Package 'krippendorffsalpha'

July 22, 2025

Type Package

Title Measuring Agreement Using Krippendorff's Alpha Coefficient

Version 2.0

Date 2022-10-25

Author John Hughes

Maintainer John Hughes <drjphughesjr@gmail.com>

Suggests parallel, pbapply, spam, testthat (>= 3.0.0)

Description Provides tools for applying Krippendorff's Alpha methodology <DOI:10.1080/19312450709336664>. Both the customary methodology and Hughes' methodology <DOI:10.48550/arXiv.2210.13265> are supported, the former being preferred for larger datasets, the latter for smaller datasets. The framework supports common and user-defined distance functions, and can accommodate any number of units, any number of coders, and missingness. Interval estimation can be done in parallel for either methodology.

License GPL (>= 2)

URL http://www.johnhughes.org

RoxygenNote 7.2.1

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2022-10-25 16:02:37 UTC

Contents

cartilage 2
confint.krippendorffsalpha 2
influence.krippendorffsalpha 3
interval.dist 4
krippendorffs.alpha 5
nominal.dist 8
plot.krippendorffsalpha 8
ratio.dist 10
summary.krippendorffsalpha 11

Index **13**

`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

```
data(cartilage)
```

Format

A data frame having 323 rows and two columns

References

Nissi, M. J., Mortazavi, S., Hughes, J., Morgan, P., and Ellermann, J. (2015). T2* relaxation time of acetabular and femoral cartilage with and without intra-articular Gd-DTPA2 in patients with femoroacetabular impingement. *American Journal of Roentgenology*, **204**(6), W695.

`confint.krippendorffsalpha` *Compute a confidence interval for Krippendorff's Alpha.*

Description

Compute a confidence interval for Krippendorff's Alpha.

Usage

```
## 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`.

Details

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

For method = "analytical", a jackknife-based interval is computed. For smaller samples the jackknife interval offers a very substantial improvement over the bootstrap interval, the latter of which offers quite poor coverage. For larger samples method = "customary" can safely be used, in which case a bootstrap interval is provided. For sufficiently large datasets the two intervals will be nearly equal, but the bootstrap approach is preferred owing to its much faster execution speed.

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

Nissi, M. J., Mortazavi, S., Hughes, J., Morgan, P., and Ellermann, J. (2015). T2* relaxation time of acetabular and femoral cartilage with and without intra-articular Gd-DTPA2 in patients with femoroacetabular impingement. *American Journal of Roentgenology*, **204**(6), W695.

See Also

krippendorffs.alpha

Examples

```
# Fit a subset of the cartilage data, using the customary methodology.
# Compute bootstrap confidence intervals using a bootstrap sample size
# of 1,000. Report the estimate of alpha, and produce a 99% interval.

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

Arguments

model	a fitted model object, the result of a call to krippendorffs.alpha.
units	a vector of integers. A DFBETA will be computed for each of the corresponding units.
coders	a vector of integers. A DFBETA will be computed for each of the corresponding coders.
...	additional arguments. These are ignored.

Details

This function computes DFBETAs for one or more units and/or one or more coders.

Value

A list comprising at most two elements.

dfbeta.units	a vector containing DFBETAs for the units specified via argument units.
dfbeta.coders	a vector containing DFBETAs for the coders specified via argument coders.

References

Young, D. S. (2017). *Handbook of Regression Methods*. CRC Press.

Krippendorff, K. (2013). Computing Krippendorff's alpha-reliability. Technical report, University of Pennsylvania.

Examples

```
# The following data were presented in Krippendorff (2013). This example
# applies Hughes' methodology to the data (method = "analytical", the default).
# DFBETAS are computed by leaving out unit 6, unit 11, coder 2, and coder 3.

nominal = matrix(c(1,2,3,3,2,1,4,1,2,NA,NA,NA,
                   1,2,3,3,2,2,4,1,2,5,NA,3,
                   NA,3,3,3,2,3,4,2,2,5,1,NA,
                   1,2,3,3,2,4,4,1,2,5,1,NA), 12, 4)
fit.nom = krippendorffs.alpha(nominal, level = "nominal", confint = FALSE)
summary(fit.nom)
(inf = influence(fit.nom, units = c(6, 11), coders = c(2, 3)))
```

---

interval.dist *Compute the squared difference between two scores.*

Description

Compute the squared difference between two scores.

