Package ‘krippendorffsalpha’

May 11, 2022

Type Package
Title Measuring Agreement Using Krippendorff's Alpha Coefficient
Version 1.1-2
Date 2022-11-05
Author John Hughes
Maintainer John Hughes <drjphughesjr@gmail.com>
Suggests parallel, pbapply, testthat (>= 3.0.0)
Description Provides tools for applying Krippendorff's Alpha methodology <DOI:10.1080/19312450709336664>. The framework supports common and user-defined distance functions, and can accommodate any number of units, any number of coders, and missingness. Bootstrap inference is permitted, and the computation can be done in parallel.
License GPL (>= 2)
URL http://www.johnhughes.org
RoxygenNote 7.1.2
NeedsCompilation no
Config/testthat/edition 3
Repository CRAN
Date/Publication 2022-05-11 18:40:02 UTC

R topics documented:

cartilage .................................................. 2
confint.krippendorffsalpha .......................... 2
influence.krippendorffsalpha ....................... 3
interval.dist ........................................ 4
krippendorffs.alpha .................................. 5
nominal.dist .......................................... 7
plot.krippendorffsalpha ............................. 8
ratio.dist ........................................... 9
summary.krippendorffsalpha ....................... 10
**cartilage**

Data from an MRI study of hip cartilage in femoroacetabular impingement.

**Description**

This data frame has exactly two columns. The first column contains raw T2* values, the second column contrast-enhanced T2* values.

**Usage**

```r
data(cartilage)
```

**Format**

A data frame having 323 rows and two columns

**References**


---

**confint.krippendorffsalpha**

Compute a confidence interval for Krippendorff’s Alpha.

**Description**

Compute a confidence interval for Krippendorff’s Alpha.

**Usage**

```r
## S3 method for class 'krippendorffsalpha'
confint(object, parm = "alpha", level = 0.95, ...)
```

**Arguments**

- `object` an object of class "krippendorffsalpha", the result of a call to `krippendorffs.alpha`.
- `parm` always ignored since there is only one parameter.
- `level` the desired confidence level for the interval. The default is 0.95.
- `...` additional arguments. These are passed to `quantile`.
influence.krippendorffsalpha

Details

This function computes a bootstrap confidence interval for alpha, assuming that krippendorffs.alpha was called with confint = TRUE.

Value

A vector with entries giving lower and upper confidence limits. These will be labelled as (1 - level) / 2 and 1 - (1 - level) / 2.

References


See Also

krippendorffs.alpha

Examples

# Fit a subset of the cartilage data. Compute bootstrap confidence intervals # using a bootstrap sample size of 1,000.

data(cartilage)
cartilage = as.matrix(cartilage[1:100, ])
fit.cart = krippendorffs.alpha(cartilage, level = "interval", confint = TRUE, control = list(bootit = 1000, parallel = FALSE))
fit.cart$alpha.hat
cconfint(fit.cart, level = 0.99)

influence.krippendorffsalpha

Compute DFBETAs for units and/or coders.

Description

Compute DFBETAs for units and/or coders.

Usage

## S3 method for class 'krippendorffsalpha'
influence(model, units, coders, ...)

interval.dist

Compute the squared difference between two scores.

Description

Compute the squared difference between two scores.

Usage

interval.dist(x, y)
krippendorffs.alpha

Arguments

x        a score.
y        a score.

Details

This function computes the squared difference between two scores. This may be an appropriate
distance function for the interval level of measurement. NA’s are handled gracefully.

Value

\((x - y)^2\), or 0 if x or y is NA.

See Also

nominal.dist, ratio.dist

---

krippendorffs.alpha  Apply Krippendorff’s Alpha.

---

Description

Apply Krippendorff’s Alpha.

