

1 Interaction model

The main question here ‘is on a global test of the null hypothesis that the mean response profiles are similar in the groups.’ To test this hypothesis, we need to consider an interaction model. If there is no effect of the treatment, the mean response profiles should be parallel. To see this, check the modeling construct. The response of i -th subject at j -th time in placebo group is

$$y_{i,j}^{(P)} = \alpha_P + \beta_P t_{i,j} + \epsilon_{i,j}$$

and in the treatment group is,

$$y_{i,j}^{(T)} = \alpha_T + \beta_T t_{i,j} + \epsilon_{i,j}$$

If there is no treatment effect, we should have $\beta_T = \beta_P$. However, it is difficult to form a statistical test for this hypothesis. Since there are many tests available for nested-type models, we will combine the above two models into one equation as

$$y_{i,j,g} = \alpha + \beta t_{i,j} + \theta g + \gamma t_{i,j} g + \epsilon_{i,j,g}$$

where $g = 0$ for the placebo group and $g = 1$ for the treatment group. Now the problem reduces to testing $\gamma = 0$. Hence, we are comparing two models: $y_{i,j,g} = \alpha + \beta t_{i,j} + \theta g + \gamma t_{i,j} g + \epsilon_{i,j}$ with $y_{i,j,g} = \alpha + \beta t_{i,j} + \theta g + \epsilon_{i,j}$. As you can see, the second model is ‘nested’ into the first model. It is easier to compare nested models as there are many tests available for this case. We can fit this model without accounting for the correlation between $(y_{i,j,g}, y_{i,k,g})$ pairs as,

Week and treatment interaction (like in ANCOVA as time is a continuous variable and V2 (the treatment group) has two levels.)

```
ls <- lm(measurement~Week*as.factor(V2), data=data_new)
model.matrix(ls)
coef(summary(ls))
gtsummary::tbl_regression(ls)
anova(ls)
```

Or like in two-way ANOVA. Check the `model.matrix` in this case. The number of predictors has expanded, as there are several possible interactions among different levels of the treatment and Week.

```
ls <- lm(measurement~as.factor(Week)*as.factor(V2), data=data_new)
model.matrix(ls)
coef(summary(ls))
gtsummary::tbl_regression(ls)
anova(ls)
```

```
ls <- lm(measurement~as.factor(Week)*as.factor(V2), data=data_new)
car::Anova(ls, type=2)
car::Anova(ls, type=3)
```

2 Infer interaction effect

It's from https://md.psych.bio.uni-goettingen.de/mv/unit/lm_cat/lm_cat_unbal_ss_explained.html

ANOVA (and R) - The ANOVA Controversy

ANOVA is a statistical process for analysing the amount of variance that is contributed to a sample by different factors. It was initially derived by R. A. Fisher in 1925, for the case of balanced data (equal numbers of observations for each level of a factor).

When data is unbalanced, there are different ways to calculate the sums of squares for ANOVA. There are at least 3 approaches, commonly called Type I, II and III sums of squares (this notation seems to have been introduced into the statistics world from the SAS package but is now widespread). Which type to use has led to an ongoing controversy in the field of statistics (for an overview, see Heer [2]). However, it essentially comes down to testing different hypotheses about the data. Type I, II and III Sums of Squares

Consider a model that includes two factors A and B; there are therefore two main effects, and an interaction, AB. The full model is represented by $SS(A, B, AB)$.

Other models are represented similarly: $SS(A, B)$ indicates the model with no interaction, $SS(B, AB)$ indicates the model that does not account for effects from factor A, and so on.

The influence of particular factors (including interactions) can be tested by examining the differences between models. For example, to determine the presence of an interaction effect, an F-test of the models $SS(A, B, AB)$ and the no-interaction model $SS(A, B)$ would be carried out.

