <gygpsicologos@gmail.com>

Repository CRAN

Date/Publication 2026-09-12 13:00:02 UTC

Contents

`boxplot3d` *Draw a 3-axis (triaxial) box plot*

Description

Computes box-and-whisker statistics on three continuous axes at once, for each group in `group`, and draws them as three linked pairwise panels (axis 1 vs 2, 2 vs 3, 1 vs 3) using base graphics.

Usage

```
boxplot3d(
  data,
  x = "x",
  y = "y",
  z = "z",
  group = "group",
  tukey = 1,
  percentile = 0,
  sd = 0,
  letter_value = 0,
  k = 1.5,
  probs = c(0.05, 0.95),
  sd_mult = c(1, 2),
  col = NULL
)
```

Arguments

`data`	A data frame with the three axes and the grouping variable.
`x`, `y`, `z`	Character. Column names of the three continuous axes.
`group`	Character. Column name of the grouping variable.
`tukey`, `percentile`, `sd`, `letter_value`	0/1 flags selecting the box/whisker convention. Exactly one must be 1.

k	Numeric. Tukey fence multiplier (default 1.5).
probs	Length-2 numeric. Lower/upper whisker percentiles for the percentile convention (default c(0.05, 0.95)).
sd_mult	Length-2 numeric. Box and whisker multipliers (in SDs) for the sd convention (default c(1, 2)).
col	Optional character vector of colors, one per group.

Details

Exactly one of tukey, percentile, sd, letter_value must be 1 (the rest 0): this selects how the box and whiskers are defined.

- **tukey** (default): box = P25-P75; whiskers reach the most extreme data point within k * IQR of the box; points beyond that fence are drawn individually as outliers.
- **percentile**: box = P25-P75; whiskers reach fixed percentiles (probs), regardless of outliers.
- **sd**: centered on the mean; box = mean +/- sd_mult[1] SD; whiskers = mean +/- sd_mult[2] SD.
- **letter_value**: nested boxes at successively finer percentile depths (Tukey's letter values), for large samples.

This base-graphics version draws three 2D projections rather than a single rotatable 3D scene. For a true interactive rotatable 3D view (same four conventions, a same-scale/z-score option, and axis-pair view buttons), see boxplot3d_interactive().

Value

Invisibly, a list with one element per group, each a list of per-axis statistics (q1, q3, center, whisker_lo, whisker_hi, plus outliers, a logical vector flagging points outside the whiskers on that axis).

Examples

```
set.seed(1)
df <- rbind(
  data.frame(x = rnorm(30, 0), y = rnorm(30, 0), z = rnorm(30, 0), group = "A"),
  data.frame(x = rnorm(30, 1), y = rnorm(30, 1), z = rnorm(30, 1), group = "B")
)
boxplot3d(df)
```

`boxplot3d_interactive` *Draw an interactive, rotatable 3-axis (triaxial) box plot*

Description

Draws a single interactive scene (via **plotly**) that can show either a rotatable 3D view with one translucent cuboid per group spanning the box on all three axes at once, or a true flat 2D panel for any one pair of axes (x-axis at the bottom, y-axis on the left – an ordinary 2D plot, not a 3D scene viewed from a fixed camera angle). Each 3D box is drawn with crisp wireframe edges (flat-shaded faces plus explicit edge lines) so its shape reads clearly instead of as a smooth blob. Each box also shows whiskers along its axes, a cross-shaped median marker (a black halo plus a smaller marker in the group's own color, so medians stay distinguishable by group even when they land close together), faint raw points, and outliers (when the method produces any) as open diamonds. In 3D mode the scene rotates with click-drag, pans with right-click or shift-drag, and zooms with the mouse wheel.

Usage

```
boxplot3d_interactive(
  data,
  x = "x",
  y = "y",
  z = "z",
  group = "group",
  tukey = 1,
  percentile = 0,
  sd = 0,
  letter_value = 0,
  k = 1.5,
  probs = c(0.05, 0.95),
  sd_mult = c(1, 2),
  scale = 0,
  grid3d = 1,
  color_by = NULL,
  col = NULL
)
```

Arguments

`data`	A data frame with the three axes and the grouping variable.
`x`, `y`, `z`	Character. Column names of the three continuous axes. These names are used verbatim as the axis titles, so pass the real column names directly (e.g. `x = "Score_ISI"`) rather than renaming your data frame's columns to generic `"x"`/`"y"`/`"z"` first.
`group`	Character. Column name of the grouping variable that defines the boxes being compared.

