Package: clinicalsignificance (via r-universe)

October 22, 2026

Type Package

Title A Toolbox for Clinical Significance Analyses in Intervention Studies

Version 3.0.0.9000

Description A clinical significance analysis can be used to determine if an intervention has a meaningful or practical effect for patients. You provide a tidy data set plus a few more metrics and this package will take care of it to make your results publication ready. Accompanying package to Claus et al. <doi:10.18637/jss.v111.i01>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Depends R (>= 4.2.0)

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports BayesFactor, bayestestR, checkmate, cli, dplyr, ggplot2, insight, lme4, purrr, rlang, snakecase, tibble, tidyr

Suggests knitr, rmarkdown, testthat (>= 3.0.0), tidyverse, vdiffr

Config/testthat/edition 3

VignetteBuilder knitr

URL https://benediktschfr.github.io/clinicalsignificance/, https://github.com/benediktschfr/clinicalsignificance

BugReports https://github.com/benediktschfr/clinicalsignificance/issues

Config/pak/sysreqs cmake make libicu-dev pari-gp

Repository https://benediktschfr.r-universe.dev

Date/Publication 2026-05-28 09:59:49 UTC

RemoteUrl https://github.com/benediktschfr/clinicalsignificance

RemoteRef HEAD

RemoteSha 7a468cf83f7f163402fff0d07c52f675f2452322

Contents

antidepressants . . . 3
anxiety . . . 4
anxiety_complete . . . 4
claus_2020 . . . 5
columbo . . . 6
cs_anchor . . . 7
cs_combined . . . 12
cs_distribution . . . 16
cs_get_augmented_data . . . 20
cs_get_cutoff . . . 23
cs_get_cutoff_descriptives . . . 24
cs_get_data . . . 24
cs_get_model . . . 25
cs_get_n . . . 26
cs_get_reliability . . . 28
cs_get_summary . . . 29
cs_percentage . . . 30
cs_statistical . . . 34
garden_club . . . 37
garden_club_sensitive . . . 38
hechler_2014 . . . 39
jacobson_1989 . . . 40
plot.cs_anchor_group_between . . . 40
plot.cs_anchor_group_within . . . 41
plot.cs_anchor_individual_within . . . 43
plot.cs_combined . . . 45
plot.cs_distribution . . . 48
plot.cs_percentage . . . 50
plot.cs_statistical . . . 53
print.cs_anchor_group_between . . . 55
print.cs_anchor_group_within . . . 56
print.cs_anchor_individual_within . . . 57
print.cs_combined . . . 57
print.cs_distribution . . . 58
print.cs_percentage . . . 59
print.cs_statistical . . . 59
summary.cs_anchor_group_between . . . 60
summary.cs_anchor_group_within . . . 61
summary.cs_anchor_individual_within . . . 62
summary.cs_combined . . . 63
summary.cs_distribution . . . 63
summary.cs_percentage . . . 64
summary.cs_statistical . . . 65
trackmania . . . 65

Index **67**

`antidepressants` *Antidepressant Data*

Description

A fictional dataset used to showcase group-based clinical significance analyses and analyses with many participants.

Usage

```
antidepressants
```

Format

A tibble with 1140 rows and 4 variables:

`patient` Patient identifier

`condition` Experimental condition

`measurement` Indicator of measurement

`mom_di` Mind over Mood Depression inventory scores (lower is better)

Details

In a fictional clinical trial, the effectiveness of a new antidepressant should be examined and depressed patients were randomized to one of four groups:

- A wait list control group that did not receive a medication
- An inactive placebo group, i.e., a group that received a placebo (inert substance without proposed clinical effect) pill
- An active placebo group, i.e., a group that received a placebo that evokes side effects like mild nausea or a dry mouth
- The antidepressant group, so the target medication of this trial that should have a clinical impact on the patients' depressive symptoms

Suppose they underwent outpatient treatment, depressive symptoms were measured before and after treatment with the Mind over Mood Depression Inventory (MoM-DI) by Greenberger & Padesky (2015), and if a patient received a pill, the clinician and the patient did not know, what type of medicaction they consumed.

Further, the minimal important difference for an improvement as measured by this instrument was agreed to be an 8 point decrease. A deterioration can be assumed if instrument scores increased by 5 points.

The functional population (i.e., non-depressed individuals) can be expected to have a mean score of M = 8 points with a standard deviation of SD = 7.

References

Greenberger, D., & Padesky, C. A. (2015). Mind over mood, second edition (2nd ed.). New York, NY: Guilford Publications.

anxiety *Anxiety Data*

Description

A fictional dataset with missings to exemplify the use of HLM method for clinical significance.

Usage

```
anxiety
```

Format

A data frame with 580 rows and 4 variables:

`subject` Participant

`treatment` Treatment. Either Placebo or Intervention

`measurement` Number of measurement

`anxiety` Anxiety score (lower is better)

Details

In a fictional clinical trial, participants were split up to belong to either a medical placebo ("Placebo") or psychotherapeutic intervention ("Intervention") group.

They underwent outpatient treatment during which they were followed for 5 measurements at which a fictional anxiety score was measured. This anxiety score may range from 0 - 60.

The functional population (i.e., non-anxious individuals) can be expected to have a mean score of M = 8 points with a standard deviation of SD = 4.

anxiety_complete *Anxiety Data (Complete)*

Description

A fictional complete dataset to exemplify the use of HLM method for clinical significance.

Usage

```
anxiety_complete
```

Format

A data frame with 580 rows and 4 variables:

`subject` Participant

`treatment` Treatment. Either Placebo or Intervention

`measurement` Number of measurement

`anxiety` Anxiety score (lower is better)

Details

In a fictional clinical trial, participants were split up to belong to either a medical placebo ("Placebo") or psychotherapeutic intervention ("Intervention") group.

They underwent outpatient treatment during which they were followed for 5 measurements at which a fictional anxiety score was measured. This anxiety score may range from 0 - 60.

The functional population (i.e., non-anxious individuals) can be expected to have a mean score of M = 8 points with a standard deviation of SD = 4.

`claus_2020` *Placebo Amplification Data*

Description

A dataset containing the data from Claus et al. (2020). In a routine inpatient setting for unipolar depressive disorders they implemented an intervention that sought to amplify the placebo response of antidepressants. In the study, two groups were compared: treatment as usual (TAU) and placebo amplification (PA). Participants were examined four times during their treatment.

Usage

```
claus_2020
```

Format

An object of class `tbl_df` with 172 rows and 9 columns.

id Participant ID

age Age

sex Sex

treatment Treatment (TAU for treatment as usual and PA for placebo amplification)

time Measurement

bdi Beck Depression Inventory (2nd Edition) score (lower is better)

shaps Snaith-Hamilton Pleasure Scale score (higher is better)

who WHO-Five Well-Being Index score (higher is better)

hamd Hamilton Rating Scale for Depression score (lower is better)

Source

https://osf.io/rc754/

References

- Claus, B. B., Scherbaum, N., & Bonnet, U. (2020). Effectiveness of an Adjunctive Psychotherapeutic Intervention Developed for Enhancing the Placebo Effect of Antidepressants Used within an Inpatient-Treatment Program of Major Depression: A Pragmatic Parallel-Group, Randomized Controlled Trial. Psychotherapy and Psychosomatics, 89(4), 258-260. https://doi.org/10.1159/000505855

---

columbo *Fictional "Columbo's Detectives" Data with Missing Values*

Description

A simulated dataset from a fictional study evaluating a special training program by Lieutenant Columbo for detectives. This dataset is specifically designed to demonstrate the handling of missing data. Missing values were introduced into the post-intervention outcome using a Missing At Random (MAR) mechanism.

Usage

```
columbo
```

Format

A tibble with 160 rows and 7 variables:

detective_id An integer representing the unique identifier for each detective.

group A factor indicating the training group (`"Columbo's Training"` or `"Control"`).

age An integer representing the detective's age.

gender A factor for the detective's gender (`"m"`, `"f"`, or `"d"`).

job_frustration A numeric score from 0-10 indicating job frustration.

time A factor for the measurement occasion (`"pre"` or `"post"`).

clearance A numeric value for the outcome, the Case Clearance Rate (in %). This variable contains `NA` values.

Source

Simulated data where missing values in the post-intervention outcome were introduced via a Missing At Random (MAR) mechanism. The probability of missingness depends on observed variables (higher `job_frustration` and lower post-intervention scores increase the likelihood of data being missing).

cs_anchor *Anchor-Based Analysis of Clinical Significance*

Description

cs_anchor() can be used to determine the clinical significance of intervention studies employing the anchor-based approach. For this, a predefined minimally important difference (MID) for an instrument is known that corresponds to an important symptom improvement for patients. The data can then be analyzed on the individual as well as the group level to estimate, if the change because of an intervention is clinically significant.

Usage

```
cs_anchor(
  data,
  id,
  time,
  outcome,
  group,
  pre = NULL,
  post = NULL,
  mid_improvement = NULL,
  mid_deterioration = NULL,
  better_is = c("lower", "higher"),
  target = c("individual", "group"),
  effect = c("within", "between"),
  bayesian = TRUE,
  prior_scale = "medium",
  reference_group = NULL,
  ci_level = 0.95
)
```

Arguments

data	A tidy data frame
id	Participant ID
time	Time variable
outcome	Outcome variable
group	Grouping variable (optional)
pre	Pre measurement (only needed if the time variable contains more than two measurements)
post	Post measurement (only needed if the time variable contains more than two measurements)
mid_improvement	Numeric, change that indicates a clinically significant improvement

`mid_deterioration`
: Numeric, change that indicates a clinically significant deterioration (optional). If `mid_deterioration` is not provided, it will be assumed to be equal to `mid_improvement`

`better_is`
: Which direction means a better outcome for the used instrument? Available are

- `"lower"` (lower outcome scores are desirable, the default) and
- `"higher"` (higher outcome scores are desirable)

`target`
: String, whether an individual or group analysis should be calculated. Available are

- `"individual"` (the default) for which every individual participant is evaluated
- `"group"` for which only the group wise effect is evaluated

`effect`
: String, if `target = "group"`, specify which effect should be calculated. Available are

- `"within"` (the default), which yields the mean pre-post intervention difference with associated confidence intervals
- `"between"`, which estimates the group wise mean difference and confidence intervals between two or more groups specified with the `group` argument at the specified measurement supplied with the `post-` argument The reference group may be supplied with `reference_group`

`bayesian`
: Logical, only relevant if `target = "group"`. Indicates if a Bayesian estimate (i.e., the median) of group differences with a credible interval should be calculated (if set to `TRUE`, the default) or a frequentist mean difference with confidence interval (if set to `FALSE`)

`prior_scale`
: String or numeric, can be adjusted to change the Bayesian prior distribution. See the documentation for `rscale` in `BayesFactor::ttestBF()` for details.

