The six different types of Intraclass Correlations (ICCs)

David Disabato

9/30/2021

Intraclass Correlations

Intraclass Correlations (ICCs) quantify how similar members of a group are to one another. The statistical technique is used in several different areas of data analysis from testing mean differences to multilevel descriptive statistics to inter-rater reliability. More recently it has become a popular scoring method for certain constructs such as emotion differentiation (Kashdan, Barrett, & McKnight (2015)). Within psychological science, ICCs were accessibly introduced by a Psychological Bulletin paper by Shrout and Fleiss (1979) [1]. They present a framework for understanding six different types of ICCs and how to calculate them.

When I was first learning advanced statistics, the term "Intraclass Correlation" kept popping up in different statistics courses and topics. While I understood the term within each area, I didn't have a framework around ICCs in general. It wasn't until I published a paper using ICCs that I started to learn what they were all about (Brown, Goodman, Disabato, Kashdan, Armeli, & Tennen, in press - check it out as its a great paper lead by a bright young scholar Brad Brown at the University of South Florida). Today, we are going to build up a schema around ICCs so that we can more deeply understand what they are all about.

In this blog post I will be explaining the difference between the six types of ICCs summarized in Shrout and Fleiss (1979). I will start with ICC(1, 1) and ICC(1, k) that are often used in multilevel analysis. Then we will proceed to ICC(2, 1), ICC(2, k), ICC(3, 1), and ICC(3, k) that are often used in inter-rater reliability. For each type of ICC, I will show estimation with both an ANOVA model and linear mixed effects model.

For the statistical programming, I will be using R - an open source computer software program. For more information about R go to https://www.r-project.org/about.html. I will also be using a package that I created called `str2str` (read as "structure to structure"), which contains a lot of simple wrapper functions for converting R objects from one structure to another. I find using these functions save a few lines of code and generally makes code easier to read. If you want to learn more about the package, you can go to the str2str documentation webpage.

```r
library(str2str)
```

ICC(1, 1) for multilevel analysis

Intraclass Correlations (ICCs) always occur in the context of grouped data. It might not be a classic form of grouped data and we might not usually use the word "group" to describe it, but the data will be grouped nonetheless. Let's start with a classic example of grouped data: one continuous variable mapped onto one grouping variable. The `InsectSprays` dataset contains one continuous variable named "count" and one grouping variable "spray" with the groups of "spray" labeled as letters. The "count" variable is the number of insects in a given plot of land and the "spray" variable is the type of insect repellent spray used. Therefore, the plots of land are grouped into sprays.

[1] Nowadays, this type of paper would be published in Psychological Methods, but that was not a journal yet and so many Quantitative Psychology papers would get published in Psycological Bulletin.

```
str(InsectSprays)
```

```
## 'data.frame':    72 obs. of  2 variables:
##  $ count: num  10 7 20 14 14 12 10 23 17 20 ...
##  $ spray: Factor w/ 6 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
N <- nrow(InsectSprays)
print(N) # cases
```

```
## [1] 72
```

```
n <- length(unique(InsectSprays$"spray"))
print(n) # groups
```

```
## [1] 6
```

As you can see, the dataset contains 72 cases and 6 groups.

```
size_by <- c(tapply(InsectSprays$"count", InsectSprays$"spray", FUN = length))
print(size_by)
```

```
##  A  B  C  D  E  F
## 12 12 12 12 12 12
```

```
size_avg <- mean(size_by)
print(size_avg)
```

```
## [1] 12
```

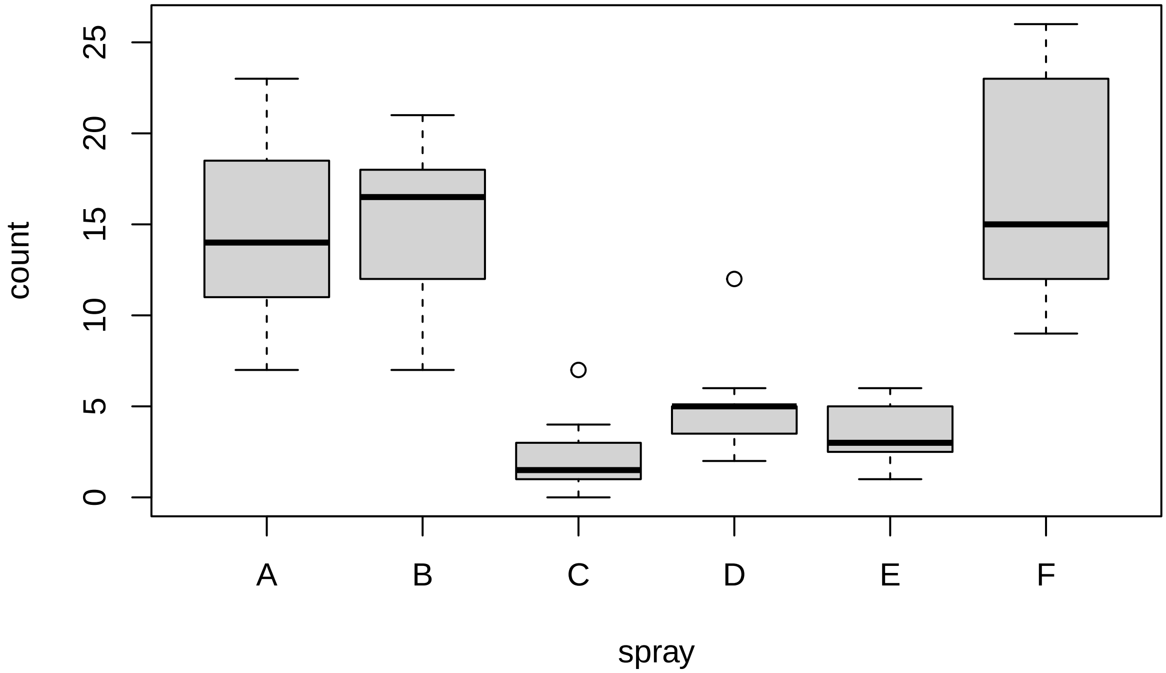
Furthermore, there are exactly 12 cases in each group, resulting in an average group size of 12. This is referred to as “balanced” grouped data and something we will be returning to later.

The table of summary statistics and boxplot graph below show meaningful mean differences across groups. Clearly, some insect repellent sprays are more effective than others. I also wanted to highlight the meaningful variance differences across groups. Not because ICCs incorperate them, but because ICCs assume equal variances across groups: homogeneity of variance. Indeed OLS regression, ANOVA, and linear mixed effect models all assume homogeneity of (within-group) variance. For educational purposes, we will be ignoring this issue, but I wanted to acknowledge that it is there.

```
count_by <- split(InsectSprays$"count", f = InsectSprays$"spray")
lv2d(lapply(X = count_by, FUN = summary), along = 1)
```

```
##   Min. 1st Qu. Median      Mean 3rd Qu. Max.
## A    7   11.50   14.0 14.500000   17.75   23
## B    7   12.50   16.5 15.333333   17.50   21
## C    0    1.00    1.5  2.083333    3.00    7
## D    2    3.75    5.0  4.916667    5.00   12
## E    1    2.75    3.0  3.500000    5.00    6
## F    9   12.50   15.0 16.666667   22.50   26
```

```r
boxplot(count ~ spray, data = InsectSprays)
```



Observed variance components

Let's start with looking at our observed variance components. By "observed", I mean there is no statistical model behind them; they are just summary statistics. We are going to split up the "count" variable into between-group and within-group scores and then take the variance of each.

```r
count_btw <- ave(InsectSprays$"count", InsectSprays$"spray", FUN = mean)
print(count_btw)
```

```
##  [1] 14.500000 14.500000 14.500000 14.500000 14.500000 14.500000 14.500000
##  [8] 14.500000 14.500000 14.500000 14.500000 14.500000 15.333333 15.333333
## [15] 15.333333 15.333333 15.333333 15.333333 15.333333 15.333333 15.333333
## [22] 15.333333 15.333333 15.333333  2.083333  2.083333  2.083333  2.083333
## [29]  2.083333  2.083333  2.083333  2.083333  2.083333  2.083333  2.083333
## [36]  2.083333  4.916667  4.916667  4.916667  4.916667  4.916667  4.916667
## [43]  4.916667  4.916667  4.916667  4.916667  4.916667  4.916667  3.500000
## [50]  3.500000  3.500000  3.500000  3.500000  3.500000  3.500000  3.500000
## [57]  3.500000  3.500000  3.500000  3.500000 16.666667 16.666667 16.666667
## [64] 16.666667 16.666667 16.666667 16.666667 16.666667 16.666667 16.666667
## [71] 16.666667 16.666667
```

```
count_wth <- InsectSprays$"count" - count_btw
print(count_wth)
```

```
##  [1] -4.50000000 -7.50000000  5.50000000 -0.50000000 -0.50000000 -2.50000000
##  [7] -4.50000000  8.50000000  2.50000000  5.50000000 -0.50000000 -1.50000000
## [13] -4.33333333  1.66666667  5.66666667 -4.33333333  0.66666667 -1.33333333
## [19]  1.66666667  1.66666667  3.66666667  5.66666667 -8.33333333 -2.33333333
## [25] -2.08333333 -1.08333333  4.91666667 -0.08333333  0.91666667 -1.08333333
## [31] -0.08333333 -1.08333333  0.91666667 -2.08333333 -1.08333333  1.91666667
## [37] -1.91666667  0.08333333  7.08333333  1.08333333 -0.91666667 -1.91666667
## [43]  0.08333333  0.08333333  0.08333333  0.08333333 -2.91666667 -0.91666667
## [49] -0.50000000  1.50000000 -0.50000000  1.50000000 -0.50000000  2.50000000
## [55] -2.50000000 -2.50000000 -0.50000000 -1.50000000  2.50000000  0.50000000
## [61] -5.66666667 -7.66666667 -1.66666667  5.33333333 -1.66666667 -0.66666667
## [67] -3.66666667 -6.66666667  9.33333333  9.33333333  7.33333333 -3.66666667
```

```
var_btw_obs <- var(count_btw)
var_wth_obs <- var(count_wth)
data.frame(var_btw_obs, var_wth_obs)
```

```
##   var_btw_obs var_wth_obs
## 1     37.5892    14.29812
```

As expected, the two observed variance components add up to the total observed variance.

```
var_tot_obs <- var(InsectSprays$"count")
data.frame(var_tot_obs, var_btw_obs + var_wth_obs, "compare" = all.equal(var_tot_obs, var_btw_obs + var_
```

```
##   var_tot_obs var_btw_obs + var_wth_obs compare
## 1    51.88732                  51.88732    TRUE
```

Now the formula for the ICC(1, 1) based on variance components is fairly simple:

$$ICC_{1,1} = \frac{\sigma^2_{between}}{\sigma^2_{between} + \sigma^2_{within}}$$

If we apply that formula to the observed variance components…

```
icc11_obs <- var_btw_obs / (var_btw_obs + var_wth_obs)
print(icc11_obs)
```

```
## [1] 0.724439
```

Let's check whether that value is correct with an established R package function:

```
library(irr)
```

```
## Loading required package: lpSolve
```

```r
InsectSprays_wide <- t(unstack(InsectSprays, form = count ~ spray))
icc_obj <- icc(InsectSprays_wide)
icc11_irr <- icc_obj[["value"]]
data.frame(icc11_obs, icc11_irr, "compare" = all.equal(icc11_obs, icc11_irr))
```

```
##   icc11_obs icc11_irr                              compare
## 1  0.724439 0.7374311 Mean relative difference: 0.01793393
```

As you can see, the `icc` function in the `irr` package does not generate the same ICC(1, 1) value as our observed ICC(1, 1) value. So the observed variance components are actually incorrect in that they are not unbiased sample estimates of the population variance components. One way to understand this is to contrast the ICC(1, 1) with the R^2 of the grouping variable "spray" predicting the continuous variable "outcome" from a one-way ANOVA model.

```r
aov_obj <- aov(count ~ spray, data = InsectSprays)
r2_aov <- summary.lm(aov_obj)[["r.squared"]]
print(r2_aov)
```

```
## [1] 0.724439
```

First thing to point out is that R^2 is NOT the same as ICC(1, 1).

```r
data.frame(r2_aov, icc11_irr, "compare" = all.equal(r2_aov, icc11_irr))
```

```
##     r2_aov icc11_irr                              compare
## 1 0.724439 0.7374311 Mean relative difference: 0.01793393
```

However, you might recongize the R^2 value. It is the same value as the (incorrect) observed ICC(1, 1).

```r
data.frame(r2_aov, icc11_obs, "compare" = all.equal(r2_aov, icc11_obs))
```

```
##     r2_aov icc11_obs compare
## 1 0.724439  0.724439    TRUE
```

This makes sense as the formula for the R^2 from a one-way ANOVA is essentially the same as the formula for the ICC(1, 1), expect that it uses sum of squares rather than variance components:

$$R^2 = \frac{SS_{between}}{SS_{between} + SS_{within}}$$

We can manually calculate R^2 to see this:

```r
aov_anova <- anova(aov_obj)
ss_btw <- aov_anova["spray", "Sum Sq"]
ss_wth <- aov_anova["Residuals", "Sum Sq"]
r2_ss <- ss_btw / (ss_btw + ss_wth)
data.frame(r2_ss, r2_aov, "compare" = all.equal(r2_ss, r2_aov))
```

```
##      r2_ss   r2_aov compare
## 1 0.724439 0.724439    TRUE
```

We can also compute the observed variance components from the sum of squares to fully see the connection. This makes sense as R^2 is *variance* explained. If we convert the sum of squares to variances, we get the same value as above.

```r
var_btw_ss <- ss_btw / (N - 1)
data.frame(var_btw_ss, var_btw_obs, "compare" = all.equal(var_btw_ss, var_btw_obs))
```

```
##   var_btw_ss var_btw_obs compare
## 1    37.5892     37.5892    TRUE
```

```r
var_wth_ss <- ss_wth / (N - 1)
data.frame(var_wth_ss, var_wth_obs, "compare" = all.equal(var_wth_ss, var_wth_obs))
```

```
##   var_wth_ss var_wth_obs compare
## 1   14.29812    14.29812    TRUE
```

```r
r2_var <- var_btw_ss / (var_btw_ss + var_wth_ss)
data.frame(r2_var, r2_aov, "compare" = all.equal(r2_var, r2_aov))
```

```
##     r2_var   r2_aov compare
## 1 0.724439 0.724439    TRUE
```

So the observed ICC(1, 1) is really the R^2 and is technically not considered an intraclass correlation.

