krippendorffsalpha: An R Package for Measuring Agreement Using Krippendorff's Alpha Coefficient

by John Hughes

Abstract R package **krippendorffsalpha** provides tools for measuring agreement using Krippendorff's α coefficient, a well-known nonparametric measure of agreement (also called inter-rater reliability and various other names). This article first develops Krippendorff's α in a natural way and situates α among statistical procedures. Then, the usage of package **krippendorffsalpha** is illustrated via analyses of two datasets, the latter of which was collected during an imaging study of hip cartilage. The package permits users to apply the α methodology using built-in distance functions for the nominal, ordinal, interval, or ratio levels of measurement. User-defined distance functions are also supported. The fitting function can accommodate any number of units, any number of coders, and missingness. Bootstrap inference is supported, and the bootstrap computation can be carried out in parallel.

Introduction

Krippendorff's α (Hayes and Krippendorff, 2007) is a well-known nonparametric measure of agreement (i.e., consistency of scoring among two or more raters for the same units of analysis (Gwet, 2014)). In R (Ihaka and Gentleman, 1996), Krippendorff's α can be applied using function `kripp.alpha` of package **irr** (Gamer et al., 2012), function `kripp.boot` of package **kripp.boot** (Proutskova and Gruszczynski, 2020), function `krippalpha` of package **icr** (Staudt and L'Ecuyer, 2020), and functions `krippen.alpha.raw` and `krippen.alpha.dist` of package **irrCAC** (Gwet, 2019). However, these packages fail to provide a number of useful features. In this article we present package **krippendorffsalpha**, which improves upon the above mentioned packages in (at least) the following ways. Package **krippendorffsalpha**

- offers commonly used built-in distance functions for the nominal, ordinal, interval, and ratio levels of measurement and also supports user-defined distance functions;
- conforms to the R idiom by providing S3 methods `confint`, `influence`, `plot`, and `summary`;
- supports embarrassingly parallel bootstrap computation; and
- supports verbose communication with the user, including the display of a progress bar during the production of the bootstrap sample.

The remainder of this article is organized as follows. In Section 2.2, we locate Krippendorff's α among statistical procedures. In Section 2.2.1, we first develop a special case of Krippendorff's α (call it α_{SED}) in a well-known parametric setting, and then we present α in its most general (i.e., nonparametric) form. In Section 2.2.2, we show that α is a type of multiresponse permutation procedure. In Section 2.2.3, we generalize α_{SED} in a fully parametric fashion, arriving at Sklar's ω. In Section 2.3, we describe our package's bootstrap inference for α and compare the performance of our procedure to that of two alternative approaches. In Section 2.4, we briefly discuss robustness and influence. In Section 2.5, we provide a thorough demonstration of **krippendorffsalpha**'s usage before concluding in Section 2.6.

Situating Krippendorff's Alpha among statistical procedures

Since Krippendorff's α is defined in terms of discrepancies (Krippendorff, 2013), at first glance, one might conclude, erroneously, that α is a measure of *dis*-agreement, and so answers the wrong question. In Sections 2.2.1–2.2.3, we will show, by examining Krippendorff's α's place among statistical procedures, that α is, in fact, a sensible measure of agreement. Also, establishing a context for α may help practitioners make educated decisions regarding α's use.

The UML class diagram (Fowler et al., 2004) shown below in Figure 1 provides a conceptual roadmap for our development. Briefly, a special case of α (which we denote as Alpha(SED) or α_{SED}) arises naturally in the context of the one-way mixed-effects ANOVA model. Alpha(SED) can then be generalized in a nonparametric fashion to arrive at Krippendorff's α as it has been presented by Hayes and Krippendorff (see Gwet (2015) for development of nonparametric α in terms of agreement rather

than discrepancies); this nonparametric form of α is a (slightly modified) multiresponse permutation procedure. Alternatively, α_{SED} can be generalized in a parametric fashion to arrive at Sklar's ω, a Gaussian copula-based methodology for measuring agreement.

