Interrater Reliability

```
# Define a function for formatting numbers
comma <- function(x, d = 2) format(x, digits = d, big.mark = ",")
```

```
library(psych)
library(lme4)
```

```
Warning in check_dep_version(): ABI version mismatch:
lme4 was built with Matrix ABI version 2
Current Matrix ABI version is 1
Please re-install lme4 from source or restore original 'Matrix' package
```

```
library(tidyverse)
library(plotly)
```

Interrater Reliability

Data analyzed in this paper: https://www.mdpi.com/2076-3425/12/8/1011/

```
slp_dat <- read.csv("https://osf.io/download/p9gqk/")
slp_vas_wide <- slp_dat |>
    select(slpID, Speaker, slp_VAS) |>
    pivot_wider(names_from = slpID, values_from = slp_VAS)
head(slp_vas_wide)
```

```
# A tibble: 6 x 22
  Speaker slp10 slp11 slp13 slp14 slp15 slp16  slp17 slp18 slp19  slp2 slp20
  <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
1 AF1      96.3 100   96.6  99.3  100   12.6  100     88.2 52.6   93.4 67.4
```

```
2 AF9      12.3  34.7  0.86  3.52  12.6  0      6.69   0    7.42  19.2  9.03
3 ALSF6    NA    30.8  5.73 73.9   73.9  4.52  45.1   41.9 26.1   39.8 17.4
4 ALSF7    94.0  96.3 61.3  64.4  100    7.74  69.9   88.2 37.4   43.3 59.4
5 ALSF9    29.5  52.2 43.6  46.8   60.6 23.9   25.2   91.6 37.1   58.7 19.4
6 ALSM1    96.3  NA   29.2  69.4   99.4 82.3   70.2   79.7 78.4   87.4 36.4
# i 10 more variables: slp21 <dbl>, slp22 <dbl>, slp23 <dbl>, slp25 <dbl>,
#   slp3 <dbl>, slp4 <dbl>, slp5 <dbl>, slp6 <dbl>, slp8 <dbl>, slp9 <dbl>
```

Plot

```
p <- ggplot(slp_dat, aes(x = slpID, y = slp_VAS, group = Speaker, color = Speaker)) +
    geom_line(alpha = 0.5) +
    labs(x = "Rater", y = "VAS Score") +
    theme_bw()
ggplotly(p)
```

For Two Raters

We'll first select just the first two raters.

```
slp_2rater <- slp_vas_wide |>
    select(slp14, slp15)
```

Nominal Agreement

To compute nominal agreement, we need to consider the ratings between two raters to be exactly the same. If we go by that definition, we have

```
with(slp_2rater, mean(slp14 == slp15))
```

```
[1] 0.1
```

If we instead relax the definition a little bit and say that agreement is reached if it is the same after rounding,

```r
slp_2round <- round(slp_2rater / 10)
slp_2round <- lapply(slp_2round, FUN = factor, levels = 0:10)
# Contingency table
table(slp_2round)
```

```
     slp15
slp14 0 1 2 3 4 5 6 7 8 9 10
   0  0 1 0 0 0 0 0 0 0 0  0
   1  0 0 0 0 0 0 0 0 0 0  0
   2  0 0 0 0 0 0 0 0 0 0  0
   3  0 0 0 0 0 0 0 0 0 0  0
   4  0 0 0 0 0 0 0 0 0 0  0
   5  0 0 0 0 0 0 1 0 0 0  0
   6  0 0 0 0 0 1 0 0 0 0  2
   7  0 0 0 0 0 0 0 1 0 0  1
   8  0 0 0 0 0 0 0 0 0 2  0
   9  0 0 0 0 0 0 0 0 1 0  2
   10 0 0 0 0 0 0 0 1 0 0  7
```

```r
p0 <- with(slp_2round, mean(slp14 == slp15))
p0
```

```
[1] 0.4
```

Cohen's Kappa

$$\kappa = \frac{P_0 - P_c}{1 - P_c}$$

$$P_c = \frac{1}{N^2} \sum_{i=1}^{c} (n_{i+})(n_{+i})$$

```r
slp_2tab <- table(slp_2round)
pc <- sum(colSums(slp_2tab) * rowSums(slp_2tab)) / sum(slp_2tab)^2
(kappa <- (p0 - pc) / (1 - pc))
```

```
[1] 0.1666667
```

Drawback: Kappa tends to be small when scores are unevenly distributed (e.g., most scores belong to certain categories). It is certainly the case above.