It is convenient to define incremental sums of squares to represent these differences. Let

$$\begin{aligned}SS(AB \mid A, B) &= SS(A, B, AB) - SS(A, B) \\SS(A \mid B, AB) &= SS(A, B, AB) - SS(B, AB) \\SS(B \mid A, AB) &= SS(A, B, AB) - SS(A, AB) \\SS(A \mid B) &= SS(A, B) - SS(B) \\SS(B \mid A) &= SS(A, B) - SS(A)\end{aligned}$$

The notation shows the incremental differences in sums of squares, for example $SS(AB \mid A, B)$ represents “the sum of squares for interaction after the main effects”, and $SS(A \mid B)$ is “the sum of squares for the A main effect after the B main effect and ignoring interactions” [1].

The different types of sums of squares then arise depending on the stage of model reduction at which they are carried out. In particular:

Type I, also called “sequential” sum of squares:

$SS(A)$ for factor A.

$SS(B \mid A)$ for factor B.

$SS(AB \mid B, A)$ for interaction AB.

This tests the main effect of factor A, followed by the main effect of factor B after the main effect of A, followed by the interaction effect AB after the main effects.

Because of the sequential nature and the fact that the two main factors are tested in a particular order, this type of sums of squares will give different results for unbalanced data depending on which main effect is considered first.

For unbalanced data, this approach tests for a difference in the weighted marginal means. In practical terms, this means that the results are dependent on the realized sample sizes, namely the proportions in the particular data set. In other words, it is testing the first factor without controlling for the other factor (for further discussion and a worked example, see Zahn [4]).

Note that this is often not the hypothesis that is of interest when dealing with unbalanced data.

Type II:

$SS(A \mid B)$ for factor A.

$SS(B \mid A)$ for factor B.

This type tests for each main effect after the other main effect.

Note that no significant interaction is assumed (in other words, you should test for interaction first ($SS(AB \mid A, B)$) and only if AB is not significant, continue with the analysis for main effects).

If there is indeed no interaction, then type II is statistically more powerful than type III.

Computationally, this is equivalent to running a type I analysis with different orders of the factors, and taking the appropriate output (the second, where one main effect is run after the other, in the example above).

Type III:

$SS(A \mid B, AB)$ for factor A.

$SS(B \mid A, AB)$ for factor B.

This type tests for the presence of a main effect after the other main effect and interaction. This approach is therefore valid in the presence of significant interactions.

However, it is often not interesting to interpret a main effect if interactions are present (generally speaking, if a significant interaction is present, the main effects should not be further analysed).

If the interactions are not significant, type II gives a more powerful test.

NOTE: when data is balanced, the factors are orthogonal, and types I, II and III all give the same results. Summary: Usually the hypothesis of interest is about the significance of one factor while controlling for the level of the other factors. This equates to using type II or III SS. In general, if there is no significant interaction effect, then type II is more powerful, and follows the principle of marginality. If interaction is present, then type II is inappropriate while type III can still be used, but results need to be interpreted with caution (in the presence of interactions, main effects are rarely interpretable).

The anova and aov Functions in R

The anova and aov functions in R implement a sequential sum of squares (type I). As indicated above, for unbalanced data, this rarely tests a hypothesis of interest, since essentially the effect of one factor is calculated based on the varying levels of the other factor. In a practical sense, this

means that the results are interpretable only in relation to the particular levels of observations that occur in the (unbalanced) data set. Fortunately, based on the above discussion, it should be clear that it is relatively straightforward to obtain type II SS in R. Type II SS in R

Since type II SS tests each main effect after the other main effects, and assumes no interactions, the correct SS can be obtained using `anova()` and varying the order of the factors.

For example, consider a data frame (`search`) for which the response variable is the time that it takes users to find a relevant answer with an information retrieval system (time). The user is assigned to one of two experimental search systems on which they run the test (`sys`). They are also assigned a number of different search queries (topic).

```
data <- read.table("H:/Documents/Course to teach/LDA/LDA/Slides/Treatment of Lead
Exposed Children Trial.txt", quote="\"", comment.char="")
```

```
library(tidyr)
data_long <- gather(data, Week, measurement, V3:V6, factor_key=TRUE)
data_long <- data_long[order(data_long$V1), ]
```

```
data_new = data_long
data_new$Week <- as.character(data_new$Week)
data_new$Week[grepl("V3", data_long$Week)] = "0"
data_new$Week[grepl("V4", data_long$Week)] = "1"
data_new$Week[grepl("V5", data_long$Week)] = "4"
data_new$Week[grepl("V6", data_long$Week)] = "6"
data_new$Week <- as.numeric(data_new$Week)
```

```
ls <- lm(measurement~as.factor(Week)*as.factor(V2), data=data_new)
head(model.matrix(ls))
```

