Package ‘SimplyAgree’

August 24, 2022

Type Package

Title Flexible and Robust Agreement and Reliability Analyses

Version 0.1.0

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. For robust analyses of agreement, limits of agreement through a bootstrap method can also be calculated.

URL https://aaroncaldwell.us/SimplyAgree/

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

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

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

VignetteBuilder knitr

Depends R (>= 3.6)

NeedsCompilation no

Author Aaron Caldwell [aut, cre]

Repository CRAN

Date/Publication 2022-08-24 07:10:16 UTC
Description

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
)
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.
agree_nest

Usage

agree_nest(
  x,
  y,
  id,
  data,
  delta,
  agree.level = 0.95,
  conf.level = 0.95,
  TOST = TRUE,
  prop_bias = 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
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 con-
            sidered for the limits of agreement calculations.

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 measure-
       ments 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


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)

Description

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.
"agee" 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


Examples

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

agree_reps

Tests for Absolute Agreement with Replicates

Description

agree_reps 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

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

Arguments

- `x`: Name of column with first measurement
- `y`: Name of other column with first measurement
- `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.

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


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)

Description

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 also provided.

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 con-
            sidered 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
tical methods in medical research, 8(2), 135-160.
ods of clinical measurement. The lancet, 327(8476), 307-310.
Biometrics, 255-268.

Examples

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

Estimate power curve for Bland-Altman limits of agreement

Description

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

Examples

```r
dataCurve <- 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**

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

#### Usage

```r
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.

Details
This function provides a Deming regression analysis wherein the sum of distances in both x and y direction is minimized. Deming 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 weighted 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

---

Simple Agreement Analysis

Description
Simple Agreement Analysis
Usage

jmvgree(
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$text a preformatted
results$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 \texttt{as.data.frame}. For example:

\begin{verbatim}
results$blandtab$asDF
\end{verbatim}

\begin{verbatim}
as.data.frame(results$blandtab)
\end{verbatim}

---

\section*{jmvgreemulti \textit{Nested/Replicate Data Agreement Analysis}}

\subsection*{Description}

Nested/Replicate Data Agreement Analysis

\subsection*{Usage}

\begin{verbatim}
jmvgreemulti(  
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"
)
\end{verbatim}

\subsection*{Arguments}

\begin{tabular}{ll}
\textbf{data} & Data  
\textbf{method1} & Name of column containing 1st Vector of data  
\textbf{method2} & Name of column containing Vector of data  
\textbf{id} & Name of column containing subject identifier  
\textbf{ciWidth} & a number between 50 and 99.9 (default: 95), the width of confidence intervals  
\textbf{agreeWidth} & a number between 50 and 99.9 (default: 95), the width of agreement limits  
\textbf{testValue} & a number specifying the limit of agreement  
\textbf{CCC} & TRUE or FALSE (default), produce CCC table  
\textbf{valEq} & .  
\textbf{plotbland} & TRUE or FALSE (default), for Bland-Altman plot  
\textbf{plotcon} & TRUE or FALSE (default), for Line of identity plot  
\textbf{prop_bias} & TRUE or FALSE  
\textbf{xlabel} & The label for the x-axis on the BA plot  
\textbf{ylabel} & The label for the y-axis on the BA plot
\end{tabular}
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,                 # Data
  method1,              # Name of column containing 1st Vector of data
  method2,              # Name of column containing Vector of data
  ciWidth = 95,         # a number between 50 and 99.9 (default: 95), the width of confidence intervals
  testValue = 1,        # Ratio of the two error variances. Default is 1.
  plotcon = FALSE,      # TRUE or FALSE (default), for Bland-Altman plot
  plotcheck = FALSE,    # for Bland-Altman plot
  weighted = FALSE,     # for Bland-Altman plot
  xlabel = "Method: 1",# for Bland-Altman plot
  ylabel = "Method: 2"  # for Bland-Altman plot
)
```

**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
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$text` a preformatted
- `results$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:

```r
results$demtab$asDF
as.data.frame(results$demtab)
```
Tables can be converted to data frames with `asDF` or `as.data.frame`. For example:

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

---

**loa_lme**

*Limits of Agreement with Linear Mixed Effects*

**Description**

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

**Usage**

```r
loa_lme(
  diff,
  avg,
  condition = NULL,
  id,
  data,
  type = "perc",
  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

### 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**

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

**Usage**

```r
doa_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 500.
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

**Methods for loa_mixed_bs objects**

Description

Methods defined for objects returned from the loa_mixed functions.

Usage

```r
## 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.

Value

- `print`: Prints short summary of the Limits of Agreement
- `plot`: Returns a plot of the limits of agreement

**Methods for powerCurve objects**

Description

Methods defined for objects returned from the powerCurve function.

Usage

```r
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.
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

Description

reli_stats produces 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,
  cv_calc = "MSE",
  conf.level = 0.95,
  other_ci = FALSE,
  type = "perc",
  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.
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 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.

replicates

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

Details

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 uses the sjstats R package approach which uses the model residuals to calculate the root mean square error which is then divided by the grand mean.

This function returns 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. The results may differ slightly from the results from ICC calculations derived from an analysis of variance. 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.

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 Predicitions (est)
"call" the matched call

References

Examples

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

Description

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

Usage

reps

Format

A data frame with 20 rows with 3 variables

  id  Subject identifier
  x   X measurement
  y   Y measurement

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 = "geom_point",
  smooth_method = NULL,
  smooth_se = TRUE,
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.
- **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.

Description

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

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

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

```r
## 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.
- `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.

---

**Description**

Methods defined for objects returned from the agree functions.

**Usage**

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

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

```r
## 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.

---

---
Value

print  Prints short summary of the Limits of Agreement
plot  Returns a plot of the data points used in the reliability analysis

Description

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

Usage

temps

trecpre_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

Index

* datasets
  reps, 24
temps, 27
agree_coef, 2
agree_nest, 3
agree_np, 5
agree_reps, 6
agree_test, 8, 25, 26
as.data.frame, 14–17
blandPowerCurve, 10, 21
check (simple_agree-methods), 24
check.loa_mermod (loa_mermod-methods), 18
check.simple_eiv (simple_eiv-methods), 25
check.simple_reli (simple_reli-methods), 26
dem_reg, 11
find_n (powerCurve-methods), 21
jmvagree, 12
jmvagreemulti, 14
jmvdeming, 15
jmvreli, 16
loa_lme, 17
loa_mermod-methods, 18
loa_mixed, 19, 19, 21
loa_mixed_bs-methods, 21
plot.loa_mermod (loa_mermod-methods), 18
plot.loa_mixed_bs (loa_mixed_bs-methods), 21
plot.powerCurve (powerCurve-methods), 21
plot.simple_agree (simple_agree-methods), 24
plot.simple_eiv (simple_eiv-methods), 25
plot.simple_reli (simple_reli-methods), 26
powerCurve-methods, 21
print.loa_mermod (loa_mermod-methods), 18
print.loa_mixed_bs (loa_mixed_bs-methods), 21
print.simple_agree (simple_agree-methods), 24
print.simple_eiv (simple_eiv-methods), 25
print.simple_reli (simple_reli-methods), 26
recpre_long (temps), 27
reli_stats, 22, 26
reps, 24
simple_agree-methods, 24
simple_eiv-methods, 25
simple_reli-methods, 26

temps, 27

28