Now, if you are like me, the definition of R^2 and the definition of ICC(1, 1) sound an awful lot alike. R^2 is "the proportion of variance explained by the grouping variable"; ICC(1, 1) is "the proportion of variance between groups". And both are sometimes interpreted as "the proportion of variance due to group membership".

Indeed, the R^2 and the model ICC(1, 1) both have the same F-value:

```r
Fvalue_aov <- aov_anova["spray", "F value"]
Fvalue_irr <- icc_obj[["Fvalue"]]
data.frame(Fvalue_aov, Fvalue_irr, "compare" = all.equal(Fvalue_aov, Fvalue_irr))
```

```
##   Fvalue_aov Fvalue_irr compare
## 1   34.70228   34.70228    TRUE
```

This makes some sense as both F-values are testing whether there are mean differences across groups. How exactly is the ICC(1, 1) different from R^2 though? To do that, we are going to calculate the variance components from a one-way ANOVA model.

ANOVA variance components

We already computed the ANOVA model above. But, we will re-calculate it for clarification and then print the traditional ANOVA table.

```r
aov_obj <- aov(count ~ spray, data = InsectSprays)
aov_anova <- anova(aov_obj)
print(aov_anova)
```

```
## Analysis of Variance Table
##
## Response: count
##           Df Sum Sq Mean Sq F value    Pr(>F)
## spray      5 2668.8  533.77  34.702 < 2.2e-16 ***
## Residuals 66 1015.2   15.38
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

Now, we know that the ANOVA model can be used to calculate the observed variance components using the sum of squares. But we showed those are incorrect variance components for ICC(1, 1). Instead, we can calculate the correct variance components with the ANOVA model using alternative formulas.

We will start with the between-group variance component.

```r
ms_btw <- aov_anova["spray", "Mean Sq"]
ms_wth <- aov_anova["Residuals", "Mean Sq"]
var_btw_aov <- (ms_btw / size_avg) - (ms_wth / size_avg)
data.frame(var_btw_aov, var_btw_obs, "compare" = all.equal(var_btw_aov, var_btw_obs))
```

```
##   var_btw_aov var_btw_obs                              compare
## 1    43.19878     37.5892 Mean relative difference: 0.129855
```

As you can see, the ANOVA variance component is not the same as the observed variance component. Instead, it is quite a bit larger than the observed variance component. We already determined that the observed variance components are incorrect. We can now explain that the observed between-group variance component ignores two statistical issues that the ANOVA between-group variance component deals with (pg. 21; Snijders & Bosker, 2012):

1) **Degrees of freedom**. The observed between-group variance component does not take into account the number of groups. The formula for a variance is $\frac{SS}{df}$, where SS = sum of squares and df = degrees of freedom [2]. When calculating the observed between-group variance component from the sum of squares above, I used `(N - 1)` as the degrees of freedom. However, that degree of freedom value is independent of how many groups the data has. We could have had 24 groups of 3 rather than 6 groups of 12 and the degrees of freedom would still have been `(N - 1)`. That doesn't really make much sense. The first number in the degrees of freedom formula can be thought of as the number of pieces of information. While the original count score has `N` = 72 pieces of information, one for each case, the between-group count score (aka `count_btw`) does not. It only has 6 values - one for each group. The between-group score simply repeats the 6 group means to fill up the dataset. So the first number should really be `n` = 6 pieces of information. This is exactly what the ANOVA model does. The between-group degrees of freedom is `n - 1` = 5.

```r
df_btw <- aov_anova["spray", "Df"]
print(df_btw)
```

```
## [1] 5
```

Indeed, another way to calculate the observed between-group variance component would be without repeating the 6 group means.

[2] Note, this is the formula for the mean square in ANOVA. Indeed, mean squares are a type of variance

```r
count_btw2 <- c(tapply(InsectSprays$"count", InsectSprays$"spray", FUN = mean))
print(count_btw2)
```

```
##         A         B         C         D         E         F
## 14.500000 15.333333  2.083333  4.916667  3.500000 16.666667
```

```r
var_btw_obs2 <- var(count_btw2)
data.frame(var_btw_obs2, var_btw_aov, "compare" = all.equal(var_btw_obs2, var_btw_aov))
```

```
##   var_btw_obs2 var_btw_aov                              compare
## 1     44.48056    43.19878 Mean relative difference: 0.02881655
```

Now this revised observed between-group variance component is closer to the ANOVA variance component, but still slighty off. This time it is larger, rather than smaller though. However, this number actually appears as the first term in the ANOVA between-grouop variance component formula.

```r
first_term <- (ms_btw / size_avg)
data.frame(first_term, var_btw_obs2, "compare" = all.equal(first_term, var_btw_obs2))
```

```
##   first_term var_btw_obs2 compare
## 1   44.48056     44.48056    TRUE
```

So the ANOVA model is actually calculating the variance of the group means. But the ANOVA model variance component was still a bit smaller. Well that is because there is a second term in the formula: we subtract out `ms_wth / size_avg`. To understand why that is, we move on to the second statistical issue.

2) **Group mean sampling error**. The group means are sample estimates, which implies they will fluctuate based on sampling error. Just like the overall sample mean has a standard error around it representing it's sampling error, each group mean does as well. This implies some of the between-group variance is due to sampling error. In other words, the observed variance of the sample group means is not equal to the "true" variance of the population group means; instead it is larger due to the extra sampling error variance. How much? Well that is inversely proportional to the size of the groups. The larger the groups, the larger the "sample" is for each estimated group mean, and the less sampling error. That is the role of the `size_avg` value in the second term `ms_wth / size_avg`. The formula is substracting out the extra sampling error variance and it makes sense that as the group size increases, the amount of sampling error variance goes down. The `ms_wth` value represents how much variance there is within-groups. While we are often interested in the within-group variance and might want a lot of it, in terms of estimating the group means, within-group variance is noise. The more variable scores are in a sample, the less precise the mean is estimated. This is true for the overall sample and for each group's "sample".

Now, the astute reader might have noticed that the term `ms_wth / size_avg` looks familiar. That is because it is very similar to the sampling error variance of the mean (aka squared standard error of the mean): $\frac{\sigma^2}{n}$. That is no coincidence. The expected amount of extra sampling error variance for the group means is the sampling error variance of the means! Once, we subtract that sampling error out, we get the correct variance component.

Let's move on to the within-group variance component.

```r
var_wth_aov <- ms_wth
print(var_wth_aov)
```

```
## [1] 15.38131
```

Well that was easy. The within-group variance component is just the mean square within. Again, it is different than the observed within-group variance component.

```r
data.frame(var_wth_aov, var_wth_obs, "compare" = all.equal(var_wth_aov, var_wth_obs))
```

```
##   var_wth_aov var_wth_obs                                compare
## 1    15.38131    14.29812 Mean relative difference: 0.07042254
```

The observed within-group variance component is smaller. This is because of the first statistical issue we discussed above: 1) degrees of freedom. When we calculated the observed within-group variance component from the sum of squares, we used `N - 1` as the degrees of freedom. We went over the first number in the degrees of freedom formula, now we will go over the second. The second number can be thought of as the number of parameters you have to estimate. To calculate our within-group count score, we needed to estimate the 6 group means. With the overall sample variance, we only need to estimate 1 mean - the overall sample mean, so having the second number in the degrees of freedom formula be 1 makes sense. However, in our case of estimating 6 means, 1 doesn't make much sense for this second number. Instead, the number 6 makes more sense. Therefore, if we recalculate our observed within-group variance component with `N - 6` = 66 degrees of freedom, we will get the correct value.

```r
var_wth_obs2 <- ss_wth / (N - n)
data.frame(var_wth_obs2, var_wth_aov, "compare" = all.equal(var_wth_obs2, var_wth_aov))
```

```
##   var_wth_obs2 var_wth_aov compare
## 1     15.38131    15.38131    TRUE
```

To sum up the degrees of freedom issue, here is a conceptual formula for degrees of freedom that is applicable to many situations:

$$df = (\#OfPiecesOfInformation) - (\#OfParametersEstimated)$$

.

The reason the within-group variance component formula is simpler is because we don't have the second issue: 2) Group mean sampling error. We are only looking within-group; thus, by definition, group mean sampling error isn't an issue. One way to think of this is that the data occur at multiple "levels" (aka multilevel data). There is a between-group "level" and a within-group "level". Sampling error at the between-group level will not directly impact variances at the within-group level. That is why in multilevel analysis, the statistical complications usually concern the between-group level more so than the within-group level.

For example, traditional multilevel modeling will decompose observed scores at multiple levels into between-group and within-group scores to obtain between-group coefficients and within-group coefficients. Statisticians have proposed multilevel structual equation modeling, which does a latent decomposition of the observed scores into between-group variables and within-group variables that takes into account 1) degrees of freedom and 2) group mean sampling error (Muthen, 1994). Quantitative psychologists have shown that it is primarily the between-group coefficients that benefit from multilevel structural equation modeling, and that the within-group coefficients are usually very similar to those from multilevel modeling (Preacher, Zyphur, & Zhang, 2010).

Now that we have our two variance components, let's go ahead and calculate the ICC(1, 1) from the ANOVA model.

```r
icc11_aov <- var_btw_aov / (var_btw_aov + var_wth_aov)
data.frame(icc11_aov, icc11_irr, "compare" = all.equal(icc11_aov, icc11_irr))
```

```
##   icc11_aov icc11_irr compare
## 1 0.7374311 0.7374311    TRUE
```

As expected, the ANOVA variance components gives us the correct ICC(1, 1) value.

Note, even though the formulas for the ANOVA variance components include the average group size, it is only slightly influenced by it. The same goes for the ICC(1, 1). To show you, let's repeat the dataset 10 times so that each group now contains 120 cases rather than 12 cases, and see how much the ICC(1, 1) value changes.

```r
tmp <- replicate(n = 10, InsectSprays, simplify = FALSE)
InsectSprays2 <- ld2d(tmp, along = 1)
aov_obj2 <- aov(count ~ spray, data = InsectSprays2)
aov_anova2 <- anova(aov_obj2)
print(aov_anova2)
```

```
## Analysis of Variance Table
##
## Response: count
##            Df Sum Sq Mean Sq F value    Pr(>F)
## spray       5  26688  5337.7  375.42 < 2.2e-16 ***
## Residuals 714  10152    14.2
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

If we compare this 10x ANOVA table to the original ANOVA table…

```r
print(aov_anova)
```

```
## Analysis of Variance Table
##
## Response: count
##           Df Sum Sq Mean Sq F value    Pr(>F)
## spray      5 2668.8  533.77  34.702 < 2.2e-16 ***
## Residuals 66 1015.2   15.38
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

…we can see that the sum of squares between and within are exactly 10x as large and the between-group degrees of freedom are the same, but the within-group degrees of freedom are not exactly 10x as large - rather `714 / 66` = 10.8181818x as large. Therefore, the within-group variance component is going to be slightly smaller relative to the between-group variance component as the average group size increases.

```r
ms_wth2 <- aov_anova2["Residuals", "Mean Sq"]
var_wth_aov2 <- ms_wth2
data.frame(var_wth_aov2, var_wth_aov, "compare" = all.equal(var_wth_aov2, var_wth_aov))
```

```
##   var_wth_aov2 var_wth_aov                            compare
## 1     14.21802    15.38131 Mean relative difference: 0.08181818
```

This will in turn, result in the between-group variance component being slightly larger. This makes sense with the idea that larger group sizes result in less group mean sampling error and a smaller amount of extra sampling error variance in the observed variance of the group means.

```
ms_btw2 <- aov_anova2["spray", "Mean Sq"]
var_btw_aov2 <- (ms_btw2 / 120) - (ms_wth2 / 120)
data.frame(var_btw_aov2, var_btw_aov, "compare" = all.equal(var_btw_aov2, var_btw_aov))
```

```
##   var_btw_aov2 var_btw_aov                              compare
## 1     44.36207    43.19878 Mean relative difference: 0.02622268
```

These differences in the variance components will result in a slightly larger ICC(1, 1).

```
icc11_aov2 <- var_btw_aov2 / (var_btw_aov2 + var_wth_aov2)
data.frame(icc11_aov2, icc11_aov, "compare" = all.equal(icc11_aov2, icc11_aov))
```

```
##   icc11_aov2 icc11_aov                              compare
## 1  0.7572892 0.7374311 Mean relative difference: 0.02622268
```

However, because the R^2 does not depend on degrees of freedom, its value will be identical.

```
ss_btw2 <- aov_anova2["spray", "Sum Sq"]
ss_wth2 <- aov_anova2["Residuals", "Sum Sq"]
r2_ss2 <- ss_btw2 / (ss_btw2 + ss_wth2)
data.frame(r2_ss, r2_ss2, "compare" = all.equal(r2_ss, r2_ss2))
```

```
##      r2_ss   r2_ss2 compare
## 1 0.724439 0.724439    TRUE
```

LME model variance components

Let’s move on to calculating the ICC(1, 1) with a linear mixed effects (LME) model. When doing multilevel analysis, this is often how the ICC(1, 1) is calculated. Perhaps because the data analyst is usually going to use linear mixed effects modeling for their prediction models (Note, linear mixed effects modeling and multilevel modeling are different names for the same thing). I specifically used the term “linear mixed effects model” rather than “multilevel modeling” because we will be using these models outside the context of multilevel analysis when we move on to ICC(2, 1), ICC(2, k), ICC(3, 1), and ICC(3, k).

We will use the R package `lme4` for the linear mixed effects models. Although we could use the R package `nlme` for the ICC(1, 1) and ICC(1, k), we could not use it for the other 4 types of ICCs since they require two non-nested grouping variables. `lme4` has no problem with multiple non-nested grouping variables.

We will specify a null, or “intercept-only” LME model where we have no predictors. Random effects are specified for the grouping variable “spray”.