Usage

krippendorffs.alpha(
  data,
  level = c("interval", "nominal", "ordinal", "ratio"),
  confint = TRUE,
  verbose = FALSE,
  control = list()
)

Arguments

data        a matrix of scores. Each row corresponds to a unit, each column a coder.
level       the level of measurement, one of "nominal", "ordinal", "interval", or "ratio";
or a user-defined distance function.
confint     logical; if TRUE, a bootstrap sample is produced.
verbose     logical; if TRUE, various messages are printed to the console. Note that if confint = TRUE a progress bar (pblapply) is displayed (if possible) during the bootstrap computation.
control     a list of control parameters.
bootit      the size of the bootstrap sample. This applies when confint = TRUE. Defaults to 1,000.
nodes  the desired number of nodes in the cluster.
parallel  logical; if TRUE (the default), bootstrapping is done in parallel.
type  one of the supported cluster types for makeCluster. Defaults to “SOCK”.

Details

This is the package’s flagship function. It applies the Krippendorff’s Alpha methodology for nominal, ordinal, interval, or ratio levels of measurement, and, if desired, produces confidence intervals. Parallel computing is supported, when applicable.

If the level of measurement is nominal, the discrete metric (nominal.dist) is employed by default. If the level of measurement is interval or ordinal, the squared-difference distance function (interval.dist) is employed by default. (For the ordinal level of measurement, using the squared-difference distance function may be inappropriate, in which case the user should supply his/her own distance function.) If the level of measurement is ratio, a ratio distance function (ratio.dist) is applied. Alternatively, the user may supply his/her own distance function. Said function must handle NA’s gracefully; see the above mentioned built-in distance functions for examples.

If argument confint is set to TRUE, bootstrapping is carried out. This is done by resampling, with replacement, the rows of data and then computing the alpha statistic for the resulting matrix. The elements of argument control are used to control the bootstrap computation.

Value

Function krippendorffs.alpha returns an object of class "krippendorffsalpha", which is a list comprising the following elements.

- boot.sample: when applicable, the bootstrap sample.
- call: the matched call.
- coders: the number of coders.
- alpha.hat: the estimate of alpha.
- confint: the value of argument confint.
- control: the list of control parameters.
- data: the matrix of scores.
- D.e: the estimate of total variation.
- D.o: the estimate of within-unit variation.
- level: the level of measurement.
- units: the number of units.
- verbose: the value of argument verbose.

References

nominal.dist

Examples

# The following data were presented in Krippendorff (2013).

nominal = matrix(c(1,2,3,2,1,4,1,2,NA,NA,NA,
                   1,2,3,2,2,4,1,2,5,NA,3,
                   NA,3,3,2,3,4,2,2,5,1,NA,
                   1,2,3,3,2,4,4,1,2,5,1,NA), 12, 4)

nominal
fit.nom = krippendorffs.alpha(nominal, level = "nominal", confint = TRUE, verbose = TRUE,
                               control = list(bootit = 100, parallel = FALSE))
summary(fit.nom)
confint(fit.nom, level = 0.99)

nominal.dist
Apply the discrete metric to two scores.

Description

Apply the discrete metric to two scores.

Usage

nominal.dist(x, y)

Arguments

x a score.
y a score.

Details

This function applies the discrete metric to two scores. This may be an appropriate distance function
for the nominal level of measurement. NA's are handled gracefully.

Value

0 if x is equal to y or if either is NA, 1 otherwise.

See Also

interval.dist, ratio.dist
plot.krippendorffalpha

Plot the results of a Krippendorff’s Alpha analysis.

Description

Plot the results of a Krippendorff’s Alpha analysis.