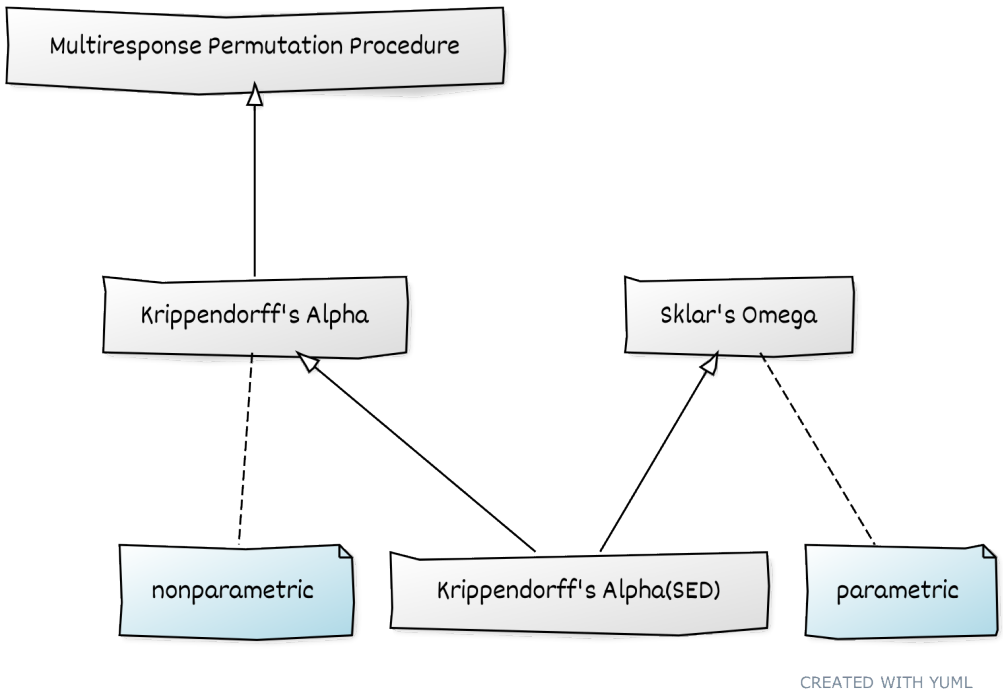

Figure 1: A UML class diagram that shows the relationships between Krippendorff's α and other statistical procedures.

Parametric genesis of Krippendorff's Alpha coefficient

In this section, we develop Krippendorff's α (Hayes and Krippendorff, 2007) in an intuitive and bottom-up fashion. Our starting point is a fully parametric model, namely, the classic one-way mixed-effects ANOVA model (Ravishanker and Dey, 2001). To ease exposition, we will consider only a balanced version of the model. We have, for n_u units and n_c coders, scores

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij}, \qquad (i = 1, \ldots, n_u;\ j = 1, \ldots, n_c),$$

where

- μ (the population mean) is a fixed real number,
- the τ_i are independent $\mathcal{N}(0, \sigma_\tau^2)$ random variables,
- the ε_{ij} are independent $\mathcal{N}(0, \sigma_\varepsilon^2)$ random variables, and
- the τ_i are independent of the ε_{ij}.

In this setup, we have n_c Gaussian codes $Y_{i1}, \ldots, Y_{in_c}$ for unit $i \in \{1, \ldots, n_u\}$. Conditional on τ_i, said codes are $\mathcal{N}(\mu + \tau_i, \sigma_\varepsilon^2)$ random variables. Since the variables share the "unit effect" τ_i, the variables are correlated. The correlation, which is usually called the *intraclass correlation*, is given by

$$\alpha = \frac{\sigma_\tau^2}{\sigma_\tau^2 + \sigma_\varepsilon^2} = 1 - \frac{\sigma_\varepsilon^2}{\sigma_\tau^2 + \sigma_\varepsilon^2}.$$

We use α to denote this quantity precisely because Krippendorff's α is the intraclass correlation for codes that conform to this model. That is, for the one-way mixed-effects ANOVA model, Krippendorff's α is the intraclass correlation. The reader may recall that the estimator of α for this model is

$$\hat{\alpha} = 1 - \frac{\widehat{\sigma_\varepsilon^2}}{\widehat{\sigma_\tau^2 + \sigma_\varepsilon^2}} = 1 - \frac{\frac{1}{n_u(n_c-1)} \sum_{i=1}^{n_u} \sum_{j=1}^{n_c} (Y_{ij} - \bar{Y}_{i\bullet})^2}{\frac{1}{n_u n_c - 1} \sum_{i=1}^{n_u} \sum_{j=1}^{n_c} (Y_{ij} - \bar{Y}_{\bullet\bullet})^2}, \tag{1}$$

where $\bar{Y}_{i\bullet}$ and $\bar{Y}_{\bullet\bullet}$ denote the arithmetic means for the ith unit and for the entire sample, respectively. The form of this estimator is not surprising, of course, since it is well-known that assuming Gaussianity leads to variance estimators involving sums of weighted squared deviations from sample arithmetic means.

We can eliminate the arithmetic means in (1) by employing the identity

$$\sum_{i=1}^{n}(x_i - \bar{x}_\bullet)^2 = \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_j)^2.$$

This gives

$$\hat{\alpha} = 1 - \frac{\frac{1}{2n_u n_c(n_c-1)}\sum_{i=1}^{n_u}\sum_{j=1}^{n_c}\sum_{k=1}^{n_c}(Y_{ij} - Y_{ik})^2}{\frac{1}{2n_u n_c(n_u n_c-1)}\sum_{i=1}^{n_u}\sum_{j=1}^{n_c}\sum_{k=1}^{n_u}\sum_{l=1}^{n_c}(Y_{ij} - Y_{kl})^2}. \tag{2}$$

Now, let $d^2(x,y) = (x-y)^2$, and rewrite (2) as

$$\hat{\alpha} = 1 - \frac{D_o}{D_e} = 1 - \frac{\frac{1}{2n_u n_c(n_c-1)}\sum_{i=1}^{n_u}\sum_{j=1}^{n_c}\sum_{k=1}^{n_c} d^2(Y_{ij}, Y_{ik})}{\frac{1}{2n_u n_c(n_u n_c-1)}\sum_{i=1}^{n_u}\sum_{j=1}^{n_c}\sum_{k=1}^{n_u}\sum_{l=1}^{n_c} d^2(Y_{ij}, Y_{kl})}, \tag{3}$$

where D_o and D_e denote observed and expected disagreement, respectively. This is Krippendorff's α for the squared Euclidean distance (which is not a metric but a Bregman divergence (Bregman, 1967)). We will henceforth refer to this version of α as Alpha(SED) or α_{SED}. As we mentioned above, this form of Krippendorff's α arises quite naturally when the data at hand conform to the one-way mixed-effects ANOVA model, for which agreement corresponds to a positive correlation. More generally, Krippendorff recommends this form of α for the interval level of measurement. For other levels of measurement, Krippendorff presents other distance functions d^2 (several possibilities are shown in Table 1). Note that package **krippendorffsalpha** supports user-defined distance functions as well as the interval, nominal, and ratio distance functions shown in the table.