```
library(lme4)
```

```
## Loading required package: Matrix
```

```r
lmer_obj <- lmer(count ~ 1 + (1 | spray), data = InsectSprays, REML = TRUE)
summary(lmer_obj)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: count ~ 1 + (1 | spray)
##    Data: InsectSprays
##
## REML criterion at convergence: 417.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0820 -0.5298 -0.1040  0.4426  2.4325
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  spray    (Intercept) 43.20    6.573
##  Residual             15.38    3.922
## Number of obs: 72, groups:  spray, 6
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    9.500      2.723   3.489
```

Now in the LME model we don't have sum of squares or mean squares. Instead we directly estimate the variance components. The model will split up the continuous variable "count" variance into a between-group variance component and a within-group variance component.

```r
var_comp <- as.data.frame(VarCorr(lmer_obj))
var_btw_lmer <- var_comp[var_comp$"grp" == "spray", "vcov"]
var_wth_lmer <- var_comp[var_comp$"grp" == "Residual", "vcov"]
data.frame(var_btw_lmer, var_wth_lmer)
```

```
##   var_btw_lmer var_wth_lmer
## 1     43.19878     15.38131
```

One other thing that confused me at first was that the correct variance components do not add up the total variance of the outcome.

```r
var_tot <- var(InsectSprays$"count")
var_tot_lmer <- var_btw_lmer + var_wth_lmer
data.frame(var_tot, var_tot_lmer, "compare" = all.equal(var_tot, var_tot_lmer))
```

```
##    var_tot var_tot_lmer                             compare
## 1 51.88732     58.58009 Mean relative difference: 0.1289866
```

The sum of the correct variance components is quite a bit larger than the observed total variance. This is again due to the degrees of freedom issue we talked about above. The observed total variance is agnostic to whether the data are grouped or not and to how many groups there are. As far as the total variance formula is concerned, the continuous variable "count" is not grouped in any way. Another way to think about the descrepency is whether we calculate the between-group variance based on the 6 group means or repeat the group means to fill up the 72 cases of the dataset. The observed total variance is based on repeating the group mean information, while the variance components are not. By repeating the group mean information, the two observed variance components add up to the total variance.

```r
print(count_btw)
```

```
##  [1] 14.500000 14.500000 14.500000 14.500000 14.500000 14.500000 14.500000
##  [8] 14.500000 14.500000 14.500000 14.500000 14.500000 15.333333 15.333333
## [15] 15.333333 15.333333 15.333333 15.333333 15.333333 15.333333 15.333333
## [22] 15.333333 15.333333 15.333333  2.083333  2.083333  2.083333  2.083333
## [29]  2.083333  2.083333  2.083333  2.083333  2.083333  2.083333  2.083333
## [36]  2.083333  4.916667  4.916667  4.916667  4.916667  4.916667  4.916667
## [43]  4.916667  4.916667  4.916667  4.916667  4.916667  4.916667  3.500000
## [50]  3.500000  3.500000  3.500000  3.500000  3.500000  3.500000  3.500000
## [57]  3.500000  3.500000  3.500000  3.500000 16.666667 16.666667 16.666667
## [64] 16.666667 16.666667 16.666667 16.666667 16.666667 16.666667 16.666667
## [71] 16.666667 16.666667
```

```r
print(var_btw_obs)
```

```
## [1] 37.5892
```

```r
var_tot_obs <- var_btw_obs + var_wth_obs
data.frame(var_tot_obs, var_tot, "compare" = all.equal(var_tot_obs, var_tot))
```

```
##   var_tot_obs  var_tot compare
## 1    51.88732 51.88732    TRUE
```

But, if we calculate the observed between-group variance component without repeating the group means, the sum of the observed variance components is much closer to the sum of the correct variance components.

```r
print(count_btw2)
```

```
##         A         B         C         D         E         F 
## 14.500000 15.333333  2.083333  4.916667  3.500000 16.666667
```

```r
print(var_btw_obs2)
```

```
## [1] 44.48056
```

```r
var_tot_obs2 <- var_btw_obs2 + var_wth_obs2
data.frame(var_tot_obs2, var_tot_lmer, "compare" = all.equal(var_tot_obs2, var_tot_lmer))
```

```
##   var_tot_obs2 var_tot_lmer                              compare
## 1     59.86187     58.58009 Mean relative difference: 0.02141219
```

However, the sum of the observed variance components is still slightly larger than the sum of the correct variance components due to the group mean sampling error issue we talked about above.

Let's go ahead and calculate the ICC(1, 1) from the LME model now.

```r
icc11_lmer <- var_btw_lmer / (var_btw_lmer + var_wth_lmer)
data.frame(icc11_lmer, icc11_aov, "compare" = all.equal(icc11_lmer, icc11_aov))
```

```
##   icc11_lmer icc11_aov compare
## 1  0.7374311 0.7374311    TRUE
```

As you can see, the ICC(1, 1) computed from the LME model is the same as the one from the ANOVA model. That will not always be the case though. The `InsectSprays` dataset is known as what is “balanced”: each group is the same size (aka has the same number of cases). Indeed the `size_by` vector is not really a variable, but rather a constant: each group has exactly 12 cases. What happens when the data are “unbalanced” from the group sizes differing? Let’s find out.

```r
keep_rows <- c(1:2, 13:16, 25:30, 37:44, 49:58, 61:72)
InsectSprays3 <- InsectSprays[keep_rows, ]
size_by3 <- c(tapply(InsectSprays3$"count", InsectSprays3$"spray", FUN = length))
print(size_by3)
```

```
##  A  B  C  D  E  F
##  2  4  6  8 10 12
```

```r
size_avg3 <- mean(size_by3)
print(size_avg3)
```

```
## [1] 7
```

I have removed some cases from the `InsectSprays` dataset such that the group sizes now range from 2 to 12 rather than all being 12: an unbalanced dataset. Let’s now calculate the variance components and ICC(1, 1) from the ANOVA model and LME model. We start with the ANOVA model:

```r
aov_obj3 <- aov(count ~ spray, data = InsectSprays3)
aov_anova3 <- anova(aov_obj3)
ms_btw3 <- aov_anova3["spray", "Mean Sq"]
ms_wth3 <- aov_anova3["Residuals", "Mean Sq"]
var_btw_aov3 <- (ms_btw3 / size_avg3) - (ms_wth3 / size_avg3)
var_wth_aov3 <- ms_wth3
icc11_aov3 <- var_btw_aov3 / (var_btw_aov3 + var_wth_aov3)
data.frame(var_btw_aov3, var_wth_aov3, icc11_aov3)
```

```
##   var_btw_aov3 var_wth_aov3 icc11_aov3
## 1      41.9727     17.11042  0.7104009
```

To make sure I did my math right, we confirm with the established `irr` R package.

```r
library(plyr)
tmp <- unstack(InsectSprays3, count ~ spray)
InsectSprays3_wide <- rbind.fill.matrix(lapply(X = tmp, FUN = t.default))
rownames(InsectSprays3_wide) <- levels(InsectSprays3$"spray")
icc11_irr3 <- icc(InsectSprays3_wide)[["value"]]
```

```
## Warning in qf(1 - alpha/2, ns - 1, ns * (nr - 1)): NaNs produced
## Warning in qf(1 - alpha/2, ns * (nr - 1), ns - 1): NaNs produced
```

```
print(icc11_irr3)
```

```
## [1] NA
```

Uh oh - we get NA back along with a warning message saying NaNs were produced. This is because the `icc` function from the `irr` package requires groups to have no missing data in order to be included in the calculation. In the wide version of our dataset `InspectSprays3_wide` there is only one group with no missing data: F.

```
print(InsectSprays3_wide)
```

```
##    1  2  3  4  5  6  7  8  9 10 11 12
## A 10  7 NA NA NA NA NA NA NA NA NA NA
## B 11 17 21 11 NA NA NA NA NA NA NA NA
## C  0  1  7  2  3  1 NA NA NA NA NA NA
## D  3  5 12  6  4  3  5  5 NA NA NA NA
## E  3  5  3  5  3  6  1  1  3  2 NA NA
## F 11  9 15 22 15 16 13 10 26 26 24 13
```

The `icc` function provides us a warning message, essentially telling us it can't calculate an ICC with only one group. The way the `icc` function in the `irr` package is set up, we are forced to only use column observations present in each group to get an estimate of any ICC.

```
library(irr)
icc11_irr3 <- icc(InsectSprays3_wide[, 1:2])[["value"]]
data.frame(icc11_aov3, icc11_irr3, "compare" = all.equal(icc11_aov3, icc11_irr3))
```

```
##   icc11_aov3 icc11_irr3                             compare
## 1  0.7104009  0.8174953 Mean relative difference: 0.150752
```

We get a different value from our manual ANOVA calculation… However, the `irr` package value is really not to be trusted since we weren't able to use all the data in `InspectSprays3_wide`. Let's transition to a different R package now. The `ICC1` function in the `multilevel` package will allow us to use all the data in `InspectSprays3_wide` by providing it the ANOVA object.

```
library(multilevel)
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:lme4':
##
##     lmList
```

```
## Loading required package: MASS
```

```
icc11_ml3 <- ICC1(aov_obj3)
data.frame(icc11_aov3, icc11_ml3, "compare" = all.equal(icc11_aov3, icc11_ml3))
```

```
##   icc11_aov3 icc11_ml3 compare
## 1  0.7104009 0.7104009    TRUE
```

As you can see, the `multilevel` R package provides us with the same value as our manual ANOVA calculation. We now move on to compute the ICC(1, 1) from the LME model with the unbalanced dataset.

```
lmer_obj3 <- lmer(count ~ 1 + (1 | spray), data = InsectSprays3, REML = TRUE)
var_comp3 <- as.data.frame(VarCorr(lmer_obj3))
var_btw_lmer3 <- var_comp3[var_comp3$"grp" == "spray", "vcov"]
var_wth_lmer3 <- var_comp3[var_comp3$"grp" == "Residual", "vcov"]
icc11_lmer3 <- var_btw_lmer3 / (var_btw_lmer3 + var_wth_lmer3)
data.frame(
   "var_btw" = c(var_btw_lmer3, var_btw_aov3),
   "var_wth" = c(var_wth_lmer3, var_wth_aov3),
   "icc11" = c(icc11_lmer3, icc11_aov3),
row.names = c("lmer","aov"))
```

```
##       var_btw  var_wth     icc11
## lmer 35.51613 17.05036 0.6756420
## aov  41.97270 17.11042 0.7104009
```

When the data are unbalanced, the ANOVA and LME estimates will differ. One way to think about unbalanced data is a missing data problem. The cases which are present in the original dataset `InsectSprays`, but not in `InsectSprays3`, are missing. We can visualize this by stacking `InsectSprays3`.

```
library(reshape)
```

```
##
## Attaching package: 'reshape'

## The following objects are masked from 'package:plyr':
##
##     rename, round_any

## The following object is masked from 'package:Matrix':
##
##     expand
```

```
tmp <- suppressWarnings(melt(InsectSprays3_wide, measure.vars = colnames(InsectSprays3_wide), na.rm = F
names(tmp) <- c("spray","observation","count")
InsectSprays4 <- tmp[order(tmp$"spray", tmp$"observation"), sort(names(tmp))]
print(InsectSprays4)
```

```
##    count observation spray
## 1     10           1     A
## 7      7           2     A
## 13    NA           3     A
## 19    NA          4     A
## 25    NA          5     A
## 31    NA          6     A
## 37    NA          7     A
## 43    NA          8     A
## 49    NA          9     A
## 55    NA         10     A
## 61    NA         11     A
## 67    NA         12     A
## 2     11          1     B
## 8     17          2     B
## 14    21          3     B
## 20    11          4     B
## 26    NA          5     B
## 32    NA          6     B
## 38    NA          7     B
## 44    NA          8     B
## 50    NA          9     B
## 56    NA         10     B
## 62    NA         11     B
## 68    NA         12     B
## 3      0          1     C
## 9      1          2     C
## 15     7          3     C
## 21     2          4     C
## 27     3          5     C
## 33     1          6     C
## 39    NA          7     C
## 45    NA          8     C
## 51    NA          9     C
## 57    NA         10     C
## 63    NA         11     C
## 69    NA         12     C
## 4      3          1     D
## 10     5          2     D
## 16    12          3     D
## 22     6          4     D
## 28     4          5     D
## 34     3          6     D
## 40     5          7     D
## 46     5          8     D
## 52    NA          9     D
## 58    NA         10     D
## 64    NA         11     D
## 70    NA         12     D
## 5      3          1     E
## 11     5          2     E
## 17     3          3     E
## 23     5          4     E
## 29     3          5     E
## 35     6          6     E
## 41     1          7     E
## 47     1          8     E
## 53     3          9     E
## 59     2           10     E
## 65    NA           11     E
## 71    NA           12     E
## 6     11            1     F
## 12     9            2     F
## 18    15            3     F
## 24    22            4     F
## 30    15            5     F
## 36    16            6     F
## 42    13            7     F
## 48    10            8     F
## 54    26            9     F
## 60    26           10     F
## 66    24           11     F
## 72    13           12     F
```

We have a “balanced” dataset, where each group has 12 observations, but there is missing data, making the observed data inbalanced. Both ANOVA and LME use case-wise deletion (with the data in the long format), so `InsectSprays4` is converted to `InsectSprays3` before the calculations are performed by each model. I will show we get the exact same results:

```r
all.equal(InsectSprays3, na.omit(InsectSprays4[c("count","spray")]),
   check.attributes = FALSE) # don't check row.names for equality
```

```
## [1] TRUE
```

```r
aov_obj4 <- aov(count ~ spray, data = InsectSprays4)
icc11_aov4 <- ICC1(aov_obj4)
data.frame(icc11_aov4, icc11_aov3, "compare" = all.equal(icc11_aov4, icc11_aov3))
```

```
##   icc11_aov4 icc11_aov3 compare
## 1  0.7104009  0.7104009    TRUE
```

```r
lmer_obj4 <- lmer(count ~ 1 + (1 | spray), data = InsectSprays4, REML = TRUE)
var_comp4 <- as.data.frame(VarCorr(lmer_obj4))
icc11_lmer4 <- var_comp4[var_comp4$"grp" == "spray", "vcov"] / (var_comp4[var_comp4$"grp" == "spray", "
data.frame(icc11_lmer4, icc11_lmer3, "compare" = all.equal(icc11_lmer4, icc11_lmer3))
```

```
##   icc11_lmer4 icc11_lmer3 compare
## 1    0.675642    0.675642    TRUE
```

From a missing data framework, the LME estimates would be considered more unbiased. LME uses direct ML estimation to allow the outcome to be missing at random, while the ANOVA model assumes the outcome to be missing completely at random [3]. However, the ANOVA estimate of the ICC(1, 1) from the unbalanced case with missing data (0.7104009) is closer to the “true” value without any missing data (0.7374311) compared to the LME estimate (0.675642). I tried to find some quantitative psychology studies comparing the ANOVA estimate and the LME estimate of the ICC(1, 1) with unbalanced data, but I could not find any. Again, I believe the LME estimates have better statistical properties, but I could be wrong.

[3] Note, linear mixed effect models only assume the *outcome* to be missing at random. It still assumes the predictors are missing completely at random.

We have now gone over the first type of ICC: ICC(1, 1). You might be wondering what the numbers refer to in the "ICC(1, 1)" name. The first number is simply a placeholder for type of ICC and is categorical. Quantitative psychologists other than Shrout and Fleiss (1979) have used letters rather than numbers for the first number (e.g., McGraw & Wong, 1996). The second number/letter refers to the number of cases the ICC represents. The number "1" refers to a single case in each group, while the letter "k" refers to the number of cases in each group. For the `InsectSprays` example, this would be `size_avg` = 12. So while ICC(1, 1) refers to the ICC of a single case in each group, ICC(1, k) will refer to the ICC of the average of the cases in each group.

ICC(1, k) for multilevel analysis

ICC(1, k) uses the same statistical model as ICC(1, 1). Let's reacquaint ourselves with the linear mixed effects model for the ICC(1, 1) as it is the same model for the ICC(1, k).

```r
print(N) # number of total counts
```

```
## [1] 72
```

```r
print(size_avg) # average number counts in each group
```

```
## [1] 12
```

```r
lmer_obj <- lmer(count ~ 1 + (1 | spray), data = InsectSprays, REML = TRUE)
summary(lmer_obj)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: count ~ 1 + (1 | spray)
##    Data: InsectSprays
##
## REML criterion at convergence: 417.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.0820 -0.5298 -0.1040  0.4426  2.4325
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  spray    (Intercept) 43.20    6.573
##  Residual             15.38    3.922
## Number of obs: 72, groups:  spray, 6
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    9.500      2.723   3.489
```

```r
var_comp <- as.data.frame(VarCorr(lmer_obj))
print(var_comp)
```

```
##        grp        var1 var2     vcov    sdcor
## 1    spray (Intercept) <NA> 43.19878 6.572578
## 2 Residual        <NA> <NA> 15.38131 3.921902
```

```r
var_btw_lmer <- var_comp[var_comp$"grp" == "spray", "vcov"]
var_wth_lmer <- var_comp[var_comp$"grp" == "Residual", "vcov"]
```

Here is the formula for the ICC(1, 1) based on the variance components:

$$ICC_{1,1} = \frac{\sigma^2_{between}}{\sigma^2_{between} + \sigma^2_{within}}$$

An alternative way to write this formula is:

$$ICC_{1,1} = \frac{\sigma^2_{between}}{\sigma^2_{between} + \frac{\sigma^2_{within}}{1}}$$

These two formulas are mathematically identical because all we are doing is dividing by 1. The 1 is from the second number/letter in the ICC naming scheme from Shrout and Fleiss (1979). It refers to the ICC being based on a single observation for that group. However, this alternative formula now generalizes to the ICC based on the average number of observations in each group. Therefore, the formula for the ICC(1, k) is:

$$ICC_{1,k} = \frac{\sigma^2_{between}}{\sigma^2_{between} + \frac{\sigma^2_{within}}{k}}$$

Because we are averaging across the observation dimension, we are decreasing the impact of the within-group variance. Similar to indices of internal consistency (e.g., Cronbach's Alpha), the more raters (e.g., items in the case of Cronbach's Alpha), the higher the reliability.

Let's go ahead and apply that formula to our insect spray example:

```r
icc1k_lmer <- var_btw_lmer / (var_btw_lmer + (var_wth_lmer / size_avg))
print(icc1k_lmer)
```

```
## [1] 0.9711835
```

Because the ICC(1, k) is often used in multilevel analysis, it is available in the `multilevel` R package. However, it is often referred to as ICC2 in the multilevel literature, so that is the name of the function:

```r
aov_obj <- aov(count ~ spray, data = InsectSprays)
icc1k_aov <- ICC2(aov_obj)
data.frame(icc1k_lmer, icc1k_aov, "compare" = all.equal(icc1k_lmer, icc1k_aov))
```

```
##   icc1k_lmer icc1k_aov compare
## 1  0.9711835 0.9711835    TRUE
```

As you can see, the values are the same. We will get the same value with the `irr` R package as well:

```r
InsectSprays_wide <- t(unstack(InsectSprays, form = count ~ spray))
icc_obj <- icc(InsectSprays_wide, model = "oneway", unit = "average")
icc1k_irr <- icc_obj[["value"]]
data.frame(icc1k_lmer, icc1k_irr, "compare" = all.equal(icc1k_lmer, icc1k_irr))
```

```
##   icc1k_lmer icc1k_irr compare
## 1  0.9711835 0.9711835    TRUE
```

The interpretation of the ICC(1, k) is the reliability of the group means. So if we wanted to create an aggregate score of the group means, the ICC(1, k) would provide us the reliability of that score. Clearly, in the case of the insect sprays example, the reliability of that score would be very high: over 95%.

I want to take a moment and explain why I am using the symbol `size_avg` rather than `k` to refer to the average group size. For the ICC(1, 1) and ICC(1, k) we have only one grouping variable. In the inspect sprays example, that grouping variable is spray. There is not "observation" grouping variable. There is nothing that makes observation #1 in group A similar to observation #1 in group B; there is nothing that makes observation #2 in group A similar to observation #2 in group B, etc. If we do have similarities across the group observations, then we might have a second grouping variable called "observation". In that case we could use ICC(2, 1), ICC(2, k), ICC(3, 1), or ICC(3, k) since they allow for two grouping variables. ICC(1, 1) and ICC(1, k) assume we only have one grouping variable. Therefore the symbol `k` does not refer to a second grouping variable, but rather the average size of the groups. I wanted to make this clear, so I used the symbol `size_avg` rather than `k` in the R code. Note that adding a second grouping variable will then increase our reliablity as ICC(1, k) is always less than or equal to ICC(2, k) and ICC(3, k).

What about when you having unbalanced data. We looked at this before for ICC(1, 1). Here is the alternative dataset we created with certain cases removed:

```r
keep_rows <- c(1:2, 13:16, 25:30, 37:44, 49:58, 61:72)
InsectSprays3 <- InsectSprays[keep_rows, ]
print(InsectSprays3)
```

```
##    count spray
## 1     10     A
## 2      7     A
## 13    11     B
## 14    17     B
## 15    21     B
## 16    11     B
## 25     0     C
## 26     1     C
## 27     7     C
## 28     2     C
## 29     3     C
## 30     1     C
## 37     3     D
## 38     5     D
## 39    12     D
## 40     6     D
## 41     4     D
## 42     3     D
## 43     5     D
## 44     5     D
## 49     3     E
## 50     5     E
## 51     3     E
## 52     5     E
## 53     3     E
## 54     6     E
## 55     1     E
## 56     1     E
## 57     3     E
## 58     2     E
## 61    11     F
## 62     9     F
## 63    15     F
## 64    22     F
## 65    15     F
## 66    16     F
## 67    13     F
## 68    10     F
## 69    26     F
## 70    26     F
## 71    24     F
## 72    13     F
```

We will also have a different average number of observations in each group.

```
size_by3 <- c(tapply(InsectSprays3$"count", InsectSprays3$"spray", FUN = length))
print(size_by3)
```

```
##  A  B  C  D  E  F
##  2  4  6  8 10 12
```

```
size_avg3 <- mean(size_by3)
print(size_avg3)
```

```
## [1] 7
```

We apply the same formula:

```
lmer_obj3 <- lmer(count ~ 1 + (1 | spray), data = InsectSprays3, REML = TRUE)
var_comp3 <- as.data.frame(VarCorr(lmer_obj3))
var_btw_lmer3 <- var_comp3[var_comp3$"grp" == "spray", "vcov"]
var_wth_lmer3 <- var_comp3[var_comp3$"grp" == "Residual", "vcov"]
icc1k_lmer3 <- var_btw_lmer3 / (var_btw_lmer3 + (var_wth_lmer3 / size_avg3))
print(icc1k_lmer3)
```

```
## [1] 0.9358196
```

We can check this result out with the `ICC` function from the `psych` package since it allows you to use a linear mixed effects model to handle the missing data.

```
library(plyr)
tmp <- unstack(InsectSprays3, count ~ spray)
InsectSprays3_wide <- rbind.fill.matrix(lapply(X = tmp, FUN = t.default))
rownames(InsectSprays3_wide) <- levels(InsectSprays3$"spray")
print(InsectSprays3_wide)
```

```
##    1  2  3  4  5  6  7  8  9 10 11 12
## A 10  7 NA NA NA NA NA NA NA NA NA NA
## B 11 17 21 11 NA NA NA NA NA NA NA NA
## C  0  1  7  2  3  1 NA NA NA NA NA NA
## D  3  5 12  6  4  3  5  5 NA NA NA NA
## E  3  5  3  5  3  6  1  1  3  2 NA NA
## F 11  9 15 22 15 16 13 10 26 26 24 13
```

```r
library(psych)
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:str2str':
##
##     m2d
```

```r
ICC_obj3 <- ICC(InsectSprays3_wide, missing = FALSE, lmer = TRUE)
icc1k_ICC3 <- ICC_obj3[["results"]][ICC_obj3[["results"]]$"type" == "ICC1k", "ICC"]
data.frame(icc1k_lmer3, icc1k_ICC3, compare = all.equal(icc1k_lmer3, icc1k_ICC3))
```

```
##   icc1k_lmer3 icc1k_ICC3                              compare
## 1   0.9358196  0.9600965 Mean relative difference: 0.02594183
```

So the value of the ICC(1, k) is actually different. This is because the ICC function in the `psych` R package does not use size_avg, but instead uses size_max - the maximum number of observations in any group. I personally don't think this makes as much sense since you are assuming you have more information in the data than you actually do. But this practice is often done with indices of internal consistency (e.g., Cronbach's Alpha) as well and is attempting to better estimate what the reliability coefficient would be if you had complete data, which could correspond better to the population parameter. The drawback is the group means in your actual sample don't actually have that high of reliability due to the missing data. Regardless, we should be able reproduce it:

```r
size_max <- max(size_by3)
icc1k_lmer3_max <- var_btw_lmer3 / (var_btw_lmer3 + (var_wth_lmer3 / size_max))
data.frame(icc1k_lmer3_max, icc1k_ICC3, compare = all.equal(icc1k_lmer3_max, icc1k_ICC3))
```

```
##   icc1k_lmer3_max icc1k_ICC3                               compare
## 1       0.9615328  0.9600965 Mean relative difference: 0.001493716
```

The values are still very slightly different - enough that it is not due to estimation error from the optimization algorithm in `lmer` or something. I don't understand why these values are different, so I am unable to resolve this confusion. Perhaps in the future, I will and update this blog post. With that unsatisfying answer, we will go on to ICC(2, 1)!

ICC(2, 1) for inter-rater reliability

While ICC(1, 1) is often associated with multilevel analysis, ICC(2, 1) is more so associated with inter-rater reliability. For ICC(2, 1), we are going to use an example of wanting an average writing rating for each student in a school. For example, say we want to rate the quality of students' written essays in response to a standardized prompt. For simplicity, we will say the essays are given a single, overall quality rating ranging from 1 to 10. We will have 6 students in this example.

Something to keep in mind is that intraclass correlations are always used with continuous ratings. Technically the ICC models assume the ratings are normally distributed. In practice, the ratings are expected to be continuous and normal "enough", for which ordinal response scales (e.g., 1 = "Very Poor"; 2 = "Poor"; 3 = "Fair"; 4 = "Good"; 5 = "Very Good") are often deemed okay. However, ICCs cannot be used with nominal ratings where the response options are not ordered. In this case, you would use Cohen's Kappa or one of its variations (see the psych::cohen.kappa function).

Here is the dataset we will be using for this example. It is actually the example data used in Shrout and Fleiss (1979).