Multiple Raters

Coefficient α

Treat each rater as an "item."

```
psych::alpha(slp_vas_wide[-1])
```

```
Number of categories should be increased  in order to count frequencies.

Warning in cor.smooth(r): Matrix was not positive definite, smoothing was done

In smc, smcs > 1 were set to 1.0

In smc, smcs > 1 were set to 1.0
In smc, smcs > 1 were set to 1.0
In smc, smcs > 1 were set to 1.0

In smc, smcs < 0 were set to .0
In smc, smcs < 0 were set to .0
In smc, smcs < 0 were set to .0
In smc, smcs < 0 were set to .0

In smc, smcs > 1 were set to 1.0

In smc, smcs < 0 were set to .0

In smc, smcs > 1 were set to 1.0
In smc, smcs > 1 were set to 1.0
In smc, smcs > 1 were set to 1.0
In smc, smcs > 1 were set to 1.0

In smc, smcs < 0 were set to .0

In smc, smcs > 1 were set to 1.0

In smc, smcs < 0 were set to .0
```

```
In smc, smcs > 1 were set to 1.0

In smc, smcs < 0 were set to .0

In smc, smcs > 1 were set to 1.0
In smc, smcs > 1 were set to 1.0
In smc, smcs > 1 were set to 1.0
In smc, smcs > 1 were set to 1.0
In smc, smcs > 1 were set to 1.0

In smc, smcs < 0 were set to .0

In smc, smcs > 1 were set to 1.0


Reliability analysis
Call: psych::alpha(x = slp_vas_wide[-1])

  raw_alpha std.alpha G6(smc) average_r S/N    ase mean sd median_r
      0.98      0.98       1      0.69  46 0.0076   62 25      0.7

    95% confidence boundaries
         lower alpha upper
Feldt     0.96  0.98  0.99
Duhachek  0.96  0.98  0.99

 Reliability if an item is dropped:
      raw_alpha std.alpha G6(smc) average_r S/N var.r med.r
slp10      0.97      0.98    0.99      0.68  43 0.014  0.70
slp11      0.97      0.98    0.99      0.68  43 0.014  0.70
slp13      0.97      0.98    0.99      0.69  44 0.015  0.70
slp14      0.97      0.98    0.99      0.69  44 0.015  0.70
slp15      0.97      0.98    0.99      0.69  44 0.014  0.70
slp16      0.98      0.98    0.99      0.70  46 0.014  0.71
slp17      0.97      0.98    0.99      0.68  43 0.015  0.69
slp18      0.97      0.98    0.99      0.69  45 0.013  0.70
slp19      0.97      0.98    0.99      0.69  45 0.014  0.71
slp2       0.97      0.98    1.00      0.69  45 0.014  0.71
slp20      0.97      0.98    0.99      0.68  43 0.015  0.69
slp21      0.97      0.98    0.99      0.68  43 0.014  0.69
slp22      0.97      0.98    0.99      0.69  44 0.015  0.70
```

```
slp23      0.97      0.98    0.99      0.68  43 0.015  0.69
slp25      0.97      0.98    0.99      0.69  45 0.014  0.71
slp3       0.98      0.98    0.99      0.70  46 0.014  0.71
slp4       0.97      0.98    0.99      0.69  45 0.014  0.71
slp5       0.97      0.98    0.99      0.69  45 0.015  0.70
slp6       0.98      0.98    0.99      0.70  47 0.010  0.71
slp8       0.97      0.98    0.99      0.69  44 0.015  0.70
slp9       0.97      0.98    1.00      0.68  43 0.014  0.69

 Item statistics
       n raw.r std.r r.cor r.drop mean sd
slp10 18  0.89  0.89  0.90   0.86   81 28
slp11 18  0.89  0.90  0.90   0.90   69 33
slp13 19  0.86  0.86  0.85   0.86   54 33
slp14 20  0.82  0.84  0.84   0.81   80 25
slp15 20  0.81  0.83  0.83   0.80   85 22
slp16 20  0.77  0.76  0.76   0.73   50 42
slp17 20  0.90  0.91  0.91   0.90   63 34
slp18 20  0.82  0.80  0.77   0.79   70 30
slp19 19  0.83  0.82  0.82   0.81   56 25
slp2  20  0.83  0.82  0.82   0.80   64 32
slp20 20  0.90  0.90  0.90   0.89   49 25
slp21 19  0.91  0.91  0.90   0.91   59 25
slp22 20  0.86  0.86  0.86   0.84   50 27
slp23 20  0.89  0.89  0.89   0.86   71 32
slp25 20  0.79  0.79  0.79   0.77   79 28
slp3  19  0.76  0.75  0.76   0.74   52 40
slp4  20  0.79  0.79  0.78   0.77   47 27
slp5  19  0.82  0.82  0.82   0.81   43 23
slp6  13  0.80  0.71  0.71   0.54   76 23
slp8  20  0.86  0.87  0.87   0.86   53 31
slp9  19  0.89  0.90  0.90   0.90   58 23
```