Level of Measurement	Distance Function
interval	$d^2(x,y) = (x-y)^2$
nominal	$d^2(x,y) = 1\{x \neq y\}$
ratio	$d^2(x,y) = \left(\frac{x-y}{x+y}\right)^2$
bipolar	$d^2(x,y) = \frac{(x-y)^2}{(x+y-2x_{\min})(2x_{\max}-x-y)}$
circular	$d^2(x,y) = \left\{\sin\left(\pi\frac{x-y}{I}\right)\right\}^2$ (I = number of equal intervals on circle)
ordinal	$d^2(x,y) = (x-y)^2$ (adjacent ranks are equidistant)

Table 1: Several distance functions that may be appropriate for use in Krippendorff's α.

Alpha as a multiresponse permutation procedure

In any case, (3) is nonparametric for arbitrary d^2 since then the estimator $\hat{\alpha}$ does not usually correspond to a well-defined population parameter α. This more general form of Krippendorff's α is, in fact, a special case of the so-called multiresponse permutation procedure (MRPP). The MRPPs form a class of permutation methods for discerning differences among two or more groups in one or more dimensions (Mielke and Berry, 2007). Note, however, that although α can be viewed as an MRPP (as we are about to show), α has been modified for the purpose of measuring agreement rather than discerning differences.

To show that Krippendorff's α is an MRPP, we first present the general form of the MRPP. Following Mielke and Berry, let $\Omega = \{\omega_1, \ldots, \omega_n\}$ be a finite sample that is representative of some population of interest, let $S_1, \ldots, S_{a+1}$ denote a partition of Ω into $a+1$ disjoint groups, and let ρ be a metric that makes sense for the objects of Ω. (Strictly speaking, ρ need not be a metric; a symmetric distance function will suffice.) To ease notation a bit, let $\rho_{jk} \equiv \rho(\omega_j, \omega_k)$. Then, the MRPP statistic can be

written as

$$\delta = \sum_{i=1}^{a} C_i \theta_i,$$

where $C_i > 0$ are group weights such that $\sum_i C_i = 1$;

$$\theta_i = \frac{1}{\binom{n_i}{2}} \sum_{j<k} \rho_{jk} 1_i\{\omega_j\} 1_i\{\omega_k\}$$

is the average distance between distinct pairs of objects in group S_i; $n_i \geq 2$ is the number of objects in group i; $l = \sum_{i=1}^{a} n_i$; $n_{a+1} = n - l \geq 0$ is the number of remaining unclassified objects in group S_{a+1}; and 1_i is the indicator function for membership in group i.

Note that this formulation is quite general since the objects in Ω can be scalars, vectors, or more exotic objects, and we are free to choose the metric and the weights. In the case of Krippendorff's α, we can produce $\delta = D_o$ by letting $\rho = d^2$ for some appropriately chosen distance function d^2 and choosing weights $C_i = 1/n_u$.

A parametric generalization of Alpha(SED)

In the preceding sections, we generalized α_{SED} in a nonparametric fashion by substituting other notions of distance for the squared Euclidean distance. Now, we will present a fully parametric generalization of α_{SED}, namely, Sklar's ω (Hughes, 2018).

The statistical model underpinning Sklar's ω is a Gaussian copula model (Xue-Kun Song, 2000). The most general form of the model is given by

$$\begin{aligned} \boldsymbol{Z} = (Z_1, \ldots, Z_n)' &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Omega}) \\ U_i = \Phi(Z_i) &\sim \mathcal{U}(0,1) \qquad (i = 1, \ldots, n) \\ Y_i = F_i^{-1}(U_i) &\sim F_i, \end{aligned} \tag{4}$$

where $\boldsymbol{\Omega}$ is a correlation matrix, Φ is the standard Gaussian cdf, and F_i is the cdf for the ith outcome Y_i. Note that $\boldsymbol{U} = (U_1, \ldots, U_n)'$ is a realization of the Gaussian copula, which is to say that the U_i are marginally standard uniform and exhibit the Gaussian correlation structure defined by $\boldsymbol{\Omega}$. Since U_i is standard uniform, applying the inverse probability integral transform to U_i produces outcome Y_i having the desired marginal distribution F_i.