`tukey`, `percentile`, `sd`, `letter_value`
: 0/1 flags selecting which convention is shown initially (switchable live via "Metodo"). Exactly one must be 1.

`k`
: Numeric. Tukey fence multiplier (default 1.5).

`probs`
: Length-2 numeric. Lower/upper whisker percentiles for the `percentile` convention (default `c(0.05, 0.95)`).

`sd_mult`
: Length-2 numeric. Box and whisker multipliers (in SDs) for the `sd` convention (default `c(1, 2)`).

`scale`
: 0/1. Which scale is shown initially: 0 = raw values, 1 = standardized (z-score). Switchable live via "Escala".

`grid3d`
: 0/1. Initial state of the background grid (default 1 = on); also togglable live from the plot itself.

`color_by`
: Optional character vector of column names to offer in the "Agrupar por" dropdown (for recoloring individual points). If `NULL` (default), `group` is offered together with any other factor/character/logical column found in `data`.

`col`
: Optional character vector of colors, one per level of `group`.

Details

This function precomputes all four whisker conventions (see `boxplot3d()` for what tukey/percentile/sd/letter_value mean), all four views (3D plus the three axis-pair 2D panels), and both the raw and standardized (z-score) versions of the data, and lets you switch between all of them live with the controls above the plot, without re-calling the function:

- **Vista**: 3D, or a flat 2D panel for axes 1-2, 2-3, or 1-3.
- **Metodo**: which box/whisker convention to draw.
- **Agrupar por**: which column colors the individual raw data points (subjects). This only recolors the points – the boxes, whiskers and medians always reflect the `group` argument, since that is what defines the statistical groups being compared. By default this offers `group` itself plus any other factor/character column found in `data`.
- **Escala**: raw values, or all three axes standardized (z-score) to a common scale.
- **Cuadricula**: on/off for the background grid.

Value

A `plotly` htmlwidget object.

Examples

```
set.seed(1)
df <- rbind(
  data.frame(x = rnorm(30, 0), y = rnorm(30, 0), z = rnorm(30, 0), group = "A"),
  data.frame(x = rnorm(30, 1), y = rnorm(30, 1), z = rnorm(30, 1), group = "B")
)

if (requireNamespace("plotly", quietly = TRUE)) {
  boxplot3d_interactive(df)
```

```
}
```

`boxplot3d_significance` *Significance testing for a 3-axis (triaxial) box plot*

Description

Compares groups on three continuous axes at once, both per-axis and jointly (3D), using a parametric test and a non-parametric test in parallel (not mutually exclusive – set both `parametric` and `nonparametric` to 1 to get both, as is the default).

Usage

```
boxplot3d_significance(
  data,
  x = "x",
  y = "y",
  z = "z",
  group = "group",
  parametric = 1,
  nonparametric = 1,
  scale = 0,
  nperm = 999,
  seed = 1
)
```

Arguments

`data`	A data frame containing the three axes and the grouping variable.
`x`, `y`, `z`	Character. Column names of the three continuous axes.
`group`	Character. Column name of the grouping variable.
`parametric`, `nonparametric`	0/1. Which family (or both) of tests to run. At least one must be 1.
`scale`	0/1. If 1, standardize (z-score) `x`, `y`, `z` before testing, so axes on different units/scales contribute comparably to the joint (3D) test. Per-axis and MANOVA results are invariant to this; PERMANOVA is not, since it is based on Euclidean distance.
`nperm`	Integer. Number of permutations for the PERMANOVA test.
`seed`	Integer. Random seed for the permutations (reproducibility).

Details

Parametric: Welch t-test (2 groups) / Welch ANOVA (>2 groups) per axis; MANOVA (Pillai's trace) for the joint 3D comparison.

Non-parametric: Wilcoxon rank-sum test (2 groups) / Kruskal-Wallis (>2 groups) per axis; a permutation-based PERMANOVA (one-way, Euclidean sum-of-squares decomposition, computed without depending on 'vegan') for the joint 3D comparison.

Rows with a missing value (`NA`) on `x`, `y`, or `z` are excluded listwise before any test is run, so the per-axis tests and both joint (3D) tests are computed on the same, identical sample.