`reference_group`
: Specify the reference group to which all subsequent groups are compared against if `target = "group"` and `effect = "within"` (optional). Otherwise, the first distinct group is chosen based on alphabetical, numerical or factor ordering.

`ci_level`
: Numeric, define the credible or confidence interval level. The default is 0.95 for a 95%-CI.

Value

An S3 object of class `cs_analysis` and `cs_anchor`

Computational details

For the individual-level analyses, the analysis is straight forward. An MID can be specified for an improvement as well as a deterioration (because these must not necessarily be identical) and the function basically counts how many patients fall within the MID range for both, improvement and deterioration, or how many patients exceed the limits of this range in either direction. A patient may than be categorized as:

- Improved, the patient demonstrated a change that is equal or greater then the MID for an improvement

- Unchanged, the patient demonstrated a change that is less than both MIDs
- Deteriorated, the patient demonstrated a change that is equal or greater then the MID for a deterioration

For group-level analyses, the whole sample is either treated as a single group or is split up by grouping presented in the data. For within group analyses, the function calculates the median change from pre to post intervention with the associated credible interval (CI). Based on the median change and the limits of this CI, a group change can be categorized in 5 distinctive categories:

- Statistically not significant, the CI contains 0
- Statistically significant but not clinically relevant, the CI does not contain 0, but the median and both CI limits are beneath the MID threshold
- Not significantly less than the threshold, the MID threshold falls within the CI but the median is still below that threshold
- Probably clinically significant effect, the median crossed the MID threshold but the threshold is still inside the CI
- Large clinically significant effect, the median crossed the MID threshold and the CI does not contain the threshold

If a between group comparison is desired, a reference group can be defined with the `reference_group` argument to which all subsequent groups are compared. This is usually an inactive comparator such as a placebo or wait-list control group. The difference between the pairwise compared groups is categorized just as the within group difference above, so the same categories apply.

The approach can be changed to a classical frequentist framework for which the point estimate then represents the mean difference and the CI a confidence interval. For an extensive overview over the differences between a Bayesian and frequentist CI, refer to Hespanhol et al. (2019).

Data preparation

The data set must be tidy, which corresponds to a long data frame in general. It must contain a patient identifier which must be unique per patient. Also, a column containing the different measurements and the outcome must be supplied. Each participant-measurement combination must be unique, so for instance, the data must not contain two "After" measurements for the same patient.

Additionally, if the measurement column contains only two values, the first value based on alphabetical, numerical or factor ordering will be used as the `pre` measurement. For instance, if the column contains the measurements identifiers `"pre"` and `"post"` as strings, then `"post"` will be sorted before `"pre"` and thus be used as the `"pre"` measurement. The function will throw a warning but generally you may want to explicitly define the `"pre"` and `"post"` measurement with arguments `pre` and `post`. In case of more than two measurement identifiers, you have to define `pre` and `post` manually since the function does not know what your pre and post intervention measurements are.

If your data is grouped, you can specify the group by referencing the grouping variable (see examples below). The analysis is then run for every group to compare group differences.

References

Hespanhol, L., Vallio, C. S., Costa, L. M., & Saragiotto, B. T. (2019). Understanding and interpreting confidence and credible intervals around effect estimates. Brazilian Journal of Physical Therapy, 23(4), 290–301. https://doi.org/10.1016/j.bjpt.2018.12.006

See Also

Main clinical signficance functions cs_combined(), cs_distribution(), cs_percentage(), cs_statistical()

Examples

```
cs_results <- antidepressants |>
  cs_anchor(patient, measurement, mom_di, mid_improvement = 8)

cs_results
plot(cs_results)

# Set argument "pre" to avoid a warning
cs_results <- antidepressants |>
  cs_anchor(
    patient,
    measurement,
    mom_di,
    pre = "Before",
    mid_improvement = 8
  )


# Inlcude the MID for deterioration
cs_results_with_deterioration <- antidepressants |>
  cs_anchor(
    patient,
    measurement,
    mom_di,
    pre = "Before",
    mid_improvement = 8,
    mid_deterioration = 5
  )

cs_results_with_deterioration
summary(cs_results_with_deterioration)
plot(cs_results_with_deterioration)


# Group the results by experimental condition
cs_results_grouped <- antidepressants |>
  cs_anchor(
    patient,
    measurement,
    mom_di,
    pre = "Before",
    group = condition,
    mid_improvement = 8,
    mid_deterioration = 5
  )

cs_results_grouped
summary(cs_results_grouped)
```

```
plot(cs_results_grouped)

# The plot method always returns a ggplot2 object, so the plot may be further
# modified with ggplot2 code, e.g., facetting to avoid overplotting of groups
plot(cs_results_grouped) +
  ggplot2::facet_wrap(~ group)


# Compute group wise results
cs_results_groupwise <- antidepressants |>
  cs_anchor(
    patient,
    measurement,
    mom_di,
    pre = "Before",
    mid_improvement = 8,
    target = "group"
  )

cs_results_groupwise
summary(cs_results_groupwise)
plot(cs_results_groupwise)


# Group wise analysis, but split by experimentawl condition
cs_results_groupwise_condition <- antidepressants |>
  cs_anchor(
    patient,
    measurement,
    mom_di,
    pre = "Before",
    group = condition,
    mid_improvement = 8,
    target = "group"
  )

cs_results_groupwise_condition
summary(cs_results_groupwise_condition)
plot(cs_results_groupwise_condition)


# Compare all groups to a predefined reference group at a predefined measurement
cs_results_groupwise_between <- antidepressants |>
  cs_anchor(
    patient,
    measurement,
    mom_di,
    post = "After",
    group = condition,
    mid_improvement = 8,
    target = "group",
    effect = "between"
  )
```

```
cs_results_groupwise_between
summary(cs_results_groupwise_between)
plot(cs_results_groupwise_between)


# Compare all groups to a predefined reference group with frequentist appraoch
cs_results_groupwise_between_freq <- antidepressants |>
  cs_anchor(
    patient,
    measurement,
    mom_di,
    post = "After",
    group = condition,
    mid_improvement = 8,
    target = "group",
    effect = "between",
    bayesian = FALSE
  )

cs_results_groupwise_between_freq
summary(cs_results_groupwise_between_freq)
plot(cs_results_groupwise_between_freq)
```

`cs_combined` *Combined Analysis of Clinical Significance*

Description

`cs_combined()` can be used to determine the clinical significance of intervention studies employing the combination of the distribution-based and statistical approach. For this, it will be assumed that the functional (non-clinical population) and patient (clinical population) scores form two distinct distributions on a continuum. `cs_combined()` calculates a cutoff point between these two populations as well as a reliable change index (RCI) based on a provided instrument reliability estimate and counts, how many of those patients that showed a reliable change (that is likely to be not due to measurement error) switched from the clinical to the functional population during intervention. Several methods for calculating the cutoff and RCI are available.

Usage

```
cs_combined(
  data,
  id,
  time,
  outcome,
  group = NULL,
  pre = NULL,
  post = NULL,
  mid_improvement = NULL,
```

```
  mid_deterioration = NULL,
  reliability = NULL,
  reliability_post = NULL,
  m_functional = NULL,
  sd_functional = NULL,
  better_is = c("lower", "higher"),
  rci_method = c("JT", "GLN", "HLL", "EN", "NK", "HA", "HLM"),
  cutoff_type = c("a", "b", "c"),
  significance_level = 0.05
)
```

Arguments

`data`	A tidy data frame
`id`	Participant ID
`time`	Time variable
`outcome`	Outcome variable
`group`	Grouping variable (optional)
`pre`	Pre measurement (only needed if the time variable contains more than two measurements)
`post`	Post measurement (only needed if the time variable contains more than two measurements)
`mid_improvement`	Numeric, change that indicates a clinically significant improvement
`mid_deterioration`	Numeric, change that indicates a clinically significant deterioration (optional). If `mid_deterioration` is not provided, it will be assumed to be equal to `mid_improvement`
`reliability`	The instrument's reliability estimate. If you selected the NK method, the here specified reliability will be the instrument's pre measurement reliability. Not needed for the HLM method.
`reliability_post`	The instrument's reliability at post measurement (only needed for the NK method)
`m_functional`	Numeric, mean of functional population.
`sd_functional`	Numeric, standard deviation of functional population
`better_is`	Which direction means a better outcome for the used instrument? Available are • `"lower"` (lower outcome scores are desirable, the default) and • `"higher"` (higher outcome scores are desirable)
`rci_method`	Clinical significance method. Available are • `"JT"` (Jacobson & Truax, 1991, the default) • `"GLN"` (Gulliksen, Lord, and Novick; Hsu, 1989, Hsu, 1995) • `"HLL"` (Hsu, Linn & Nord; Hsu, 1989) • `"EN"` (Edwards & Nunnally; Speer, 1992)

- "NK" (Nunnally & Kotsch, 1983), requires a reliability estimate at post measurement. If this is not supplied, reliability and reliability_post are assumed to be equal
- "HA" (Hageman & Arrindell, 1999)
- "HLM" (Hierarchical Linear Modeling; Raudenbush & Bryk, 2002), requires at least three measurements per patient

`cutoff_type` Cutoff type. Available are "a", "b", and "c". Defaults to "a" but "c" is usually recommended. For "b" and "c", summary data from a functional population must be given with arguments `m_functional` and `sd_functional`.