To see that the one-way mixed-effects ANOVA model (and hence α_{SED}) is a special case of Sklar's ω, let the copula correlation matrix $\boldsymbol{\Omega}$ be block diagonal, where the ith block corresponds to the ith unit $(i = 1, \ldots, n_u)$ and has a compound symmetry structure. That is,

$$\boldsymbol{\Omega} = \operatorname{diag}(\boldsymbol{\Omega}_i),$$

where

$$\boldsymbol{\Omega}_i = \begin{array}{c} \\ c_1 \\ c_2 \\ \vdots \\ c_{n_c} \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \ldots & c_{n_c} \end{array} \\ \begin{pmatrix} 1 & \omega & \ldots & \omega \\ \omega & 1 & \ldots & \omega \\ \vdots & \vdots & \ddots & \vdots \\ \omega & \omega & \ldots & 1 \end{pmatrix} \end{array}.$$

Complete the specification by letting F_{ij} $(i = 1, \ldots, n_u;\ j = 1, \ldots, n_c)$ be the cdf for the Gaussian distribution with mean μ and variance σ^2. Then $\omega = \alpha$, the intraclass correlation coefficient.

Inference for Krippendorff's Alpha

Mielke and Berry describe hypothesis testing for MRPPs. Specifically, they discuss three approaches: permutation, Monte Carlo resampling, and Pearson type III moment approximation. The latter has significant advantages. For Krippendorff's α, though, we are interested not in hypothesis testing but in interval estimation. This can be done straightforwardly and efficiently using Monte Carlo resampling. Since D_e is invariant to permutation of the scores, our resampling procedure focuses on D_o only. The algorithm proceeds as follows.

1. Collect the scores in an $n_u \times n_c$ matrix, $\mathbf{A}$, where each row corresponds to a unit.

2. For $i \in \{1, \ldots, n_b\}$, form matrix $\mathbf{A}_i$ by sampling, with replacement, n_u rows from $\mathbf{A}$.
3. For each $\mathbf{A}_i$, compute $D_o^{(i)}$ using the same distance function d^2 that was used to compute $\hat{\alpha}$.
4. For each $D_o^{(i)}$, compute $\hat{\alpha}_i = 1 - D_o^{(i)} / D_e$.

The resulting collection $\{\hat{\alpha}_1, \ldots, \hat{\alpha}_{n_b}\}$ is a bootstrap sample for $\hat{\alpha}$, sample quantiles of which are estimated confidence limits for α.

We carried out a number of realistic simulation experiments and found that this approach to interval estimation performs well in a wide variety of circumstances. When the true distribution of $\hat{\alpha}$ is (at least approximately) symmetric, Gwet's closed-form expression for $\hat{\mathbb{V}}(\hat{\alpha})$, which is implemented (for categorical data only) in package **irrCAC**, also performs well. By contrast, we found that the bootstrapping procedure recommended by Krippendorff (2016), which is implemented in packages **kripp.boot** and **icr**, generally performs rather poorly, producing intervals that are far too narrow (e.g., 95% intervals achieve 74% coverage).

Robustness and interpretation

For some levels of measurement, one may, in the interest of robustness, be tempted to replace squares with absolute values (in the distance function d^2). This would be advantageous if one aimed to do hypothesis testing. But for Krippendorff's α, using absolute values instead of squares proves disastrous, for the resulting estimator $\hat{\alpha}$ is substantially negatively biased and tends to lead to erroneous inference regarding agreement. All is not lost, however, since package **krippendorffsalpha** provides a means of investigating the influence on $\hat{\alpha}$ of any unit or coder (see the next section for examples).

Illustrations

Here we illustrate the use of **krippendorffsalpha** by applying Krippendorff's α to a couple of datasets. We will interpret the results according to the ranges given in Table 2, but we suggest—as do Krippendorff and others (Artstein and Poesio, 2008; Landis and Koch, 1977)—that an appropriate reliability threshold may be context-dependent.

Range of Agreement	Interpretation
$\alpha \leq 0.2$	Slight Agreement
$0.2 < \alpha \leq 0.4$	Fair Agreement
$0.4 < \alpha \leq 0.6$	Moderate Agreement
$0.6 < \alpha \leq 0.8$	Substantial Agreement
$\alpha > 0.8$	Near-Perfect Agreement

Table 2: Guidelines for interpreting values of an agreement coefficient.

Nominal data analyzed previously by Krippendorff

Consider the following data, which appear in (Krippendorff, 2013). These are nominal values (in $\{1, \ldots, 5\}$) for twelve units and four coders. The dots represent missing values.

Note that the scores for all units except the sixth are constant or nearly so. This suggests near-perfect agreement, and so we should expect $\hat{\alpha}$ to be greater than 0.8.

To apply Krippendorff's α to these data, first we load package **krippendorffsalpha**.

```
R> library(krippendorffsalpha)

krippendorffsalpha: Measuring Agreement Using Krippendorff's Alpha Coefficient
Version 1.1 created on 2021-01-13.
copyright (c) 2020-2021, John Hughes
For citation information, type citation("krippendorffsalpha").
Type help(package = krippendorffsalpha) to get started.
```

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}
c_1	1	2	3	3	2	1	4	1	2	•	•	•
c_2	1	2	3	3	2	2	4	1	2	5	•	3
c_3	•	3	3	3	2	3	4	2	2	5	1	•
c_4	1	2	3	3	2	4	4	1	2	5	1	•

Figure 2: Some example nominal outcomes for twelve units and four coders, with seven missing values.