Usage

```
interval.dist(x, y)
```

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"),
  method = c("analytical", "customary"),
  confint = TRUE,
  verbose = FALSE,
  control = list()
)
```

Arguments

data	a matrix of scores. Each row corresponds to a unit, each column to a coder.
level	the level of measurement, one of "nominal", "ordinal", "interval", or "ratio"; or a user-defined distance function.
method	the methodology to apply, either "analytical" or "customary".

`confint` logical; if `TRUE`, a confidence interval is computed. For `method = "analytical"` the interval is a jackknife interval. For `method = "customary"` the interval is a bootstrap interval.

`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 or jackknife computation.

`control` a list of control parameters.

- `bootit` the size of the bootstrap sample. This applies when `confint = TRUE` and `method = "customary"`. Defaults to 1,000.
- `nodes` the desired number of nodes in the cluster.
- `parallel` logical; if `TRUE` (the default), bootstrapping or jackknife estimation is done in parallel (for `confint = TRUE`).
- `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.

Argument `method` is used to choose between the customary Alpha methodology and the analytical methodology developed by Hughes: `method = "analytical"` or `method = "customary"`. For smaller samples Hughes' methodology should be strongly preferred because that approach reduces bias for point estimation and provides much better performing confidence intervals—jackknife intervals, to be precise. For large samples Krippendorff's customary methodology can safely be used for inference, and speeds computation considerably relative to Hughes' jackknife method.

If argument `confint` is set to `TRUE`, a confidence interval is computed. For Hughes' methodology a jackknife interval is produced. For the customary methodology a bootstrap interval is produced. The bootstrap 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 interval computation.

Value

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

`alpha.hat` the estimate of alpha.

`boot.sample` when applicable, the bootstrap sample.

`call` the matched call.

coders	the number of coders.
confint	the value of argument confint.
control	the list of control parameters.
data	the matrix of scores, where rows represent units and columns represent coders.
eta.hat	when method = "analytical", $log(MSA/MSE)$.
L	when method = "analytical", the lower 95% confidence limit for alpha.
level	the level of measurement, or a user-dfined distance function.
MSA	when method = "analytical", the estimate of between-unit variation.
MSE	the estimate of within-unit variation.
MST	when method = "customary", the estimate of total variation.
method	the value of argument method.
n_	when method = "analytical", the average number of scores per row of the data matrix.
se	when method = "analytical", the jackknife standard error.
U	when method = "analytical", the upper 95% confidence limit for alpha.
units	the number of units.
verbose	the value of argument verbose.

References

Krippendorff, K. (2013). Computing Krippendorff's alpha-reliability. Technical report, University of Pennsylvania.

Hughes, J. (2022). Toward improved inference for Krippendorff's Alpha agreement coefficient. arXiv.

Examples

```
# The following data were presented in Krippendorff (2013). This example
# applies Hughes' methodology (the default) to these data. A jackknife
# confidence interval is produced (confint = TRUE). The fit is then
# summarized, and a 99% interval is given.

nominal = matrix(c(1,2,3,3,2,1,4,1,2,NA,NA,NA,
                   1,2,3,3,2,2,4,1,2,5,NA,3,
                   NA,3,3,3,2,3,4,2,2,5,1,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(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.krippendorffsalpha *Plot the results of a Krippendorff's Alpha analysis.*

Description

Plot the results of a Krippendorff's Alpha analysis.

Usage

```
## S3 method for class 'krippendorffsalpha'
plot(
  x,
  y = NULL,
  level = 0.95,
  type = 7,
  density = TRUE,
  lty.density = 1,
  lty.estimate = 1,
  lty.interval = 2,
  col.density = "black",
  col.estimate = "orange",
  col.interval = "blue",
  lwd.density = 3,
  lwd.estimate = 3,
  lwd.interval = 3,
  ...
)
```

Arguments

x	an object of class "krippendorffsalpha", 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.
lwd.estimate	the line width for the estimate of alpha. The default is 3.
lwd.interval	the line width for the confidence limits. The default is 3.
...	additional arguments. These are passed to hist.

Details

This function plots the results of a Krippendorff's Alpha analysis, assuming that krippendorffs.alpha was called with method = "customary" and confint = TRUE. Otherwise there is no bootstrap sample to work with. The plot is highly customizable.

This function plots a histogram of the bootstrap sample, (optionally) a kernel density estimate, and vertical lines marking the lower and upper confidence limits.

References

Krippendorff, K. (2013). Computing Krippendorff's alpha-reliability. Technical report, University of Pennsylvania.

See Also

krippendorffs.alpha

Examples

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

nominal = matrix(c(1,2,3,3,2,1,4,1,2,NA,NA,NA,
                   1,2,3,3,2,2,4,1,2,5,NA,3,
                   NA,3,3,3,2,3,4,2,2,5,1,NA,
                   1,2,3,3,2,4,4,1,2,5,1,NA), 12, 4)
fit.nom = krippendorffs.alpha(nominal, level = "nominal", method = "customary", confint = TRUE,
                          verbose = TRUE, control = list(bootit = 1000, parallel = FALSE))
dev.new()
plot(fit.nom, main = "Results for Nominal Data", xlab = "Bootstrap Estimates", density = FALSE)
```

---

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

$(x-y)^2/(x+y)^2$, or 0 if x or y 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

```
## 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 data geometry is described, then the call signature is printed, then the values of the control parameters (defaults and/or values supplied in the call) are printed. Finally, a table of estimates is shown. If applicable, the table includes confidence limits.

References

Nissi, M. J., Mortazavi, S., Hughes, J., Morgan, P., and Ellermann, J. (2015). T2* relaxation time of acetabular and femoral cartilage with and without intra-articular Gd-DTPA2 in patients with femoroacetabular impingement. *American Journal of Roentgenology*, **204**(6), W695.

See Also

krippendorffs.alpha

Examples

```
# Fit a subset of the cartilage data, using the customary methodology.
# 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 = "ratio", method = "customary", 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*, *8*, *10*

krippendorffs.alpha, *2–4*, 5, *9–11*

makeCluster, *6*

nominal.dist, *5*, *6*, 8, *10*

pblapply, *6*
plot.krippendorffsalpha, 8

quantile, *2*, *9*, *11*

ratio.dist, *5*, *6*, *8*, 10

summary.krippendorffsalpha, 11