`significance_level`
Significance level alpha, defaults to `0.05`. If you choose the "HA" method, this value corresponds to the maximum risk of misclassification

Value

An S3 object of class `cs_analysis` and `cs_combined`

Categories

Each individual's change can then be categorized into the following groups:

- Recovered, i.e., the individual showed a reliable change in the beneficial direction and changed from the clinical to the functional population
- Improved, i.e., the individual showed a reliable change in the beneficial direction but did not change populations
- Unchanged, i.e., the individual showed no reliable change
- Deteriorated, i.e., the individual showed a reliable change in the disadvantageous direction but did not change populations
- Harmed, i.e., the individual showed a reliable change in the disadvantageous direction and switched from the functional to the clinincal population

Computational details

There are three available cutoff types, namely a, b, and c which can be used to "draw a line" or separate the functional and clinical population on a continuum. a as a cutoff is defined as the mean of the clinical population minus two times the standard deviation (SD) of the clinical population. b is defined as the mean of the functional population plus also two times the SD of the clinical population. This is true for "negative" outcomes, where a lower instrument score is desirable. For "positive" outcomes, where higher scores are beneficial, a is the mean of the clinical population plus $2 \cdot$ SD of the clinical population and b is mean of the functional population minus $2 \cdot$ SD of the clinical population. The summary statistics for the clinical population are estimated from the provided data at pre measurement.

c is defined as the midpoint between both populations based on their respective mean and SD. In order to calculate b and c, descriptive statistics for the functional population must be provided.

From the provided data, a region of change is calculated in which an individual change may likely be due to an inherent measurement of the used instrument. This concept is also known as the minimally detectable change (MDC).

Data preparation

The data set must be tidy, which corresponds to a long data frame in general. It must contain a patient identifier which must be unique per patient. Also, a column containing the different measurements and the outcome must be supplied. Each participant-measurement combination must be unique, so for instance, the data must not contain two "After" measurements for the same patient.

Additionally, if the measurement column contains only two values, the first value based on alphabetical, numerical or factor ordering will be used as the `pre` measurement. For instance, if the column contains the measurements identifiers `"pre"` and `"post"` as strings, then `"post"` will be sorted before `"pre"` and thus be used as the `"pre"` measurement. The function will throw a warning but generally you may want to explicitly define the `"pre"` and `"post"` measurement with arguments `pre` and `post`. In case of more than two measurement identifiers, you have to define `pre` and `post` manually since the function does not know what your pre and post intervention measurements are.

If your data is grouped, you can specify the group by referencing the grouping variable (see examples below). The analysis is then run for every group to compare group differences.

See Also

Main clinical signficance functions `cs_anchor()`, `cs_distribution()`, `cs_percentage()`, `cs_statistical()`

Examples

```
cs_results <- claus_2020 |>
  cs_combined(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    reliability = 0.80
  )

cs_results
summary(cs_results)
plot(cs_results)


# You can choose a different cutoff but must provide summary statistics for the
# functional population
cs_results_c <- claus_2020 |>
  cs_combined(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    reliability = 0.80,
    m_functional = 8,
    sd_functional = 8,
    cutoff_type = "c"
  )
```

```
cs_results_c
summary(cs_results_c)
plot(cs_results_c)


# You can group the analysis by providing a grouping variable in the data
cs_results_grouped <- claus_2020 |>
  cs_combined(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    group = treatment,
    reliability = 0.80,
    m_functional = 8,
    sd_functional = 8,
    cutoff_type = "c"
  )

cs_results_grouped
summary(cs_results_grouped)
plot(cs_results_grouped)
```

`cs_distribution` *Distribution-Based Analysis of Clinical Significance*

Description

`cs_distribution()` can be used to determine the clinical significance of intervention studies employing the distribution-based approach. For this, the reliable change index is estimated from the provided data and a known reliability estimate which indicates, if an observed individual change is likely to be greater than the measurement error inherent for the used instrument. In this case, a reliable change is defined as clinically significant. Several methods for calculating this RCI can be chosen.

Usage

```
cs_distribution(
  data,
  id,
  time,
  outcome,
  group = NULL,
  pre = NULL,
  post = NULL,
  reliability = NULL,
  reliability_post = NULL,
```

```
  better_is = c("lower", "higher"),
  rci_method = c("JT", "GLN", "HLL", "EN", "NK", "HA", "HLM"),
  significance_level = 0.05
)
```

Arguments

`data`	A tidy data frame
`id`	Participant ID
`time`	Time variable
`outcome`	Outcome variable
`group`	Grouping variable (optional)
`pre`	Pre measurement (only needed if the time variable contains more than two measurements)
`post`	Post measurement (only needed if the time variable contains more than two measurements)
`reliability`	The instrument's reliability estimate. If you selected the NK method, the here specified reliability will be the instrument's pre measurement reliability. Not needed for the HLM method.
`reliability_post`	The instrument's reliability at post measurement (only needed for the NK method)
`better_is`	Which direction means a better outcome for the used instrument? Available are • `"lower"` (lower outcome scores are desirable, the default) and • `"higher"` (higher outcome scores are desirable)
`rci_method`	Clinical significance method. Available are • `"JT"` (Jacobson & Truax, 1991, the default) • `"GLN"` (Gulliksen, Lord, and Novick; Hsu, 1989, Hsu, 1995) • `"HLL"` (Hsu, Linn & Nord; Hsu, 1989) • `"EN"` (Edwards & Nunnally; Speer, 1992) • `"NK"` (Nunnally & Kotsch, 1983), requires a reliability estimate at post measurement. If this is not supplied, reliability and reliability_post are assumed to be equal • `"HA"` (Hageman & Arrindell, 1999) • `"HLM"` (Hierarchical Linear Modeling; Raudenbush & Bryk, 2002), requires at least three measurements per patient
`significance_level`	Significance level alpha, defaults to `0.05`. If you choose the `"HA"` method, this value corresponds to the maximum risk of misclassification

Value

An S3 object of class `cs_analysis` and `cs_distribution`

Computational details

From the provided data, a region of change is calculated in which an individual change may likely be due to an inherent measurement of the used instrument. This concept is also known as the minimally detectable change (MDC).

Categories

Each individual's change may then be categorized into one of the following three categories:

- Improved, the change is greater than the RCI in the beneficial direction
- Unchanged, the change is within a region that may attributable to measurement error
- Deteriorated, the change is greater than the RCI, but in the disadvantageous direction

Most of these methods are developed to deal with data containing two measurements per individual, i.e., a pre intervention and post intervention measurement. The Hierarchical Linear Modeling (`rci_method = "HLM"`) method can incorporate data for multiple measurements an can thus be used only for at least three measurements per participant.

Data preparation

The data set must be tidy, which corresponds to a long data frame in general. It must contain a patient identifier which must be unique per patient. Also, a column containing the different measurements and the outcome must be supplied. Each participant-measurement combination must be unique, so for instance, the data must not contain two "After" measurements for the same patient.

Additionally, if the measurement column contains only two values, the first value based on alphabetical, numerical or factor ordering will be used as the `pre` measurement. For instance, if the column contains the measurements identifiers `"pre"` and `"post"` as strings, then `"post"` will be sorted before `"pre"` and thus be used as the `"pre"` measurement. The function will throw a warning but generally you may want to explicitly define the `"pre"` and `"post"` measurement with arguments `pre` and `post`. In case of more than two measurement identifiers, you have to define `pre` and `post` manually since the function does not know what your pre and post intervention measurements are.

If your data is grouped, you can specify the group by referencing the grouping variable (see examples below). The analysis is then run for every group to compare group differences.

References

- Jacobson, N. S., & Truax, P. (1991). Clinical significance: A statistical approach to defining meaningful change in psychotherapy research. Journal of Consulting and Clinical Psychology, 59(1), 12–19. https://doi.org/10.1037//0022-006X.59.1.12
- Hsu, L. M. (1989). Reliable changes in psychotherapy: Taking into account regression toward the mean. Behavioral Assessment, 11(4), 459–467.
- Hsu, L. M. (1995). Regression toward the mean associated with measurement error and the identification of improvement and deterioration in psychotherapy. Journal of Consulting and Clinical Psychology, 63(1), 141–144. https://doi.org/10.1037//0022-006x.63.1.141
- Speer, D. C. (1992). Clinically significant change: Jacobson and Truax (1991) revisited. Journal of Consulting and Clinical Psychology, 60(3), 402–408. https://doi.org/10.1037/0022-006X.60.3.402

- Nunnally, J. C., & Kotsch, W. E. (1983). Studies of individual subjects: Logic and methods of analysis. British Journal of Clinical Psychology, 22(2), 83–93. https://doi.org/10.1111/j.2044-8260.1983.tb00582.x
- Hageman, W. J., & Arrindell, W. A. (1999). Establishing clinically significant change: increment of precision and the distinction between individual and group level analysis. Behaviour Research and Therapy, 37(12), 1169–1193. https://doi.org/10.1016/S0005-7967(99)00032-7
- Raudenbush, S. W., & Bryk, A. S. (2002). Hierarchical Linear Models - Applications and Data Analysis Methods (2nd ed.). Sage Publications.

See Also

Main clinical signficance functions `cs_anchor()`, `cs_combined()`, `cs_percentage()`, `cs_statistical()`

Examples

```
antidepressants |>
  cs_distribution(patient, measurement, mom_di, reliability = 0.80)


# Turn off the warning by providing the pre measurement time
cs_results <- antidepressants |>
  cs_distribution(
    patient,
    measurement,
    mom_di,
    pre = "Before",
    reliability = 0.80
  )

summary(cs_results)
plot(cs_results)


# If you use data with more than two measurements, you always have to define a
# pre and post measurement
cs_results <- claus_2020 |>
  cs_distribution(
    id,
    time,
    hamd,
    pre = 1,
    post = 4,
    reliability = 0.80
  )

cs_results
summary(cs_results)
plot(cs_results)


# Set the rci_method argument to change the RCI method
```

```
cs_results_ha <- claus_2020 |>
  cs_distribution(
    id,
    time,
    hamd,
    pre = 1,
    post = 4,
    reliability = 0.80,
    rci_method = "HA"
  )

cs_results_ha
summary(cs_results_ha)
plot(cs_results_ha)


# Group the analysis by providing a grouping variable
cs_results_grouped <- claus_2020 |>
  cs_distribution(
    id,
    time,
    hamd,
    pre = 1,
    post = 4,
    group = treatment,
    reliability = 0.80
  )

cs_results_grouped
summary(cs_results_grouped)
plot(cs_results_grouped)


# Use more than two measurements
cs_results_hlm <- claus_2020 |>
  cs_distribution(
    id,
    time,
    hamd,
    rci_method = "HLM"
  )

cs_results_hlm
summary(cs_results_hlm)
plot(cs_results_hlm)
```

---

cs_get_augmented_data *Extract Augmented Data from a cs_analysis Object*

Description

This function returns the patient-wise results, containing the considered pre and post intervention value, its raw change as well as all other change estimates calculated during the clinical significance analysis with the individual's clinical significance category. This function is only useful for individual level analyses because the group level analyses only yield group level results.