Value

A data frame with one row per axis plus one row for the joint 3D comparison, with the test names, p-values, and significance stars.

Examples

```
set.seed(1)
df <- rbind(
  data.frame(x = rnorm(30, 0), y = rnorm(30, 0), z = rnorm(30, 0), group = "A"),
  data.frame(x = rnorm(30, 1), y = rnorm(30, 1), z = rnorm(30, 1), group = "B")
)
boxplot3d_significance(df, nperm = 199)
```

---

chisq_plot3d *Draw a chi-square association plot (standardized residuals per cell)*

Description

Builds a contingency table from `row` and `col` – each of which can be either a single categorical column name or several column names combined into one bigger grouping (e.g. `col = c("sexo", "compra")` produces a "sexo x compra" nested column: Varon-Si, Varon-No, Mujer-Si, Mujer-No) – and draws ONE 3D bar per cell whose height and color both encode the cell's standardized (Pearson) residual: how many standard deviations the observed count is above (red, excess) or below (blue, deficit) what independence would predict. Two translucent reference planes at +/-1.96 and +/-2.58 mark the conventional p < .05 / p < .01 thresholds, so cells with a real deviation are immediately visible without reading a separate table.

Usage

```
chisq_plot3d(
  data,
  row,
  col,
  opacity = 0.85,
  sep = " - ",
  view = c("3d", "2d"),
```

```
  lang = c("es", "en")
)
```

Arguments

`data`	A data frame.
`row`, `col`	Character vector(s) of column name(s) in `data`. Length 1 for a simple two-way table; length > 1 to combine several categorical columns into one side of the table (e.g. `col = c("sexo", "compra")`). At most one of `row`/`col` may have length > 1 (needed for the Simpson's-paradox diagnostics; see Details).
`opacity`	Numeric in (0, 1]. Initial bar opacity (also a live slider). Default 0.85.
`sep`	Character used to join combined-column level labels. Default `" - "`.
`view`	Initial camera: `"3d"` (default) or `"2d"` (flattened top-down orthographic view; also switchable live via the "3D"/"2D" buttons).
`lang`	Language for all plot text (title, axis/legend labels, hover text, on-plot annotation) and for this function's own messages/errors: `"es"` (default) or `"en"`.

Details

Two extra diagnostics are computed automatically and attached to the returned object (and summarized in the on-plot annotation):

- **Low-frequency warning** (Cochran's rule): flags the table when any cell's expected count is below 1, or when more than 20% of cells have an expected count below 5 – in either case the usual chi-square approximation is unreliable and Fisher's exact test or collapsing categories is recommended.
- **Simpson's paradox warning**, only when `row` or `col` was given as more than one column (i.e. one side is "subdivided"): a *descriptive* flag comparing the sign of each cell's residual in the collapsed (marginal) table against its sign within each level of the subdividing variable, plus a *formal* test (Gail & Simon, 1985, qualitative interaction test) applied per cell of the collapsed table, treating each stratum's standardized residual as an approximately N(0,1) effect estimate and testing whether its sign is consistent across strata.

A "3D / 2D" button pair switches the camera between the default perspective view and a flattened, straight-down orthographic view (looking along the z axis), so the bars read as a plain 2D heatmap grid – no data changes, only the camera.

Value

A `plotly` htmlwidget object. The full numeric results (the contingency table, the `chisq.test()` object, expected counts, standardized residuals, the low-frequency note, and – when applicable – the Simpson's-paradox diagnostics) are attached as the `"stats"` attribute.

Examples

```
set.seed(1)
df <- data.frame(
  tipo = sample(c("A", "B", "C"), 300, replace = TRUE, prob = c(0.4, 0.35, 0.25)),
  sexo = sample(c("Varon", "Mujer"), 300, replace = TRUE),
```

```
  compra = sample(c("Si", "No"), 300, replace = TRUE)
)
if (requireNamespace("plotly", quietly = TRUE)) {
  chisq_plot3d(df, row = "tipo", col = c("sexo", "compra"))
}
```

`irt_plot3d` *Draw a 3D Item Response Theory (IRT) map: characteristic curves + double histogram*

Description

Fits an Item Response Theory model with `mirt::mirt()` – dichotomous items (0/1) as 1PL/2PL/3PL, polytomous items (e.g. Likert 1-5 or 1-10) as the Graded Response Model (GRM) – and draws every item's characteristic curve(s) as a 3D "cordillera": x is the shared trait scale (theta), y separates items (one depth slot per item, ordered by difficulty), z is the probability of endorsing that response level or higher. A polytomous item with k response categories contributes k-1 boundary curves (one per threshold between consecutive categories). Every curve has its own legend entry (click to toggle, double-click to isolate) and its own checkbox in an items-by-response-options matrix placed above the plot, so a curve can also be toggled by row/column instead of hunting through the legend – there is no separate "azar" (guessing) parameter for polytomous thresholds, since it does not exist in the graded model.

