Package ‘SimplyAgree’

February 24, 2025

Type Package

Title Flexible and Robust Agreement and Reliability Analyses

Version 0.2.1

Maintainer Aaron Caldwell <arcaldwell49@gmail.com>

Description Reliability and agreement analyses often have limited software support. Therefore, this package was created to make agreement and reliability analyses easier for the average researcher. The functions within this package include simple tests of agreement, agreement analysis for nested and replicate data, and provide robust analyses of reliability. In addition, this package contains a set of functions to help when planning studies looking to assess measurement agreement.

URL https://aaroncaldwell.us/SimplyAgree/

BugReports https://github.com/arcaldwell49/SimplyAgree/issues

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports ggplot2, emmeans, lme4, boot, stats, dplyr, magrittr, tidyselect, tidyr, stringr, jmvcore, quantreg, patchwork, insight, nlme, purrr, Matrix, MASS, lifecycle

Suggests knitr, rmarkdown, testthat, readr, covr, mgcv, ggeffects, deming, pbkrtest

VignetteBuilder knitr

Depends R (>= 3.6)

NeedsCompilation no

Author Aaron Caldwell [aut, cre]

Repository CRAN

Date/Publication 2025-02-24 16:40:11 UTC

Contents

`agreement_limit` *Limits of Agreement*

Description

[Maturing]

A function for calculating for Bland-Altman limits of agreement based on the difference between two measurements (difference = x-y). Please note that the package developer recommends reporting/using tolerance limits (see `"tolerance_limit"` function).

Usage

```
agreement_limit(
  x,
  y,
  id = NULL,
  data,
  data_type = c(”simple”, ”nest”, ”reps”),
  loa_calc = c(”mover”, ”blandaltman”),
  agree.level = 0.95,
  alpha = 0.05,
  prop_bias = FALSE,
  log_tf = FALSE,
  log_tf_display = c(”ratio”, ”sympercent”),
  lmer_df = c(”satterthwaite”, ”asymptotic”),
  lmer_limit = 3000
)
```

Arguments

`x` Name of column with first measurement

`y` Name of other column with the other measurement to compare to the first.

`id` Column with subject identifier. Default is "id" if no entry is provided.

`data` Data frame with all data.

`data_type` The type of data structure. Options include "simple" (all independent data points), "nest" (nested data) and "reps" (replicated data points).

`loa_calc` The method by which the limits of agreement confidence intervals are calculated. Options are "mover" (Methods of Recovering Variances method) or "blandlatman" (Bland-Altman method).

`agree.level` the agreement level required. Default is 95%. The proportion of data that should lie between the thresholds, for 95% limits of agreement this should be 0.95.

`alpha` The alpha-level for confidence levels.

`prop_bias` Logical indicator (TRUE/FALSE) of whether proportional bias should be considered for the limits of agreement calculations.

`log_tf` Calculate limits of agreement using log-transformed data.

`log_tf_display` The type of presentation for log-transformed results. The differences between methods can be displayed as a "ratio" or "sympercent".

`lmer_df` Degrees of freedom method, only matters for if data_type is "nest". Default is "satterthwaite". The "asymptotic" method is faster but more liberal.

`lmer_limit` Sample size limit for degrees of freedom method. If number of observations exceeds this limit, then the "asymptotic" method is utilized.

Details

The limits of agreement (LoA) are calculated in this function are based on the method originally detailed by Bland & Atlman (1986 & 1999). The `loa_calc` allow users to specify the calculative method for the LoA which can be based on Bland-Altman (1999) (`loa_calc = ”blandaltman”`), or by the more accurate MOVER method of Zou (2013) and Donner & Zou (2012) (`loa_calc = ”mover”`).

Value

Returns single loa class object with the results of the agreement analysis.

- `loa`: A data frame containing the Limits of Agreement.
- `call`:The matched call.

References

MOVER methods:

Zou, G. Y. (2013). Confidence interval estimation for the Bland–Altman limits of agreement with multiple observations per individual. Statistical methods in medical research, 22(6), 630-642.

Donner, A., & Zou, G. Y. (2012). Closed-form confidence intervals for functions of the normal mean and standard deviation. Statistical Methods in Medical Research, 21(4), 347-359.

Bland & Altman methods:

Bland, J. M., & Altman, D. (1986). Statistical methods for assessing agreement between two methods of clinical measurement. The Lancet, 327(8476), 307-310.

Bland, J. M., & Altman, D. (1999). Measuring agreement in method comparison studies. Statistical methods in medical research, 8(2), 135-160.

Bland, J. M., & Altman, D. G. (1996). Statistics notes: measurement error proportional to the mean. BMJ, 313(7049), 106.

Examples

```
data('reps')

# Simple
agreement_limit(x = "x", y ="y", data = reps)

# Replicates
agreement_limit(x = "x", y ="y", data = reps, id = "id", data_type = "rep")

# Nested
agreement_limit(x = "x", y ="y", data = reps, id = "id", data_type = "nest")
```

`agree_coef` *Agreement Coefficents*

Description

[Maturing]

agree_coef produces inter-rater reliability or "agreement coefficients" as described by Gwet.

Usage

```
agree_coef(
  wide = TRUE,
  col.names = NULL,
  measure,
  item,
  id,
  data,
  weighted = FALSE,
  conf.level = 0.95
)
```

Arguments

`wide` Logical value (TRUE or FALSE) indicating if data is in a "wide" format. Default is TRUE.

`col.names` If wide is equal to TRUE then col.names is a list of the column names containing the measurements for reliability analysis.

`measure` Name of column containing the measurement of interest.

`item` Name of column containing the items. If this is an inter-rater reliability study then this would indicate the rater (e.g., rater1, rater2, rater3, etc).

`id` Column with subject identifier.

`data` Data frame with all data.

`weighted` Logical value (TRUE or FALSE) indicating whether to weight the responses. If TRUE (default is FALSE) then quadratic weights are utilized. This option should be set to TRUE for ordinal or continuous responses.

`conf.level` the confidence level required. Default is 95%.

Value

Returns single data frame of inter-rater reliability coefficients.