Usage

```
cs_get_augmented_data(x, ...)

## Default S3 method:
cs_get_augmented_data(x, ...)

## S3 method for class 'cs_distribution'
cs_get_augmented_data(x, ...)

## S3 method for class 'cs_statistical'
cs_get_augmented_data(x, ...)

## S3 method for class 'cs_combined'
cs_get_augmented_data(x, ...)

## S3 method for class 'cs_percentage'
cs_get_augmented_data(x, ...)

## S3 method for class 'cs_anchor_individual_within'
cs_get_augmented_data(x, ...)
```

Arguments

x	A cs_analysis object
...	Additional arguments

Value

A tibble with augmented data

See Also

Extractor functions cs_get_data(), cs_get_model(), cs_get_n(), cs_get_reliability(), cs_get_summary()

Examples

```
# Augmented data can be extracted for every individual approach
anchor_results <- claus_2020 |>
  cs_anchor(
    id,
    time,
    bdi,
```

```
    pre = 1,
    post = 4,
    mid_improvement = 9
  )


distribution_results <- claus_2020 |>
  cs_distribution(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    reliability = 0.80
  )


distribution_results_hlm <- claus_2020 |>
  cs_distribution(
    id,
    time,
    bdi,
    rci_method = "HLM"
  )


statistical_results <- claus_2020 |>
  cs_statistical(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    m_functional = 8,
    sd_functional = 8
  )


combined_results <- claus_2020 |>
  cs_combined(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    m_functional = 8,
    sd_functional = 8,
    reliability = 0.80
  )


cs_get_augmented_data(anchor_results)
cs_get_augmented_data(distribution_results)
```

```
cs_get_augmented_data(distribution_results_hlm)
cs_get_augmented_data(statistical_results)
cs_get_augmented_data(combined_results)
```

`cs_get_cutoff` *Get Used Cutoff And Type From A cs_analysis Object*

Description

Get Used Cutoff And Type From A cs_analysis Object

Usage

```
cs_get_cutoff(x, with_descriptives = FALSE)
```

Arguments

`x`	A cs_analysis object
`with_descriptives`	Logical indicating whether you want to retrieve only the cutoff type and value or the summary statistics on which it is based on. The default is `FALSE`.

Value

A tibble with cutoff information

Examples

```
cs_results <- claus_2020 |>
  cs_statistical(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    m_functional = 8,
    sd_functional = 8,
    cutoff_type = "c"
  )

cs_get_cutoff(cs_results)
cs_get_cutoff(cs_results, with_descriptives = TRUE)
```

cs_get_cutoff_descriptives *Get Descriptives Used In The Cutoff Calculation*

Description

Get Descriptives Used In The Cutoff Calculation

Usage

```
cs_get_cutoff_descriptives(x)
```

Arguments

x	A clinisig object

Value

A tibble with means and standard deviations of the clinical and functional population

cs_get_data *Get Data From A cs_analysis Object*

Description

Get Data From A cs_analysis Object

Usage

```
cs_get_data(x, dataset = "data")
```

Arguments

x	A cs_analysis object.
dataset	The dataset you wish to retrieve. Available options are • "original" (the raw original dataset) • "wide" (the original dataset in wide format) • "data" (the dataset which is used in the calculations). The default is "data"

Value

A tibble

See Also

Extractor functions cs_get_augmented_data(), cs_get_model(), cs_get_n(), cs_get_reliability(), cs_get_summary()

Examples

```
cs_results <- claus_2020 |>
  cs_anchor(id, time, bdi, mid_improvement = 9, pre = 1, post = 4)

cs_get_data(cs_results)
cs_get_data(cs_results, dataset = "wide")
cs_get_data(cs_results, dataset = "original")
```

---

cs_get_model *Get The HLM Model From A cs_analysis Object*

Description

With cs_get_model() you can extract the fitted HLM model for the distribution-based approach. This is useful to either diagnose the model further (beacuse all assumptions of HLMs apply in this case) or plot the results differently.

Usage

```
cs_get_model(x)
```

Arguments

x	A cs_analysis object

Value

A model of class lmerMod

See Also

Extractor functions cs_get_augmented_data(), cs_get_data(), cs_get_n(), cs_get_reliability(), cs_get_summary()

Examples

```
cs_results <- claus_2020 |>
  cs_distribution(id, time, bdi, rci_method = "HLM")

cs_get_model(cs_results)
```

cs_get_n *Get Number Of Participants From A cs_analysis Object*

Description

With `cs_get_n()` one can extract the number of participants used in a clinical significance analysis from a `cs_analysis`object. This may depend on the clinical significance approach and if missing values were present in the dataset. For all individual analyses, missing values are handled by list-wise deletion. Consequently, individuals with a missing pre or post intervention score will be omitted from the analyses.

Usage

```
cs_get_n(x, which = "all")
```

Arguments

`x`	A cs_analysis object
`which`	Which n should be returned? Available options are • `"all"`, (the default) returns the number of participants in both, the original and used data set • `"original"`, number of participants in the original dataset • `"used"`, number of participants in the used data set, so after conversion to wide format and omitting cases with missing values

Value

A tibble with number of participants

See Also

Extractor functions `cs_get_augmented_data()`, `cs_get_data()`, `cs_get_model()`, `cs_get_reliability()`, `cs_get_summary()`

Examples

```
# n can be extracted for every approach
cs_results_anchor <- claus_2020 |>
  cs_anchor(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    mid_improvement = 9
  )

cs_results_distribution <- claus_2020 |>
```

```
  cs_distribution(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    reliability = 0.80
  )

cs_results_statistical <- claus_2020 |>
  cs_statistical(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    m_functional = 8,
    sd_functional = 8,
    cutoff_type = "c"
  )

cs_results_combined <- claus_2020 |>
  cs_combined(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    reliability = 0.80,
    m_functional = 8,
    sd_functional = 8,
    cutoff_type = "c"
  )

cs_results_percentage <- claus_2020 |>
  cs_percentage(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    pct_improvement = 0.3
  )


cs_get_n(cs_results_anchor)
cs_get_n(cs_results_distribution)
cs_get_n(cs_results_statistical)
cs_get_n(cs_results_combined)
cs_get_n(cs_results_percentage)


# Get your desired n
```

```
cs_get_n(cs_results_anchor, which = "all")
cs_get_n(cs_results_anchor, which = "original")
cs_get_n(cs_results_anchor, which = "used")
```

`cs_get_reliability` *Get Reliability Of A cs_analysis Object*

Description

Get Reliability Of A cs_analysis Object

Usage

```
cs_get_reliability(x)
```

Arguments

x	A cs_analysis object

Value

A tibble showing the reliability

See Also

Extractor functions `cs_get_augmented_data()`, `cs_get_data()`, `cs_get_model()`, `cs_get_n()`, `cs_get_summary()`

Examples

```
cs_results <- claus_2020 |>
  cs_distribution(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    reliability = 0.80
  )


cs_get_reliability(cs_results)
```

cs_get_summary *Get A Summary Table From A cs_analysis Object*

Description

Retrieve the summary table in a tidy tibble format. This is especially useful to plot the results or conduct sensitivity analyses.

Usage

```
cs_get_summary(x, ...)

## Default S3 method:
cs_get_summary(x, which = c("individual", "group"), ...)

## S3 method for class 'cs_anchor_group_within'
cs_get_summary(x, ...)

## S3 method for class 'cs_anchor_group_between'
cs_get_summary(x, ...)
```

Arguments

x	An object of class cs_analysis
...	Additional arguments passed to the respective method
which	Which level of summary table to return. This is only necessary for method "HA" since two summary tables are reported. Available are • individual, the default • group, group level results according to Hageman & Arrindell (1999)

Value

A tibble with clinical significance categories

References

- Hageman, W. J., & Arrindell, W. A. (1999). Establishing clinically significant change: increment of precision and the distinction between individual and group level analysis. Behaviour Research and Therapy, 37(12), 1169–1193. https://doi.org/10.1016/S0005-7967(99)00032-7

See Also

Extractor functions cs_get_augmented_data(), cs_get_data(), cs_get_model(), cs_get_n(), cs_get_reliability()

Examples

```
anchor_results <- claus_2020 |>
  cs_anchor(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    mid_improvement = 8
  )

cs_get_summary(anchor_results)


# Get summary table for a group level analysis
anchor_results_grouped <- claus_2020 |>
  cs_anchor(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    mid_improvement = 8,
    target = "group"
  )

cs_get_summary(anchor_results_grouped)


# Get group-wise summary table for the Hageman & Arrindell method
combined_results <- claus_2020 |>
  cs_combined(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    m_functional = 8,
    sd_functional = 8,
    reliability = 0.80,
    rci_method = "HA"
  )

cs_get_summary(combined_results)
cs_get_summary(combined_results, which = "group")
```

`cs_percentage` *Percentage-Change Analysis of Clinical Significance*

Description

`cs_percentage()` can be used to determine the clinical significance of intervention studies employing the percentage-change approach. For this, each individuals relative change compared to the pre intervention measurement and if this change exceeds a predefined change in percent points, this change is then deemed clinically significant.

Usage

```
cs_percentage(
  data,
  id,
  time,
  outcome,
  group = NULL,
  pre = NULL,
  post = NULL,
  pct_improvement = NULL,
  pct_deterioration = NULL,
  better_is = c("lower", "higher")
)
```

Arguments

`data`	A tidy data frame
`id`	Participant ID
`time`	Time variable
`outcome`	Outcome variable
`group`	Grouping variable (optional)
`pre`	Pre measurement (only needed if the time variable contains more than two measurements)
`post`	Post measurement (only needed if the time variable contains more than two measurements)
`pct_improvement`	Numeric, percent change that indicates a clinically significant improvement
`pct_deterioration`	Numeric, percent change that indicates a clinically significant deterioration (optional). If this is not set, `pct_deterioration` will be assumed to be equal to `pct_improvement`
`better_is`	Which direction means a better outcome for the used instrument? Available are • `"lower"` (lower outcome scores are desirable, the default) and • `"higher"` (higher outcome scores are desirable)

Value

An S3 object of class `cs_analysis` and `cs_percentage`

Computational details

Each participants change is calculated and then divided by the pre intervention score to estimate the individual's percent change. A percent change for an improvement as well as a deterioration can be provided separately and if `pct_deterioration` is not set, it will be assumed to be the same as `pct_improvement`.

