The goals of this post are to (1) examine what ANOVAs are and are not, (2) demonstrate what the various types of ANOVAs are, and (3) familiarize you with how R does ANOVA.

Here are some assumptions I make about you, the reader, in this post:

1. You're familiar with the ideas of multi-factor ANOVAs (what a main effect is, what interactions are…).
2. You know some `R` – how to fit a linear model, how to wrangle some data.
3. ~~You are IID and normally distributed.~~

What *is* ANOVA?

ANOVA tables are a way of summarizing a model – any model – by presenting the results grouped by the model's terms / effects. These tables contain a test statistic (often *F*), which represents the combined significance of all the parameters associated with a term. This test is sometimes called the omnibus test, and is often accompanied by a measure of effect size, informing us how much of the variation in the outcome can be attributed to this term. In other words, these answer the question: *how much of the variation in \(Y\) is related to the variation in \(X\)?*

This type of model summary is often paired with **factorial designs** – designs in which all the predictors (independent variables) are categorical, and all possible combinations between the levels of the different categories are represented in the data (often in a balanced way).

In fact, ANOVAs and factorial designs are paired so often that the two have (wrongfully) become synonymous and ANOVAs are commonly seen as "a type of model" that is applicable to factorial data. A consequence of this is that it is common knowledge[1] that ANOVAs are:

1. A type of model, that is a special case of linear regression (i.e. with a conditionally normal outcome variable), where
2. All predictors are categorical, and
3. The model is maximal – it has all possible main effects and interactions.

These 3 points are in fact ***false***.

Let's look at some examples…

ANOVAs in R

Toy Data

```
d <- tibble::tribble(
  ~id, ~group,  ~X, ~Z,       ~Rx, ~condition,     ~Y,
   1L,   "Gb", 102, 1L, "Placebo",       "Ca", 584.07,
   2L,   "Ga",  52, 1L, "Placebo",       "Cb", 790.29,
   3L,   "Gb", 134, 2L, "Dose100",       "Ca", 875.76,
   4L,   "Gb", 128, 3L, "Dose100",       "Cb", 848.37,
   5L,   "Ga",  78, 1L, "Dose250",       "Ca", 270.42,
   6L,   "Gb", 150, 2L, "Dose250",       "Cb", 999.87,
   7L, "Ga", 73, 1L, "Placebo",       "Ca",  364.1,
   8L,   "Ga",  87, 7L, "Placebo",       "Cb", 420.84,
   9L,   "Gb", 115, 6L, "Dose100",       "Ca", 335.78,
  10L,   "Gb", 113, 4L, "Dose100",       "Cb",    627,
  11L,   "Gc", 148, 3L, "Dose250",       "Ca", 607.79,
  12L,   "Gc",  82, 3L, "Dose250",       "Cb", 329.32,
  13L,   "Ga", 139, 1L, "Placebo",       "Ca", 335.56,
  14L,   "Ga",  65, 2L, "Placebo",       "Cb", 669.04,
  15L,   "Gb", 139, 1L, "Dose100",       "Ca", 405.04,
  16L,   "Gc",  96, 1L, "Dose100",       "Cb", 367.15,
  17L,   "Gb",  50, 5L, "Dose250",       "Ca",  27.37,
  18L,   "Gc",  90, 2L, "Dose250",       "Cb", 468.69,
  19L, "Ga", 90, 2L, "Placebo",       "Ca", 584.67,
  20L,   "Ga", 116, 2L, "Placebo",       "Cb", 277.71,
  21L,   "Gb",  78, 2L, "Dose100",       "Ca", 266.01,
  22L,   "Gb",  60, 1L, "Dose100",       "Cb",   0.04,
  23L,   "Gc", 112, 4L, "Dose250",       "Ca", 593.25,
  24L, "Ga", 63, 4L, "Dose250",       "Cb", 512.26,
  25L, "Ga", 89, 1L, "Placebo",       "Ca", 635.57,
  26L,   "Ga",  97, 2L, "Placebo",       "Cb", 468.69,
  27L,   "Gc",  76, 3L, "Dose100",       "Ca", 514.66,
  28L,   "Gb",  83, 1L, "Dose100",       "Cb", 264.87,
  29L,   "Gc",  84, 4L, "Dose250",       "Ca", 220.34,
  30L,   "Gb",  88, 1L, "Dose250",       "Cb", 216.54
  )
d$id <- factor(d$id)
d$group <- factor(d$group)
d$Rx <- factor(d$Rx, levels = c("Placebo", "Dose100", "Dose250"))
d$condition <- factor(d$condition)

dplyr::glimpse(d)
#> Rows: 30
#> Columns: 7
#> $ id        1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 1~
#> $ group     Gb, Ga, Gb, Gb, Ga, Gb, Ga, Ga, Gb, Gb, Gc, Gc, Ga, Ga,
Gb, ~
#> $ X         102, 52, 134, 128, 78, 150, 73, 87, 115, 113, 148, 82,
139, ~
#> $ Z         1, 1, 2, 3, 1, 2, 1, 7, 6, 4, 3, 3, 1, 2, 1, 1, 5, 2,
2, 2, ~
#> $ Rx        Placebo, Placebo, Dose100, Dose100, Dose250, Dose250,
Placeb~
#> $ condition  Ca, Cb, Ca, Cb, Ca, Cb, Ca, Cb, Ca, Cb, Ca, Cb, Ca, Cb,
Ca, ~
#> $ Y          584.07, 790.29, 875.76, 848.37, 270.42, 999.87, 364.10,
420.~
m <- lm(Y ~ group + X, data = d)
```