To obtain type I SS:

```
anova(lm(measurement~as.factor(Week)*as.factor(V2), data=data_new))
```

If the data is unbalanced, you will obtain slightly different results if you instead use:

```
anova(lm(measurement~as.factor(Week)*as.factor(V2), data=data_new))
```

The type II SS is obtained by using the second line of output from each of the above commands (since in type I SS, the second component will be the second factor, after the first factor). That is, you obtain the type II SS results for topic from the first command, and the results for `sys` from the second.

Type III SS in R

This is slightly more involved than the type II results. First, it is necessary to set the contrasts option in R. Because the multi-way ANOVA model is over-parameterised, it is necessary to choose a contrasts setting that sums to zero, otherwise the ANOVA analysis will give incorrect results with respect to the expected hypothesis. (The default contrasts type does not satisfy this requirement.)

```
options(contrasts = c(\contr.sum", "contr.poly"))
```

Next, store the model:

```
model <- lm(measurement~as.factor(Week)*as.factor(V2), data=data_new)
```

Finally, call the drop1 function on each model component:

```
drop1(model, .~., test="F")
```

The results give the type III SS, including the p-values from an F-test.

Type II and III SS Using the car Package

A somewhat easier way to obtain type II and III SS is through the car package. This defines a new function, Anova(), which can calculate type II and III SS directly.

Type II, using the same data set defined above:

```
car::Anova(lm(measurement~as.factor(Week)*as.factor(V2), data=data_new, type=2))
```

Type III:

```
library(car)
```

```
Anova(lm(measurement~as.factor(Week)*as.factor(V2), data=data_new, contrasts=list(topic=contr.sum, sys=contr.sum)), type=3)
```

NOTE: Again, due to the way in which the SS are calculated when incorporating the interaction effect, for type III you must specify the contrasts option to obtain sensible results (an explanation is given here). "here" refers to the references that follow in the original article. [cf](<https://mcfromnz.wordpress.com/2011/03/02/anova-type-iiiiii-ss-explained/>)

Hintergrund: Typen von Quadratsummen (http://www.unh.edu/halelab/BIOL933/lectures/lect_16_reading.pdf)

If there is a Type I SS and a Type II SS, it begs the question: Might there be more? You're in luck! In fact, there are four types of sums of squares, each with their associated statistics. These four types, of course, are called Types I, II, III, and IV (Goodnight 1978). Though we are going to use only Type I and Type II SS during this course, here's a brief description of all four.

Type I (sequential or incremental SS)

Type I sums of squares are determined by considering each source (factor) sequentially, in the order they are listed in the model. The Type I SS may not be particularly useful for analyses of unbalanced, multi-way structures but may be useful for balanced data and nested models. Type I SS are also useful for parsimonious polynomial models (i.e. regressions), allowing the simpler components (e.g. linear) to explain as much variation as possible before resorting to models of higher complexity (e.g. quadratic, cubic, etc.). Also, comparing Type I and other types of sums of squares provides some information regarding the magnitude of imbalance in the data. Types II and III SS are also known as partial sums of squares, in which each effect is adjusted for other effects.

Type III

Type III is also a partial SS approach, but it's a little easier to explain than Type II; so we'll start here. In this model, every effect is adjusted for all other effects. The Type III SS

will produce the same SS as a Type I SS for a data set in which the missing data are replaced by the leastsquares estimates of the values. The Type III SS correspond to Yates' weighted squares of means analysis. One use of this SS is in situations that require a comparison of main effects even in the presence of interactions (something the Type II SS does not do and something, incidentally, that many statisticians say should not be done anyway!). In particular, the main effects A and B are adjusted for the interaction A*B, as long as all these terms are in the model. If the model contains only main effects, then you will find that the Type II and Type III analyses are the same.