Categories

Each individual's change may then be categorized into one of the following three categories:

- Improved, the change is greater than the predefined percent change in the beneficial direction
- Unchanged, the change is within the predefined percent change
- Deteriorated, the change is greater than the predefined percent change, but in the disadvantageous direction

Data preparation

The data set must be tidy, which corresponds to a long data frame in general. It must contain a patient identifier which must be unique per patient. Also, a column containing the different measurements and the outcome must be supplied. Each participant-measurement combination must be unique, so for instance, the data must not contain two "After" measurements for the same patient.

Additionally, if the measurement column contains only two values, the first value based on alphabetical, numerical or factor ordering will be used as the `pre` measurement. For instance, if the column contains the measurements identifiers `"pre"` and `"post"` as strings, then `"post"` will be sorted before `"pre"` and thus be used as the `"pre"` measurement. The function will throw a warning but generally you may want to explicitly define the `"pre"` and `"post"` measurement with arguments `pre` and `post`. In case of more than two measurement identifiers, you have to define `pre` and `post` manually since the function does not know what your pre and post intervention measurements are.

If your data is grouped, you can specify the group by referencing the grouping variable (see examples below). The analysis is then run for every group to compare group differences.

See Also

Main clinical signficance functions `cs_anchor()`, `cs_combined()`, `cs_distribution()`, `cs_statistical()`

Examples

```
cs_results <- claus_2020 |>
  cs_percentage(
    id,
    time,
    hamd,
    pre = 1,
    post = 4,
    pct_improvement = 0.3
  )

cs_results
summary(cs_results)
```

```
plot(cs_results)


# You can set different thresholds for improvement and deterioration
cs_results_2 <- claus_2020 |>
  cs_percentage(
    id,
    time,
    hamd,
    pre = 1,
    post = 4,
    pct_improvement = 0.3,
    pct_deterioration = 0.2
  )

cs_results_2
summary(cs_results_2)
plot(cs_results_2)


# You can group the analysis by providing a group column from the data
cs_results_grouped <- claus_2020 |>
  cs_percentage(
    id,
    time,
    hamd,
    pre = 1,
    post = 4,
    pct_improvement = 0.3,
    group = treatment
  )

cs_results_grouped
summary(cs_results_grouped)
plot(cs_results_grouped)


# The analyses can be performed for positive outcomes as well, i.e., outcomes
# for which a higher value is beneficial
cs_results_who <- claus_2020 |>
  cs_percentage(
    id,
    time,
    who,
    pre = 1,
    post = 4,
    pct_improvement = 0.3,
    better_is = "higher"
  )

cs_results_who
summary(cs_results_who)
plot(cs_results_who)
```

```
plot(cs_results_who, show = category)
```

`cs_statistical` *Statistical Analysis of Clinical Significance*

Description

`cs_statistical()` can be used to determine the clinical significance of intervention studies employing the statistical approach. For this, it will be assumed that the functional (non-clinical population) and patient (clinical population) scores form two distinct distributions on a continuum. `cs_statistical()` calculates a cutoff point between these two populations and counts, how many patients changed from the clinical to the functional population during intervention. Several methods for calculating this cutoff are available.

Usage

```
cs_statistical(
  data,
  id,
  time,
  outcome,
  group = NULL,
  pre = NULL,
  post = NULL,
  m_functional = NULL,
  sd_functional = NULL,
  reliability = NULL,
  better_is = c("lower", "higher"),
  cutoff_method = c("JT", "HA"),
  cutoff_type = c("a", "b", "c"),
  significance_level = 0.05
)
```

Arguments

`data`	A tidy data frame
`id`	Participant ID
`time`	Time variable
`outcome`	Outcome variable
`group`	Grouping variable (optional)
`pre`	Pre measurement (only needed if the time variable contains more than two measurements)
`post`	Post measurement (only needed if the time variable contains more than two measurements)
`m_functional`	Numeric, mean of functional population.

`sd_functional` Numeric, standard deviation of functional population

`reliability` The instrument's reliability estimate. If you selected the NK method, the here specified reliability will be the instrument's pre measurement reliability. Not needed for the HLM method.

`better_is` Which direction means a better outcome for the used instrument? Available are

- `"lower"` (lower outcome scores are desirable, the default) and
- `"higher"` (higher outcome scores are desirable)

`cutoff_method` Cutoff method, Available are

- `"JT"` (Jacobson & Truax, 1991, the default)
- `"HA"` (Hageman & Arrindell, 1999)

`cutoff_type` Cutoff type. Available are `"a"`, `"b"`, and `"c"`. Defaults to `"a"` but `"c"` is usually recommended. For `"b"` and `"c"`, summary data from a functional population must be given with arguments `m_functional` and `sd_functional`.

`significance_level`
Significance level alpha, defaults to `0.05`. If you choose the `"HA"` method, this value corresponds to the maximum risk of misclassification

Value

An S3 object of class `cs_analysis` and `cs_statistical`

Computational details

There are three available cutoff types, namely a, b, and c which can be used to "draw a line" or separate the functional and clinical population on a continuum. a as a cutoff is defined as the mean of the clinical population minus two times the standard deviation (SD) of the clinical population. b is defined as the mean of the functional population plus also two times the SD of the clinical population. This is true for "negative" outcomes, where a lower instrument score is desirable. For "positive" outcomes, where higher scores are beneficial, a is the mean of the clinical population plus $2 \cdot$ SD of the clinical population and b is mean of the functional population minus $2 \cdot$ SD of the clinical population. The summary statistics for the clinical population are estimated from the provided data at pre measurement.

c is defined as the midpoint between both populations based on their respective mean and SD. In order to calculate b and c, descriptive statistics for the functional population must be provided.

Categories

Individual patients can be categorized into one of the following groups:

- Improved, i.e., one changed from the clinical to the functional population
- Unchanged, i.e., one can be seen as a member of the same population pre and post intervention
- Deteriorated, i.e., one changed from the functional to the clinical population during intervention

Data preparation

The data set must be tidy, which corresponds to a long data frame in general. It must contain a patient identifier which must be unique per patient. Also, a column containing the different measurements and the outcome must be supplied. Each participant-measurement combination must be unique, so for instance, the data must not contain two "After" measurements for the same patient.

Additionally, if the measurement column contains only two values, the first value based on alphabetical, numerical or factor ordering will be used as the `pre` measurement. For instance, if the column contains the measurements identifiers `"pre"` and `"post"` as strings, then `"post"` will be sorted before `"pre"` and thus be used as the `"pre"` measurement. The function will throw a warning but generally you may want to explicitly define the `"pre"` and `"post"` measurement with arguments `pre` and `post`. In case of more than two measurement identifiers, you have to define `pre` and `post` manually since the function does not know what your pre and post intervention measurements are.

If your data is grouped, you can specify the group by referencing the grouping variable (see examples below). The analysis is then run for every group to compare group differences.

References

- Jacobson, N. S., & Truax, P. (1991). Clinical significance: A statistical approach to defining meaningful change in psychotherapy research. Journal of Consulting and Clinical Psychology, 59(1), 12–19. https://doi.org/10.1037//0022-006X.59.1.12
- Hageman, W. J., & Arrindell, W. A. (1999). Establishing clinically significant change: increment of precision and the distinction between individual and group level analysis. Behaviour Research and Therapy, 37(12), 1169–1193. https://doi.org/10.1016/S0005-7967(99)00032-7

See Also

Main clinical signficance functions `cs_anchor()`, `cs_combined()`, `cs_distribution()`, `cs_percentage()`

Examples

```
# By default, cutoff type "a" is used
cs_results <- claus_2020 |>
  cs_statistical(id, time, hamd, pre = 1, post = 4)

cs_results
summary(cs_results)
plot(cs_results)


# You can choose a different cutoff type but need to provide additional
# population summary statistics for the functional population
cs_results_c <- claus_2020 |>
  cs_statistical(
    id,
    time,
    hamd,
    pre = 1,
    post = 4,
    m_functional = 8,
```

```
    sd_functional = 8,
    cutoff_type = "c"
  )

cs_results_c
summary(cs_results_c)
plot(cs_results_c)


# You can use a different method to calculate the cutoff
cs_results_ha <- claus_2020 |>
  cs_statistical(
    id,
    time,
    hamd,
    pre = 1,
    post = 4,
    m_functional = 8,
    sd_functional = 8,
    reliability = 0.80,
    cutoff_type = "c",
    cutoff_method = "HA"
  )

cs_results_ha
summary(cs_results_ha)
plot(cs_results_ha)


# And you can group the analysis by providing a grouping variable from the data
cs_results_grouped <- claus_2020 |>
  cs_statistical(
    id,
    time,
    hamd,
    pre = 1,
    post = 4,
    m_functional = 8,
    sd_functional = 8,
    cutoff_type = "c",
    group = treatment
  )

cs_results_grouped
summary(cs_results_grouped)
plot(cs_results_grouped)
```

---

garden_club *Fictional "Garden Club" Pre-Post Intervention Data*

Description

A simulated dataset from a linear mixed-effects model (LMM) representing a standard two-group (CBT vs. Antienvyssant) pre-post intervention study. The data is ideal for demonstrating the analysis of longitudinal data with repeated measures.

Usage

```
garden_club
```

Format

A tibble with 150 rows and 6 variables:

id An integer representing the unique participant identifier.

group A character vector indicating the treatment group (`"CBT"` or `"Antienvyssant"`).

age An integer representing the participant's age.

gender A character vector for the participant's gender (`"m"`, `"f"`, or `"d"`).

time An integer for the measurement occasion (`0` = pre-test, `1` = post-test).

gei_score A numeric value for the outcome, the score on the fictional Gardener Envy Inventory (GEI), ranging from 0 to 50.