This is a multiple regression model with a covariate `X` and a 3-level factor `group`. We can summarize the results in a coefficient table (aka a “regression” table):

```
summary(m)
#>
#> Call:
#> lm(formula = Y ~ group + X, data = d)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -372.51 -166.47  -53.67  134.57  451.16
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  118.608    145.557   0.815  0.42256
#> groupGb     -102.591     94.853  -1.082  0.28937
#> groupGc      -92.384    107.302  -0.861  0.39712
#> X              4.241      1.504   2.820  0.00908 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 218.8 on 26 degrees of freedom
#> Multiple R-squared:  0.2383, Adjusted R-squared:  0.1504
#> F-statistic: 2.711 on 3 and 26 DF,  p-value: 0.06556
```

Or we can look at an ANOVA table:

```
anova(m)
#> Analysis of Variance Table
#>
#> Response: Y
#>           Df  Sum Sq Mean Sq F value   Pr(>F)
#> group      2    8783    4391  0.0918 0.912617
#> X          1  380471  380471  7.9503 0.009077 **
#> Residuals 26 1244265   47856
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

While the `group` term had 2 parameters in the coefficient table, it now has a single test with a `Df` of 2 - the test represents the *total* significance of the two parameters combined! That is, these *F*-tests are **omnibus tests** for all the parameters of a given term.

Note that the model being summarized here is neither completely factorial (`X` is a continuous covariate), nor maximal (the `group:X` interaction is missing)!

By default `R` calculates *type 1* sums of squares (SS) - these are also called *sequential SS*, because each term’s SS is calculated for its addition on top of the PREVIOUS terms - sequentially! In our example, the effect of `X` represents only what `X` explains on top of what group was already able to explain / predict - it is the *unique* contribution of X, controlling for group; the effect of group however does *not* represent its unique contribution, but instead its *total* contribution.

This means that because the following models have the same terms *in a different order* they will produce different *type 1* ANOVA tables!