References

Gwet, K.L. (2014, ISBN:978-0970806284). “Handbook of Inter-Rater Reliability,” 4th Edition. Advanced Analytics, LLC. Gwet, K. L. (2008). “Computing inter-rater reliability and its variance in the presence of high agreement," British Journal of Mathematical and Statistical Psychology, 61, 29-48.

Examples

```
data('reps')
agree_coef(data = reps, wide = TRUE, col.names = c("x","y"), weighted = TRUE)
```

agree_nest *Tests for Absolute Agreement with Nested Data*

Description

[Superseded]

Development on agree_nest() is complete, and for new code we recommend switching to agreement_limit(), which is easier to use, has more features, and still under active development.

agree_nest produces an absolute agreement analysis for data where there is multiple observations per subject but the mean varies within subjects as described by Zou (2013). Output mirrors that of agree_test but CCC is calculated via U-statistics.

Usage

```
agree_nest(
  x,
  y,
  id,
  data,
  delta,
  agree.level = 0.95,
  conf.level = 0.95,
  TOST = TRUE,
  prop_bias = FALSE,
  ccc = TRUE
)
```

Arguments

x	Name of column with first measurement
y	Name of other column with the other measurement to compare to the first.
id	Column with subject identifier
data	Data frame with all data
delta	The threshold below which methods agree/can be considered equivalent, can be in any units. Equivalence Bound for Agreement.
agree.level	the agreement level required. Default is 95%. The proportion of data that should lie between the thresholds, for 95% limits of agreement this should be 0.95.
conf.level	the confidence level required. Default is 95%.
TOST	Logical indicator (TRUE/FALSE) of whether to use two one-tailed tests for the limits of agreement. Default is TRUE.
prop_bias	Logical indicator (TRUE/FALSE) of whether proportional bias should be considered for the limits of agreement calculations.
ccc	Calculate concordance correlation coefficient.

Value

Returns single simple_agree class object with the results of the agreement analysis.

- `loa`: A data frame of the limits of agreement including the average difference between the two sets of measurements, the standard deviation of the difference between the two sets of measurements and the lower and upper confidence limits of the difference between the two sets of measurements.
- `h0_test`: Decision from hypothesis test.
- `ccc.xy`: Lin's concordance correlation coefficient and confidence intervals using U-statistics. Warning: if underlying value varies this estimate will be inaccurate.
- `call`: the matched call.
- `var_comp`: Table of Variance Components.
- `class`: The type of simple_agree analysis.

References

Zou, G. Y. (2013). Confidence interval estimation for the Bland–Altman limits of agreement with multiple observations per individual. Statistical methods in medical research, 22(6), 630-642.

King, TS and Chinchilli, VM. (2001). A generalized concordance correlation coefficient for continuous and categorical data. Statistics in Medicine, 20, 2131:2147.

King, TS; Chinchilli, VM; Carrasco, JL. (2007). A repeated measures concordance correlation coefficient. Statistics in Medicine, 26, 3095:3113.

Carrasco, JL; Phillips, BR; Puig-Martinez, J; King, TS; Chinchilli, VM. (2013). Estimation of the concordance correlation coefficient for repeated measures using SAS and R. Computer Methods and Programs in Biomedicine, 109, 293-304.

Examples

```
data('reps')
agree_nest(x = "x", y = "y", id = "id", data = reps, delta = 2)
```

`agree_np` *Nonparametric Test for Limits of Agreement*

Description

[Stable]

`agree_np` A non-parametric approach to limits of agreement. The hypothesis test is based on binomial proportions within the maximal allowable differences, and the limits are calculated with quantile regression.

Usage

```
agree_np(
  x,
  y,
  id = NULL,
  data,
  delta = NULL,
  prop_bias = FALSE,
  TOST = TRUE,
  agree.level = 0.95,
  conf.level = 0.95
)
```

Arguments

`x` Name of column with first measurement.

`y` Name of other column with the other measurement to compare to the first.

`id` Column with subject identifier with samples are taken in replicates.

`data` Data frame with all data.

`delta` The threshold below which methods agree/can be considered equivalent and this argument is required. Equivalence Bound for Agreement or Maximal Allowable Difference.

`prop_bias` Logical indicator (TRUE/FALSE) of whether proportional bias should be considered for the limits of agreement calculations.

`TOST` Logical indicator (TRUE/FALSE) of whether to use two one-tailed tests for the limits of agreement. Default is TRUE.

`agree.level` the agreement level required. Default is 95%. The proportion of data that should lie between the thresholds, for 95% limits of agreement this should be 0.95.

`conf.level` the confidence level required. Default is 95%.

Value

Returns simple_agree object with the results of the agreement analysis.

- `loa`: A data frame of the limits of agreement.
- `agree`: A data frame of the binomial proportion of results in agreement.
- `h0_test`: Decision from hypothesis test.
- `qr_mod`: The quantile regression model.
- `call`: The matched call

References

Bland, J. M., & Altman, D. G. (1999). Measuring agreement in method comparison studies. In Statistical Methods in Medical Research (Vol. 8, Issue 2, pp. 135–160). SAGE Publications. doi:10.1177/096228029900800204

Examples

```
data('reps')
agree_np(x = "x", y = "y", id = "id", data = reps, delta = 2)
```

agree_reps *Tests for Absolute Agreement with Replicates*

Description

[Superseded]

Development on `agree_reps()` is complete, and for new code we recommend switching to `agreement_limit()`, which is easier to use, has more features, and still under active development.

agree_nest produces an absolute agreement analysis for data where there is multiple observations per subject but the mean does not vary within subjects as described by Zou (2013). Output mirrors that of agree_test but CCC is calculated via U-statistics.

Usage

```
agree_reps(
  x,
  y,
  id,
  data,
  delta,
  agree.level = 0.95,
  conf.level = 0.95,
  prop_bias = FALSE,
  TOST = TRUE,
  ccc = TRUE
)
```

Arguments

`x` Name of column with first measurement

`y` Name of other column with the other measurement to compare to the first.

`id` Column with subject identifier

`data` Data frame with all data

`delta` The threshold below which methods agree/can be considered equivalent, can be in any units. Equivalence Bound for Agreement.

`agree.level` the agreement level required. Default is 95%. The proportion of data that should lie between the thresholds, for 95% limits of agreement this should be 0.95.