Usage

```
irt_plot3d(
  data,
  items = names(data),
  guessing = FALSE,
  theta_method = c("EAP", "MAP", "WLE", "ML"),
  opacity = 0.85,
  bin_width = 0.5,
  hist_shift = 1,
  view = c("3d", "2d"),
  lang = c("es", "en"),
  col_subj = "#eda100",
  col_items = NULL
)
```

Arguments

`data`	A data frame with one row per subject and one column per item. Each item column must be numeric/integer-coded: 2 distinct values for a dichotomous item (fit as 1PL/2PL/3PL), 3 or more for a polytomous item (fit as a Graded Response Model). Items can be mixed (some dichotomous, some polytomous) in the same call.

items	Character vector of column names in data to include. Default: all columns.
guessing	Logical. If TRUE, dichotomous items are fit as 3PL (with a guessing/"azar" parameter); if FALSE (default), as 2PL. Polytomous items never get a guessing parameter – it does not exist in the Graded Response Model – so this is silently inapplicable to them (a message notes this when relevant).
theta_method	Ability-estimation method passed to mirt::fscores(): one of "EAP" (default), "MAP", "WLE", "ML".
opacity	Numeric in (0, 1]. Initial opacity of curves and histogram bars (also a live slider). Default 0.85.
bin_width	Width of the shared theta bins used for both histograms. Default 0.5.
hist_shift	Numeric in [0, 1]. Initial position of the double-histogram panel along the item-depth axis (also a live slider): 1 (default) is fully inserted at the front; 0 rests it against the back wall, out of the way of the curves.
view	Initial camera: "3d" (default) or "2d" (flattened orthographic view; also switchable live via the "3D"/"2D" buttons).
lang	Language for all plot text (title, axis/legend labels, hover text, on-plot annotation) and for this function's own messages/errors: "es" (default) or "en".
col_subj	Color for the subject (ability) histogram. Default "#eda100".
col_items	Optional named (or positional, matching items) character vector of colors, one per item. Default: a fixed 6-color palette recycled/extended with grDevices::hcl.colors() for more items.

Details

A flat panel is inserted into the scene, sharing the same x (theta) axis as the curves and the same "0" point as z = 0 (probability 0): a histogram of the sample's estimated ability (theta, from mirt::fscores()) grows upward from that shared line, and directly below it a histogram of item/threshold difficulty grows downward from the same line, each item/threshold "hanging" from theta = 0 and stacking with others that fall in the same theta bin. A thin line connects each curve's own P = 0.5 crossing point (theta_50) to where it lands in that difficulty histogram – for a 3PL item with guessing, theta_50 is NOT the raw difficulty parameter b (see Details). A second slider slides this whole panel along the item-depth axis, from tucked against the back wall (its "resting" position, out of the way) to fully inserted at the front of the scene (the default).

For a dichotomous 3-parameter item, P(theta) = c + (1-c) / (1 + exp(-a*(theta-b))), and solving P(theta) = 0.5 gives theta_50 = b - (1/a) * log(0.5 / (0.5 - c)), which equals b only when c = 0 (2PL/Rasch); the vertical drop line and the difficulty histogram both use this real crossing point, not b directly. For a polytomous (graded) threshold, the boundary curve P(X >= k | theta) = 1 / (1 + exp(-a*(theta-b_k))) has no guessing term, so its crossing point is b_k itself.

A "3D / 2D" button pair switches the camera between the default perspective view and a flattened, straight-on orthographic view (looking along the item-depth axis), so every curve reads on the same (theta, P) plane, like a classic 2D item-characteristic-curve plot – no data changes, only the camera.

Value

A plotly htmlwidget object. The fitted mirt model object, the extracted IRT-parameterized coefficients, the per-subject theta estimates, and a tidy data frame with one row per drawn curve (item, threshold label, a, b, c, theta_50) are attached as the "stats" attribute.