Now, we create the dataset as a matrix such that each row corresponds to a unit and each column corresponds to a coder.

```
R> 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)
R> nominal

      [,1] [,2] [,3] [,4]
 [1,]    1    1   NA    1
 [2,]    2    2    3    2
 [3,]    3    3    3    3
 [4,]    3    3    3    3
 [5,]    2    2    2    2
 [6,]    1    2    3    4
 [7,]    4    4    4    4
 [8,]    1    1    2    1
 [9,]    2    2    2    2
[10,]   NA    5    5    5
[11,]   NA   NA    1    1
[12,]   NA    3   NA   NA
```

Next, we apply Krippendorff's α for the nominal level of measurement. If argument `level` is set to `"nominal"`, the discrete metric $d^2(x, y) = 1\{x \neq y\}$ is used by default. We do a bootstrap with sample size $n_b = 1{,}000$ (argument `confint` defaults to `TRUE`, and control parameter `bootit` defaults to 1,000). We set control parameter `parallel` equal to `FALSE` because the dataset is too small to warrant parallelization of the bootstrap computation. Finally, we set argument `verbose` equal to `TRUE` so that a progress bar is shown during the bootstrap computation. The computation took less than one second.

```
R> set.seed(42)
R> fit.full = krippendorffs.alpha(nominal, level = "nominal", control = list(parallel = FALSE),
+                                 verbose = TRUE)

  |++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=00s
```

As is customary in R, one can view a summary by passing the fit object to `summary.krippendorffsalpha`, an S3 method. If `krippendorffs.alpha` was called with `confint = TRUE`, `summary` displays a 95% confidence interval by default. The confidence level can be specified using argument `conf.level`. In any case, the quantile method (Davison and Hinkley, 1997) is used to estimate the confidence limits. Any arguments passed to `summary.krippendorffsalpha` via `...` are passed on to R's `quantile` function. This allows the user to control, for example, how the sample quantiles are computed.

```
R> summary(fit.full)

Krippendorff's Alpha

Data: 12 units x 4 coders

Call:
```

```
krippendorffs.alpha(data = nominal, level = "nominal", verbose = TRUE,
    control = list(parallel = FALSE))

Control parameters:

parallel FALSE
bootit   1000

Results:

      Estimate  Lower Upper
alpha   0.7429 0.4644     1
```

We see that $\hat{\alpha} = 0.74$ and $\alpha \in (0.46, 1.00)$. This point estimate indicates only substantial agreement, which is not what we expected. At least the interval is consistent with near-perfect agreement, but we should not take this interval too seriously since the interval is rather wide (owing to the small size of the dataset).

Perhaps the substantial disagreement for the sixth unit was influential enough to yield $\hat{\alpha} \leq 0.8$. We can use `influence.krippendorffsalpha`, another S3 method, to investigate. This function, like other R versions of `influence` (e.g., `influence.lm`, `influence.glm`), computes DFBETA statistics (Young, 2017), as illustrated below.

```
R> (inf.6 = influence(fit.full, units = 6))

$dfbeta.units
         6
-0.1141961
```

Leaving out the sixth unit yields a DFBETA statistic of -0.11, which implies that $\hat{\alpha}$ would have been 0.86. This is consistent with our initial hypothesis.

```
R> fit.full$alpha.hat - inf.6$dfbeta.units

    alpha
0.8571429
```

Let us call `krippendorffs.alpha` again to get a new interval.

```
R> fit.sub = krippendorffs.alpha(nominal[-6, ], level = "nominal",
+                                control = list(parallel = FALSE))
confint(fit.sub)

    0.025     0.975
0.6616541 1.0000000
```

We see that excluding the sixth unit leads to $\alpha \in (0.66, 1.00)$. The new 95% interval was returned by S3 method `confint.krippendorffsalpha`, whose `level` argument defaults to 0.95, in keeping with R's other `confint` methods. Note that `confint.krippendorffsalpha`, like `summary.krippendorffsalpha`, passes any `...` arguments on to the `quantile` function.

We conclude this example by producing a visual display of our results (Figure 3). The figure was produced via a call to S3 method `plot.krippendorffsalpha`, which in turn calls `hist` and `abline`, and does not show a kernel density estimate. Function `plot.krippendorffsalpha` is capable of producing highly customized plots; see the package documentation for details. Since $\hat{\alpha}$ is close to 1 and the dataset is small, the bootstrap distribution is substantially skewed to the left. Thus, these data provide a textbook example of the importance of bootstrapping.

```
R> plot(fit.sub, xlim = c(0, 1), xlab = "Bootstrap Estimates", main = "Nominal Data",
+       density = FALSE)
```

Since the dataset used in this example has missing values, we take this opportunity to explain how the package handles missingness. First, the scores for a given unit of analysis are included in the computation only if two or more scores are present for that unit. Otherwise, the unit's row of the data matrix is simply ignored. Second, if two or more scores are present for a given unit, each `NA` for that unit is ignored in the computations for that row. This is handled both by the loop (adjusted denominator) and by the distance function, which should return 0 if either of its arguments is `NA`. In the next example, we illustrate this by way of a user-defined distance function, and of course, the package's built-in distance functions take the same approach.