`conf.level` the confidence level required. Default is 95%.

`prop_bias` Logical indicator (TRUE/FALSE) of whether proportional bias should be considered for the limits of agreement calculations.

`TOST` Logical indicator (TRUE/FALSE) of whether to use two one-tailed tests for the limits of agreement. Default is TRUE.

`ccc` Calculate concordance correlation coefficient.

Value

Returns single list with the results of the agreement analysis.

- `loa`: a data frame of the limits of agreement including the average difference between the two sets of measurements, the standard deviation of the difference between the two sets of measurements and the lower and upper confidence limits of the difference between the two sets of measurements.
- `h0_test`: Decision from hypothesis test.
- `ccc.xy`: Lin's concordance correlation coefficient and confidence intervals using U-statistics.
- `call`: The matched call.
- `var_comp`: Table of Variance Components.
- `class`: The type of simple_agree analysis.

References

Zou, G. Y. (2013). Confidence interval estimation for the Bland–Altman limits of agreement with multiple observations per individual. Statistical methods in medical research, 22(6), 630-642.

King, TS and Chinchilli, VM. (2001). A generalized concordance correlation coefficient for continuous and categorical data. Statistics in Medicine, 20, 2131:2147.

King, TS; Chinchilli, VM; Carrasco, JL. (2007). A repeated measures concordance correlation coefficient. Statistics in Medicine, 26, 3095:3113.

Carrasco, JL; Phillips, BR; Puig-Martinez, J; King, TS; Chinchilli, VM. (2013). Estimation of the concordance correlation coefficient for repeated measures using SAS and R. Computer Methods and Programs in Biomedicine, 109, 293-304.

Examples

```
data('reps')
agree_reps(x = "x", y = "y", id = "id", data = reps, delta = 2)
```

`agree_test` *Tests for Absolute Agreement*

Description

[Superseded]

Development on `agree_test()` is complete, and for new code we recommend switching to `agreement_limit()`, which is easier to use, has more features, and still under active development.

The agree_test function calculates a variety of agreement statistics. The hypothesis test of agreement is calculated by the method described by Shieh (2019). Bland-Altman limits of agreement, and confidence intervals, are also provided (Bland & Altman 1999; Bland & Altman 1986). In addition, the concordance correlation coefficient (CCC; Lin 1989) is additional part of the output.

Usage

```
agree_test(
  x,
  y,
  delta,
  conf.level = 0.95,
  agree.level = 0.95,
  TOST = TRUE,
  prop_bias = FALSE
)
```

Arguments

`x` Vector with first measurement

`y` Vector with second measurement

`delta` The threshold below which methods agree/can be considered equivalent, can be in any units. Often referred to as the "Equivalence Bound for Agreement" or "Maximal Allowable Difference".

`conf.level` the confidence level required. Default is 95%.

`agree.level` the agreement level required. Default is 95%. The proportion of data that should lie between the thresholds, for 95% limits of agreement this should be 0.95.

`TOST` Logical indicator (TRUE/FALSE) of whether to use two one-tailed tests for the limits of agreement. Default is TRUE.

`prop_bias` Logical indicator (TRUE/FALSE) of whether proportional bias should be considered for the limits of agreement calculations.

Value

Returns single list with the results of the agreement analysis.

- `shieh_test`: The TOST hypothesis test as described by Shieh.
- `ccc.xy`: Lin's concordance correlation coefficient and confidence intervals.
- `s.shift`: Scale shift from x to y.
- `l.shift`: Location shift from x to y.
- `bias`: a bias correction factor that measures how far the best-fit line deviates from a line at 45 degrees. No deviation from the 45 degree line occurs when bias = 1. See Lin 1989, page 258.
- `loa`: Data frame containing the limits of agreement calculations
- `h0_test`: Decision from hypothesis test.
- `call`: the matched call

References

Shieh (2019). Assessing Agreement Between Two Methods of Quantitative Measurements: Exact Test Procedure and Sample Size Calculation, Statistics in Biopharmaceutical Research, doi:10.1080/19466315.2019.1677495

Bland, J. M., & Altman, D. G. (1999). Measuring agreement in method comparison studies. Statistical methods in medical research, 8(2), 135-160.

Bland, J. M., & Altman, D. (1986). Statistical methods for assessing agreement between two methods of clinical measurement. The lancet, 327(8476), 307-310.

Lawrence, I., & Lin, K. (1989). A concordance correlation coefficient to evaluate reproducibility. Biometrics, 255-268.

Examples

```
data('reps')
agree_test(x=reps$x, y=reps$y, delta = 2)
```

`blandPowerCurve` *Power Curve for Bland-Altman Limits of Agreement*

Description

[Maturing] This function calculates the power for the Bland-Altman method under varying parameter settings and for a range of sample sizes.

Usage

```
blandPowerCurve(
  samplesizes = seq(10, 100, 1),
  mu = 0,
  SD,
  delta,
  conf.level = 0.95,
  agree.level = 0.95
)
```

Arguments

`samplesizes` vector of samples sizes at which to estimate power.

`mu` mean of differences

`SD` standard deviation of differences

`delta` The threshold below which methods agree/can be considered equivalent, can be in any units. Equivalence Bound for Agreement. More than one delta can be provided.

`conf.level` the confidence level(s) required. Default is 95%. More than one confidence level can be provided.

`agree.level` the agreement level(s) required. Default is 95%. The proportion of data that should lie between the thresholds, for 95% limits of agreement this should be 0.95. More than one confidence level can be provided.

Value

A dataframe is returned containing the power analysis results. The results can then be plotted with the plot.powerCurve function.

References

Lu, M. J., et al. (2016). Sample Size for Assessing Agreement between Two Methods of Measurement by Bland-Altman Method. The international journal of biostatistics, 12(2), doi:10.1515/ijb20150039

Examples

```
powerCurve <- blandPowerCurve(samplesizes = seq(10, 200, 1),
mu = 0,
SD = 3.3,
delta = 8,
conf.level = .95,
agree.level = .95)
# Plot the power curve
plot(powerCurve, type = 1)
# Find at what N power of .8 is achieved
find_n(powerCurve, power = .8)

# If the desired power is not found then
## Sample size range must be expanded
```

---

dem_reg *Deming Regression*

Description

[Stable]

A function for fitting a straight line to two-dimensional data (i.e., X and Y) that are measured with error.