Type II

Type II partial SS are a little more difficult to understand. Generally, the Type II SS for an effect U, which may be a main effect or interaction, is adjusted for an effect V if and only if V does not contain U. Specifically, for a two-factor structure with interaction, the main effects A and B are not adjusted for the A*B interaction because the interaction contains both A and B. Factor A is adjusted for B because the symbol B does not contain A. Similarly, B is adjusted for A. Finally, the A*B interaction is adjusted for each of the two main effects because neither main effect contains both A and B. Put another way, the Type II SS are adjusted for all factors that do not contain the complete set of letters in the effect. In some ways, you could think of it as a sequential, partial SS; in that it allows lower-order terms explain as much variation as possible, adjusting for one another, before letting higher-order terms take a crack at it.

Type IV

The Type IV functions were designed primarily for situations where there are empty cells, also known as "radical" data loss. The principles underlying the Type IV sums of squares are quite involved and can be discussed only in a framework using the general construction of estimable functions. It should be noted that the Type IV functions are not necessarily unique when there are empty cells but are identical to those provided by Type III when there are no empty cells.

3 Restricted maximum likelihood (REML)

REML is useful for various purposes. The main recipe for REML is to adjust the likelihood in a way that new or 'restricted' likelihood becomes a function of only a smaller set of parameters and becomes independent of the rest. It somewhat mimics the unbiased estimation of the error. In matrix-vector representation, we get $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ and define $\mathbf{H} = \mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$. Since $\mathbf{H}\mathbf{X} = \mathbf{0}$, we have $\mathbf{H}\mathbf{y} = \mathbf{H}\boldsymbol{\epsilon}$ and thus $\mathbf{H}\mathbf{y} \sim \text{MVN}(\mathbf{0}, \mathbf{H}\boldsymbol{\Sigma}\mathbf{H}^T)$ if $\boldsymbol{\epsilon} \sim \text{MVN}(\mathbf{0}, \boldsymbol{\Sigma})$. Thus, after the linear transformation $\mathbf{H}\mathbf{y}$ becomes independent of \mathbf{X} .

Why it makes sense? The linear model decomposes the outcome \mathbf{y} into two part $\mathbf{X}\boldsymbol{\beta} = \sum_{i=1}^p \beta_i \mathbf{x}_i$ (a linear combination of the predictors i.e. the columns \mathbf{x}_i 's in \mathbf{X}) and $\boldsymbol{\epsilon}$ is the error or the part of \mathbf{y} that cannot be explained by a linear combination of \mathbf{x}_i 's (this 'definitely' means that if there is a part (in \mathbf{y}) that involves quadratic contribution of the predictors, it will be part of $\boldsymbol{\epsilon}$ in the above specification.). Now \mathbf{H} (also called the orthogonal projection matrix with respect to \mathbf{X}) is constructed such that $\mathbf{H}\mathbf{X} = \mathbf{0}$ and $(\mathbf{I} - \mathbf{H})\mathbf{X} = \mathbf{X}$, meaning that $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + (\mathbf{I} - \mathbf{H})\boldsymbol{\epsilon} + \mathbf{H}\boldsymbol{\epsilon}$. Now, $(\mathbf{I} - \mathbf{H})\boldsymbol{\epsilon} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\epsilon}$.

Thus, $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + (\mathbf{I} - \mathbf{H})\boldsymbol{\epsilon} + \mathbf{H}\boldsymbol{\epsilon} = \mathbf{X}(\boldsymbol{\beta} + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\epsilon}) + \mathbf{H}\boldsymbol{\epsilon}$, which can be represented as $\mathbf{X}\boldsymbol{\theta} + \mathbf{H}\boldsymbol{\epsilon}$. Hence, $\mathbf{H}\boldsymbol{\epsilon}$ is the part that cannot be represented by a linear combination of columns

of \mathbf{X} . Since ϵ contains the information of variance. We can only use $\mathbf{H}\epsilon = \mathbf{H}\mathbf{y}$ to compute the variance.