```
anova(lm(Y ~ group + X, data = d))
#> Analysis of Variance Table
#>
#> Response: Y
#>           Df  Sum Sq Mean Sq F value   Pr(>F)
#> group      2    8783    4391  0.0918 0.912617
#> X          1  380471  380471  7.9503 0.009077 **
#> Residuals 26 1244265   47856
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
anova(lm(Y ~ X + group, data = d))
#> Analysis of Variance Table
#>
#> Response: Y
#>           Df  Sum Sq Mean Sq F value  Pr(>F)
#> X          1  325745  325745  6.8067 0.01486 *
#> group      2   63509   31754  0.6635 0.52353
#> Residuals 26 1244265   47856
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can recreate the ANOVA table above by building a sequence of models, and comparing them:

```
m0 <- lm(Y ~ 1, data = d)
m1 <- lm(Y ~ group, data = d)

anova(m0, m1, m)
#> Analysis of Variance Table
#>
#> Model 1: Y ~ 1
#> Model 2: Y ~ group
#> Model 3: Y ~ group + X
#>   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
#> 1     29 1633519
#> 2     27 1624737  2      8783 0.0918 0.912617
#> 3     26 1244265  1    380471 7.9503 0.009077 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
anova(m) # same SS values
#> Analysis of Variance Table
#>
#> Response: Y
#>           Df  Sum Sq Mean Sq F value   Pr(>F)
#> group      2    8783    4391  0.0918 0.912617
#> X          1  380471  380471  7.9503 0.009077 **
#> Residuals 26 1244265   47856
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Simultaneous Sum of Squares

There are also *type 2 SS* - also called *simultaneous SS*, because the SS of each term is calculated for its addition over all other terms - its unique contribution while controlling for the other terms. Type 2 SS can be obtained with the `Anova()` function from the {`car`} package:

```
car::Anova(m, type = 2)
#> Anova Table (Type II tests)
#> 
#> Response: Y
#>            Sum Sq Df F value   Pr(>F)
#> group       63509  2  0.6635 0.523533
#> X          380471  1  7.9503 0.009077 **
#> Residuals 1244265 26
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can recreate the above ANOVA table by building two sequences of models, and comparing them:

```
m_sans_X <- lm(Y ~ group, data = d)
m_sans_group <- lm(Y ~ X, data = d)

anova(m_sans_group, m) # Same SS as the type 2 test for group
#> Analysis of Variance Table
#> 
#> Model 1: Y ~ X
#> Model 2: Y ~ group + X
#>   Res.Df     RSS Df Sum of Sq      F Pr(>F)
#> 1     28 1307774
#> 2     26 1244265  2     63509 0.6635 0.5235
anova(m_sans_X,     m) # Same SS as the type 2 test for X
#> Analysis of Variance Table
#> 
#> Model 1: Y ~ group
#> Model 2: Y ~ group + X
#>   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
#> 1     27 1624737
#> 2     26 1244265  1    380471 7.9503 0.009077 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Because the order of terms is usually of little importance, type 1 tests are rarely used in the analysis of factorial designs…

Unfortunately, they are R’s default…

Adding Interactions

Things get a bit more complicated when interactions are involved, as type 2 SS treat interactions differently than main effects:

```
m_int <- lm(Y ~ group * X, data = d)
```

The SS of each main effect is calculated over all other main effects (simultaneously, as we've already seen) but *without* accounting for the interaction, while the SS of the interaction term is calculated on top of the underlying main effects (sequentially). So we get the unique contribution of each main effect when controlling only for the other main effect, and the unique contribution of the interaction controlling for the already-included contribution of the main effects.