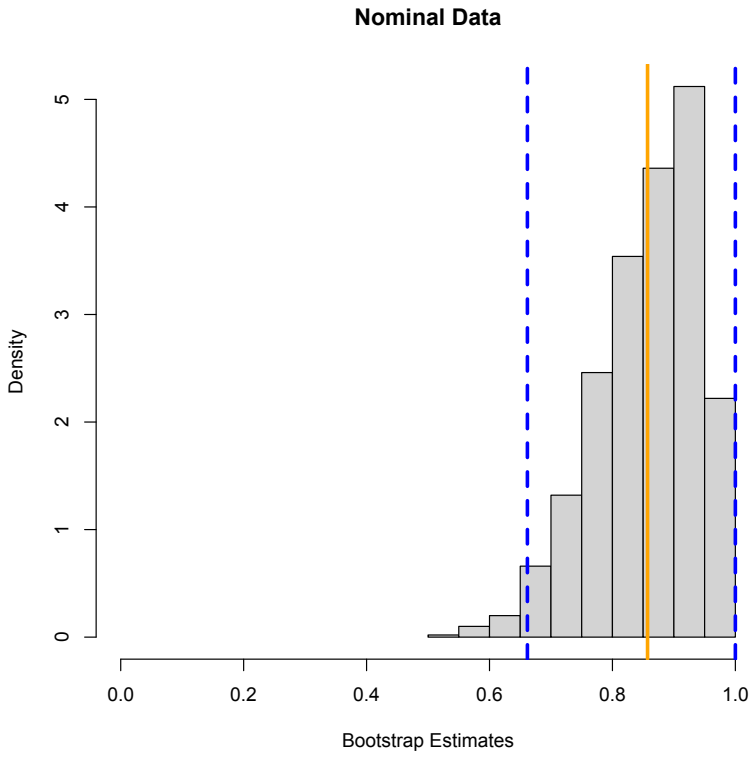

Figure 3: A plot of the results from our analysis of the nominal data. The histogram shows the bootstrap sample, the solid orange line marks the value of $\hat{\alpha}$, and the dashed blue lines mark the 95% confidence limits.

Interval data from an imaging study of hip cartilage

The data for this example, some of which appear in Figure 4, are 323 pairs of T2* relaxation times (a magnetic resonance quantity) for femoral cartilage (Nissi et al., 2015) in patients with femoroacetabular impingement (Figure 5), a hip condition that can lead to osteoarthritis. One measurement was taken when a contrast agent was present in the tissue, and the other measurement was taken in the absence of the agent. The aim of the study was to determine whether raw and contrast-enhanced T2* measurements agree closely enough to be interchangeable for the purpose of quantitatively assessing cartilage health.

	u_1	u_2	u_3	u_4	u_5	...	u_{319}	u_{320}	u_{321}	u_{322}	u_{323}
c_1	27.3	28.5	29.1	31.2	33.0	...	19.7	21.9	17.7	22.0	19.5
c_2	27.8	25.9	19.5	27.8	26.6	...	18.3	23.1	18.0	25.7	21.7

Figure 4: Raw and contrast-enhanced T2* values for femoral cartilage.

First, we load the cartilage data, which are included in the package. The cartilage data are stored in a data frame; we convert the data frame to a matrix, which is the format required by `krippendorffs.alpha`.

```
R> data(cartilage)
R> cartilage = as.matrix(cartilage)
```

Now, we compute $\hat{\alpha}$ for the interval level of measurement, i.e., squared Euclidean distance. We also produce a bootstrap sample of size 10,000. Since this dataset is much larger than the dataset analyzed in the preceding section, we parallelize the bootstrap computation. We use three CPU cores (of the four available on the author's computer). Setting argument `verbose` to `TRUE` causes the fitting function to display a progress bar once again. The computation took five seconds to complete.

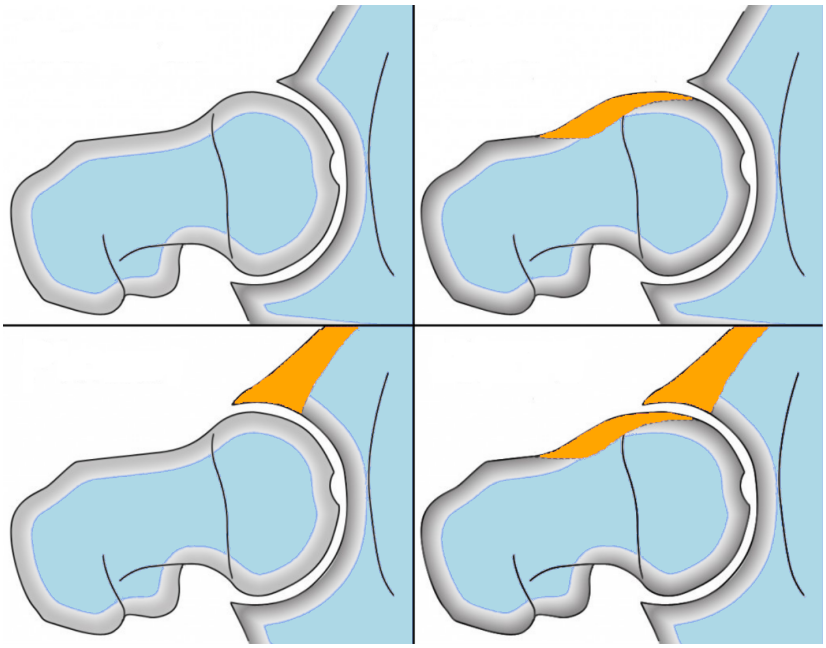

Figure 5: An illustration of femoroacetabular impingement (FAI). Top left: normal hip joint. Top right: cam type FAI (deformed femoral head). Bottom left: pincer type FAI (deformed acetabulum). Bottom right: mixed type (both deformities present).

```
R> set.seed(12)
R> fit.sed = krippendorffs.alpha(cartilage, level = "interval", verbose = TRUE,
+                                control = list(bootit = 10000, parallel = TRUE,
+                                nodes = 3))

Control parameter 'type' must be "SOCK", "PVM", "MPI", or "NWS". Setting it to "SOCK".

  |++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=05s
```