Usage

```r
## S3 method for class 'krippendorffalpha'
plot(
x,             # an object of class "krippendorffalpha", the result of a call to krippendorffs.alpha.
y = NULL,      # always ignored.
level = 0.95,  # the desired confidence level for the interval. The default is 0.95.
type = 7,      # the method used to compute sample quantiles. This argument is passed to quantile. The default is 7.
density = TRUE, # logical; if TRUE, a kernel density estimate is plotted.
lty.density = 1, # the line type for the kernel density estimate. The default is 1.
lty.estimate = 1, # the line type for the estimate of alpha. The default is 1.
lty.interval = 2, # the line type for the confidence limits. The default is 2.
col.density = "black", # the color for the kernel density estimate. The default is black.
col.estimate = "orange", # the color for the estimate of alpha. The default is orange.
col.interval = "blue", # the color for the confidence limits. The default is blue.
lwd.density = 3, # the line width for the kernel density estimate. The default is 3.
lwd.estimate = 3,
lwd.interval = 3,
...)
```

Arguments

- `x`: an object of class "krippendorffalpha", the result of a call to `krippendorffs.alpha`.
- `y`: always ignored.
- `level`: the desired confidence level for the interval. The default is 0.95.
- `type`: the method used to compute sample quantiles. This argument is passed to `quantile`. The default is 7.
- `density`: logical; if TRUE, a kernel density estimate is plotted.
- `lty.density`: the line type for the kernel density estimate. The default is 1.
- `lty.estimate`: the line type for the estimate of alpha. The default is 1.
- `lty.interval`: the line type for the confidence limits. The default is 2.
- `col.density`: the color for the kernel density estimate. The default is black.
- `col.estimate`: the color for the estimate of alpha. The default is orange.
- `col.interval`: the color for the confidence limits. The default is blue.
- `lwd.density`: the line width for the kernel density estimate. The default is 3.
ratio.dist

Apply a ratio distance function to two scores.

Description

Apply a ratio distance function to two scores.

Usage

    ratio.dist(x, y)

Arguments

    x a score.
    y a score.

Details

This function applies a ratio distance function to two scores. This may be an appropriate distance function for the ratio level of measurement. NA's are handled gracefully.
Value

\[(x - y)^2/(x + y)^2, \text{ or } 0 \text{ if } x \text{ or } y \text{ is NA.}\]

See Also

`interval.dist`, `nominal.dist`

---

**summary.krippendorffsalpha**

*Print a summary of a Krippendorff’s Alpha fit.*

---

**Description**

Print a summary of a Krippendorff’s Alpha fit.

**Usage**

```r
## S3 method for class 'krippendorffsalpha'
summary(object, conf.level = 0.95, digits = 4, ...)
```

**Arguments**

- `object` an object of class "krippendorffsalpha", the result of a call to `krippendorffs.alpha`
- `conf.level` the confidence level for the confidence intervals. The default is 0.95.
- `digits` the number of significant digits to display. The default is 4.
- `...` additional arguments. These are passed to `quantile`.

**Details**

This function prints a summary of the fit. First the values of the control parameters (defaults and/or values supplied in the call) are printed. Then a table of estimates is shown. If applicable, the table includes confidence intervals.

**References**


**See Also**

`krippendorffs.alpha`
Examples

# Fit a subset of the cartilage data. Compute bootstrap confidence intervals
# using a bootstrap sample size of 1,000. Display a summary of the results,
# including a 99% confidence interval. Also plot the results.

data(cartilage)
cartilage = as.matrix(cartilage[1:100, ])
fit.cart = krippendorffs.alpha(cartilage, level = "interval", confint = TRUE,
                               control = list(bootit = 1000, parallel = FALSE))
summary(fit.cart, conf.level = 0.99)
dev.new()
plot(fit.cart, xlim = c(0.7, 0.9), xlab = "Bootstrap Estimates",
     main = "Results for Cartilage Data")
Index

* datasets
  cartilage, 2

  cartilage, 2
  confint.krippendorffsalpha, 2

  hist, 9

  influence.krippendorffsalpha, 3
  interval.dist, 4, 6, 7, 10

  krippendorffs.alpha, 2–4, 5, 8–10

  makeCluster, 6

  nominal.dist, 5, 6, 7, 10

  pblapply, 5
  plot.krippendorffsalpha, 8

  quantile, 2, 8, 10

  ratio.dist, 5–7, 9

  summary.krippendorffsalpha, 10