Examples

```
set.seed(1)
n <- 200
theta <- rnorm(n)
logistic <- function(x) 1 / (1 + exp(-x))
df <- data.frame(
  i1 = rbinom(n, 1, logistic(1.2 * (theta - (-0.5)))),
  i2 = rbinom(n, 1, logistic(1.4 * (theta - 0.3))),
  i3 = rbinom(n, 1, logistic(0.9 * (theta - 0.8)))
)
if (requireNamespace("mirt", quietly = TRUE) &&
    requireNamespace("plotly", quietly = TRUE)) {
  irt_plot3d(df, items = c("i1", "i2", "i3"))
}
```

`manova_plot3d` *Draw more-than-2 groups as bivariate 3D "mountains" with pairwise post-hoc tests*

Description

The many-groups counterpart to `ttest_plot3d()`. Each group (there must be more than 2) is drawn as its own bivariate density surface ("mountain") over the (`x`, `z`) plane, all in the same 3D scene. An overall one-way MANOVA (Wilks' lambda) tests whether the groups differ jointly on both variables; a "Comparar" menu picks which pair of groups to highlight (their overlap is shaded, same convention as `ttest_plot3d()`), and a "Metodo" menu reports that pair's post-hoc result under Bonferroni, Tukey (HSD), DMS (Fisher's LSD) or Dunnett.

Usage

```
manova_plot3d(
  data,
  x = "x",
  z = "z",
  group = "group",
  control = NULL,
  conf.level = 0.95,
  n_grid = 50,
  opacity = 0.5,
  col = NULL,
  col_overlap = "#4a3aa7"
)
```

Arguments

`data`	A data frame with the two continuous axes and the grouping variable.
`x`, `z`	Character. Column names of the two continuous axes.

group	Character. Column name of the grouping variable. Must have more than 2 levels (with exactly 2, use ttest_plot3d() instead).
control	Character. The reference group for Dunnett's test. If NULL (default), the first level of group is used and a message reports which one that is.
conf.level	Confidence level (default 0.95).
n_grid	Integer. Grid resolution per axis for the density surfaces (default 50).
opacity	Initial opacity (0-1) of the group surfaces (default 0.5); adjustable afterwards with the transparency slider.
col	Character vector of colors, one per group. Defaults to the Okabe-Ito palette.
col_overlap	Color used to mark the overlap of the currently compared pair.

Details

Tukey and Dunnett do not have a standard closed-form multivariate version, so the 4 post-hoc methods are computed axis-by-axis with the real underlying R routines (stats::TukeyHSD(), stats::pairwise.t.test() with Bonferroni or no correction, and Dunnett via the multcomp package) – the same per-axis convention ttest_plot3d() already uses for its two t-tests. Each pair's unadjusted, 2-variable Hotelling's T-squared (as in ttest_plot3d()) is also reported for every method, as a joint summary of that one pair.

Dunnett compares every group only against a fixed reference/control group, not all pairs against each other – selecting a pair that does not include control while "Metodo" is set to Dunnett shows a note instead of a p-value for that pair.

Value

A plotly htmlwidget object (auto-displayed if the result isn't assigned). Full statistics – manova (Wilks' lambda test), means, covariances, and posthoc (a data frame with one row per pair per method: per-axis p-values, the pair's Hotelling T2 p-value, and whether that method applies to that pair) – are attached as the "stats" attribute.

Examples

```
set.seed(1)
df <- rbind(
  data.frame(x = rnorm(25, 0, 1.2), z = rnorm(25, 0, 1.1), group = "Control"),
  data.frame(x = rnorm(25, 2, 1.3), z = rnorm(25, 1.4, 1.2), group = "A"),
  data.frame(x = rnorm(25, 2.5, 1.1), z = rnorm(25, 3.2, 1.3), group = "B")
)
manova_plot3d(df, control = "Control")
```

paired_plot3d *Draw a paired (pre/post) comparison on two variables as a 3D "spaghetti" plot*

Description

A repeated-measures counterpart to `ttest_plot3d()`: instead of two independent groups, the same subjects are measured twice (e.g. before and after an intervention) on two continuous variables. Each subject is drawn as a point at their "Pre" measurement and a point at their "Post" measurement, joined by a line, in a 3D scene where `x` and `z` are the two variables and the third axis ("Momento") separates Pre from Post. Both paired Student's t-tests (one per variable) and a joint one-sample (paired) Hotelling's T-squared test on the vector of differences are reported.