```
car::Anova(m_int, type = 2)
#> Anova Table (Type II tests)
#>
#> Response: Y
#>           Sum Sq Df F value    Pr(>F)
#> group      63509  2  1.3555 0.2768607
#> X         380471  1 16.2410 0.0004884 ***
#> group:X   682026  2 14.5566 7.246e-05 ***
#> Residuals 562240 24
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Is the same as[2]:

```
anova(m_sans_group, m)     # Same SS as the type 2 test for group
#> Analysis of Variance Table
#>
#> Model 1: Y ~ X
#> Model 2: Y ~ group + X
#>   Res.Df     RSS Df Sum of Sq      F Pr(>F)
#> 1     28 1307774
#> 2     26 1244265  2     63509 0.6635 0.5235
anova(m_sans_X,     m)     # Same SS as the type 2 test for X
#> Analysis of Variance Table
#>
```

```
#> Model 1: Y ~ group
#> Model 2: Y ~ group + X
#>   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
#> 1     27 1624737
#> 2     26 1244265  1    380471 7.9503 0.009077 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
anova(m,            m_int) # Same SS as the type 2 test for group:X
#> Analysis of Variance Table
#>
#> Model 1: Y ~ group + X
#> Model 2: Y ~ group * X
#>   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
#> 1     26 1244265
#> 2     24  562240  2    682026 14.557 7.246e-05 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In designs of higher order, each “order” is tested in a similar *simultaneous-sequential* manner. E.g., in a 3-way design, all main effects (1st order) are tested *simultaneously* (accounting for one another), **then** all 2-way interactions (2nd order) are tested *simultaneously* (accounting for the main effects *and* one another), and **then** the 3-way interaction is tested (accounting for all main effects and 2-way interactions).

Type 2 vs Type 3

There is another type of simultaneous SS - the type 3 test, which does not treat interactions differently than main effects: the SS for each main effect / interaction is calculated as its contribution on top of all other main effects AND interactions. So the effect of `X` is its unique contribution while controlling both for `group` *and* for `group:X`!

However, remember that we previously saw that these methods actually produce omnibus tests for the combined effect of the *parameters of each term*. But in the `m_int` model the parameters labeled `X`, `groupGb`, and `groupGc` are no longer parameters of the main effects - instead they are parameters of simple (i.e., conditional) effects!

```
summary(m_int)
#>
#> Call:
#> lm(formula = Y ~ group * X, data = d)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -370.18  -67.15   33.68  111.35  172.14
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   833.130    173.749   4.795 6.99e-05 ***
#> groupGb     -1308.898    233.218  -5.612 8.90e-06 ***
#> groupGc      -752.718    307.747  -2.446   0.0222 *
#> X              -4.041      1.942  -2.081   0.0482 *
#> groupGb:X      13.041      2.419   5.390 1.55e-05 ***
#> groupGc:X       7.731      3.178   2.432   0.0228 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 153.1 on 24 degrees of freedom
#> Multiple R-squared:  0.6558, Adjusted R-squared:  0.5841
#> F-statistic: 9.146 on 5 and 24 DF,  p-value: 5.652e-05
```

- `X` is the slope of `X` ***when `group=Ga`*** (`Ga` is the reference level)
- `groupGb` is the difference between `group=Ga` and `group=Gb` ***when `X=0`***
- `groupGc` is the difference between `group=Ga` and `group=Gc` ***when `X=0`***

(Pay attention to the “when” - this is what makes them conditional.)

We can see that changing the reference group changes the test for `X`:

```
d$group <- relevel(d$group, ref = "Gb")
m_int2 <- lm(Y ~ group * X, data = d)

car::Anova(m_int, type = 3)
#> Anova Table (Type III tests)
#>
#> Response: Y
#>             Sum Sq Df F value    Pr(>F)
#> (Intercept) 538630  1 22.9922 6.994e-05 ***
#> group       738108  2 15.7536 4.269e-05 ***
#> X           101495  1  4.3325   0.04823 *
#> group:X     682026  2 14.5566 7.246e-05 ***
#> Residuals   562240 24
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
car::Anova(m_int2, type = 3)
#> Anova Table (Type III tests)
#>
#> Response: Y
#>             Sum Sq Df F value    Pr(>F)
#> (Intercept) 219106  1  9.3528  0.005402 **
#> group       738108  2 15.7536 4.269e-05 ***
#> X           910646  1 38.8722 1.918e-06 ***
#> group:X     682026  2 14.5566 7.246e-05 ***
#> Residuals   562240 24
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How can we resolve this?

Centering

By centering our predictors! Centering is transforming our data in such a way that 0 represents the overall mean. When this is done, conditioning on 0 is the same as conditioning on the overall mean = looking at the main effect!