Usage

```
dem_reg(
  x,
  y,
  id = NULL,
  data,
  conf.level = 0.95,
  weighted = FALSE,
  weights = NULL,
  error.ratio = 1,
  keep_data = FALSE
)
```

Arguments

`x` Name of column with first measurement.

`y` Name of other column with the other measurement to compare to the first.

`id` Column with subject identifier.

`data` Data frame with all data.

`conf.level` The confidence level required. Default is 95%.

`weighted` Logical indicator (TRUE/FALSE) for whether to use weighted Deming regression. Default is FALSE.

`weights` an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector.

`error.ratio` Ratio of the two error variances. Default is 1. This argument is ignored if subject identifiers are provided.

`keep_data` Logical indicator (TRUE/FALSE). If TRUE, the jacknife samples are returned; default is FALSE. Users may wish to set to FALSE if data is especially large.

Details

This function provides a Deming regression analysis wherein the sum of distances in both x and y direction is minimized. Deming regression, also known as error-in-variable regression, is useful in situations where both X & Y are measured with error. The use of Deming regression is beneficial when comparing to methods for measuring the same continuous variable.

Currently, the dem_reg function covers simple Deming regression and weighted Deming regression. Weighted Deming regression can be used by setting the weighted argument to TRUE. The weights can be provided by the user or can be calculated within function.

If the data are measured in replicates, then the measurement error can be directly derived from the data. This can be accomplished by indicating the subject identifier with the id argument. When the replicates are not available in the data, then the ratio of error variances (y/x) can be provided with the error.ratio argument.

Value

The function returns a simple_eiv (eiv meaning "error in variables") object.

- `call`: The matched call.
- `model`: Data frame presenting the results from the Deming regression analysis.
- `resamples`: List containing resamples from jacknife procedure.

References

Linnet, K. (1990) Estimation of the linear relationship between the measurements of two methods with proportional errors. Statistics in Medicine, 9, 1463-1473.

Linnet, K. (1993). Evaluation of regression procedures for methods comparison studies. Clinical chemistry, 39, 424-432.

jmvagree *Simple Agreement Analysis*

Description

Simple Agreement Analysis

Usage

```
jmvagree(
  data,
  method1,
  method2,
  ciWidth = 95,
  agreeWidth = 95,
  testValue = 2,
  CCC = TRUE,
  plotbland = TRUE,
  plotcon = FALSE,
  plotcheck = FALSE,
  prop_bias = FALSE,
  xlabel = "Average of Both Methods",
  ylabel = "Difference between Methods"
)
```

Arguments

`data`	Data
`method1`	Name of column containing 1st Vector of data
`method2`	Name of column containing Vector of data
`ciWidth`	a number between 50 and 99.9 (default: 95), the width of confidence intervals
`agreeWidth`	a number between 50 and 99.9 (default: 95), the width of agreement limits
`testValue`	a number specifying the limit of agreement
`CCC`	`TRUE` or `FALSE` (default), produce CCC table
`plotbland`	`TRUE` or `FALSE` (default), for Bland-Altman plot
`plotcon`	`TRUE` or `FALSE` (default), for Bland-Altman plot
`plotcheck`	`TRUE` or `FALSE` (default), assumptions plots
`prop_bias`	`TRUE` or `FALSE`
`xlabel`	The label for the x-axis on the BA plot
`ylabel`	The label for the y-axis on the BA plot

Value

A results object containing:

`results$text`	a html
`results$blandtab`	a table
`results$ccctab`	a table
`results$plotba`	an image
`results$plotcon`	an image
`results$plotcheck`	an image

Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

`results$blandtab$asDF`

`as.data.frame(results$blandtab)`

`jmvagreemulti` *Nested/Replicate Data Agreement Analysis*

Description

Nested/Replicate Data Agreement Analysis

Usage

```
jmvagreemulti(
  data,
  method1,
  method2,
  id,
  ciWidth = 95,
  agreeWidth = 95,
  testValue = 2,
  CCC = TRUE,
  valEq = FALSE,
  plotbland = FALSE,
  plotcon = FALSE,
  prop_bias = FALSE,
  xlabel = "Average of Both Methods",
  ylabel = "Difference between Methods"
)
```

Arguments

`data`	Data
`method1`	Name of column containing 1st Vector of data
`method2`	Name of column containing Vector of data
`id`	Name of column containing subject identifier
`ciWidth`	a number between 50 and 99.9 (default: 95), the width of confidence intervals
`agreeWidth`	a number between 50 and 99.9 (default: 95), the width of agreement limits
`testValue`	a number specifying the limit of agreement
`CCC`	`TRUE` or `FALSE` (default), produce CCC table
`valEq`	.
`plotbland`	`TRUE` or `FALSE` (default), for Bland-Altman plot
`plotcon`	`TRUE` or `FALSE` (default), for Line of identity plot
`prop_bias`	`TRUE` or `FALSE`
`xlabel`	The label for the x-axis on the BA plot
`ylabel`	The label for the y-axis on the BA plot

Value

A results object containing:

`results$text`	a preformatted
`results$blandtab`	a table
`results$ccctab`	a table
`results$plotba`	an image
`results$plotcon`	an image

Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

`results$blandtab$asDF`

`as.data.frame(results$blandtab)`

`jmvdeming` *Deming Regression*

Description

Deming Regression

Usage

```
jmvdeming(
  data,
  method1,
  method2,
  ciWidth = 95,
  testValue = 1,
  plotcon = FALSE,
  plotcheck = FALSE,
  weighted = FALSE,
  xlabel = "Method: 1",
  ylabel = "Method: 2"
)
```

Arguments

data	Data
method1	Name of column containing 1st Vector of data
method2	Name of column containing Vector of data
ciWidth	a number between 50 and 99.9 (default: 95), the width of confidence intervals
testValue	Ratio of the two error variances. Default is 1.
plotcon	TRUE or FALSE (default), for Bland-Altman plot
plotcheck	TRUE or FALSE (default), assumptions plots
weighted	TRUE or FALSE
xlabel	The label for the x-axis
ylabel	The label for the y-axis