```
StudentRatings <- data.frame(
   "Ms. W" = c(9, 6, 8, 7, 10, 6),
   "Ms. X" = c(2, 1, 4, 1, 5, 2),
   "Ms. Y" = c(5, 3, 6, 2, 6, 4),
   "Ms. Z" = c(8, 2, 8, 6, 9, 7),
   row.names = c("ashley","bailey","chester","daniel","emily","frank"),
   check.names = FALSE
)
print(StudentRatings)
```

```
##         Ms. W Ms. X Ms. Y Ms. Z
## ashley      9     2     5     8
## bailey      6     1     3     2
## chester     8     4     6     8
## daniel      7     1     2     6
## emily      10     5     6     9
## frank       6     2     4     7
```

As you can see, each row corresponds to a different student who wrote an essay and each column refers to a different teacher who rated the quality of the students' essays. The values in the dataset are the ratings themselves from 1 to 10. Right away, you might notice that this dataset is structured differently than InsectSprays. There are two important differences:

1) The InsectSpray dataset is in the "long" format for grouped data, while the StudentRatings dataset is in the "wide" format for grouped data. In the "long" format there is a variable for the group labels. For InsectSprays this is the "spray" variable labeled as letters. For StudentRatings, this information is in the rows names specifying which row refers to which teacher's ratings. We can restructure the data from wide to long though.

```
library(reshape)
tmp <- as.data.frame(t(as.matrix(StudentRatings, force.rownames = TRUE)))
StudentRatings2 <- rev(melt(tmp, measure.vars = colnames(tmp),
   variable_name = "student"))
print(StudentRatings2)
```

```
##    value student
## 1      9  ashley
## 2      2  ashley
## 3      5  ashley
## 4      8  ashley
## 5      6  bailey
## 6      1  bailey
## 7      3  bailey
## 8      2  bailey
## 9      8 chester
## 10     4 chester
## 11     6 chester
## 12     8 chester
## 13     7  daniel
## 14      1  daniel
## 15      2  daniel
## 16      6  daniel
## 17     10   emily
## 18      5   emily
## 19      6   emily
## 20      9   emily
## 21      6   frank
## 22      2   frank
## 23      4   frank
## 24      7   frank
```

Now the StudentRatings2 dataset looks similar to the InsectSprays dataset where the "value" variable is like the "count" variable and the "student" variable is like the "spray" variable. However, something is missing… which brings us to the second difference between the datasets.

2) The InsectSpray dataset has one grouping variable, while the StudentRatings dataset has two grouping variables. The InsectSpray dataset just has the spray grouping variable. But the StudentRatings dataset has a student grouping variable *and* a teacher grouping variable. Indeed, we have both rownames *and* colnames in StudentRatings. The rownames represent one grouping variable and the colnames represent another. You could technically say that the InsectSprays dataset has a second grouping variable, which we created when we converted the data from long to wide format to get InsectSprays_wide.

```
print(InsectSprays_wide)
```

```
##   [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## A   10    7   20   14   14   12   10   23   17    20    14    13
## B   11   17   21   11   16   14   17   17   19    21     7    13
## C    0    1    7    2    3    1    2    1    3     0     1     4
## D    3    5   12    6    4    3    5    5    5     5     2     4
## E    3    5    3    5    3    6    1    1    3     2     6     4
## F   11    9   15   22   15   16   13   10   26    26    24    13
```

We have a row for each spray and then the columns are unspecified. The columns would arguably need to correspond to separate groups to say the columns represent another grouping variable. Perhaps each row corresponds to a particular plot of land that received different sprays throughout the year. However given the InsectSprays dataset does not provide any information about this potential second grouping variable, its safer to assume each value in the dataset corresponds to a different plot of land and that the columns in InsectSprays_wide are not separate groups.

Now StudentRatings2 has the student grouping variable, but the teacher grouping variable went away! It is certainly still there, it is just that our restructuring got rid of it. If we use a different function for restructuring the data, we can retain the teacher grouping variable in addition to the teacher grouping variable.

```
library(str2str)
StudentRatings3 <- stack2(data = StudentRatings, select.nm = names(StudentRatings),
   rtn.el.nm = "rating", rtn.vrbnames.nm = "teacher", rtn.rownames.nm = "student",
   order.by.rownames = FALSE, keep.nm = NULL) # need keep = NULL due to str2str bug
print(StudentRatings3)
```

```
##    student teacher rating
## 1   ashley   Ms. W      9
## 2   bailey   Ms. W      6
## 3  chester   Ms. W      8
## 4   daniel   Ms. W      7
## 5    emily   Ms. W     10
## 6    frank   Ms. W      6
## 7   ashley   Ms. X      2
## 8   bailey   Ms. X      1
## 9  chester   Ms. X      4
## 10  daniel   Ms. X      1
## 11   emily   Ms. X      5
## 12   frank   Ms. X      2
## 13  ashley   Ms. Y      5
## 14  bailey   Ms. Y      3
## 15 chester   Ms. Y      6
## 16  daniel   Ms. Y      2
## 17   emily   Ms. Y      6
## 18   frank   Ms. Y      4
## 19  ashley   Ms. Z      8
## 20  bailey   Ms. Z      2
## 21 chester   Ms. Z      8
## 22  daniel   Ms. Z      6
## 23   emily   Ms. Z      9
## 24   frank   Ms. Z      7
```

We now have two grouping variables: 1) students, 2) teachers. Once we have *two* grouping variables we have the option to move beyond ICC(1, 1) and on to ICC(2, 1) (or ICC(3, 1) as we will see later on). ICC(2, 1) is not possible with only one grouping variable: two grouping variables are needed. (To prevent any confusion, the number 2 in ICC(2, 1) does not refer to two grouping variables, but is simply a category label and does not carry any quantitative meaning. Indeed sometimes the category label A is used instead of 2 resulting in ICC(A, 1): Liljequist, Elfving, & Skavberg-Roaldsen, 2019)) With two grouping variables, we don't have just one between-group variance component, but two separate between-group variance components. This means we will have three variance components: 1) between-student, 2) between-teacher, 3) within-groups.

Let's go ahead and calculate these variance components. We will start with the ANOVA model. Although we won't be going over the incorrect observed variance components, the same two statistical issues are still at play: 1) degrees of freedom and 2) group mean sampling error.

ANOVA variance components

Because we will need them in just a moment, let's calculate the same descriptive information about the `TeacherRating3` dataset we did for the `InsectSprays` dataset.

```r
N <- nrow(StudentRatings3)
print(N)
```

```
## [1] 24
```

```r
n <- length(unique(StudentRatings3$"student"))
print(n)
```

```
## [1] 6
```

```
k <- length(unique(StudentRatings3$"teacher"))
print(k)
```

```
## [1] 4
```

We have 24 cases in the dataset. There are 6 students and 4 teachers. Notice, instead of calculating `size_avg` as the average number of ratings for each student, we have `k` as the number of teachers. They give the exact same number, but the use of the term `k` highlights that we have a second grouping variable now.

The formula for the ANOVA model with the students ratings dataset will include the rating variable on the left hand side and the two grouping variables on the right hand side.

```
aov_obj <- aov(rating ~ student + teacher, data = StudentRatings3)
aov_anova <- anova(aov_obj)
print(aov_anova)
```

```
## Analysis of Variance Table
##
## Response: rating
##           Df Sum Sq Mean Sq F value    Pr(>F)
## student    5 56.208  11.242  11.027 0.0001346 ***
## teacher    3 97.458  32.486  31.866 9.454e-07 ***
## Residuals 15 15.292   1.019
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can again calculate the correct variance components from the ANOVA model. We will start with the between-student variance component:

```
ms_student <- aov_anova["student", "Mean Sq"]
ms_within <- aov_anova["Residuals", "Mean Sq"]
var_student_aov <- (ms_student / k) - (ms_within / k)
print(var_student_aov)
```

```
## [1] 2.555556
```

Followed by the between-teacher variance component:

```
ms_teacher <- aov_anova["teacher", "Mean Sq"]
var_teacher_aov <- (ms_teacher / n) - (ms_within / n)
print(var_teacher_aov)
```

```
## [1] 5.244444
```

And then the within-groups variance component is simply the mean square within like the ICC(1, 1):

```
var_within_aov <- ms_within
print(var_within_aov)
```

```
## [1] 1.019444
```

Before we talked about ICC(1, 1) being defined as “the proportion of variance between groups” or interpreted as “the proportion of variance due to group membership”. The definition of ICC(2, 1) is not so straight forward because we have two separate grouping variables. And in the case of inter-rater reliability, we are focused on one grouping variable being the participants we want scores for and the other grouping variable being the raters that are treated as a means to an end of obtaining reliable participant scores. We don’t really care about the raters per se and are usually not interested in scores for the raters. In the context of our example, the participants are our students we want writing quality scores for and the raters are teachers who we are using to obtain reliable writing scores.

Therefore, the key grouping variable of interest is students and the key between-group variance component is the between-students one. We want “the proportion of variance between *students*”. So we are now specifying which grouping variable we are focusing on since there are multiple.

Let’s now return to the formula for the ICC(1, 1) based on variance components:

$$ICC_{1,1} = \frac{\sigma^2_{between}}{\sigma^2_{between} + \sigma^2_{within}}$$

One way to think about this formula is that it is the between-group variance of interest divided the total variance. With the insect spray example, we only had one between-group variance - between-sprays variance - which naturally was in the numerator. Then the denominator was the sum of all the variance components. We can apply that same logic to the student rating example and get the ICC(2, 1) formula based on variance components:

$$ICC_{2,1} = \frac{\sigma^2_{student}}{\sigma^2_{student} + \sigma^2_{teacher} + \sigma^2_{within}}$$

The between-student variance is in the numerator and the denominator is the sum of all the variance components.

Other than “proportion of variance between students”, another way to define the ICC(2, 1) is as a form of reliability. Reliability coefficient formulas usually have the following structure:

$$Reliability = \frac{\sigma^2_{true}}{\sigma^2_{true} + \sigma^2_{error}}$$

In the student ratings example, the “true” variance is the between-student variance and the “error” variance is the sum of the between-teacher and within-groups variances. It makes sense that the between-teacher variance would be treated as error because we don’t want teachers giving systematically different ratings than each other. We want them to be similar to each other, or *reliable*. Indeed, that is the whole idea of inter-rater reliability: do different raters, in this case teachers, give similar ratings for the same student. Systematically different ratings cause between-teacher variance. I say *systematically* because teacher ratings are bound to differ due to semi-random processes such as their mood on that particular day or whether they recently graded older students’ essays that are of higher quality, etc. Those semi-randomly different ratings would be in the within-groups variance since it is unrelated to students or teachers. After all, random is defined as unrelated to anything else in the model.

Let’s go ahead and calculate the ICC(2, 1):

```r
icc21_aov <- var_student_aov / (var_student_aov + var_teacher_aov + var_within_aov)
print(icc21_aov)
```

```
## [1] 0.2897638
```

We can check to make sure we have the correct value with the `irr` package.

```
icc_obj <- icc(ratings = StudentRatings, model = "twoway", type = "agreement")
icc21_irr <- icc_obj[["value"]]
data.frame(icc21_aov, icc21_irr, "compare" = all.equal(icc21_aov, icc21_irr))
```

```
##   icc21_aov icc21_irr compare
## 1 0.2897638 0.2897638    TRUE
```

The multilevel package does not have a function for calculating the ICC(2, 1) function as it is not traditionally used in multilevel analysis. However, the psych package does.

```
ICC_obj <- ICC(x = StudentRatings, lmer = TRUE)
icc21_ICC <- ICC_obj[["results"]][ICC_obj[["results"]]$"type" == "ICC2", "ICC"]
data.frame(icc21_aov, icc21_ICC, "compare" = all.equal(icc21_aov, icc21_ICC))
```

```
##   icc21_aov icc21_ICC                                 compare
## 1 0.2897638 0.2897642 Mean relative difference: 1.429381e-06
```

The psych package is using a linear mixed effects model to compute the ICC(2, 1) (when lmer = TRUE), so the value is slightly different. The negligable difference in the later decimal points are due to estimation error from the linear mixed effects model optimization algorithm. The ICC function can also provide us with the variance components to ensure we calculated them correctly.

```
varcomp_ICC <- d2v(ICC_obj[["lme"]]["variance"])
names(varcomp_ICC)[1:2] <- c("student","teacher")
print(varcomp_ICC)
```

```
##  student  teacher Residual    Total
## 2.555563 5.244451 1.019443 8.819457
```

Indeed, these are the same variance components we calculated (except for negligable differences due to the linear mixed effects model optimization algorithm):

```
data.frame(
   "var_student" = c(var_student_aov, varcomp_ICC["student"]),
   "var_teacher" = c(var_teacher_aov, varcomp_ICC["teacher"]),
   "var_within" = c(var_within_aov, varcomp_ICC["Residual"]),
   "icc11" = c(icc21_aov, icc21_ICC),
row.names = c("manual","psych"))
```

```
##        var_student var_teacher var_within     icc11
## manual    2.555556    5.244444   1.019444 0.2897638
## psych     2.555563    5.244451   1.019443 0.2897642
```

Let's now calculate the ICC(2, 1) with linear mixed effects modeling ourselves to discover that our variance components will then match exactly those from the psych R package.

