Krippendorff’s alpha

Cooper Stansbury

12 May, 2020

Software Dependencies

```
library("irr")
library("boot")
```

Loading Data

Note that blanks are explicitly represented im this example, though the software documentation for `kripp.alpha()` in the “irr” package (shown in Appendix I) shows non-annotated blanks as `NA`. This makes a substantial difference (shown below). This data structure must be transposed before input into the `kripp.alpha()` function.

```
rater.data <- read.csv("ANNOTATIONS.csv")
head(rater.data)
```

```
##   KATHLEEN KAYCEE   LIZ
## 1    blank  blank blank
## 2    blank  blank blank
## 3    blank      A blank
## 4    blank      A blank
## 5    blank  blank blank
## 6    blank  blank blank
```

Krippendorff’s alpha

This is the `kripp.alpha()` function from: https://cran.r-project.org/web/packages/irr/irr.pdf run on the data where blanks are explcitly coded (not `NA`). We provide both nominal and ordinal estimates. From the supplementary material of the foillowing publication we find this a note indicating the lack of confidence intervals for this function, as well as small estimation errors due to the lack of `NA` values. It is worth noting that similar behavior is seen in the Python package `krippendorff`.

1. Zapf A, Castell S, Morawietz L, Karch A. Measuring inter-rater reliability for nominal data – which coefficients and confidence intervals are appropriate? BMC Medical Research Methodology. 2016 Aug 5;16(1):93.

"In R (R Core Team, Vienna, Austria) there is the package irr (version 0.84) from Gamer et al. [2], which calculates Fleiss' K and Krippendorff's alpha, but both without confidence intervals. There is a small error in the estimation of the coincidence matrix for Krippendorff's alpha if there are no missing values. In the upcoming actualized version this error will be corrected (personal communication). An R-program for the calculation of Krippendorffs alpha with the standard bootstrap confidence interval as applied by us was written by Gruszczynski and can be downloaded via GitHub [3]."

```
kripp.alpha(t(rater.data), "nominal")
```

```
## Warning in kripp.alpha(t(rater.data), "nominal"): NAs introduced by coercion
```

```
##  Krippendorff's alpha
##
##  Subjects = 9951
##    Raters = 3
##     alpha = 0.584
```

```
kripp.alpha(t(rater.data), "ordinal")
```

```
## Warning in kripp.alpha(t(rater.data), "ordinal"): NAs introduced by coercion
```

```
##  Krippendorff's alpha
##
##  Subjects = 9951
##    Raters = 3
##     alpha = 0.607
```