Value

A results object containing:

results$text	a html
results$demtab	a table
results$plotcon	an image
results$plotcheck	an image

Tables can be converted to data frames with asDF or as.data.frame. For example:

results$demtab$asDF

as.data.frame(results$demtab)

jmvreli *Reliability Analysis*

Description

Reliability Analysis

Usage

```
jmvreli(data, vars, ciWidth = 95, desc = FALSE, plots = FALSE)
```

Arguments

`data`	the data as a data frame
`vars`	a list of the column names containing the measurements for reliability analysis.
`ciWidth`	a number between 50 and 99.9 (default: 95), the width of confidence intervals
`desc`	`TRUE` or `FALSE` (default), provide table of variance components
`plots`	`TRUE` or `FALSE` (default), plot data

Value

A results object containing:

`results$text`	a html
`results$icctab`	a table
`results$vartab`	a table
`results$plots`	an image

Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

`results$icctab$asDF`

`as.data.frame(results$icctab)`

loa-methods *Methods for loa objects*

Description

Methods defined for objects returned from the agreement_limit function.

Usage

```
## S3 method for class 'loa'
print(x, digits = 4, ...)

## S3 method for class 'loa'
plot(
  x,
  geom = c("geom_point", "geom_bin2d", "geom_density_2d", "geom_density_2d_filled",
    "stat_density_2d"),
  delta = NULL,
  ...
)

## S3 method for class 'loa'
check(x)
```

Arguments

`x`	object of class `loa` as returned from a agreement_limit function.
`digits`	The number of digits to print.
`...`	further arguments passed through, see description of return value for details. `agreement_limit`.
`geom`	String naming the type of geometry to display the data points. Default is "geom_point". Other options include: "geom_bin2d", "geom_density_2d", "geom_density_2d_filled", and "stat_density_2d".
`delta`	The maximal allowable difference.

Value

`print` Prints short summary of the Limits of Agreement.

`plot` Returns a plot of the limits of agreement.

`check` Returns plots testing the assumptions of a Bland-Altman analysis. P-values for the normality and heteroskedascity tests are provided as captions to the plot.

`loa_lme` *Limits of Agreement with Linear Mixed Effects*

Description

[Stable]

This function allows for the calculation of (parametric) bootstrapped limits of agreement when there are multiple observations per subject. The package author recommends using `tolerance_limit` as an alternative to this function.

Usage

```
loa_lme(
  diff,
  avg,
  condition = NULL,
  id,
  data,
  type = c("perc", "norm", "basic"),
  conf.level = 0.95,
  agree.level = 0.95,
  replicates = 999,
  prop_bias = FALSE,
  het_var = FALSE
)
```

Arguments

`diff`	Column name of the data frame that includes the difference between the 2 measurements of interest.
`avg`	Column name of the data frame that includes the average of the 2 measurements of interest.
`condition`	Column name indicating different conditions subjects were tested under. This can be left missing if there are no differing conditions to be tested.
`id`	Column name indicating the subject/participant identifier
`data`	A data frame containing the variables within the model.
`type`	A character string representing the type of bootstrap confidence intervals. Only "norm", "basic", and "perc" currently supported. Bias-corrected and accelerated, bca, is the default. See ?boot::boot.ci for more details.
`conf.level`	The confidence level required. Default is 95%.
`agree.level`	The agreement level required. Default is 95%.
`replicates`	The number of bootstrap replicates. Passed on to the boot function. Default is 999.
`prop_bias`	Logical indicator (default is FALSE) of whether proportional bias should be considered for the limits of agreement calculations.
`het_var`	Logical indicator (default is FALSE) of whether to assume homogeneity of variance in each condition.

Value

Returns single list with the results of the agreement analysis.

- `var_comp`: Table of variance components
- `loa`: A data frame of the limits of agreement including the average difference between the two sets of measurements, the standard deviation of the difference between the two sets of measurements and the lower and upper confidence limits of the difference between the two sets of measurements.
- `call`: The matched call.

References

Parker, R. A., Weir, C. J., Rubio, N., Rabinovich, R., Pinnock, H., Hanley, J., McLoughan, L., Drost, E.M., Mantoani, L.C., MacNee, W., & McKinstry, B. (2016). "Application of mixed effects limits of agreement in the presence of multiple sources of variability: exemplar from the comparison of several devices to measure respiratory rate in COPD patients". PLOS One, 11(12), e0168321. doi:10.1371/journal.pone.0168321

`loa_mermod-methods` *Methods for loa_mermod objects*

Description

Methods defined for objects returned from the loa_lme.

Usage

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

## S3 method for class 'loa_mermod'
plot(
  x,
  x_label = "Average of Both Methods",
  y_label = "Difference Between Methods",
  geom = "geom_point",
  smooth_method = NULL,
  smooth_se = TRUE,
  ...
)

## S3 method for class 'loa_mermod'
check(x)
```

Arguments

`x`	object of class `loa_mermod`.
`...`	further arguments passed through, see description of return value for details. `loa_mixed`.
`x_label`	Label for x-axis.
`y_label`	Label for y-axis.
`geom`	String naming the type of geometry to display the data points. Default is "geom_point". Other options include: "geom_bin2d", "geom_density_2d", "geom_density_2d_filled", and "stat_density_2d".
`smooth_method`	Smoothing method (function) to use, accepts either NULL or a character vector, e.g. "lm", "glm", "gam", "loess" or a function. Default is NULL, which will not include a trend line.
`smooth_se`	Display confidence interval around smooth?

Value

`print` Prints short summary of the Limits of Agreement

`plot` Returns a plot of the limits of agreement

loa_mixed *Mixed Effects Limits of Agreement*

Description

[Deprecated]

`loa_mixed()` is outdated, and for new code we recommend switching to `loa_lme()` or `tolerance_limit`, which are easier to use, have more features, and are still under active development.

This function allows for the calculation of bootstrapped limits of agreement when there are multiple observations per subject.

Usage