The R code to fit the model $\mathbf{y}_i \sim \mathbf{x}_i\beta + \epsilon_i$ with $\epsilon_i \sim \text{MVN}(0, \Sigma)$, where \mathbf{y}_i is the array of 4 measurements in the lead study and \mathbf{x}_i is the design matrix with a one column and the Week variable. Here (1|V1) part lets the function know which measurements are together as in which measurements are coming from the same subjects and V1 has the subject indicators. This will help to define the error variance. If no structure for Σ is assumed, the following is the code for that. In case there is any known structure, we can incorporate with some modifications.

```
res <- lme4::lmer(measurement ~ Week + (1|V1), data = data_new, REML=T)
```

Note that the above is just an example of REML. Any method that marginalizes the likelihood by representing it as a function of a restricted set of parameters for running the analysis, it comes under REML.

If more structures in Σ are assumed, then lme in nlme is more suitable as one can consider a pre-specified correlation structure for analysis.

```
library(nlme)
res <- lme(measurement ~ Week, random = ~1|V1, data = data_new)
res.AR1 <- update(res, correlation = corAR1())
res.ARMA11 <- update(res, corr = corARMA(p = 1, q = 1))
```

Assumptions on the correlation structure may lead to differences in the fixed effect estimates.

4 Generalized least square and Mahalanobish distance

When error variance Σ is given, the generalized least square objective is $f(\beta) = \sum_{i=1}^N (\mathbf{y}_i - \mathbf{X}_i\beta)^T \Sigma^{-1} (\mathbf{y}_i - \mathbf{X}_i\beta)$, where each \mathbf{y}_i is n dimensional and Σ is $n \times n$.

Let the eigen-decomposition of Σ be $\mathbf{U}\mathbf{D}\mathbf{U}^T$, where \mathbf{D} is a diagonal matrix with the eigenvalues and \mathbf{U} is the eigenvector matrix where each column is different eigenvectors. Another way is $\Sigma = \sum_{i=1}^n d_i \mathbf{u}_i \mathbf{u}_i^T$, where d_i is the i -th diagonal entry in \mathbf{D} and \mathbf{u}_i is i -th column in \mathbf{U} . Since these are eigenvectors, by definition these vectors are orthogonal. We usually set $d_1 \geq d_2 \geq \dots \geq d_n$. Thus, the interpretation of \mathbf{u}_i 's is that the greatest amount of the variable of the error will be along \mathbf{u}_1 , then along \mathbf{u}_2 , and so on.

Above representation leads to $\Sigma^{-1} = \sum_{i=1}^n \frac{1}{d_i} \mathbf{u}_i \mathbf{u}_i^T$ and thus, $f(\beta) = \sum_{i=1}^N (\mathbf{y}_i - \mathbf{X}_i\beta)^T \Sigma^{-1} (\mathbf{y}_i - \mathbf{X}_i\beta) = \sum_{i=1}^N \sum_{j=1}^n \frac{1}{d_j} \{\mathbf{u}_j^T (\mathbf{y}_i - \mathbf{X}_i\beta)\}^2$ (Verify!). Here, $\mathbf{u}_j^T (\mathbf{y}_i - \mathbf{X}_i\beta)$ is the contribution of error of i -th replication along j -th eigenvector \mathbf{u}_j . Hence, the contribution of the error along \mathbf{u}_1 has a lesser contribution to the loss with respect to \mathbf{u}_2 . The contribution of the error along \mathbf{u}_n has the maximum weight. This means that the minimization process to find β will put more importance on reducing the contribution of the resulting error along, \mathbf{u}_n which makes sense as the variance of the error has the least contribution in that direction.

$(\mathbf{y}_i - \mathbf{X}_i\beta)^T \Sigma^{-1} (\mathbf{y}_i - \mathbf{X}_i\beta)$ is also called Mahalanobish distance between \mathbf{y}_i and $\mathbf{X}_i\beta$. The main difference between ℓ_2 -distance ($\|\mathbf{y}_i - \mathbf{X}_i\beta\|_2$) and this distance is that the latter is influenced by the stretch of Σ to different directions.