LME variance components

Again, we will need the data in long format:

```
print(StudentRatings3)
```

```
##    student teacher rating
## 1   ashley   Ms. W      9
## 2   bailey   Ms. W      6
## 3  chester   Ms. W      8
## 4   daniel   Ms. W      7
## 5    emily   Ms. W     10
## 6    frank   Ms. W      6
## 7   ashley   Ms. X      2
## 8   bailey   Ms. X      1
## 9  chester   Ms. X      4
## 10  daniel   Ms. X      1
## 11   emily   Ms. X      5
## 12   frank   Ms. X      2
## 13  ashley   Ms. Y      5
## 14  bailey   Ms. Y      3
## 15 chester   Ms. Y      6
## 16  daniel   Ms. Y      2
## 17   emily   Ms. Y      6
## 18   frank   Ms. Y      4
## 19  ashley   Ms. Z      8
## 20  bailey   Ms. Z      2
## 21 chester   Ms. Z      8
## 22  daniel   Ms. Z      6
## 23   emily   Ms. Z      9
## 24   frank   Ms. Z      7
```

Again, we will specify a null, or "intercept-only" LME model where we have no predictors. However, this time we have two different grouping variables, so we will have two different sets of random effects: one for students and one for teachers.

```
lmer_obj <- lmer(rating ~ 1 + (1 | student) + (1 | teacher), data = StudentRatings3,
   REML = TRUE)
summary(lmer_obj)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rating ~ 1 + (1 | student) + (1 | teacher)
##    Data: StudentRatings3
##
## REML criterion at convergence: 91.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5153 -0.3782  0.3378  0.5360  0.8607
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  student  (Intercept) 2.556    1.599
##  teacher  (Intercept) 5.244    2.290
##  Residual             1.019    1.010
## Number of obs: 24, groups:  student, 6; teacher, 4
```

```
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    5.292      1.334   3.967
```

Again, in the LME model we don't have sum of squares or mean squares. Instead we directly estimate the variance components. The model will split up the "rating" variance into a between-group variance component for students, a between-group variance component for teachers, and a within-group variance component.

```
var_comp <- as.data.frame(VarCorr(lmer_obj))
print(var_comp)
```

```
##        grp        var1 var2     vcov    sdcor
## 1  student (Intercept) <NA> 2.555563 1.598613
## 2  teacher (Intercept) <NA> 5.244451 2.290077
## 3 Residual        <NA> <NA> 1.019443 1.009675
```

```
var_student_lmer <- var_comp[var_comp$"grp" == "student", "vcov"]
var_teacher_lmer <- var_comp[var_comp$"grp" == "teacher", "vcov"]
var_within_lmer <- var_comp[var_comp$"grp" == "Residual", "vcov"]
```

Let's go and compare these to the variance components from the `ICC` function in the `psych` R package:

```
var_lmer <- c(var_student_lmer, var_teacher_lmer, var_within_lmer)
var_ICC <- varcomp_ICC[-length(varcomp_ICC)]
var_compare <- unlist(Map(var_lmer, var_ICC, f = all.equal, check.attributes = FALSE))
var_all <- rbind.data.frame(var_lmer, var_ICC, var_compare)
names(var_all) <- c("student","teacher","within")
row.names(var_all) <- c("lmer","ICC","compare")
print(var_all)
```

```
##          student  teacher   within
## lmer    2.555563 5.244451 1.019443
## ICC     2.555563 5.244451 1.019443
## compare 1.000000 1.000000 1.000000
```

The 1.0s in the last row indicate the "lmer" and "ICC" rows are exactly identical. Let's go ahead and calculate the ICC(2, 1) value itself.

```
icc21_lmer <- var_student_lmer / (var_student_lmer + var_teacher_lmer + var_within_lmer)
print(icc21_lmer)
```

```
## [1] 0.2897642
```

```
data.frame(icc21_lmer, icc21_ICC, "compare" = all.equal(icc21_lmer, icc21_ICC))
```

```
##   icc21_lmer icc21_ICC compare
## 1  0.2897642 0.2897642    TRUE
```

The value of the ICC(2, 1) itself means that about 30% of the overall variance in student ratings is due to different students. It also means that any given teacher's ratings of a group of students is about 30% reliable. I don't know about you, but 30% reliability does not sound very good. I would not take any teacher ratings seriously if they only had 30% reliability.

One way to improve the reliability of teacher ratings is to use the average of four teachers' raters rather than just one teacher's rating. The ICC(2, 1) provides the reliability of a single teacher's rating, similar to how the ICC(1, 1) refers to a single observation in the group. ICC(2, k) will refer to the average score for each student across the four teachers, similar to how ICC(1, k) refers to the average of the observations in each group.

ICC(2, k) for inter-rater reliability

ICC(2, k) has the same relationship to ICC(2, 1) as ICC(1, k) has to ICC(1, 1). ICC(2, k) is the inter-rater reliability of the average score across the k raters. In our student ratings example, ICC(2, 1) was the inter-rater reliability of a single teacher's rating. ICC(2, k) is the inter-rater reliability of the score from averaging the four teacher's ratings.

Now the formula for the ICC(2, 1) was:

$$ICC_{2,1} = \frac{\sigma^2_{student}}{\sigma^2_{student} + \sigma^2_{teacher} + \sigma^2_{within}}$$

We are going to make the same change we made to the ICC(1, 1) formula and divide the variance components unique to the denominator by the number of teachers associated with the ICC(2, 1): 1 teacher. We also now have two variance components unique to the denominator that are treated as measurement error: 1) between-teacher variance component and 2) within-groups variance component. Both get divided by the number of teachers.

$$ICC_{2,1} = \frac{\sigma^2_{student}}{\sigma^2_{student} + \frac{\sigma^2_{teacher}}{1} + \frac{\sigma^2_{within}}{1}}$$

We can now replace the `1` with `k` for the ICC(2, k) formula. Note, I am not using `size_avg` anymore to highlight that we have the second grouping variable and `k` represents the number of groups for the second grouping variable (i.e., teachers).

$$ICC_{2,k} = \frac{\sigma^2_{student}}{\sigma^2_{student} + \frac{\sigma^2_{teacher}}{k} + \frac{\sigma^2_{within}}{k}}$$

If we apply that formula, we get:

```r
icc2k_lmer <- var_student_lmer / (var_student_lmer + (var_teacher_lmer / k) + (var_within_lmer / k))
print(icc2k_lmer)
```

```
## [1] 0.620051
```

We can check this with the `irr` R package:

```r
icc_obj <- icc(StudentRatings, model = "twoway", type = "agreement", unit = "average")
icc2k_irr <- icc_obj[["value"]]
data.frame(icc2k_lmer, icc2k_irr, "compare" = all.equal(icc2k_lmer, icc2k_irr))
```

```
##   icc2k_lmer icc2k_irr                                 compare
## 1   0.620051 0.6200505 Mean relative difference: 7.646633e-07
```

The negligable difference is due to the `icc` function in the `irr` package using an ANOVA model rather than a linear mixed effects model. We can also compare our ICC(2, k) value to that from the `ICC` function in the `psych` package which will use a linear mixed effects model when `lmer` = TRUE.

```
ICC_obj <- ICC(StudentRatings, lmer = TRUE)
icc2k_ICC <- ICC_obj[["results"]][ICC_obj[["results"]]$"type" == "ICC2k", "ICC"]
data.frame(icc2k_lmer, icc2k_ICC, compare = all.equal(icc2k_lmer, icc2k_ICC))
```

```
##   icc2k_lmer icc2k_ICC compare
## 1   0.620051  0.620051    TRUE
```

Now we have exactly identical ICC(2, k) values.

The interpretation of the ICC(2, k) means that about 60% of the variance in the average student rating score is reliable. So while any given teacher's rating was only 30% reliable, we doubled our reliability by combining the ratings from all four teachers. This is to be expected as the whole idea behind reliability is that as you average over more and more observations, reliability increases. Now, the reliability is still not great and a student could make a reasonable case for a bad grade on their written essay being due to measurement error.

Rater Bias

Another way to improve the reliability of teacher ratings is to use the same four teachers to grade the essays for all students in the school. Let's look at why that is. The ICC(2, 1) and ICC(2, k) assume that the four teachers we have in the data - Ms. W, Ms. X, Ms. Y, Ms. Z - are a random sample of all the teachers in the school district. In other words, the ICC(2, 1) and ICC(2, k) assume we are NOT using the same teachers to grade all student essays. So there might be 40 different teachers in the school district that we have available to provide student essay ratings. The data we have is from one round of ratings done by these particular 4 teachers on 6 students. With ICC(2, 1) and ICC(2, k), we are assuming the ratings are set up such that another round of ratings for the next set of 6 students might be done by 4 completely different teachers. So we might have 10 different rounds of ratings where each round involves 4 different teachers and 6 different students. At the end of the 10 rounds, we would have 4 ratings by each of the 60 students. However, most student ratings are made by different teachers. Because of this, the teacher ratings need to *agree* with each other to ensure students ratings made by different teachers are comparable. This is why the ICC(2, 1) and ICC(2, k) are sometimes referred to as the ICCs based on *absolute agreement*. Another way to think about it is that we not only care about the covariance structure of the ratings, but also the mean structure of the ratings.

By absolute agreement, we mean that each teacher's idea of a 6 out of 10 essay quality is the same, each teachers idea of a 3 out of 10 essay quality is the same, each teacher's idea of a 9 out of 10 essay quality is the same, etc. We can't have some teacher's being more lenient grader than others, because they wouldn't have absolute agreement. Similarly, We can't have some teachers being harsher graders than others, because we wouldn't have absolute agreement. Let's see whether we might have some more lenient vs. harsher graders in the dataset.

```
print(StudentRatings)
```

```
##         Ms. W Ms. X Ms. Y Ms. Z
## ashley      9     2     5     8
## bailey      6     1     3     2
## chester     8     4     6     8
## daniel      7     1     2     6
## emily      10     5     6     9
## frank       6     2     4     7
```

We can see this by taking the mean of each teacher's ratings:

```
colMeans(StudentRatings)
```

```
##    Ms. W    Ms. X    Ms. Y    Ms. Z
## 7.666667 2.500000 4.333333 6.666667
```

We can see the teachers clearly differ on how lenient vs. harsh they are. Ms. W is the most lenient grader giving an average of about 7.5 out of 10 for the 6 students as a set. Mr. X is the harshest grader giving an average of about 2.5 out of 10 for the 6 students as a set. For example, chester's essay was rated as an 8 out of 10 by Ms. W and a 4 out of 10 by Ms. X. Clearly, these two teachers do not have the same idea of what an 8 out of 10 essay quality means or what a 4 out of 10 essay quality means. This results in chester potentially getting a very different rating depending on what teacher happens to be grading his essay. This is exactly what we don't want and is the reason the inter-rater reliability based on the ICC(2, 1) is so low and the ICC(2, k) is subpar. This is sometimes refered to as rater bias, or in our case, teacher bias. We assume somewhere in the middle is most accurate, e.g., 6 out of 10 rating. Thus, Ms. X's ratings are downwardly biased while Ms. W's ratings are upwardly biased. It makes sense that bias in the raters would reduce our inter-rater reliability.

The large differences in the teacher means results in a large between-teacher variance component, and thus teacher bias. Remember, the between-teacher variance component is treated as measurement error with ICC(2, 1) and ICC(2, k). The larger the between-teacher variance, the more measurement error, and the lower the reliability. If we have many different teachers grading different sets of students' essays, then we are concerned about teacher bias and will want absolute agreement across the teacher ratings.

As I mentioned earlier, one way to improve our inter-rater reliability is to use the same 4 teachers to grade all 60 students' essays. Why does that change the reliability? Because now we don't need abolute agreement across the 40 teachers. Now any teacher who is grading chester's essay, is also grading the other 59 students' essays. So if Ms. W is a lenient grader, all of the students will get that lenient grading. In other words, Ms. W is biased to all the students, not just the 6 out of 60 she happened to grade. Similarly, if Ms. X is a harsh grader, all of the students will get that harsh grading. In other words, Mr. X is biased to all the students, not just the 6 out of 60 she happened to grade. We essentially can ignore teacher biases because all 60 students are being graded by the same 4 teachers.

This is exactly what the ICC(3, 1) and ICC(3, k) do, which we will now go over.

ICC(3, 1) for inter-rater reliability

The ICC(3, 1) is the same as ICC(2, 1) in that it requires two different grouping variables, but it now treats the raters - in our example the teachers - as fixed (rather than random). By fixed, we mean that the same teachers are rating all the students. This includes any potential future students from the next year that we want to compare. If we use these 4 teachers to grade this year's 60 students, but then we use another 4 teachers to grade next year's 60 students, to compare across years we would need to use ICC(2, 1) instead of ICC(3, 1). Of course, if we don't want to compare across years and only within years, then we could use ICC(3, 1).

While ICC(3, 1) does not care about any teacher biases (aka the mean structure of the ratings), there are still sources of potential unreliability. It is possible that these four teachers are not *consistent* in their biases across students. For example, Ms. W might be more lenient towards some students rather than others. This can be thought of as an interaction between the teacher and the student. The teacher's level of leniency or harshness depends on the student. Similarly, Ms. X might be harsher towards some students rather than others. While she has an "average" level of harshness, her harshness might differ depending on whether she likes that student, how good their handwriting is, whether they know the typical grades the student gets on past writing assignments, etc. We would not want that and this type of interaction bias would reduce inter-rater reliability.