Confidence Intervals

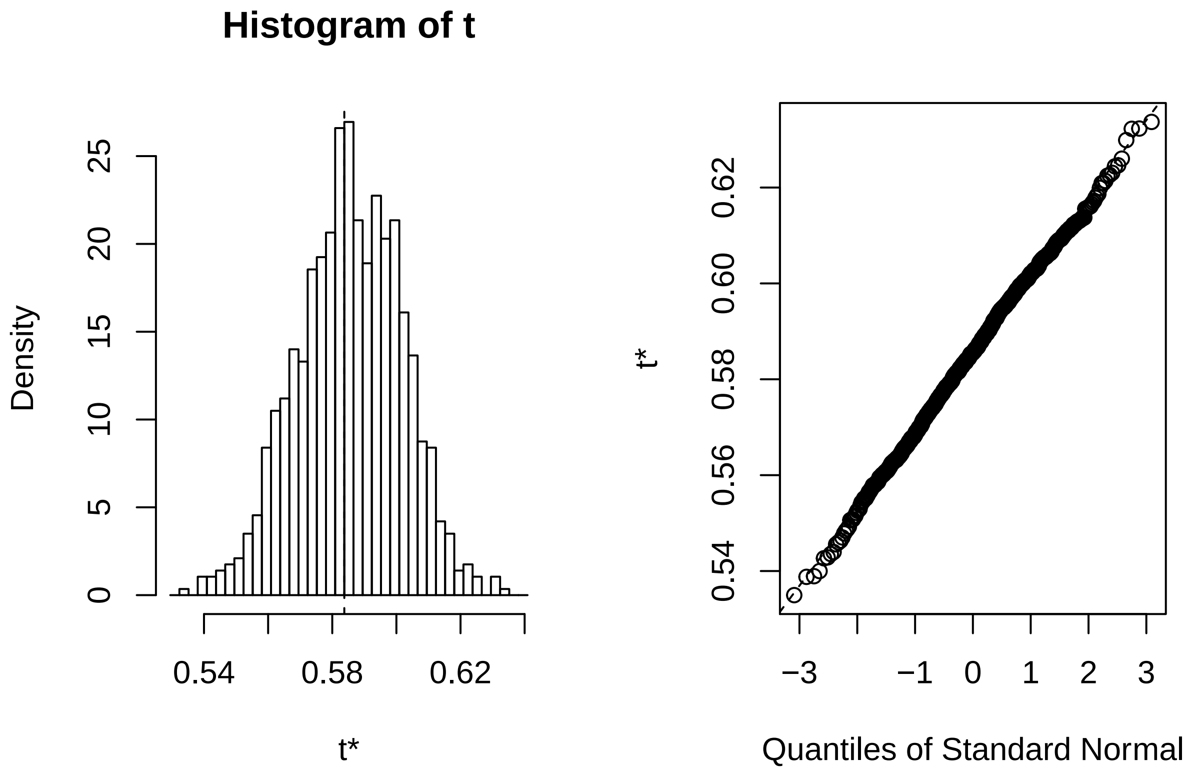
We'll rely on this post (https://stackoverflow.com/questions/41944703) to using bootstrapiing to build confidence intervals, since the "irr" function doesn't return them. Below is the bootstrapped confidence interval for Krippendorff's alpha:

```
alpha.boot <- function(d,w) {
      data <- t(d[w,])
      kripp.alpha(data)$value
}

b <- boot(data = rater.data, statistic = alpha.boot, R = 1000)
bs
```

```
## Error in eval(expr, envir, enclos): object 'bs' not found
```

```
plot(b)
```



```
boot.ci(b, type = "perc")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = b, type = "perc")
##
## Intervals :
## Level     Percentile
## 95%   ( 0.5530,  0.6156 )
## Calculations and Intervals on Original Scale
```

Differences in Coding

Here we repeat the above, but with blank annotations represented as NA values. Note the surprising results when done this way.

```
rater.data <- read.csv("ANNOTATIONS_NA_CODED.csv")
head(rater.data)
```

```
##   KATHLEEN KAYCEE LIZ
## 1       NA     NA  NA
## 2       NA      NA  NA
## 3       NA       2  NA
## 4       NA       2  NA
## 5       NA      NA  NA
## 6       NA      NA  NA
```

```r
kripp.alpha(t(rater.data), "nominal")
```

```
##  Krippendorff's alpha
##
##  Subjects = 9951
##    Raters = 3
##     alpha = 0.185
```

```r
kripp.alpha(t(rater.data), "ordinal")
```

```
##  Krippendorff's alpha
##
##  Subjects = 9951
##    Raters = 3
##     alpha = 0.185
```

Appendix I From Documentation

Example from: https://www.rdocumentation.org/packages/irr/versions/0.84.1/topics/kripp.alpha

```r
nmm<-matrix(c(1,1,NA,1,2,2,3,2,3,3,3,3,3,3,3,3,2,2,2,2,1,2,3,4,4,4,4,4,
1,1,2,1,2,2,2,2,NA,5,5,5,NA,NA,1,1,NA,NA,3,NA),nrow=4)
# first assume the default nominal classification
kripp.alpha(nmm)
```

```
##  Krippendorff's alpha
##
##  Subjects = 12
##    Raters = 4
##     alpha = 0.743
```

```r
# now use the same data with the other three methods
kripp.alpha(nmm,"ordinal")
```

```
##  Krippendorff's alpha
##
##  Subjects = 12
##    Raters = 4
##     alpha = 0.815
```

```r
kripp.alpha(nmm,"interval")
```

```
##  Krippendorff's alpha
##
##  Subjects = 12
##    Raters = 4
##     alpha = 0.849
```

```
kripp.alpha(nmm,"ratio")
```

```
##  Krippendorff's alpha
##
##  Subjects = 12
##    Raters = 4
##     alpha = 0.797
```