Source

Simulated data based on a linear mixed-effects model (LMM) with fixed effects for group and time, and random intercepts and slopes for participants.

See Also

`garden_club_sensitive` for a dataset designed for sensitivity analysis.

`garden_club_sensitive` *Fictional "Garden Club" Data for MID Sensitivity Analysis*

Description

A simulated dataset specifically designed to be "sensitive" to the choice of a Minimal Important Difference (MID). The mean change score is deliberately set to be near common MID thresholds, with a small standard deviation, making it perfect for demonstrating MID-based sensitivity analyses.

Usage

```
garden_club_sensitive
```

Format

A tibble with 200 rows and 3 variables:

id An integer representing the unique participant identifier.

time A numeric value for the measurement occasion (1 = pre, 2 = post).

gei_score A numeric value for the outcome, the score on the fictional Gardener Envy Inventory (GEI).

Source

Simulated data where the mean change score is precisely calibrated to fall within a range of plausible Minimal Important Difference (MID) values.

See Also

`garden_club` for a more standard pre-post dataset.

---

hechler_2014 *Chronic Pain Data*

Description

A reduced version of the data collected by Hechler et al. (2014)

Usage

```
hechler_2014
```

Format

A tibble with 208 rows and 3 variables:

`patient` Patient identifier

`measurement` Indicator of measurement

`disability` Pain-related disability as measured with the PPDI (lower is better)

References

Hechler, T., Ruhe, A.-K., Schmidt, P., Hirsch, J., Wager, J., Dobe, M., Krummenauer, F., & Zernikow, B. (2014). Inpatient Based Intensive Interdisciplinary Pain Treatment for Highly Impaired Children with Severe Chronic Pain: Randomized Controlled Trial of Efficacy and Economic Effects. Pain, 155(1), 118–128. https://doi.org/10.1016/j.pain.2013.09.015

jacobson_1989 *Marital Therapy Data*

Description

A dataset containing the data from Jacobson et al. (1989). The purpose of the study was to examine two forms of behavioral marital therapy,

Usage

```
jacobson_1989
```

Format

An object of class `tbl_df` with 60 rows and 4 columns.

subject Subject ID

time Measurement

das Dyadic Adjustment Scale score (higher is better)

gds Global Distress Scale score (lower is better)

References

- Jacobson, N. S., Schmaling, K. B., Holtzworth-Munroe, A., Katt, J. L., Wood, L. F., & Follette, V. M. (1989). Research-structured vs clinically flexible versions of social learning-based marital therapy. Behaviour Research and Therapy, 27(2), 173-180. https://doi.org/10.1016/0005-7967(89)90076-4
- Jacobson, N. S., & Truax, P. (1991). Clinical significance: A statistical approach to defining meaningful change in psychotherapy research. Journal of Consulting and Clinical Psychology, 59(1), 12-19. https://doi.org/10.1037//0022-006X.59.1.12

plot.cs_anchor_group_between *Plot an Object of Class cs_anchor_group_between*

Description

This function creates a generic group level clinical significance plot by plotting the between group change with the associated uncertainty interval around the estimated change on the y-axis.

Usage

```
## S3 method for class 'cs_anchor_group_between'
plot(
  x,
  x_lab = "Group",
  y_lab = "Mean Intervention Effect\n(with 95%-CI)",
  ...
)
```

Arguments

x	An object of class cs_anchor_group_between
x_lab	String, x axis label, defaults to "Group"
y_lab	String, y axis label, defaults to "Mean Intervention Effect (with 95%-CI)"
...	Additional arguments

Value

A ggplot2 plot

Examples

```
cs_results <- antidepressants |>
  cs_anchor(
    patient,
    measurement,
    mom_di,
    mid_improvement = 8,
    target = "group",
    group = condition,
    effect = "between",
    post = "After"
  )


# Plot the results "as is"
plot(cs_results)


# Change the axis labels
plot(cs_results, x_lab = "Condition", y_lab = "Treatment Effect")
```

---

plot.cs_anchor_group_within

Plot an Object of Class cs_anchor_group_within

Description

This function creates a generic group level clinical significance plot by plotting the within group change with the associated uncertainty interval around the estimated change on the y-axis.

Usage

```
## S3 method for class 'cs_anchor_group_within'
plot(
  x,
  x_lab = "Group",
  y_lab = "Mean Intervention Effect\n(with 95%-CI)",
  ...
)
```

Arguments

x	An object of class cs_anchor_group_within
x_lab	String, x axis label. Defaults to "Group"
y_lab	String, y axis label, defaults to "Mean Intervention Effect (with 95%-CI)"
...	Additional arguments

Value

A ggplot2 plot

Examples

```
cs_results <- antidepressants |>
  cs_anchor(
    patient,
    measurement,
    mom_di,
    mid_improvement = 8,
    target = "group",
    group = condition
  )


# Plot the results "as is"
plot(cs_results)


# Change the axis labels
plot(cs_results, x_lab = "Condition", y_lab = "Treatment Effect")
```

plot.cs_anchor_individual_within *Plot an Object of Class cs_anchor_individual_within*

Description

This function creates a generic clinical significance plot by plotting the patients' pre intervention value on the x-axis and the post intervention score on the y-axis.

Usage

```
## S3 method for class 'cs_anchor_individual_within'
plot(
  x,
  x_lab = "Pre",
  y_lab = "Post",
  color_lab = "Group",
  lower_limit,
  upper_limit,
  show,
  point_alpha = 1,
  mid_fill = "grey10",
  mid_alpha = 0.1,
  overplotting = 0.02,
  ...
)
```

Arguments

x	An object of class `cs_distribution`
x_lab	String, x axis label. Default is `"Pre"`.
y_lab	String, x axis label. Default is `"Post"`.
color_lab	String, color label (if colors are displayed). Default is `"Group"`
lower_limit	Numeric, lower plotting limit. Defaults to 2% smaller than minimum instrument score
upper_limit	Numeric, upper plotting limit. Defaults to 2% larger than maximum instrument score
show	Unquoted category name. You have several options to color different features. Available are • `improved` (shows improved participants) • `unchanged` (shows unchanged participants) • `deteriorated` (shows deteriorated participants)
point_alpha	Numeric, transparency adjustment for points. A value between 0 and 1 where 1 corresponds to not transparent at all and 0 to fully transparent.

mid_fill	String, a color (name or HEX code) for the percentage range fill
mid_alpha	Numeric, controls the transparency of the percentage fill. This can be any value between 0 and 1, defaults to 0.1
overplotting	Numeric, control amount of overplotting. Defaults to 0.02 (i.e., 2% of range between lower and upper limit).
...	Additional arguments

Value

A ggplot2 plot

Examples

```
cs_results <- antidepressants |>
  cs_anchor(
    patient,
    measurement,
    pre = "Before",
    mom_di,
    mid_improvement = 8
  )


# Plot the results "as is"
plot(cs_results)


# Change the axis labels
plot(cs_results, x_lab = "Before Intervention", y_lab = "After Intervention")


# Show the individual categories
plot(cs_results, show = category)


# Show a specific category
plot(cs_results, show = improved)


# Show groups as specified in the data
cs_results_grouped <- antidepressants |>
  cs_anchor(
    patient,
    measurement,
    pre = "Before",
    mom_di,
    mid_improvement = 8,
    group = condition
  )

plot(cs_results_grouped)
```

```
# To avoid overplotting, generic ggplot2 code can be used to facet the plot
library(ggplot2)
plot(cs_results_grouped) +
  facet_wrap(~ group)


# Adjust the transparency of individual data points
plot(cs_results, point_alpha = 0.3)


# Adjust the fill and transparency of the "unchanged" (PCC) region
plot(cs_results, mid_fill = "firebrick", mid_alpha = 0.2)


# Control the overplotting
plot(cs_results, overplotting = 0.1)


# Or adjust the axis limits by hand
plot(cs_results, lower_limit = 0, upper_limit = 80)
```

`plot.cs_combined` *Plot an Object of Class cs_combined*

Description

This function creates a generic clinical significance plot by plotting the patients' pre intervention value on the x-axis and the post intervention score on the y-axis.

Usage

```
## S3 method for class 'cs_combined'
plot(
  x,
  x_lab = NULL,
  y_lab = NULL,
  color_lab = "Group",
  lower_limit,
  upper_limit,
  show,
  point_alpha = 1,
  trajectory_alpha = 1,
  rci_fill = "grey10",
  rci_alpha = 0.1,
  overplotting = 0.02,
  ...
)
```

Arguments

x	An object of class cs_distribution
x_lab	String, x axis label. Default is "Pre".
y_lab	String, x axis label. Default is "Post".
color_lab	String, color label (if colors are displayed). Default is "Group"
lower_limit	Numeric, lower plotting limit. Defaults to 2% smaller than minimum instrument score
upper_limit	Numeric, upper plotting limit. Defaults to 2% larger than maximum instrument score
show	Unquoted category name. You have several options to color different features. Available are • category (shows all categories at once) • recovered (shows recovered participants) • improved (shows improved participants) • unchanged (shows unchanged participants) • deteriorated (shows deteriorated participants) • harmed (shows harmed participants)
point_alpha	Numeric, transparency adjustment for points. A value between 0 and 1 where 1 corresponds to not transparent at all and 0 to fully transparent.
trajectory_alpha	Numeric, transparency adjustment for trajectories. A value between 0 and 1 where 1 corresponds to not transparent at all and 0 to fully transparent.
rci_fill	String, a color (name or HEX code) for RCI fill
rci_alpha	Numeric, controls the transparency of the RCI. This can be any value between 0 and 1, defaults to 0.1
overplotting	Numeric, control amount of overplotting. Defaults to 0.02 (i.e., 2% of range between lower and upper limit).
...	Additional arguments

Value

A ggplot2 plot

Examples

```
cs_results <- antidepressants |>
  cs_combined(
    patient,
    measurement,
    pre = "Before",
    mom_di,
    reliability = 0.80,
    m_functional = 15,
    sd_functional = 8,
```

```
    cutoff_type = "c"
  )


# Plot the results "as is"
plot(cs_results)


# Change the axis labels
plot(cs_results, x_lab = "Before Intervention", y_lab = "After Intervention")


# Show the individual categories
plot(cs_results, show = category)


# Show a specific
plot(cs_results, show = recovered)


# Show groups as specified in the data
cs_results_grouped <- antidepressants |>
  cs_combined(
    patient,
    measurement,
    pre = "Before",
    mom_di,
    reliability = 0.80,
    m_functional = 15,
    sd_functional = 8,
    cutoff_type = "c",
    group = condition
  )

plot(cs_results_grouped)


# To avoid overplotting, generic ggplot2 code can be used to facet the plot
library(ggplot2)
plot(cs_results_grouped) +
  facet_wrap(~ group)


# Adjust the transparency of individual data points
plot(cs_results, point_alpha = 0.3)


# Adjust the fill and transparency of the "unchanged" (RCI) region
plot(cs_results, rci_fill = "firebrick", rci_alpha = 0.2)


# Control the overplotting
plot(cs_results, overplotting = 0.1)
```

```
# Or adjust the axis limits by hand
plot(cs_results, lower_limit = 0, upper_limit = 80)
```

`plot.cs_distribution` *Plot an Object of Class cs_distribution*

Description

This function creates a generic clinical significance plot by plotting the patients' pre intervention value on the x-axis and the post intervention score on the y-axis.