```
loa_mixed(
  diff,
  condition,
  id,
  data,
  plot.xaxis = NULL,
  delta,
  conf.level = 0.95,
  agree.level = 0.95,
  replicates = 1999,
  type = "bca"
)
```

Arguments

`diff`	column name of the data frame that includes the continuous measurement of interest.
`condition`	column name indicating different conditions subjects were tested under.
`id`	column name indicating the subject/participant identifier
`data`	A data frame containing the variables within the model.
`plot.xaxis`	column name indicating what to plot on the x.axis for the Bland-Altman plots. If this argument is missing or set to NULL then no plot will be produced.
`delta`	The threshold below which methods agree/can be considered equivalent, can be in any units. Equivalence Bound for Agreement.
`conf.level`	the confidence level required. Default is 95%.
`agree.level`	the agreement level required. Default is 95%.

`replicates` the number of bootstrap replicates. Passed on to the boot function. Default is 1999.

`type` A character string representing the type of bootstrap confidence intervals. Only "norm", "basic", "bca", and "perc" currently supported. Bias-corrected and accelerated, bca, is the default. See ?boot::boot.ci for more details.

Value

Returns single list with the results of the agreement analysis.

- `var_comp`: Table of variance components
- `loa`: a data frame of the limits of agreement including the average difference between the two sets of measurements, the standard deviation of the difference between the two sets of measurements and the lower and upper confidence limits of the difference between the two sets of measurements.
- `h0_test`: Decision from hypothesis test.
- `bland_alt.plot`: Simple Bland-Altman plot. Red line are the upper and lower bounds for shieh test; grey box is the acceptable limits (delta). If the red lines are within the grey box then the shieh test should indicate ’reject h0’, or to reject the null hypothesis that this not acceptable agreement between x & y.
- `conf.level`: Returned as input.
- `agree.level`: Returned as input.

References

Parker, R. A., Weir, C. J., Rubio, N., Rabinovich, R., Pinnock, H., Hanley, J., McLoughan, L., Drost, E.M., Mantoani, L.C., MacNee, W., & McKinstry, B. (2016). "Application of mixed effects limits of agreement in the presence of multiple sources of variability: exemplar from the comparison of several devices to measure respiratory rate in COPD patients". Plos One, 11(12), e0168321. doi:10.1371/journal.pone.0168321

---

loa_mixed_bs-methods *Methods for loa_mixed_bs objects*

Description

Methods defined for objects returned from the loa_mixed functions.

Usage

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

## S3 method for class 'loa_mixed_bs'
plot(x, ...)
```

Arguments

`x`	object of class `loa_mixed_bs` as returned from `loa_mixed`
`...`	further arguments passed through, see description of return value for details. `loa_mixed`.

Value

`print` Prints short summary of the Limits of Agreement

`plot` Returns a plot of the limits of agreement

`powerCurve-methods` *Methods for powerCurve objects*

Description

Methods defined for objects returned from the powerCurve function.

Usage

```
find_n(x, power = 0.8)

## S3 method for class 'powerCurve'
plot(x, ...)
```

Arguments

`x`	object of class `powerCurve`
`power`	Level of power (value between 0 and 1) for find_n to find the sample size.
`...`	further arguments passed through, see description of return value for details. `blandPowerCurve`.

Value

`plot` Returns a plot of the limits of agreement (type = 1) or concordance plot (type = 2)

`find_n` Find sample size at which desired power is achieved

`reli_stats` *Reliability Statistics*

Description

[Stable]

The reli_stats and reli_aov functions produce reliability statistics described by Weir (2005). This includes intraclass correlation coefficients, the coefficient of variation, and the standard MSE of measurement.

Usage

```
reli_stats(
  measure,
  item,
  id,
  data,
  wide = FALSE,
  col.names = NULL,
  se_type = c("MSE", "ICC1", "ICC2", "ICC3", "ICC1k", "ICC2k", "ICC3k"),
  cv_calc = c("MSE", "residuals", "SEM"),
  conf.level = 0.95,
  other_ci = FALSE,
  type = c("chisq", "perc", "norm", "basic"),
  replicates = 1999
)

reli_aov(
  measure,
  item,
  id,
  data,
  wide = FALSE,
  col.names = NULL,
  se_type = c("MSE", "ICC1", "ICC2", "ICC3", "ICC1k", "ICC2k", "ICC3k"),
  cv_calc = c("MSE", "residuals", "SEM"),
  conf.level = 0.95,
  other_ci = FALSE,
  type = c("chisq", "perc", "norm", "basic"),
  replicates = 1999
)
```

Arguments

`measure`	Name of column containing the measurement of interest.
`item`	Name of column containing the items. If this is a test-retest reliability study then this would indicate the time point (e.g., time1,time2, time3, etc.).

id Column with subject identifier.

data Data frame with all data.

wide Logical value (TRUE or FALSE) indicating if data is in a "wide" format. Default is TRUE.

col.names If wide is equal to TRUE then col.names is a list of the column names containing the measurements for reliability analysis.

se_type Type of standard error calculation. The default is to use the mean square error (MSE). Otherwise, the total sums of squares and the ICC are utilized to estimate the SEM, SEE, and SEP.

cv_calc Coefficient of variation (CV) calculation. This function allows for 3 versions of the CV. "MSE" is the default.

conf.level the confidence level required. Default is 95%.

other_ci Logical value (TRUE or FALSE) indicating whether to calculate confidence intervals for the CV, SEM, SEP, and SEE. Note: this will dramatically increase the computation time.

type A character string representing the type of confidence intervals for the CV, SEM, SEP, and SEE. Only "norm", "basic", and "perc" currently supported for parametric bootstrap CI. An approximate method, chisq, is the default. See ?boot::boot.ci for more details on bootstrap methods.

replicates The number of bootstrap replicates. Passed on to the boot function. Default is 1999.

Details

These functions return intraclass correlation coefficients and other measures of reliability (CV, SEM, SEE, and SEP). The estimates of variances for any of the measures are derived from linear mixed models. When other_ci is set to TRUE, then a parametric bootstrap approach to calculating confidence intervals is used for the CV, SEM, SEE, and SEP.

reli_stats uses a linear mixed model to estimate variance components. In some cases there are convergence issues. When this occurs it is prudent to use reli_aov which instead utilizes sums of squares approach. The results may differ slightly between the functions. If reli_aov is used then rows with missing observations (e.g., if a participant has a missing observation) will be dropped.