For covariables this is easy enough:

```
d$X_c <- d$X - mean(d$X) # or scale(d$X, center = TRUE, scale = FALSE)
```

But how do we center a factor??

The answer is - use some type of orthogonal coding, for example `contr.sum()` (effects coding). This makes the coefficients harder to interpret [3], but we're not looking at those anyway!

```
contrasts(d$group) <- contr.sum
```

Now when looking at type 3 tests, the main effects terms actually are main effects!

```
m_int3 <- lm(Y ~ group*X_c, data = d)
car::Anova(m_int3, type = 3)
#> Anova Table (Type III tests)
#>
#> Response: Y
#>              Sum Sq Df  F value    Pr(>F)
#> (Intercept) 4743668  1 202.4902 3.401e-13 ***
#> group         19640  2   0.4192   0.66231
#> X_c          143772  1   6.1371   0.02067 *
#> group:X_c    682026  2  14.5566 7.246e-05 ***
#> Residuals    562240 24
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Remember: ***ABC - Always Be Centering*** (your predictors) - type 3 ANOVA tables make little sense without centering!

Recreate the Type 3 ANOVA

Unfortunately, we can't just build a model without an interaction term and use it to recreate the type 3 ANOVA. Instead, we need to actually build the model matrix (i.e., the design matrix), and drop the correct columns:

```
mm <- model.matrix(m_int2)
head(mm)
#>   (Intercept) groupGa groupGc   X groupGa:X groupGc:X
#> 1           1       0       0 102         0         0
#> 2           1       1       0  52        52         0
#> 3           1       0       0 134         0         0
#> 4           1       0       0 128         0         0
#> 5           1       1       0  78        78         0
#> 6           1       0       0 150         0         0
```

A type 3 test for a term, is equal to the comparison between a model without the parameters associated with that term and the **full model**:

```
m_sans_group <- lm(Y ~ mm[,-(2:3)], data = d)
m_sans_X <-     lm(Y ~ mm[,-4],     data = d)
m_sans_int <-   lm(Y ~ mm[,-(5:6)], data = d)

anova(m_sans_group, m_int2) # Same SS as type 3 for group
#> Analysis of Variance Table
#>
#> Model 1: Y ~ mm[, -(2:3)]
#> Model 2: Y ~ group * X
#>   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
#> 1     26 1300348
#> 2     24  562240  2    738108 15.754 4.269e-05 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
anova(m_sans_X,     m_int2) # Same SS as type 3 for X
#> Analysis of Variance Table
#>
#> Model 1: Y ~ mm[, -4]
#> Model 2: Y ~ group * X
#>   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
#> 1     25 1472886
#> 2     24  562240  1    910646 38.872 1.918e-06 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
anova(m_sans_int,   m_int2) # Same SS as type 3 for group:X
#> Analysis of Variance Table
#>
#> Model 1: Y ~ mm[, -(5:6)]
#> Model 2: Y ~ group * X
#>   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
#> 1     26 1244265
#> 2     24  562240  2    682026 14.557 7.246e-05 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Compare to:

```
car::Anova(m_int2, type = 3)
#> Anova Table (Type III tests)
#>
#> Response: Y
#>             Sum Sq Df F value    Pr(>F)
#> (Intercept) 219106  1  9.3528  0.005402 **
#> group       738108  2 15.7536 4.269e-05 ***
#> X           910646  1 38.8722 1.918e-06 ***
#> group:X     682026  2 14.5566 7.246e-05 ***
#> Residuals   562240 24
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Okay, that’s some ugly stuff. Let us never look at that again.

Balanced vs. Unbalanced Data

The distinction between types 1, 2 and 3 SS is only relevant when there is some dependency between predictors (aka some collinearity). In our example, we can see that `group` and `X` are somewhat co-linear (VIF / tolerance are not strictly 1):

```
performance::check_collinearity(m)
#> # Check for Multicollinearity
#>
#> Low Correlation
```