Usage

```
## S3 method for class 'cs_distribution'
plot(
  x,
  x_lab = NULL,
  y_lab = NULL,
  color_lab = "Group",
  lower_limit,
  upper_limit,
  show,
  point_alpha = 1,
  trajectory_alpha = 1,
  rci_fill = "grey10",
  rci_alpha = 0.1,
  overplotting = 0.02,
  ...
)
```

Arguments

`x`	An object of class `cs_distribution`
`x_lab`	String, x axis label. Default is `"Pre"`.
`y_lab`	String, x axis label. Default is `"Post"`.
`color_lab`	String, color label (if colors are displayed). Default is `"Group"`
`lower_limit`	Numeric, lower plotting limit. Defaults to 2% smaller than minimum instrument score
`upper_limit`	Numeric, upper plotting limit. Defaults to 2% larger than maximum instrument score
`show`	Unquoted category name. You have several options to color different features. Available are • `category` (shows all categories at once)

- `improved` (shows improved participants)
- `unchanged` (shows unchanged participants)
- `deteriorated` (shows deteriorated participants)

`point_alpha` Numeric, transparency adjustment for points. A value between 0 and 1 where 1 corresponds to not transparent at all and 0 to fully transparent.

`trajectory_alpha` Numeric, transparency adjustment for trajectories. A value between 0 and 1 where 1 corresponds to not transparent at all and 0 to fully transparent.

`rci_fill` String, a color (name or HEX code) for RCI fill

`rci_alpha` Numeric, controls the transparency of the RCI. This can be any value between 0 and 1, defaults to 0.1

`overplotting` Numeric, control amount of overplotting. Defaults to 0.02 (i.e., 2% of range between lower and upper limit).

`...` Additional arguments

Value

A ggplot2 plot

Examples

```
cs_results <- antidepressants |>
  cs_distribution(
    patient,
    measurement,
    pre = "Before",
    mom_di,
    reliability = 0.80
  )


# Plot the results "as is"
plot(cs_results)


# Change the axis labels
plot(cs_results, x_lab = "Before Intervention", y_lab = "After Intervention")


# Show the individual categories
plot(cs_results, show = category)


# Show a specific
plot(cs_results, show = improved)


# Show groups as specified in the data
cs_results_grouped <- antidepressants |>
```

```
  cs_distribution(
    patient,
    measurement,
    pre = "Before",
    mom_di,
    reliability = 0.80,
    group = condition
  )

plot(cs_results_grouped)


# To avoid overplotting, generic ggplot2 code can be used to facet the plot
library(ggplot2)
plot(cs_results_grouped) +
  facet_wrap(~ group)


# Adjust the transparency of individual data points
plot(cs_results, point_alpha = 0.3)


# Adjust the fill and transparency of the "unchanged" (RCI) region
plot(cs_results, rci_fill = "firebrick", rci_alpha = 0.2)


# Control the overplotting
plot(cs_results, overplotting = 0.1)


# Or adjust the axis limits by hand
plot(cs_results, lower_limit = 0, upper_limit = 80)
```

`plot.cs_percentage` *Plot an Object of Class cs_percentage*

Description

This function creates a generic clinical significance plot by plotting the patients' pre intervention value on the x-axis and the post intervention score on the y-axis.

Usage

```
## S3 method for class 'cs_percentage'
plot(
  x,
  x_lab = "Pre",
  y_lab = "Post",
  color_lab = "Group",
  lower_limit,
```

```
  upper_limit,
  show,
  point_alpha = 1,
  pct_fill = "grey10",
  pct_alpha = 0.1,
  overplotting = 0.02,
  ...
)
```

Arguments

x	An object of class `cs_distribution`
x_lab	String, x axis label. Default is `"Pre"`.
y_lab	String, x axis label. Default is `"Post"`.
color_lab	String, color label (if colors are displayed). Default is `"Group"`
lower_limit	Numeric, lower plotting limit. Defaults to 2% smaller than minimum instrument score
upper_limit	Numeric, upper plotting limit. Defaults to 2% larger than maximum instrument score
show	Unquoted category name. You have several options to color different features. Available are • `improved` (shows improved participants) • `unchanged` (shows unchanged participants) • `deteriorated` (shows deteriorated participants)
point_alpha	Numeric, transparency adjustment for points. A value between 0 and 1 where 1 corresponds to not transparent at all and 0 to fully transparent.
pct_fill	String, a color (name or HEX code) for the percentage range fill
pct_alpha	Numeric, controls the transparency of the percentage fill. This can be any value between 0 and 1, defaults to 0.1
overplotting	Numeric, control amount of overplotting. Defaults to 0.02 (i.e., 2% of range between lower and upper limit).
...	Additional arguments

Value

A ggplot2 plot

Examples

```
cs_results <- antidepressants |>
  cs_percentage(
    patient,
    measurement,
    pre = "Before",
    mom_di,
    pct_improvement = 0.4
```

```
  )


# Plot the results "as is"
plot(cs_results)


# Change the axis labels
plot(cs_results, x_lab = "Before Intervention", y_lab = "After Intervention")


# Show the individual categories
plot(cs_results, show = category)


# Show a specific category
plot(cs_results, show = improved)


# Show groups as specified in the data
cs_results_grouped <- antidepressants |>
  cs_percentage(
    patient,
    measurement,
    pre = "Before",
    mom_di,
    pct_improvement = 0.4,
    group = condition
  )

plot(cs_results_grouped)


# To avoid overplotting, generic ggplot2 code can be used to facet the plot
library(ggplot2)
plot(cs_results_grouped) +
  facet_wrap(~ group)


# Adjust the transparency of individual data points
plot(cs_results, point_alpha = 0.3)


# Adjust the fill and transparency of the "unchanged" (PCC) region
plot(cs_results, pct_fill = "firebrick", pct_alpha = 0.2)


# Control the overplotting
plot(cs_results, overplotting = 0.1)


# Or adjust the axis limits by hand
plot(cs_results, lower_limit = 0, upper_limit = 80)
```

`plot.cs_statistical` *Plot an Object of Class cs_statistical*

Description

This function creates a generic clinical significance plot by plotting the patients' pre intervention value on the x-axis and the post intervention score on the y-axis.

Usage

```
## S3 method for class 'cs_statistical'
plot(
  x,
  x_lab = "Pre",
  y_lab = "Post",
  color_lab = "Group",
  include_cutoff = TRUE,
  lower_limit,
  upper_limit,
  show,
  point_alpha = 1,
  overplotting = 0.02,
  ...
)
```

Arguments

`x`	An object of class `cs_statistical`
`x_lab`	String, x axis label. Default is `"Pre"`.
`y_lab`	String, x axis label. Default is `"Post"`.
`color_lab`	String, color label (if colors are displayed). Default is `"Group"`
`include_cutoff`	Logical, whether to include the population cutoff. Default is `TRUE`.
`lower_limit`	Numeric, lower plotting limit. Defaults to 2% smaller than minimum instrument score
`upper_limit`	Numeric, upper plotting limit. Defaults to 2% larger than maximum instrument score
`show`	Unquoted category name. You have several options to color different features. Available are • `category` (shows all categories at once) • `clinical_pre` (shows participants with clinical scores pre intervention) • `functional_post` (shows participants with functional scores post intervention) • `unchanged` (shows unchanged participants)

point_alpha	Numeric, transparency adjustment for points. A value between 0 and 1 where 1 corresponds to not transparent at all and 0 to fully transparent.
overplotting	Numeric, control amount of overplotting. Defaults to 0.02 (i.e., 2% of range between lower and upper limit).
...	Additional arguments

Value

A ggplot2 plot

Examples

```
cs_results <- antidepressants |>
  cs_statistical(
    patient,
    measurement,
    pre = "Before",
    mom_di,
    m_functional = 15,
    sd_functional = 8,
    cutoff_type = "c"
  )


# Plot the results "as is"
plot(cs_results)


# Change the axis labels
plot(cs_results, x_lab = "Before Intervention", y_lab = "After Intervention")


# Show the individual categories
plot(cs_results, show = category)


# Show groups as specified in the data
cs_results_grouped <- antidepressants |>
  cs_statistical(
    patient,
    measurement,
    pre = "Before",
    mom_di,
    m_functional = 15,
    sd_functional = 8,
    cutoff_type = "c",
    group = condition
  )

plot(cs_results_grouped)
```

```
# To avoid overplotting, generic ggplot2 code can be used to facet the plot
library(ggplot2)
plot(cs_results_grouped) +
  facet_wrap(~ group)


# Adjust the transparency of individual data points
plot(cs_results, point_alpha = 0.3)


# Control the overplotting
plot(cs_results, overplotting = 0.1)


# Or adjust the axis limits by hand
plot(cs_results, lower_limit = 0, upper_limit = 80)
```

`print.cs_anchor_group_between` *Print Method for the Anchor-Based Approach for Groups (Between)*

Description

Print Method for the Anchor-Based Approach for Groups (Between)