Intraclass Correlation

One-way ANOVA for nested design

Let's say we have data where each person is rated by two different raters:

```
slp_vas_nested <- slp_dat |>
    mutate(SpeakerID = as.numeric(as.factor(Speaker))) |>
    # Select only 10 speakers
```

```
    filter(SpeakerID <= 10) |>
    group_by(Speaker) |>
    # Filter specific raters
    filter(row_number() %in% (SpeakerID[1] * 2 - (1:0)))
```

We have a design with raters nested within ratees. With this design, we cannot distinguish rater effect from random error. We can now run a one-way random-effect ANOVA, which is the same as a random-intercept multilevel model:

```
m1 <- lmer(slp_VAS ~ 1 + (1 | Speaker), data = slp_vas_nested)
summary(m1)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: slp_VAS ~ 1 + (1 | Speaker)
   Data: slp_vas_nested

REML criterion at convergence: 179.6

Scaled residuals:
     Min       1Q   Median       3Q      Max
-1.50049 -0.77609 -0.05948  0.87584  1.35589

Random effects:
 Groups   Name        Variance Std.Dev.
 Speaker  (Intercept) 308.9    17.58
 Residual             816.8    28.58
Number of obs: 19, groups:  Speaker, 10

Fixed effects:
            Estimate Std. Error t value
(Intercept)   53.613      8.627   6.214
```

```
# Extract variance components (Ratee, error)
vc_m1 <- as.data.frame(VarCorr(m1))
```

ICC for single rating: $\frac{\sigma^2_{\text{ratee}}}{\sigma^2_{\text{ratee}}+\sigma^2_E}$

```
vc_m1$vcov[1] / (vc_m1$vcov[1] + vc_m1$vcov[2])
```

```
[1] 0.2744076
```

ICC for average rating: $\frac{\sigma^2_{\text{ratee}}}{\sigma^2_{\text{ratee}}+\sigma^2_E/k}$, where k is the number of raters per ratee

```
vc_m1$vcov[1] / (vc_m1$vcov[1] + vc_m1$vcov[2] / 2)
```

```
[1] 0.4306434
```

Two-way ANOVA

> - For consistency (relative decision), rater effect is not error, because the rater bias is applying to everyone and does not change the rank order.
> - For agreement (absolute decision), rater effect is error as it changes the absolute scores.

```
m2 <- lmer(slp_VAS ~ 1 + (1 | Speaker) + (1 | slpID), data = slp_dat)
summary(m2)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: slp_VAS ~ 1 + (1 | Speaker) + (1 | slpID)
   Data: slp_dat

REML criterion at convergence: 3544.1

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.3281 -0.5813  0.0862  0.6611  2.5747

Random effects:
 Groups   Name        Variance Std.Dev.
 slpID    (Intercept) 132.8    11.53
 Speaker  (Intercept) 585.7    24.20
 Residual             291.2    17.07
Number of obs: 403, groups:  slpID, 21; Speaker, 20

Fixed effects:
            Estimate Std. Error t value
(Intercept)   61.882      6.028   10.27
```

```
vc_m2 <- as.data.frame(VarCorr(m2))
```