Usage

```
paired_plot3d(
  data,
  x_pre,
  x_post,
  z_pre,
  z_post,
  x_name = NULL,
  z_name = NULL,
  conf.level = 0.95,
  opacity = 0.75,
  col_pre = "#2a78d6",
  col_post = "#eb6834",
  col_dir = c("#1baf7a", "#d03b3b", "#9a9890")
)
```

Arguments

`data`	A data frame in wide format: one row per subject, with separate columns for the Pre and Post measurement of each variable.
`x_pre`, `x_post`	Character. Column names of the first variable's Pre and Post measurements.
`z_pre`, `z_post`	Character. Column names of the second variable's Pre and Post measurements.
`x_name`, `z_name`	Character. Axis titles for the two variables (default: derived from `x_pre`/`z_pre` by stripping a trailing `_pre`/`_Pre` suffix if present).
`conf.level`	Confidence level for the paired t-tests (default 0.95).
`opacity`	Initial opacity (0-1) of the subject lines (default 0.75); adjustable afterwards with the transparency slider.
`col_pre`, `col_post`	Colors for the Pre and Post markers.
`col_dir`	Length-3 character vector of colors for the "Direccion de cambio" coloring scheme: both axes increase, both decrease, mixed change (in that order).

Value

A `plotly` htmlwidget object (auto-displayed if the result isn't assigned). The full statistics – `n`, per-variable Pre/Post means and SDs, `t_x` and `t_z` (paired `stats::t.test()` results), and `hotelling_paired` (a list with `T2`, `F`, `df1`, `df2`, `p.value`) – are attached as the `"stats"` attribute.

Examples

```
set.seed(1)
n <- 20
df <- data.frame(
  x_pre = rnorm(n, 60, 9), z_pre = rnorm(n, 18, 4)
)
df$x_post <- df$x_pre - rnorm(n, 9, 6)
df$z_post <- df$z_pre - rnorm(n, 6, 4)
paired_plot3d(df, x_pre = "x_pre", x_post = "x_post",
              z_pre = "z_pre", z_post = "z_post")
```

`ttest_plot` *Draw a t-distribution curve with the observed statistic and critical region*

Description

A didactic plot: the t-distribution density curve for the relevant degrees of freedom, the critical (rejection) region shaded according to `alternative` and `conf.level`, and the observed t-statistic marked with a vertical line – so it's visually clear whether the observed value falls inside the rejection region.

Usage

```
ttest_plot(
  x = NULL,
  y = NULL,
  mu = 0,
  paired = FALSE,
  alternative = c("two.sided", "less", "greater"),
  var.equal = FALSE,
  conf.level = 0.95,
  t = NULL,
  df = NULL,
  col_curve = "#2a78d6",
  col_critical = "#d03b3b",
  col_observed = "#0b0b0b",
  main = NULL,
  lang = c("es", "en")
)
```

Arguments

x	Numeric vector (first sample), or NULL if using the manual t/df mode.
y	Optional numeric vector (second sample), for a two-sample or paired test.
mu	Number. Null-hypothesis mean for a one-sample test (default 0).
paired	Logical. Paired t-test (default FALSE).
alternative	One of "two.sided" (default), "less", "greater".
var.equal	Logical. Assume equal variances in a two-sample test (default FALSE, i.e. Welch).
conf.level	Confidence level defining the critical region (default 0.95, i.e. alpha = 0.05).
t, df	Numeric. Manual mode: the observed t-statistic and degrees of freedom, used instead of computing a test from x/y. Both must be supplied together.
col_curve, col_critical, col_observed	Colors for the density curve, the shaded critical region, and the observed-statistic marker.
main	Optional plot title.
lang	One of "es" (default) or "en", translating the axis labels, legend text, and error message.

Details

Two ways to use it:

- **From data**: pass x (and optionally y) as you would to stats::t.test() – one-sample, two-sample, or paired – and the test is run internally.
- **Manual**: pass t and df directly (skipping x/y entirely), useful for teaching a specific scenario without needing raw data.

Value

Invisibly, a list with t, df, p.value, alternative, alpha, critical (the critical value(s)), and stars.