Usage

```
## S3 method for class 'cs_anchor_group_between'
print(x, ...)
```

Arguments

`x`	An object of class `cs_anchor_group_between`
`...`	Additional arguments

Value

No return value, called for side effects

Examples

```
cs_results <- claus_2020 |>
  cs_anchor(
    id,
    time,
    bdi,
    post = 4,
    mid_improvement = 7,
    group = treatment,
```

```
    target = "group",
    effect = "between"
  )

cs_results
```

`print.cs_anchor_group_within` *Print Method for the Anchor-Based Approach for Groups (Within)*

Description

Print Method for the Anchor-Based Approach for Groups (Within)

Usage

```
## S3 method for class 'cs_anchor_group_within'
print(x, ...)
```

Arguments

`x`	An object of class `cs_anchor_group_within`
`...`	Additional arguments

Value

No return value, called for side effects

Examples

```
cs_results <- claus_2020 |>
  cs_anchor(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    mid_improvement = 7,
    target = "group"
  )

cs_results
```

print.cs_anchor_individual_within *Print Method for the Anchor-Based Approach for Individuals*

Description

Print Method for the Anchor-Based Approach for Individuals

Usage

```
## S3 method for class 'cs_anchor_individual_within'
print(x, ...)
```

Arguments

x	An object of class cs_anchor_individual_within
...	Additional arguments

Value

No return value, called for side effects

Examples

```
cs_results <- claus_2020 |>
  cs_distribution(id, time, hamd, pre = 1, post = 4, reliability = 0.8)
cs_results
```

print.cs_combined *Print Method for the Combined Approach*

Description

Print Method for the Combined Approach

Usage

```
## S3 method for class 'cs_combined'
print(x, ...)
```

Arguments

x	An object of class cs_combined
...	Additional arguments

Value

No return value, called for side effects

Examples

```
cs_results <- claus_2020 |>
  cs_combined(id, time, hamd, pre = 1, post = 4, reliability = 0.8)

cs_results
```

`print.cs_distribution` *Print Method for the Distribution-Based Approach*

Description

Print Method for the Distribution-Based Approach

Usage

```
## S3 method for class 'cs_distribution'
print(x, ...)
```

Arguments

`x`	An object of class `cs_distribution`
`...`	Additional arguments

Value

No return value, called for side effects

Examples

```
cs_results <- claus_2020 |>
  cs_distribution(id, time, hamd, pre = 1, post = 4, reliability = 0.8)

cs_results
```

print.cs_percentage *Print Method for the Percentange-Change Approach*

Description

Print Method for the Percentange-Change Approach

Usage

```
## S3 method for class 'cs_percentage'
print(x, ...)
```

Arguments

x	An object of class cs_percentage
...	Additional arguments

Value

No return value, called for side effects

Examples

```
cs_results <- claus_2020 |>
  cs_percentage(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    pct_improvement = 0.5
  )

cs_results
```

print.cs_statistical *Print Method for the Statistical Approach*

Description

Print Method for the Statistical Approach

Usage

```
## S3 method for class 'cs_statistical'
print(x, ...)
```

Arguments

x	An object of class cs_distribution
...	Additional arguments

Value

No return value, called for side effects

Examples

```
cs_results <- claus_2020 |>
  cs_statistical(
    id,
    time,
    hamd,
    pre = 1,
    post = 4,
    m_functional = 8,
    sd_functional = 7
  )

cs_results
```

---

summary.cs_anchor_group_between

Summary Method for the Anchor-Based Approach for Groups (Between)

Description

Summary Method for the Anchor-Based Approach for Groups (Between)

Usage

```
## S3 method for class 'cs_anchor_group_between'
summary(object, ...)
```

Arguments

object	An object of class cs_anchor_group_between
...	Additional arguments

Value

No return value, called for side effects only

Examples

```
cs_results <- antidepressants |>
  cs_anchor(
    patient,
    measurement,
    post = "After",
    mom_di,
    mid_improvement = 8,
    target = "group",
    effect = "between",
    group = condition
  )

summary(cs_results)
```

`summary.cs_anchor_group_within` *Summary Method for the Anchor-Based Approach for Groups (Within)*

Description

Summary Method for the Anchor-Based Approach for Groups (Within)

Usage

```
## S3 method for class 'cs_anchor_group_within'
summary(object, ...)
```

Arguments

`object`	An object of class `cs_anchor_group_within`
`...`	Additional arguments

Value

No return value, called for side effects only

Examples

```
cs_results <- claus_2020 |>
  cs_anchor(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    mid_improvement = 8,
    target = "group"
  )
```

```
summary(cs_results)
```

`summary.cs_anchor_individual_within` *Summary Method for the Anchor-Based Approach*

Description

Summary Method for the Anchor-Based Approach

Usage

```
## S3 method for class 'cs_anchor_individual_within'
summary(object, ...)
```

Arguments

`object`	An object of class `cs_anchor_individual_within`
`...`	Additional arguments

Value

No return value, called for side effects only

Examples

```
cs_results <- claus_2020 |>
  cs_anchor(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    mid_improvement = 7
  )

cs_results
```

summary.cs_combined *Summary Method for the Combined Approach*

Description

Summary Method for the Combined Approach

Usage

```
## S3 method for class 'cs_combined'
summary(object, ...)
```

Arguments

object	An object of class cs_combined
...	Additional arguments

Value

No return value, called for side effects only

Examples

```
cs_results <- claus_2020 |>
  cs_combined(id, time, hamd, pre = 1, post = 4, reliability = 0.8)

summary(cs_results)
```

summary.cs_distribution *Summary Method for the Distribution-Based Approach*

Description

Summary Method for the Distribution-Based Approach

Usage

```
## S3 method for class 'cs_distribution'
summary(object, ...)
```

Arguments

object	An object of class cs_distribution
...	Additional arguments

Value

No return value, called for side effects only

Examples

```
cs_results <- claus_2020 |>
  cs_distribution(id, time, hamd, pre = 1, post = 4, reliability = 0.8)

summary(cs_results)
```

---

summary.cs_percentage *Summary Method for the Percentage-Change Approach*

Description

Summary Method for the Percentage-Change Approach

Usage

```
## S3 method for class 'cs_percentage'
summary(object, ...)
```

Arguments

object	An object of class cs_percentage
...	Additional arguments

Value

No return value, called for side effects only

Examples

```
cs_results <- claus_2020 |>
  cs_percentage(
    id,
    time,
    bdi,
    pre = 1,
    post = 4,
    pct_improvement = 0.5
  )

summary(cs_results)
```

summary.cs_statistical *Summary Method for the Statistical Approach*

Description

Summary Method for the Statistical Approach

Usage

```
## S3 method for class 'cs_statistical'
summary(object, ...)
```

Arguments

object	An object of class cs_distribution
...	Additional arguments

Value

No return value, called for side effects only

Examples

```
cs_results <- claus_2020 |>
  cs_statistical(
    id,
    time,
    hamd,
    pre = 1,
    post = 4,
    m_functional = 8,
    sd_functional = 7
  )

summary(cs_results)
```

trackmania *Fictional "Trackmania" Racing Data for Interrupted Time-Series*

Description

A simulated dataset for an interrupted time-series (ITS) design. The data follows two groups of race car drivers ("Training" and "Control") over 10 lap attempts. An intervention occurs after the 5th attempt, designed to affect both the immediate performance (jump) and the subsequent learning rate (slope change) of the Training group.

Usage

```
trackmania
```

Format

A tibble with 400 rows and 4 variables:

driver_id An integer representing the unique identifier for each driver.

group A character vector indicating the experimental group (`"Training"` or `"Control"`).

attempt An integer representing the lap attempt number (from 1 to 10).

lap_time_sec A numeric value for the outcome, the lap time in seconds.

Source

Simulated data from a time-series model including fixed effects for a pre-intervention slope, a post-intervention jump, and a post-intervention slope change, plus random effects for driver talent (intercept) and learning rate (slope).

Index

∗ **datasets**
  antidepressants, 3
  anxiety, 4
  anxiety_complete, 4
  claus_2020, 5
  columbo, 6
  garden_club, 37
  garden_club_sensitive, 38
  hechler_2014, 39
  jacobson_1989, 40
  trackmania, 65
∗ **get**
  cs_get_augmented_data, 20
  cs_get_data, 24
  cs_get_model, 25
  cs_get_n, 26
  cs_get_reliability, 28
  cs_get_summary, 29
∗ **main**
  cs_anchor, 7
  cs_combined, 12
  cs_distribution, 16
  cs_percentage, 30
  cs_statistical, 34

antidepressants, 3
anxiety, 4
anxiety_complete, 4

BayesFactor::ttestBF(), *8*

claus_2020, 5
columbo, 6
cs_anchor, 7, *15*, *19*, *32*, *36*
cs_combined, *10*, 12, *19*, *32*, *36*
cs_distribution, *10*, *15*, 16, *32*, *36*
cs_get_augmented_data, 20, *25, 26*, *28, 29*
cs_get_cutoff, 23
cs_get_cutoff_descriptives, 24
cs_get_data, *21*, 24, *25, 26*, *28, 29*
cs_get_model, *21*, 25, *25, 26*, *28, 29*
cs_get_n, *21*, *25*, 26, *28, 29*
cs_get_reliability, *21*, *25, 26*, 28, *29*
cs_get_summary, *21*, *25, 26*, *28*, 29
cs_percentage, *10*, *15*, *19*, 30, *36*
cs_statistical, *10*, *15*, *19*, *32*, 34

garden_club, 37, *39*
garden_club_sensitive, 38, *38*

hechler_2014, 39

jacobson_1989, 40

plot.cs_anchor_group_between, 40
plot.cs_anchor_group_within, 41
plot.cs_anchor_individual_within, 43
plot.cs_combined, 45
plot.cs_distribution, 48
plot.cs_percentage, 50
plot.cs_statistical, 53
print.cs_anchor_group_between, 55
print.cs_anchor_group_within, 56
print.cs_anchor_individual_within, 57
print.cs_combined, 57
print.cs_distribution, 58
print.cs_percentage, 59
print.cs_statistical, 59

summary.cs_anchor_group_between, 60
summary.cs_anchor_group_within, 61
summary.cs_anchor_individual_within, 62
summary.cs_combined, 63
summary.cs_distribution, 63
summary.cs_percentage, 64
summary.cs_statistical, 65

trackmania, 65