A call of function `summary.krippendorffsalpha` produced the output shown below.

```
R> summary(fit.sed)

Krippendorff's Alpha

Data: 323 units x 2 coders

Call:

krippendorffs.alpha(data = cartilage, level = "interval", verbose = TRUE,
    control = list(bootit = 10000, parallel = TRUE, nodes = 3))

Control parameters:

bootit   10000
parallel TRUE
nodes    3
type     SOCK

Results:

      Estimate Lower  Upper
alpha   0.8369 0.808 0.8648
```

We see that $\hat{\alpha} = 0.84$ and $\alpha \in (0.81, 0.86)$. Thus these data suggest that raw T2* measurements agree almost perfectly with contrast-enhanced T2* measurements, perhaps rendering gadolinium-based contrast agents (GBCAs) unnecessary in T2*-based cartilage assessment. This finding could

have clinical significance since the use of GBCAs is not free of risk to patients, especially pregnant women and patients with impaired kidney function. For much additional information regarding the potential risks associated with the use of GBCAs, we refer the interested reader to the University of California, San Francisco's policy on MRI with contrast: https://radiology.ucsf.edu/patient-care/patient-safety/contrast/mri-with-contrast-gadolinium-policy.

Figure 6 provides a visual display of the cartilage results. The histogram and kernel density estimate show the expected large-sample behavior of $\hat{\alpha}$, i.e., the estimator is approximately Gaussian-distributed and has a small variance.

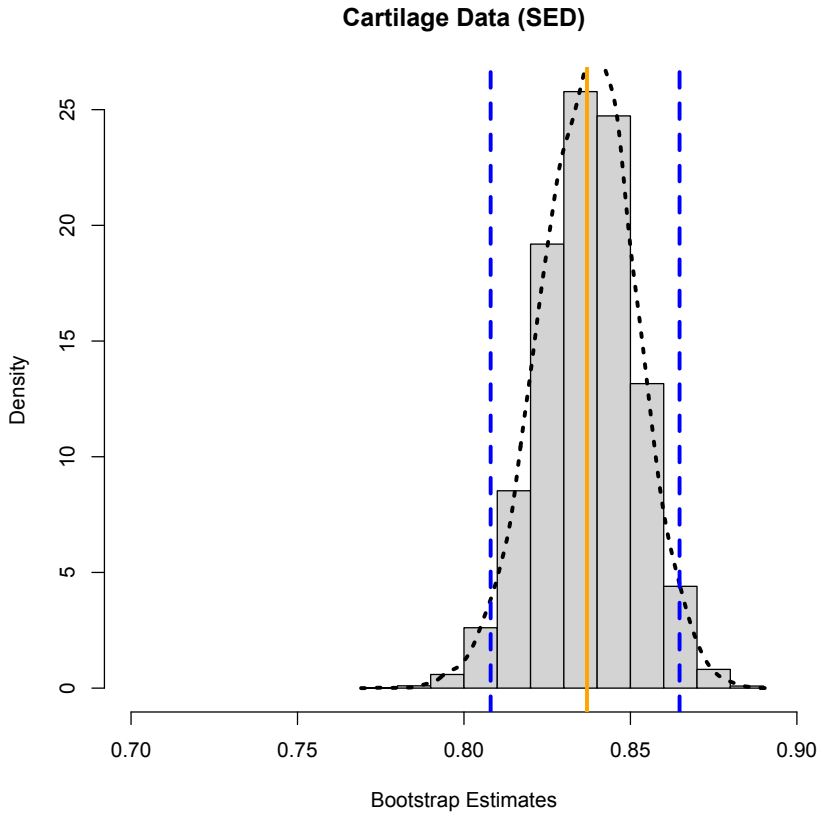

Figure 6: A plot of the results from our analysis of the cartilage data. The histogram and kernel density estimate (dotted black curve) show the bootstrap sample, the solid orange line marks the value of $\hat{\alpha}$, and the dashed blue lines mark the 95% confidence limits.

We mentioned above that attempting to robustify Krippendorff's α by using absolute values in place of squares may prove problematic. This is evident for the cartilage data, as we now demonstrate.