For this reason, the ICC(3, 1) is sometimes referred to as the ICC based on *relative consistency*. We don't care whether the teacher's abolsutely agree with one another, just that they are relatively consistent. Returning to chester as an example, if Ms. W rates him as above average for her ratings, then we would want Ms. X to also rate him as above average for her ratings. Their ratings for chester might be completely different (aka the mean structure can be different), but their rank ordering we want to be the same (aka the covariance structure should be the same). As you can see, chester was rated as above Ms. W's average of 7.67 by being given a rating of 8 and was rated as above Ms. X's average of 2.50 by being given a rating of 4. However, let's see just how close their rank ordering really is. We can do this by examining the z-scores for chester across teachers.

```
print(StudentRatings)
```

```
##         Ms. W Ms. X Ms. Y Ms. Z
## ashley      9     2     5     8
## bailey      6     1     3     2
## chester     8     4     6     8
## daniel      7     1     2     6
## emily      10     5     6     9
## frank       6     2     4     7
```

```
means <- apply(X = StudentRatings, MARGIN = 2, FUN = mean)
sds <- apply(X = StudentRatings, MARGIN = 2, FUN = sd)
print(means); print(sds)
```

```
##    Ms. W    Ms. X    Ms. Y    Ms. Z
## 7.666667 2.500000 4.333333 6.666667
```

```
##    Ms. W    Ms. X    Ms. Y    Ms. Z
## 1.632993 1.643168 1.632993 2.503331
```

```
z_scores <- setNames((8 - means[["Ms. W"]]) / sds[["Ms. W"]], nm = "Ms. W")
append(z_scores, nm = "Ms. X") <- (4 - means[["Ms. X"]]) / sds[["Ms. X"]]
print(z_scores)
```

```
##     Ms. W     Ms. X
## 0.2041241 0.9128709
```

As you can see, the z-score for Ms. X is quite a bit larger than the z-score for Ms. W. Therefore, Ms. X is giving chester a *relatively* better rating than Ms. W, even though Ms. W is giving chester an *absolutely* better rating. We can also do this for the other two teachers.

```
append(z_scores, nm = "Ms. Y") <- (6 - means[["Ms. Y"]]) / sds[["Ms. Y"]]
append(z_scores, nm = "Ms. Z") <- (8 - means[["Ms. Z"]]) / sds[["Ms. Z"]]
print(z_scores)
```

```
##     Ms. W     Ms. X     Ms. Y     Ms. Z
## 0.2041241 0.9128709 1.0206207 0.5326236
```

Again some decent discrepencies. The fact that the z-scores for chester range from 0.20 to 1.02 will reduce the inter-rater reliability. This source of interaction bias is taken into account for both ICC(2, 1) and ICC(3, 1).

Let's go ahead and calculcate the ICC(3, 1) value itself with an ANOVA model.

ANOVA variance components

The ANOVA model needed for ICC(3, 1) is actually the exact same as the one needed for ICC(2, 1). I will repeat the model below, but it is identical to the one we already computed.

```
aov_obj <- aov(rating ~ student + teacher, data = StudentRatings3)
aov_anova <- anova(aov_obj)
print(aov_anova)
```

```
## Analysis of Variance Table
##
## Response: rating
##           Df Sum Sq Mean Sq F value    Pr(>F)
## student    5 56.208  11.242  11.027 0.0001346 ***
## teacher    3 97.458  32.486  31.866 9.454e-07 ***
## Residuals 15 15.292   1.019
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can calculate the variance components from the ANOVA model. Again, they are identical to the ones we already calculated for ICC(2, 1). We start with the between-student variance components:

```
ms_student <- aov_anova["student", "Mean Sq"]
ms_within <- aov_anova["Residuals", "Mean Sq"]
var_student_aov <- (ms_student / k) - (ms_within / k)
print(var_student_aov)
```

```
## [1] 2.555556
```

Followed by the between-teacher variance component:

```
ms_teacher <- aov_anova["teacher", "Mean Sq"]
var_teacher_aov <- (ms_teacher / n) - (ms_within / n)
print(var_teacher_aov)
```

```
## [1] 5.244444
```

And then the within-groups variance component is simply the mean square within like before:

```
var_within_aov <- ms_within
print(var_within_aov)
```

```
## [1] 1.019444
```

Let's now return to the formula for the ICC(2, 1) based on variance components:

$$ICC_{2,1} = \frac{\sigma^2_{student}}{\sigma^2_{student} + \sigma^2_{teacher} + \sigma^2_{within}}$$

The between-student variance is in the numerator and the denominator is the sum of all the variance components. The difference with ICC(3, 1) is that we don't care about rater bias (aka teacher bias).

Therefore, we don't include the between-teacher variance component in the denominator. The denominator is no longer the sum of all the variance components, but just the sum of the between-student variance component and the within-group variance component.

$$ICC_{3,1} = \frac{\sigma^2_{student}}{\sigma^2_{student} + \sigma^2_{within}}$$

Now remember I commented that we still care about the interaction bias. We haven't talked about it this way yet, but the within-group variance component is capturing the interaction bias from teachers being more lenient or harsher depending on the student. However, we can't really tease this apart from other sources of within-group variance because each teacher only rates each student once. The cell sample size for each student-teacher combination is n = 1 and you can't estimate a variance with cell sample sizes of 1. Of course, not all of the within-group variance is the interaction bias. Some of the within-group variance is due to random measurement error. Some of this could be theoretically "true" randomness, but it likely includes a lot of unconscious influences the teachers are unaware of: whether they think the essay topic is interesting, how many hours into grading a particular essay is rated, their mood that day, the weather outside, etc.

Note that the ICC(3, 1) still uses the traditional formula for a reliability coefficient:

$$Reliability = \frac{\sigma^2_{true}}{\sigma^2_{true} + \sigma^2_{error}}$$

We have simply changed what is included in the error variance for the formula.

Let's go ahead and calculate the ICC(3, 1):

```
icc31_aov <- var_student_aov / (var_student_aov + var_within_aov)
print(icc31_aov)
```

```
## [1] 0.7148407
```

We can check to make sure we have the correct value with the `irr` package.

```
icc_obj <- icc(ratings = StudentRatings, model = "twoway", type = "consistency")
icc31_irr <- icc_obj[["value"]]
data.frame(icc31_aov, icc31_irr, "compare" = all.equal(icc31_aov, icc31_irr))
```

```
##   icc31_aov icc31_irr compare
## 1 0.7148407 0.7148407    TRUE
```

And then also with the `psych` package.

```
ICC_obj <- ICC(x = StudentRatings, lmer = TRUE)
icc31_ICC <- ICC_obj[["results"]][ICC_obj[["results"]]$"type" == "ICC3", "ICC"]
data.frame(icc31_aov, icc31_ICC, "compare" = all.equal(icc31_aov, icc31_ICC))
```

```
##   icc31_aov icc31_ICC                                compare
## 1 0.7148407 0.7148415 Mean relative difference: 1.114474e-06
```

Similar to before, the negligable difference in the later decimal points is due to the estimation error in the optimization algorithm for the linear mixed effects model the `ICC` function from the `psych` package uses when `lmer` = TRUE. The `psych` package also can provide us the variance components to ensure we calculated them correctly with the ANOVA model.

```
varcomp_ICC <- d2v(ICC_obj[["lme"]]["variance"])
names(varcomp_ICC)[1:2] <- c("student","teacher")
data.frame(
   "var_student" = c(var_student_aov, varcomp_ICC["student"]),
   "var_teacher" = c(var_teacher_aov, varcomp_ICC["teacher"]),
   "var_within" = c(var_within_aov, varcomp_ICC["Residual"]),
   "icc11" = c(icc31_aov, icc31_ICC),
row.names = c("manual","psych"))
```

```
##        var_student var_teacher var_within     icc11
## manual    2.555556    5.244444   1.019444 0.7148407
## psych     2.555563    5.244451   1.019443 0.7148415
```

Let's calculate the ICC(3, 1) with a linear mixed effects model now.

LME variance components

Again, it is the exact same model we used for the ICC(2, 1):

```
lmer_obj <- lmer(rating ~ 1 + (1 | student) + (1 | teacher), data = StudentRatings3,
   REML = TRUE)
summary(lmer_obj)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rating ~ 1 + (1 | student) + (1 | teacher)
##    Data: StudentRatings3
##
## REML criterion at convergence: 91.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5153 -0.3782  0.3378  0.5360  0.8607
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  student  (Intercept) 2.556    1.599
##  teacher  (Intercept) 5.244    2.290
##  Residual             1.019    1.010
## Number of obs: 24, groups:  student, 6; teacher, 4
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    5.292      1.334   3.967
```

As well as the exact same variance components:

```
var_comp <- as.data.frame(VarCorr(lmer_obj))
print(var_comp)
```

```
##        grp        var1 var2     vcov    sdcor
## 1  student (Intercept) <NA> 2.555563 1.598613
## 2  teacher (Intercept) <NA> 5.244451 2.290077
## 3 Residual        <NA> <NA> 1.019443 1.009675
```

```r
var_student_lmer <- var_comp[var_comp$"grp" == "student", "vcov"]
var_teacher_lmer <- var_comp[var_comp$"grp" == "teacher", "vcov"]
var_within_lmer <- var_comp[var_comp$"grp" == "Residual", "vcov"]
var_aov <- c(var_student_aov, var_teacher_aov, var_within_aov)
var_lmer <- c(var_student_lmer, var_teacher_lmer, var_within_lmer)
var_compare <- rbind.data.frame(var_aov, var_lmer)
names(var_compare) <- c("student","teacher","within")
row.names(var_compare) <- c("aov","lmer")
print(var_compare)
```

```
##       student  teacher   within
## aov  2.555556 5.244444 1.019444
## lmer 2.555563 5.244451 1.019443
```

As expected, we get the same variance components and ICC(3, 1) value whether we use the ANOVA model or the LME model (outside of estimation error due to the optimization algorithm in the linear mixed effects model).

```r
icc31_lmer <- var_student_lmer / (var_student_lmer + var_within_lmer)
data.frame(icc31_aov, icc31_lmer, "compare" = all.equal(icc31_aov, icc31_lmer))
```

```
##   icc31_aov icc31_lmer                                 compare
## 1 0.7148407  0.7148415 Mean relative difference: 1.114474e-06
```

The value of the ICC(3, 1) itself means that about 70% of the overall variance in student ratings is due to different students - if we can assume the same teachers will be rating all of them. It also means that any one of the four teacher's ratings of a group of students is about 70% reliable. This is a much more acceptable degree of inter-rater reliability compared with the 30% we had before with ICC(2, 1). Clearly, using the same four teachers to grade all students' essays has a major reliability advantage.

Let's now move on to ICC(3, k) where we will see the highest inter-rater reliability.

ICC(3, k) for inter-rater reliability

We use the same statistical model for ICC(3, k) as we did for ICC(3, 1), so we don't need to recalculate that. Let's re-visit the formula for ICC(3, 1):

$$ICC_{3,1} = \frac{\sigma^2_{student}}{\sigma^2_{student} + \sigma^2_{within}}$$

. It is the formula for ICC(2, 1) without the between-teacher variance component since we are ignoring teacher bias. For ICC(3, k), we are again going to divide the terms unique to the denominator by k. In this case, just the within-groups variance, resulting in the formula:

$$ICC_{3,k} = \frac{\sigma^2_{student}}{\sigma^2_{student} + \frac{\sigma^2_{within}}{k}}$$

Let's go ahead and apply that formula to our example:

```r
icc3k_lmer <- var_student_lmer / (var_student_lmer + (var_within_lmer / k))
print(icc3k_lmer)
```

```
## [1] 0.9093159
```

Again, we can check this with the `irr` and `psych` R packages:

```
icc_obj <- icc(StudentRatings, model = "twoway", type = "consistency", unit = "average")
icc3k_irr <- icc_obj[["value"]]
data.frame(icc3k_lmer, icc3k_irr, "compare" = all.equal(icc3k_lmer, icc3k_irr))
```

```
##   icc3k_lmer icc3k_irr                                compare
## 1  0.9093159 0.9093155 Mean relative difference: 3.544171e-07
```

```
ICC_obj <- ICC(StudentRatings, lmer = TRUE)
icc3k_ICC <- ICC_obj[["results"]][ICC_obj[["results"]]$"type" == "ICC3k", "ICC"]
data.frame(icc3k_lmer, icc3k_ICC, compare = all.equal(icc3k_lmer, icc3k_ICC))
```

```
##   icc3k_lmer icc3k_ICC compare
## 1  0.9093159 0.9093159    TRUE
```

The interpretation of the ICC(3, k) means that about 90% of the variance in the average score is reliable - if we can assume we are using the same teachers for all students. So while any given teacher's rating was about 70% reliable, the reliability increased to 90% by averaging the raters from all four teachers. We would consider this reliability to be great and feel confident that these four teachers could reliability grade all the students' essays.

Now something you might have been curious about is how the **intra**class correlation relates to the more traditional Pearson's correlation. Using similar terminology, we can think of the Pearson correlation as the **inter**class correlation. The **intra**class correlation is quantifying the correlation of observations within the same class (aka group). The **inter**class correlation is quantifying the correlation of observations between different classes (aka groups). The **intra**class correlation is most useful when the raters are random rather than fixed. When there is a larger population of raters that we don't have data on, but want to make inferences about, that is what ICC(2, 1) and ICC(2, k) are for. However, for ICC(3, 1) and ICC(3, k), we have data on the entire population of raters because those are the only raters we are using. In this case, we can actually use the **inter**class correlation.

For example, we could take the average inter-class correlation (aka Pearson's correlation) between the raters:

```
tmp <- cor(StudentRatings)
cor_all <- tmp[lower.tri(tmp)]
cor_avg <- mean(cor_all)
print(cor_avg)
```

```
## [1] 0.7603077
```

While this can be useful for people who know how to interpret correlations, it is not a reliability coefficient like the ICCs were. However, we can use the spearman brown formula to convert the average inter-class correlation into a reliability coefficient:

```
sb_coef <- (k * cor_avg) / (1 + (k - 1) * cor_avg)
print(sb_coef)
```

```
## [1] 0.9269436
```

We can check this with the `CTT` R package:

```r
library(CTT)
```

```
##
## Attaching package: 'CTT'

## The following objects are masked from 'package:psych':
##
##     polyserial, reliability
```

```r
sb_coef_CTT <- unlist(spearman.brown(cor_avg, input = k, n.or.r = "n"), use.names = FALSE)
data.frame(sb_coef, sb_coef_CTT, "compare" = all.equal(sb_coef, sb_coef_CTT))
```

```
##     sb_coef sb_coef_CTT compare
## 1 0.9269436   0.9269436    TRUE
```

Note, this reliability coefficient is different than the ICC(3, k).