```
#>
#>   Term  VIF Increased SE Tolerance
#>  group 1.08         1.04      0.92
#>      X 1.08         1.04      0.92
```

In a factorial design, we might call this dependence / collinearity among our predictors an “unbalanced design” (the number of observations differs between cells), and when the predictors are completely independent we would call this a “balanced design” (equal number of observations in all cells).

Let’s look at two examples:

Balanced data

We can see that `Rx` and `condition` are balanced:

```
table(d$Rx, d$condition)
#>
#>           Ca Cb
#>   Placebo  5  5
#>   Dose100  5  5
#>   Dose250  5  5
chisq.test(d$Rx, d$condition)$statistic # Chisq is exactly 0
#> X-squared
#>         0
```

And so type 2 and type 3 ANOVAs are identical:

```
contrasts(d$Rx) <- contr.sum
contrasts(d$condition) <- contr.sum

m_balanced <- lm(Y ~ condition * Rx, data = d)

anova(m_balanced)
#> Analysis of Variance Table
#>
#> Response: Y
#>             Df  Sum Sq Mean Sq F value Pr(>F)
#> condition     1   13666   13666  0.2162 0.6461
#> Rx            2   41379   20690  0.3273 0.7240
#> condition:Rx  2   61444   30722  0.4860 0.6210
#> Residuals    24 1517030   63210
car::Anova(m_balanced, type = 2)
#> Anova Table (Type II tests)
#>
#> Response: Y
#>               Sum Sq Df F value Pr(>F)
#> condition      13666  1  0.2162 0.6461
#> Rx             41379  2  0.3273 0.7240
#> condition:Rx   61444  2  0.4860 0.6210
#> Residuals    1517030 24
car::Anova(m_balanced, type = 3)
#> Anova Table (Type III tests)
```

```
#>
#> Response: Y
#>               Sum Sq Df  F value    Pr(>F)    
#> (Intercept)  6422803  1 101.6112 4.204e-10 ***
#> condition      13666  1   0.2162    0.6461    
#> Rx             41379  2   0.3273    0.7240    
#> condition:Rx   61444  2   0.4860    0.6210    
#> Residuals    1517030 24                       
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Unbalanced data

However, `condition` and `group` are NOT balanced:

```
table(d$group, d$condition)
#>     
#>      Ca Cb
#>   Gb  6  6
#>   Ga  5  6
#>   Gc  4  3
chisq.test(d$group, d$condition)$statistic # Chisq is NOT 0
#> Warning in chisq.test(d$group, d$condition): Chi-squared
approximation may be
#> incorrect
#> X-squared 
#> 0.2337662
```