The CV calculation has 3 versions. The "MSE" uses the "mean squared error" from the linear mixed model used to calculate the ICCs. The "SEM" option instead uses the SEM calculation and expresses CV as a ratio of the SEM to the overall mean. The "residuals" option u uses the model residuals to calculate the root mean square error which is then divided by the grand mean.

The CV, SEM, SEE, and SEP values can have confidence intervals produced if the other_ci argument is set to TRUE. For the CV, the default method (`type = "chisq"`) is Vangal's modification of the McKay approximation. For the other measures, a simple chi-squared approximation is utilized (Hann & Meeker, 1991). All other methods are bootstrapping based methods (see `?boot::boot`). The reli_stats functions utilizes a parametric bootstrap while the reli_aov function utilizes an ordinary (non-parametric) bootstrap method.

Value

Returns single list with the results of the agreement analysis.

- `icc`: Table of ICC results
- `lmer`: Linear mixed model from lme4
- `anova`: Analysis of Variance table
- `var_comp`: Table of Variance Components
- `n.id`: Number of subjects/participants
- `n.items`: Number of items/time points
- `cv`: Coefficient of Variation
- `SEM`: List with Standard MSE of Measurement estimate (est)
- `SEE`: List with Standard MSE of the Estimate estimate (est)
- `SEP`: List with Standard MSE of Predictions (est)
- `call`: the matched call

References

Weir, J. P. (2005). Quantifying test-retest reliability using the intraclass correlation coefficient and the SEM. The Journal of Strength & Conditioning Research, 19(1), 231-240.

Shrout, P.E. and Fleiss, J.L. (1976). Intraclass correlations: uses in assessing rater reliability. Psychological Bulletin, 86, 420-3428.

McGraw, K. O. and Wong, S. P. (1996). Forming inferences about some intraclass correlation coefficients. Psychological Methods, 1, 30-46. See errata on page 390 of same volume.

Hahn, G. J., & Meeker, W. Q. (2011). Statistical intervals: a guide for practitioners (Vol. 92). John Wiley & Sons. pp. 55-56.

Vangel, M. G. (1996). Confidence intervals for a normal coefficient of variation. The American Statistician, 50(1), 21-26.

Examples

```
data('reps')
reli_stats(data = reps, wide = TRUE, col.names = c("x","y"))
```

`reps` *reps*

Description

A fake data set of a agreement study where both measures have replicates.

The data set published in the original Bland & Altman paper on agreement.

Usage

```
reps

ba1986
```

Format

A data frame with 20 rows with 3 variables

id Subject identifier

x X measurement

y Y measurement

A data frame with 17 rows with 5 variables

id Subject identifier

wright1 PERF measurement #1 using Wright device

wright2 PERF measurement #2 using Wright device

mini1 PERF measurement #1 using Mini device

mini2 PERF measurement #2 using Mini device

References

Bland, J. M., & Altman, D. (1986). Statistical methods for assessing agreement between two methods of clinical measurement. The Lancet, 327(8476), 307-310.

---

simple_agree-methods *Methods for simple_agree objects*

Description

Methods defined for objects returned from the agree functions.

Usage

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

## S3 method for class 'simple_agree'
plot(
  x,
  type = 1,
  x_name = "x",
  y_name = "y",
 geom = c("geom_point", "geom_bin2d", "geom_density_2d", "geom_density_2d_filled",
    "stat_density_2d"),
  smooth_method = NULL,
  smooth_se = TRUE,
  ...
)

check(x)

## S3 method for class 'simple_agree'
check(x)
```

Arguments

`x`	object of class `simple_agree` as returned from a function starting with 'agree'
`...`	further arguments passed through, see description of return value for details. `agree_test`.
`type`	Type of plot to output. Default (1) is Bland-Altman plot while type=2 will produce a line-of-identity plot.
`x_name`	Name/label for x values (first measurement)
`y_name`	Name/label for y values (second measurement)
`geom`	String naming the type of geometry to display the data points. Default is "geom_point". Other options include: "geom_bin2d", "geom_density_2d", "geom_density_2d_filled", and "stat_density_2d".
`smooth_method`	Smoothing method (function) to use, accepts either NULL or a character vector, e.g. "lm", "glm", "gam", "loess" or a function. Default is NULL, which will not include a trend line.
`smooth_se`	Display confidence interval around smooth?

Value

`print` Prints short summary of the Limits of Agreement

`plot` Returns a plot of the limits of agreement (type = 1) or concordance plot (type = 2)

`check` Returns 2 plots, p_norm and p_het, testing the assumptions of a Bland-Altman analysis. P-values for the normality and heteroskedasticity tests are provided as captions to the plot.

`simple_eiv-methods` *Methods for simple_eiv objects*

Description

Methods defined for objects returned from the error-in-variables models (e.g., dem_reg).

Usage

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

## S3 method for class 'simple_eiv'
plot(x, x_name = "x", y_name = "y", ...)

## S3 method for class 'simple_eiv'
check(x)
```

Arguments

`x` object of class `simple_eiv` from the dem_reg function.

`...` further arguments passed through, see description of return value. for details. `agree_test`.

`x_name` Name/label for x values (first measurement)

`y_name` Name/label for y values (second measurement)

Value

`print` Prints short summary of the error-in-variables (e.g., Deming) regression model.

`plot` Returns a plot of the deming regression line, the line-of-identity, and the raw data.

`check` Returns plots of the optimized residuals.

`simple_reli-methods` *Methods for simple_reli objects*

Description

Methods defined for objects returned from the agree functions.