First, define a new distance function as follows. Note that any user-defined distance function must deal explicitly with NAs if the data at hand exhibit missingness. There are no missing values in the cartilage data, but we illustrate the handling of NA anyway.

```
R> L1.dist = function(x, y)
+ {
+    d = abs(x - y)
+    if (is.na(d))
+        d = 0
+    d
+ }
```

Now we call `krippendorffs.alpha`, supplying our new distance function via the `level` argument.

```
R> fit.L1 = krippendorffs.alpha(cartilage, level = L1.dist, verbose = TRUE,
+                               control = list(bootit = 10000, parallel = TRUE,
+                               nodes = 3))

Control parameter 'type' must be "SOCK", "PVM", "MPI", or "NWS". Setting it to "SOCK".
```

```
  |++++++++++++++++++++++++++++++++++++++++++++++++++| 100% elapsed=05s
```

The results are summarized below. These results strongly suggest that only moderate to substantial agreement exists between raw T2* measurements and contrast-enhanced T2* measurements. This contradicts not only our α_{SED} analysis but also a Sklar's ω analysis that assumed a non-central t marginal distribution to accommodate slight asymmetry.

```
R> summary(fit.L1)

Krippendorff's Alpha

Data: 323 units x 2 coders

Call:

krippendorffs.alpha(data = cartilage, level = L1.dist, verbose = TRUE,
    control = list(bootit = 10000, parallel = TRUE, nodes = 3))

Control parameters:

bootit   10000
parallel TRUE
nodes    3
type     SOCK

Results:

      Estimate  Lower Upper
alpha   0.6125 0.5761 0.648
```

Summary and discussion

In this article, we described Krippendorff's α methodology for measuring agreement and illustrated the use of R package **krippendorffsalpha**. We first established α's context among statistical procedures. Specifically, the one-way mixed-effects ANOVA model provides a natural, intuitive genesis for α as the intraclass correlation coefficient. This form of α can be generalized in a parametric fashion to arrive at Sklar's ω, or in a nonparametric fashion to arrive at the form of α presented by Krippendorff, which is a special case of the multiresponse permutation procedure.

We demonstrated the use of **krippendorffsalpha** version 1.1 by analyzing two datasets: a nominal dataset previously analyzed by Krippendorff, and a sample of raw and contrast-enhanced T2* values from an MRI study of hip cartilage. These analyses highlighted the benefits of the package, which include the use of S3 methods, parallel bootstrap computation, support for user-defined distance functions, and a means of identifying influential units and/or coders.

Computational details

The results in this paper were obtained using R 4.0.3 for macOS and the **pbapply** 1.4-2 package. R itself and all packages used (save **kripp.boot**) are available from the Comprehensive R Archive Network (CRAN) at http://CRAN.R-project.org. Package **krippendorffsalpha** may be downloaded from CRAN or from the author's GitHub repository, which can be found at https://github.com/drjphughesjr/krippendorffsalpha. Information about the author's other R packages can be found at http://www.johnhughes.org/software.html.

John Hughes
Department of Statistics
The Pennsylvania State University
University Park, PA
USA
drjphughesjr@gmail.com

Bibliography

R. Artstein and M. Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008. [p417]

L. M. Bregman. The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967. [p415]

A. C. Davison and D. V. Hinkley. *Bootstrap Methods and their Application*, volume 1. Cambridge University Press, 1997. [p418]

M. Fowler, C. Kobryn, and K. Scott. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Object Technology Series. Addison-Wesley, 2004. [p413]

M. Gamer, J. Lemon, and I. F. P. Singh. *irr: Various Coefficients of Interrater Reliability and Agreement*, 2012. URL https://CRAN.R-project.org/package=irr. R package version 0.84. [p413]

K. L. Gwet. *Handbook of Inter-Rater Reliability: The Definitive Guide to Measuring the Extent of Agreement Among Raters*. Advanced Analytics, LLC, Gaithersburg, MD, 4th edition, 2014. [p413]

K. L. Gwet. On Krippendorff's alpha coefficient. October 2015. [p413, 417]

K. L. Gwet. *irrCAC: Computing Chance-Corrected Agreement Coefficients (CAC)*, 2019. URL https://CRAN.R-project.org/package=irrCAC. R package version 1.0. [p413]

A. F. Hayes and K. Krippendorff. Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, 1(1):77–89, 2007. [p413, 414]

J. Hughes. Sklar's Omega: A Gaussian copula-based framework for assessing agreement. *arXiv: Methodology*, 2018. [p416]

R. Ihaka and R. Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5:299–314, 1996. [p413]

K. Krippendorff. Computing Krippendorff's alpha-reliability. Technical report, University of Pennsylvania, 2013. [p413, 417]

K. Krippendorff. Bootstrapping distributions for Krippendorff's alpha. Technical report, The University of Pennsylvania, 2016. [p417]

J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174, 1977. [p417]

P. W. Mielke and K. J. Berry. *Permutation Methods: A Distance Function Approach*. Springer Series in Statistics. Springer, New York, 2nd edition, 2007. [p415, 416]

M. J. Nissi, S. Mortazavi, J. Hughes, P. Morgan, and J. Ellermann. 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, 2015. [p420]

P. Proutskova and M. Gruszczynski. *kripp.boot: Bootstrap Krippendorff's Alpha Intercoder Reliability Statistic*, 2020. URL https://github.com/MikeGruz/kripp.boot. R package version 1.0.0. [p413]

N. Ravishanker and D. Dey. *A First Course in Linear Model Theory*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2001. [p414]

A. Staudt and P. L'Ecuyer. *icr: Compute Krippendorff's Alpha*, 2020. URL https://CRAN.R-project.org/package=icr. R package version 0.6.2. [p413]

P. Xue-Kun Song. Multivariate dispersion models generated from Gaussian copula. *Scandinavian Journal of Statistics*, 27(2):305–320, 2000. [p416]

D. S. Young. *Handbook of Regression Methods*. CRC Press, 2017. [p419]