And so type 2 and type 3 ANOVAs are NOT identical:

```
contrasts(d$group) <- contr.sum
contrasts(d$condition) <- contr.sum

m_unbalanced <- lm(Y ~ condition * group, data = d)

anova(m_unbalanced)
#> Analysis of Variance Table
#> 
#> Response: Y
#>                 Df  Sum Sq Mean Sq F value Pr(>F)
#> condition        1   13666   13666  0.2087 0.6519
#> group            2    7165    3582  0.0547 0.9469
#> condition:group  2   41204   20602  0.3146 0.7330
#> Residuals       24 1571485   65479               
car::Anova(m_unbalanced, type = 2)
#> Anova Table (Type II tests)
#> 
#> Response: Y
#>                  Sum Sq Df F value Pr(>F)
#> condition         12048  1  0.1840 0.6718
#> group              7165  2  0.0547 0.9469
#> condition:group   41204  2  0.3146 0.7330
#> Residuals       1571485 24               
```

```
car::Anova(m_unbalanced, type = 3)
#> Anova Table (Type III tests)
#>
#> Response: Y
#>                   Sum Sq Df F value    Pr(>F)
#> (Intercept)      5858845  1 89.4773 1.439e-09 ***
#> condition           3452  1  0.0527    0.8203
#> group               8913  2  0.0681    0.9344
#> condition:group    41204  2  0.3146    0.7330
#> Residuals        1571485 24
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As the deviation from perfect balance (i.e. independence) is larger, so will the differences between the types increase.

Why Does This Happen?

For type 1 SS, each effect is estimated sequentially - so in the model `Y ~ A * B`, the effect of B is its unique contribution after controlling for and partialling out the effect of A. But if A and B are independent, then there is nothing from A to partial out of B, and vice versa - so the order does not matter!

The distinction between types 2 and 3 comes from how they estimate main effects in the presence of interactions: Type 2 SS looks at the SS between the marginal means of `group`, across the levels of `condition`. So the mean of the first group is:

$$\bar{Y}_{1.} = \frac{\sum{Y_{1.}}}{N_{1.}} = \frac{\sum{Y_{11}}+\sum{Y_{12}}}{N_{11}+N_{12}}$$

Type 3 SS however looks at the SS between the means of `group`, weighted by `condition`. So the mean of group `a` is

$$\bar{Y}_{1.} = \frac{\frac{\sum{Y_{11}}}{N_{11}} + \frac{\sum{Y_{12}}}{N_{12}}}{2}$$

This makes type 3 SS invariant to the cell frequencies! (However, both equations would give the same result when the data are balanced.)

A lot has been said about type 2 vs type 3. I will not go into the weeds here, but it is important to note that

1. Most statistical softwares (SAS, Stata, SPSS, …) default to type 3 SS with orthogonal factor coding (but covariables are *not* mean-centered in most cases by default) (see Langsrud, 2003). This makes `R` inconsistent as we've seen it defaults to type 1 ANOVA and treatment coding.
2. Often in factorial designs, any imbalance in the design is incidental, so it is often beneficial to have a method that is invariant to such imbalances. (Though this might not be true if the data is observational.)
3. Coefficient tables give results that are analogous to type 3 SS when all terms are covariables:

```
m_covs <- lm(Y ~ X * Z, data = d)
```

```
car::Anova(m_covs, type = 2)
#> Anova Table (Type II tests)
#> 
#> Response: Y
#>            Sum Sq Df F value  Pr(>F)  
#> X          324676  1  6.7657 0.01513 *
#> Z            2430  1  0.0506 0.82373  
#> X:Z         57640  1  1.2011 0.28315  
#> Residuals 1247704 26                  
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
car::Anova(m_covs, type = 3)
#> Anova Table (Type III tests)
#> 
#> Response: Y
#>               Sum Sq Df F value Pr(>F)
#> (Intercept)    83130  1  1.7323 0.1996
#> X               5866  1  0.1222 0.7294
#> Z              59917  1  1.2486 0.2740
#> X:Z            57640  1  1.2011 0.2831
#> Residuals    1247704 26               
summary(m_covs) # same p-values as type 3
#> 
#> Call:
#> lm(formula = Y ~ X * Z, data = d)
#> 
#> Residuals:
#>     Min      1Q  Median      3Q     Max 
#> -384.97 -153.72  -23.98  141.28  422.79 
#> 
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)  366.352    278.348   1.316    0.200
#> X              1.014      2.900   0.350    0.729
#> Z           -112.681    100.843  -1.117    0.274
#> X:Z            1.175      1.072   1.096    0.283
#> 
#> Residual standard error: 219.1 on 26 degrees of freedom
#> Multiple R-squared:  0.2362, Adjusted R-squared:  0.1481 
#> F-statistic:  2.68 on 3 and 26 DF,  p-value: 0.06772
```

Note once again that the model being summarized here is not factorial at all - both `Z` and `X` are continuous covariables!

ANOVA Made Easy

Making sure that our factors are orthogonally coded is a pain in the @$$.

Thankfully, we have the `afex` package which turns this:

```
d$group <- factor(d$group)
d$condition <- factor(d$condition)
```

```
contrasts(d$group) <- contr.sum
contrasts(d$condition) <- contr.sum
m_lm <- lm(Y ~ group * condition, d)
car::Anova(m_lm, type = 3)
#> Anova Table (Type III tests)
#> 
#> Response: Y
#>                  Sum Sq Df F value    Pr(>F)    
#> (Intercept)     5858845  1 89.4773 1.439e-09 ***
#> group              8913  2  0.0681    0.9344    
#> condition          3452  1  0.0527    0.8203    
#> group:condition   41204  2  0.3146    0.7330    
#> Residuals       1571485 24                      
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Into this:

```
afex::aov_car(Y ~ group * condition + Error(id), data = d)
#> Anova Table (Type 3 tests)
#> 
#> Response: Y
#>            Effect    df      MSE    F  ges p.value
#> 1           group 2, 24 65478.53 0.07 .006    .934
#> 2       condition 1, 24 65478.53 0.05 .002    .820
#> 3 group:condition 2, 24 65478.53 0.31 .026    .733
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

Much easier!

Other Types of Models

So far we’ve seen how ANOVAs are applied to linear OLS models. However the idea of omnibus tests per term can be extended to many other types of models. For models where SS cannot be calculated, analogous methods based on deviance or likelihood are used instead (read more in the `car::Anova()` docs). Here are some examples:

GLMs

Logistic

```
m_logistic <- glm(condition ~ group * X_c, data = d,
                  family = binomial())
car::Anova(m_logistic, type = 2)
#> Analysis of Deviance Table (Type II tests)
#> 
#> Response: condition
#>           LR Chisq Df Pr(>Chisq)
#> group      0.15679  2     0.9246
#> X_c        0.75134  1     0.3861
#> group:X_c  1.10225  2     0.5763
```

Poisson

```
m_poisson <- glm(Z ~ group * X_c, data = d,
                 family = poisson())
car::Anova(m_poisson, type = 3)
#> Analysis of Deviance Table (Type III tests)
#> 
#> Response: Z
#>           LR Chisq Df Pr(>Chisq)
#> group      0.88074  2     0.6438
#> X_c        0.04370  1     0.8344
#> group:X_c  0.11357  2     0.9448
```

Ordinal

```
m_ordinal <- ordinal::clm(group ~ X_c * condition, data = d)
anova(m_ordinal)
#> Type I Analysis of Deviance Table with Wald chi-square tests
#> 
#>               Df  Chisq Pr(>Chisq)
#> X_c            1 0.4469     0.5038
#> condition      1 0.1584     0.6907
#> X_c:condition  1 0.6129     0.4337
```

Note that all of the models summarized here with an ANOVA table do not have a continuous (conditionally normal) outcome, and not *all* of their predictors are categorical. We may have been inclined to summarize these models with a coefficient table, but it is equally valid to present an ANOVA table!

(G)LMMs

```
m_mixed <- lmerTest::lmer(Y ~ X_c * group + (group | Z), data = d)
anova(m_mixed, type = 2)
#> Type II Analysis of Variance Table with Satterthwaite's method
#>           Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)    
#> X_c       290878  290878     1 23.645 14.4210 0.0008951 ***
#> group       7198    3599     2  9.520  0.1784 0.8393243    
#> X_c:group 582071  291036     2 21.938 14.4288 0.0001001 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Also for GLMMs:

```
m_mixed2 <- lme4::glmer(condition ~ X_c * group + (1 | Z), data = d,
                        family = binomial())
car::Anova(m_mixed2, type = 3)
#> Analysis of Deviance Table (Type III Wald chisquare tests)
#> 
#> Response: condition
#>              Chisq Df Pr(>Chisq)
#> (Intercept) 0.0886  1     0.7660
#> X_c         1.1917  1     0.2750
#> group       0.0829  2     0.9594
#> X_c:group   0.9972  2     0.6074
```

Concluding Remarks

Although we might be more inclined to summarize our model with an ANOVA table when our model contains categorical predictors, especially when interactions are involved, we’ve seen that (contrary to what we may have learned) ANOVA tables do not require any special data (factorial, balanced, normal outcome), and can be used as an alternative to a coefficient table.