Usage

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

## S3 method for class 'simple_reli'
plot(x, ...)

## S3 method for class 'simple_reli'
check(x)
```

Arguments

`x` object of class `simple_reli` as returned from the reli_stats function

`...` further arguments passed through, see description of return value for details. `reli_stats`.

Value

`print` Prints short summary of the Limits of Agreement

`plot` Returns a plot of the data points used in the reliability analysis

`temps` *Data*

Description

A dataset from a study on the reliability of human body temperature at different times of day before and after exercise.

Usage

```
temps

recpre_long
```

Format

A data frame with 60 rows and 10 variables:

id Subject identifier

trial_num order in which the experimental trial was completed

trial_condition Environmental condition and metabolic heat production

tod Time of Day

trec_pre Rectal temperature before the beginning of the trial

trec_post Rectal temperature at the end of the trial

trec_delta Change in rectal temperature

teso_pre Esophageal temperature before the beginning of the trial

teso_post Esophageal temperature at the end of the trial

teso_delta Change in esophageal temperature

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 30 rows and 6 columns.

Source

Ravanelli N, Jay O. The Change in Core Temperature and Sweating Response during Exercise Are Unaffected by Time of Day within the Wake Period. Med Sci Sports Exerc. 2020 Dec 1. doi: 10.1249/MSS.0000000000002575. Epub ahead of print. PMID: 33273272.

tolerance_delta-methods

Methods for tolerance_delta objects

Description

Methods defined for objects returned from the tolerance_delta function(s).

Usage

```
## S3 method for class 'tolerance_delta'
print(x, digits = 4, ...)

## S3 method for class 'tolerance_delta'
plot(
  x,
 geom = c("geom_point", "geom_bin2d", "geom_density_2d", "geom_density_2d_filled",
    "stat_density_2d"),
  delta = NULL,
  ...
)

## S3 method for class 'tolerance_delta'
check(x)
```

Arguments

x	object of class tolerance_delta as returned from a agreement_limit function.
digits	The number of digits to print.
...	further arguments passed through, see description of return value for details. tolerance_limit.
geom	String naming the type of geometry to display the data points. Default is "geom_point". Other options include: "geom_bin2d", "geom_density_2d", "geom_density_2d_filled", and "stat_density_2d".
delta	The maximal allowable difference.

Value

print Prints short summary of the tolerance limits.

plot Returns a plot of the tolerance limits.

check Returns plots testing the assumptions of the model. P-values for the normality and heteroskedasticity tests are provided as captions to the plot.

`tolerance_limit` *Tolerance Limits from an Agreement Study*

Description

[Maturing]

A function for calculating tolerance limits for the difference between two measurements (difference = x-y). This is a procedure that should produce results similar to the Bland-Altman limits of agreement. See vignettes for more details.

Usage

```
tolerance_limit(
  data,
  x,
  y,
  id = NULL,
  condition = NULL,
  time = NULL,
  pred_level = 0.95,
  tol_level = 0.95,
  tol_method = c("approx", "perc"),
  prop_bias = FALSE,
  log_tf = FALSE,
  log_tf_display = c("ratio", "sympercent"),
  cor_type = c("sym", "car1", "ar1", "none"),
  correlation = NULL,
  weights = NULL,
  keep_model = TRUE,
  replicates = 999
)
```

Arguments

`data` A data frame containing the variables.

`x` Name of the column for the first measurement.

`y` Name of the column for the second measurement.

`id` Name of the column for the subject ID.

`condition` Name of the column indicating different conditions subjects were tested under. This can be left missing if there are no differing conditions to be tested.

`time` Name of the column indicating the time points. Only necessary if the data is from time series or repeated measures collection.

`pred_level` Prediction level for the prediction interval. Default is 95%.

`tol_level` Tolerance level for the tolerance limit (i.e., the CI of the prediction limit). Default is 95%.

`tol_method` Method for calculating the tolerance interval. Options are "approx" for a chi-square based approximation and "perc" for a parametric percentile bootstrap method.

`prop_bias` Whether to include a proportional bias term in the model. Determines whether proportional bias should be considered for the prediction/tolerance limits calculations.

`log_tf` Calculate limits of agreement using log-transformed data.

`log_tf_display` The type of presentation for log-transformed results. The differences between methods can be displayed as a "ratio" or "sympercent".

`cor_type` The type of correlation structure. "sym" is for Compound Symmetry, "car1" is for continuous autocorrelation structure of order 1, or "ar1" for autocorrelation structure of order 1.

`correlation` an optional corStruct object describing the within-group correlation structure that overrides the default setting. See the documentation of corClasses for a description of the available corStruct classes. If a grouping variable is to be used, it must be specified in the form argument to the corStruct constructor. Defaults to NULL.

`weights` an optional varFunc object or one-sided formula describing the within-group heteroskedasticity structure that overrides the default setting. If given as a formula, it is used as the argument to varFixed, corresponding to fixed variance weights. See the documentation on varClasses for a description of the available varFunc classes.

`keep_model` Logical indicator to retain the GLS model. Useful when working with large data and the model is very large.

`replicates` The number of bootstrap replicates. Passed on to the boot function. Default is 999.

Details

The tolerance limits calculated in this function are based on the papers by Francq & Govaerts (2016), Francq, et al. (2019), and Francq, et al. (2020). When `tol_method` is set to "approx", the tolerance limits are calculated using the approximation detailed in Francq et al. (2020). However, these are only an approximation and conservative. Therefore, as suggested by Francq, et al. (2019), a parametric bootstrap approach can be utilized to calculate percentile tolerance limits (`tol_method = "perc"`).

Value

Returns single `tolerance_delta` class object with the results of the agreement analysis with a prediction interval and tolerance limits.

- `limits`: A data frame containing the prediction/tolerance limits.
- `model`: The GLS model; NULL if keep_model set to FALSE.
- `call`: The matched call.

References

Francq, B. G., & Govaerts, B. (2016). How to regress and predict in a Bland–Altman plot? Review and contribution based on tolerance intervals and correlated-errors-in-variables models. Statistics in mMdicine, 35(14), 2328-2358.

Francq, B. G., Lin, D., & Hoyer, W. (2019). Confidence, prediction, and tolerance in linear mixed models. Statistics in Medicine, 38(30), 5603-5622.

Francq, B. G., Berger, M., & Boachie, C. (2020). To tolerate or to agree: A tutorial on tolerance intervals in method comparison studies with BivRegBLS R Package. Statistics in Medicine, 39(28), 4334-4349.

Examples

```
data('reps')

# Simple
tolerance_limit(x = "x", y ="y", data = reps)

# Nested
tolerance_limit(x = "x", y ="y", data = reps, id = "id")
```

Index