ICC for single rating:

- Agreement: $\frac{\sigma^2_{\text{ratee}}}{\sigma^2_{\text{ratee}}+\sigma^2_{\text{rater}}+\sigma^2_E}$

```
# Note: ratee is in position 2, but may be different in different data sets
vc_m2$vcov[2] / (vc_m2$vcov[1] + vc_m2$vcov[2] + vc_m2$vcov[3])
```

```
[1] 0.5800175
```

- Consistency: $\frac{\sigma^2_{\text{ratee}}}{\sigma^2_{\text{ratee}}+\sigma^2_E}$

```
# Note: ratee is in position 2, but may be different in different data sets
vc_m2$vcov[2] / (vc_m2$vcov[2] + vc_m2$vcov[3])
```

```
[1] 0.6678742
```

ICC for average rating:

- Agreement: $\frac{\sigma^2_{\text{ratee}}}{\sigma^2_{\text{ratee}}+(\sigma^2_{\text{rater}}+\sigma^2_E)/k}$

```
# Note: ratee is in position 2, but may be different in different data sets
vc_m2$vcov[2] / (vc_m2$vcov[2] + (vc_m2$vcov[1] + vc_m2$vcov[3]) / 21)
```

```
[1] 0.966669
```

- Consistency: $\frac{\sigma^2_{\text{ratee}}}{\sigma^2_{\text{ratee}}+\sigma^2_E/k}$

```
vc_m2$vcov[2] / (vc_m2$vcov[2] + vc_m2$vcov[3] / 21)
```

```
[1] 0.9768674
```

i Questions

Q1
In practice, for someone to be seen by a pathologist, which coefficient from the above is relevant? And what is its value?
Answer:

Q2
While it is highly unlikely that someone would be able to get opinions from $k = 21$ pathologists, and take their average, to inform their status on dysarthria, how many pathologists would be needed to have an interrater agreement of at least .90?
Hint: change the value of k in the formula to get updated numbers.
Answer:

You can use the `psych::ICC()` function for these calculations. This requires wide-format data.

```
psych::ICC(slp_vas_wide[-1])
```

```
Call: psych::ICC(x = slp_vas_wide[-1])

Intraclass correlation coefficients
                         type  ICC  F df1 df2       p lower bound upper bound
Single_raters_absolute   ICC1 0.58 30  19 400 1.6e-64        0.43        0.75
Single_random_raters     ICC2 0.58 43  19 380 1.0e-82        0.43        0.75
Single_fixed_raters      ICC3 0.67 43  19 380 1.0e-82        0.53        0.81
Average_raters_absolute ICC1k 0.97 30  19 400 1.6e-64        0.94        0.98
Average_random_raters   ICC2k 0.97 43  19 380 1.0e-82        0.94        0.98
Average_fixed_raters    ICC3k 0.98 43  19 380 1.0e-82        0.96        0.99

 Number of subjects = 20     Number of Judges =  21
See the help file for a discussion of the other 4 McGraw and Wong estimates,
```

! Be aware that the terminology in the above output may cause confusion. Here is the translation:

- `Single_raters_absolute`, ICC(1, 1): ICC for single rating for one-way ANOVA (assuming each ratee is rated by a different set of raters)
- `Average_raters_absolute`, ICC(1, k): ICC for average rating for one-way ANOVA (assuming each ratee is rated by a different set of raters)
- `Single_random_raters`, ICC(2, 1): ICC for interrater *agreement* for single rating for two-way ANOVA ("random" here really means agreement)
- `Average_random_absolute`, ICC(2, k): ICC for interrater *agreement* for average rating for two-way ANOVA ("random" here really means agreement)
- `Single_fixed_raters`, ICC(3, 1): ICC for interrater *consistency* for single rating for two-way ANOVA ("fixed" here really means consistency)
- `Average_fixed_absolute`, ICC(3, k): ICC for interrater *consistency* for average rating for two-way ANOVA ("fixed" here really means consistency)

For this example, ICC(1, 1) and ICC(1, k) are not relevant, as it is a crossed (not nested) design.