Examples

```
set.seed(1)
a <- rnorm(20, 5, 1)
b <- rnorm(20, 6, 1)
ttest_plot(a, b)

# manual mode: no data needed
ttest_plot(t = 2.4, df = 28, alternative = "greater")
```

`ttest_plot3d` *Draw two groups as bivariate 3D "mountains" with their overlap and joint test*

Description

A 3D counterpart to `ttest_plot()` for comparing 2 groups on 2 continuous variables at once. Each group is drawn as a single bivariate density surface ("mountain") over the (`x`, `z`) plane – fitted from that group's own sample mean and covariance, `y` is density – rather than as two separate 1D curves. The region where the two surfaces overlap is shaded, and both the per-axis Student's t-tests (one for `x`, one for `z`) and the joint two-sample Hotelling's T-squared test are reported, so you can read the result either axis-by-axis or as a single combined test.

Usage

```
ttest_plot3d(
  data,
  x = "x",
  z = "z",
  group = "group",
  var.equal = FALSE,
  conf.level = 0.95,
  n_grid = 60,
  opacity = 0.55,
  col = c("#2a78d6", "#eb6834"),
  col_overlap = "#4a3aa7"
)
```

Arguments

`data`	A data frame with the two continuous axes and the grouping variable.
`x`, `z`	Character. Column names of the two continuous axes.
`group`	Character. Column name of the grouping variable. Must have exactly 2 levels.
`var.equal`	Logical, passed to the per-axis `stats::t.test()` calls (default `FALSE`, i.e. Welch).
`conf.level`	Confidence level for the per-axis t-tests (default 0.95).
`n_grid`	Integer. Grid resolution per axis for the density surfaces and the numerical overlap estimate (default 60).
`opacity`	Initial opacity (0-1) of the two group surfaces (default 0.55); adjustable afterwards with the transparency slider.
`col`	Length-2 character vector of colors, one per group.
`col_overlap`	Color used to mark the region where the two surfaces overlap.

Details

Hotelling's T-squared is the multivariate generalization of Student's t: it tests whether the two groups' mean vectors (on x and z together) differ, accounting for the covariance between the two variables, and collapses to a single F-statistic and p-value. The per-axis t-tests can disagree with each other (significant on one axis, not the other) and can also disagree with the joint Hotelling result; that disagreement is informative, not a bug, so all three are reported rather than picking one.

The plot has 4 live controls, all pre-computed so they respond instantly without needing R again:

- A "Vista" menu switches between the full 3D scene and 3 flat 2D views (same convention as `boxplot3d_interactive()`'s axis-pair panels, built as a true flat second cartesian layer, not a camera trick): **Eje x** and **Eje z** each show that one variable's two group curves (density vs. value) with the shared region shaded, and **Ambos ejes** shows a top-down contour view of both mountains at once, with the same shaded overlap footprint.
- A transparency slider adjusts the opacity of the two group surfaces in the 3D view (the overlap surface always stays solid, so the shared region remains readable at any transparency level).
- A "Reportar" dropdown chooses which of the reported statistics show in the on-plot text box (all of them, only the per-axis t-tests, only Hotelling, or only the overlap percentage); each group's mean and SD on both variables are always shown underneath.
- A button pair switches the 3D surfaces between raw units and standardized (z-score) units. Standardizing only changes what the surfaces look like – the t-test, Hotelling and overlap numbers are mathematically identical either way, since all three are invariant to a shared linear rescaling of the axes – so the text box does not change when you switch scales. This toggle only affects the 3D view.

Value

A `plotly` htmlwidget object (auto-displayed if the result isn't assigned). The full statistics – `means` and `covariances` per group, `overlap` (the estimated overlap coefficient, 0-1), `t_x` and `t_z` (the `stats::t.test()` results for each axis), and `hotelling` (a list with `T2`, `F`, `df1`, `df2`, `p.value`) – are attached as the `"stats"` attribute, e.g. `attr(ttest_plot3d(df), "stats")$hotelling`.

Examples

```
set.seed(1)
df <- rbind(
  data.frame(x = rnorm(30, 0, 1.3), z = rnorm(30, 0, 1.1), group = "A"),
  data.frame(x = rnorm(30, 3, 1.4), z = rnorm(30, 2.6, 1.2), group = "B")
)
ttest_plot3d(df)
```

Index