```r
data.frame(icc3k_lmer, sb_coef, "compare" = all.equal(icc3k_lmer, sb_coef))
```

```
##   icc3k_lmer   sb_coef                              compare
## 1  0.9093159 0.9269436 Mean relative difference: 0.01938576
```

So how is it that ICC(3, k) can also be thought of reliability based on inter-class **correlations** then? Well technically ICC(3, k) can be thought of as reliability based on inter-class **covariances**. The ICC(3, k) works with the raw datums (i.e., covariance), not their standardized versions (i.e., correlation). Let's instead, compute the average inter-class covariance:

```r
tmp <- cov(StudentRatings)
cov_all <- tmp[lower.tri(tmp)]
cov_avg <- mean(cov_all)
print(cov_avg)
```

```
## [1] 2.555556
```

Not very interpretable, but we can use the formula for Cronbach's Alpha to convert the average inter-item covariance into a reliability coefficient:

```r
var_all <- apply(X = StudentRatings, MARGIN = 2, FUN = var)
var_avg <- mean(var_all)
alpha_coef <- (k * cov_avg) / (var_avg + (k - 1) * cov_avg)
print(alpha_coef)
```

```
## [1] 0.9093155
```

Now notice that we get the same value as the ICC(3, k). That is because ICC(3, k) and Cronbach's Alpha are actually the same reliability coefficient!

```
data.frame(icc3k_lmer, alpha_coef, all.equal(icc3k_lmer, alpha_coef))
```

```
##   icc3k_lmer alpha_coef      all.equal.icc3k_lmer..alpha_coef.
## 1  0.9093159  0.9093155 Mean relative difference: 3.544171e-07
```

(Again, the negligable difference is due to estimation error from the optimization algorithm used by the linear mixed effects model.) Furthermore, what we calculated above with the average inter-item correlation was Standardized Cronbach's Alpha. We can use the `psych` R package to confirm:

```
alpha_obj <- alpha(StudentRatings)
alpha_coef_psych <- alpha_obj[["total"]][["raw_alpha"]]
data.frame(alpha_coef, alpha_coef_psych, "compare" = all.equal(alpha_coef, alpha_coef_psych))
```

```
##   alpha_coef alpha_coef_psych compare
## 1  0.9093155        0.9093155    TRUE
```

```
sb_coef_psych <- alpha_obj[["total"]][["std.alpha"]]
data.frame(sb_coef, sb_coef_psych, "compare" = all.equal(sb_coef, sb_coef_psych))
```

```
##     sb_coef sb_coef_psych compare
## 1 0.9269436     0.9269436    TRUE
```

Therefore, researchers will sometimes use Cronbach's Alpha to estimate inter-rater reliability when the ratings are continuous variables and the raters are the same for all participants. After all, Cronbach's Alpha is the same as ICC(3, k)!

Summary

To summarize, I have made a table that compares the six different types of ICCs:

```
library(knitr)
sum_tab <- data.frame("grouping variables" = character(6), "study raters" = character(6), "type of relia
   row.names = c("ICC(1, 1)","ICC(2, 1)","ICC(3, 1)","ICC(1, k)","ICC(2, k)","ICC(3, k)"), check.names =
sum_tab["ICC(1, 1)",] <- c("one","random","absolute agreement","yes","single")
sum_tab["ICC(2, 1)",] <- c("two","random","absolute agreement","yes","single")
sum_tab["ICC(3, 1)",] <- c("two","fixed","relative consistency","no","single")
sum_tab["ICC(1, k)",] <- c("one","random","absolute agreement","yes","average")
sum_tab["ICC(2, k)",] <- c("two","random","absolute agreement","yes","average")
sum_tab["ICC(3, k)",] <- c("two","fixed","relative consistency","no","average")
kable(sum_tab)
```

	grouping variables	study raters	type of reliability	mean structure	score used
ICC(1, 1)	one	random	absolute agreement	yes	single
ICC(2, 1)	two	random	absolute agreement	yes	single
ICC(3, 1)	two	fixed	relative consistency	no	single
ICC(1, k)	one	random	absolute agreement	yes	average
ICC(2, k)	two	random	absolute agreement	yes	average
ICC(3, k)	two	fixed	relative consistency	no	average

You can see this same summary of the differences between the six types of ICCs from printing the return object from the `ICC` function in the `psych` R package. The rownames of the data.frame printed provide very similar information I put in the table above.

```
ICC(StudentRatings, lmer = TRUE)
```

```
## Call: ICC(x = StudentRatings, lmer = TRUE)
##
## Intraclass correlation coefficients
##                          type  ICC    F df1 df2       p lower bound upper bound
## Single_raters_absolute   ICC1 0.17  1.8   5  18 0.16477      -0.133        0.72
## Single_random_raters     ICC2 0.29 11.0   5  15 0.00013       0.019        0.76
## Single_fixed_raters      ICC3 0.71 11.0   5  15 0.00013       0.342        0.95
## Average_raters_absolute ICC1k 0.44  1.8   5  18 0.16477      -0.884        0.91
## Average_random_raters   ICC2k 0.62 11.0   5  15 0.00013       0.071        0.93
## Average_fixed_raters    ICC3k 0.91 11.0   5  15 0.00013       0.676        0.99
##
##  Number of subjects = 6     Number of Judges =  4
## See the help file for a discussion of the other 4 McGraw and Wong estimates,
```

As you can see, there are also formulas for constructing confidence intervals for the ICCs. These are asymmetrical as the ICC is bounded from 0 to 1 (although you can get values less than 0 in a sample due to more within-group variance than expected for the group sizes). Perhaps a future blog post will go into the formulas for the confidence intervals. You can read more about them in McGraw, Kenneth, & Wong (1996).

So hopefully now you have a better sense for what Intraclass Correlations (ICCs) are and what the six different types of ICCs are. For those interested in more in-depth reading on the topic, I recommend Liljequist, Elfving, & Shavberg-Roaldson, 2019, which provides a pedagogical discussion of the differences between ICC(1, 1), ICC(2, 1), and ICC(3, 1).

Epilogue: ICCs as emotion differentiation scores

I also wanted to take this moment to comment on using the ICC to create scores for emotion differentiation in psychological science (Kashdan, Barrett, & McKnight (2015)). Emotion differentiation is how well a person is able to distinguish the different emotions they are feeling. Usually this is restricted to a particular valence of emotion: positive or negative. For example, can a person identify that they are feeling anxious rather than angry, or sad rather than guilty, etc. (negative emotions). When emotion differentiation was first quantified with data from experience sampling methods, researchers used the average inter-class correlation (aka Pearson's correlation) between a person's responses to the negative emotion items in the experience sampling survey (Barrett, Gross, Conner, Benvenuto, 2001; Demiralp et al., 2012). This was most similar to the ICC(3, k) as I showed above. However, when **intra**-class correlations started being used rather than **inter**-class correlation, the ICC(2, k) was used (Tugade, Fredrickson, & Barrett, 2004; Kashdan, Ferssizidis, Collins, & Muraven, 2010). This scoring resulted in mean differences between the negative emotion items being taken into account as well as correlations between the items.

For example, say a person completed an experience sampling survey with items asking them to rate the intensity of their negative emotions in the moment on a response scale from 0 = "not at all" to 5" = "extremely". And say a person tends to not report guilt or sadness most of the time. However, when the person does report guilt, they tend to also report sadness. They will report a 4 out of 5 for guilt and a 2 out of 5 for sadness. The average inter-item correlation and the ICC(3, k) will not quantify the fact that the person's response for sadness tends to be half that for guilt. Instead it will only quantify the fact that whenever the person reports non-zero guilt, they tend to report non-zero sadness as well. The ICC(2, k) on the other hand will quantify not only only the fact that whenever the person reports non-zero guilt, they tend to report non-zero sadness, but also the fact the person's response for sadness tends to half that for guilt. That is because the ICC(2, k) is taking into account the mean structure - mean differences across the negative emotion items. In terms of inter-rater reliability language (although it conceptually does not apply here), the item bias would be taken into account. This makes sense to me since a person is theoretically doing some differentiation if they can identify they are feeling sadness less than guilt, even if they might not be able to fully differentiate between the two emotions.

However, some researchers have proposed returning to the conceptual underpinnings of the average interclass correlation and using the ICC(3, k) rather than the ICC(2, k) (Erbas, Ceulemans, Lee Pe, Koval, & Kuppens, 2014; Erbas, Ceulemans, Blanke, Sels, Fischer, & Kuppens, 2019). They argue that the mean structure of the emotions should not be taken into account as it doesn't theoretically measure emotion differentiation. I admit I don't really understand their argument as it seems to treat emotions as light switches that are either on or off and ignore their degree of intensity. I would think someone with very high emotion differentiation has excellent self-awareness of the emotions they are feeling **and** the intensities they are feeling those emotions at. But perhaps I am missing something.

Regardless of which type of ICC is used to measure emotion differentiation, I do think there is a larger conceptual problem with the measurement. The ICC cannot distinguish between a person genuinely experiencing multiple emotions at the same time and someone experiencing a single emotion, but reporting multiple emotions because they lack self-awareness. This seems like a rather big problem since people likely experience multiple emotions all the time - that is part of the human experience. Therefore, it might be that the ICC is capturing a person's tendancy to experience multiple negative emotions at the same time rather than negative emotion differentiation, per se. This would theoretically make sense with the findings that negative emotion differentiation tends to predict worse mental health (Sean & Coifman, in press. People with mental health difficulties often have negative emotion regulation problems and are probably more likely to be experiencing multiple negative emotions at the same time. Ultimately, it is an unfalsifiable question as the person's reports on the experience sampling survey cannot distinguish these two cases. And if we tried to use qualitative interviews to ask people whether they are genuintely experiencing multiple emotions or experiencing a single emotion and reporting multiple emotions because they lack self-awareness, they would only be able to tell us if they were high on emotion differentiation since people low on emotion differentiation wouldn't be accurate reporters!

I think that Starr, Hershenberg, Li, & Shaw (2017) put it nicely in their discussion section on page 626:

"Notably, although research has typically asummed that low ED scores reflect an inability to cognitively discriminate between emotions, they may also report a genuine propensity toward experiencing multiple emotions in clusters rather than individually... Although our approach to assessing NED and PED (using EMAs to calculate intraclass correlations among momentary emotions) is a widely accepted method (Selby et al., 2014; Shrout & Fleiss, 1979; Tugade et al., 2004), future researchers should develop techniques that better discriminate between the experience and the discernment of multiple concurrent emotions."

I don't have the solution to what better measurement techniques we should use other than the ICC to measure emotion differentiation. But I agree with Starr and colleagues that ideally researchers would be able to find something that teases apart genuine concurrent emotions and the inability to discern emotions. Perhaps someone reading this blog post will figure it out!

References

Barrett, L. F., Gross, J., Christensen, T. C., & Benvenuto, M. (2001). Knowing what you're feeling and knowing what to do about it: Mapping the relation between emotion differentiation and emotion regulation. *Cognition & Emotion, 15*(6), 713-724.

Brown, B. A., Goodman, F. R., Disabato, D. J., Kashdan, T. B., Armeli, S., & Tennen, H. (in press). Does negative emotion differentiation influence how people choose to regulate their distress after stressful events? A four-year daily diary study. *Emotion.*

Demiralp, E., Thompson, R. J., Mata, J., Jaeggi, S. M., Buschkuehl, M., Barrett, L. F., ... & Jonides, J. (2012). Feeling blue or turquoise? Emotional differentiation in major depressive disorder. *Psychological*

Science, 23(11), 1410-1416.

Erbas, Y., Ceulemans, E., Blanke, E. S., Sels, L., Fischer, A., & Kuppens, P. (2019). Emotion differentiation dissected: Between-category, within-category, and integral emotion differentiation, and their relation to well-being. *Cognition and Emotion, 33*(2), 258-271.

Erbas, Y., Ceulemans, E., Lee Pe, M., Koval, P., & Kuppens, P. (2014). Negative emotion differentiation: Its personality and well-being correlates and a comparison of different assessment methods. *Cognition and Emotion, 28*(7), 1196-1213.

Kashdan, T. B., Barrett, L. F., & McKnight, P. E. (2015). Unpacking emotion differentiation: Transforming unpleasant experience by perceiving distinctions in negativity. *Current Directions in Psychological Science, 24*(1), 10-16.

Kashdan, T. B., Ferssizidis, P., Collins, R. L., & Muraven, M. (2010). Emotion differentiation as resilience against excessive alcohol use: An ecological momentary assessment in underage social drinkers. *Psychological Science, 21*(9), 1341-1347.

Liljequist, D., Elfving, B., & Skavberg Roaldsen, K. (2019). Intraclass correlation–A discussion and demonstration of basic features. *PloS one, 14*(7), e0219854.

McGraw, K. O., & Wong, S. P. (1996). Forming inferences about some intraclass correlation coefficients. *Psychological Methods, 1*(1), 30-46

Muthén, B. (1994). Multilevel covariance structure analysis. *Sociological Methods & Research, 22*, 376-398.

Preacher, K. J., Zyphur, M. J., & Zhang, Z. (2010). A general multilevel SEM framework for assessing multilevel mediation. Psychological methods, 15(3), 209-233.

Seah, T. H., & Coifman, K. G. (in press). Emotion differentiation and behavioral dysregulation in clinical and nonclinical samples: A meta-analysis. *Emotion.*

Shrout, P. E., & Fleiss, J. L. (1979). Intraclass correlations: uses in assessing rater reliability. *Psychological Bulletin, 86*(2), 420.

Snijders, T. A. B., & Bosker, R. J. (2012). *Multilevel analysis: An introduction to basic and advanced multilevel modeling - second edition.* Thousand Oaks, CA: Sage Publications.

Starr, L. R., Hershenberg, R., Li, Y. I., & Shaw, Z. A. (2017). When feelings lack precision: Low positive and negative emotion differentiation and depressive symptoms in daily life. *Clinical Psychological Science, 5*(4), 613-631.

Tugade, M. M., Fredrickson, B. L., & Feldman Barrett, L. (2004). Psychological resilience and positive emotional granularity: Examining the benefits of positive emotions on coping and health. *Journal of Personality, 72*(6), 1161-1190.