

A3. ANOVA Types (I, II, III)

Why the sums of squares matter when your data are unbalanced

Dr. Paul Schmidt

To install and load all the packages used in this chapter, run the following code:

```
for (pkg in c("car", "broom", "glue", "tidyverse")) {
  if (!require(pkg, character.only = TRUE)) install.packages(pkg)
}

library(car)
library(broom)
library(glue)
library(tidyverse)
```

When running an analysis of variance on a linear model, R actually offers three different flavours, traditionally labelled Type I, Type II and Type III. For perfectly balanced data they all return the same numbers, which is why the distinction is often glossed over in introductory courses. The moment the data are unbalanced — and real experiments almost always end up unbalanced, even if only because a plot was lost or a plant died — the three types can give noticeably different p-values for the same effect. This chapter explains what the three types actually compute, shows a small example where the differences become visible, and gives practical recommendations for which one to use.

A Quick Overview

The three types differ in *how the sum of squares for each term is calculated*, not in the underlying model. The model is fit once with `lm()`. The “type” only decides which comparisons of nested models are used to attribute variation to each term.

Type	R call	How SS are computed	Typical use
I (sequential)	<code>stats::anova(mod)</code>	Each term added on top of the previous ones, in the order they appear in the formula.	Balanced data, or genuinely hierarchical / nested models where term order reflects a causal sequence.
II (hierarchical)	<code>car::Anova(mod, type = "II")</code>	Each main effect adjusted for all other main effects, but not for interactions that contain it.	Unbalanced data with only main effects, or when interactions are present but not of primary interest. Recommended default for most applied analyses [1].

Type	R call	How SS are computed	Typical use
III (marginal)	<code>car::Anova(mod, type = "III")</code>	Each term adjusted for all other terms, including higher-order interactions containing it.	When interactions are present and main effects should be tested “at the margin”. Requires sum-to-zero contrasts to be meaningful.

Two points are worth emphasising before we look at any data. First, `stats::anova()` always computes Type I sequential sums of squares — it has no `type` argument, and changing the order of predictors in the formula will change the result. Second, `car::Anova()` (note the capital A) is the standard tool for Type II and Type III and is essentially required for the latter, because correct Type III tests need a particular kind of contrast coding (more on this below).

Why Balance Matters

When a dataset is balanced — every factor combination has the same number of observations — the main effects are *orthogonal* to each other. Orthogonality means that the variation explained by factor A does not overlap with the variation explained by factor B, so it does not matter in which order we attribute it. All three ANOVA types give the same sums of squares and the same p-values.

Unbalanced data destroy this orthogonality. The variation explained by A and by B now shares some common ground, and the three types differ in how they handle this shared portion:

- **Type I** gives all shared variation to whichever term comes first in the formula.
- **Type II** tests each main effect after removing the contribution of the other main effects, ignoring interactions.
- **Type III** tests each main effect after removing the contribution of all other terms, including interactions.

A Concrete 2 by 2 Example

To make this visible, we construct a small unbalanced two-factor dataset. The two factors `diet` and `supp` each have two levels, and the cell counts are deliberately uneven:

```
set.seed(42)

dat <- tibble(
  diet = rep(c("low", "high"), times = c(14, 10)),
  supp = c(rep(c("A", "B"), times = c(10, 4)), # low: 10 A, 4 B
           rep(c("A", "B"), times = c(3, 7))), # high: 3 A, 7 B
  response = c(
    rnorm(10, mean = 10, sd = 1.5), # low + A
    rnorm(4, mean = 12, sd = 1.5), # low + B
    rnorm(3, mean = 14, sd = 1.5), # high + A
    rnorm(7, mean = 17, sd = 1.5) # high + B
  )
)
```

```
) %>%
  mutate(across(c(diet, supp), as.factor))

xtabs(~ diet + supp, data = dat)
```

```
      supp
diet    A  B
high    3  7
low    10  4
```

The design matrix is clearly unbalanced: the combination *low diet + supplement A* has 10 observations, whereas *high diet + supplement A* has only 3. We now fit the two-way model with interaction:

```
mod <- lm(response ~ diet * supp, data = dat)
```

Type I Depends on Term Order

Let us first see what `stats::anova()` does. We fit the same model but swap the order of the two factors in the formula, and compare the Type I tables side by side:

```
mod_ds <- lm(response ~ diet * supp, data = dat)
mod_sd <- lm(response ~ supp * diet, data = dat)

anova(mod_ds)
```

```
Analysis of Variance Table
```

```
Response: response
      Df Sum Sq Mean Sq F value    Pr(>F)
diet    1  95.469   95.469  27.3858 4.031e-05 ***
supp    1  17.563   17.563   5.0381  0.03627 *
diet:supp 1    0.002    0.002  0.0006  0.98005
Residuals 20  69.722    3.486
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mod_sd)
```

```
Analysis of Variance Table
```

```
Response: response
      Df Sum Sq Mean Sq F value    Pr(>F)
supp    1  61.275   61.275  17.5770 0.0004485 ***
diet    1  51.758   51.758  14.8470 0.0009915 ***
supp:diet 1    0.002    0.002  0.0006 0.9800473
Residuals 20  69.722    3.486
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The sums of squares and p-values for `diet` and `supp` differ between the two tables, simply because of term order. Whichever factor is listed first “absorbs” the shared variation. This is exactly the behaviour that makes Type I unattractive as a default: a scientifically meaningless decision (which factor to write first) changes the reported test statistic.

Type II and Type III Do Not

In contrast, `car::Anova()` produces order-independent tables:

```
Anova(mod_ds, type = "II")
```

```
Anova Table (Type II tests)

Response: response
      Sum Sq Df F value    Pr(>F)
diet    51.758  1 14.8470 0.0009915 ***
supp    17.563  1  5.0381 0.0362658 *
diet:supp 0.002  1  0.0006 0.9800473
Residuals 69.722 20
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(mod_ds, type = "III")
```

```
Anova Table (Type III tests)

Response: response
      Sum Sq Df F value    Pr(>F)
(Intercept) 597.21  1 171.3119 2.888e-11 ***
diet         24.95  1   7.1576  0.01454 *
supp         7.25  1   2.0785  0.16486
diet:supp    0.00  1   0.0006  0.98005
Residuals   69.72 20
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The Type II and Type III tables are the same regardless of whether the formula reads `diet * supp` or `supp * diet`. However, the two tables differ from each other, because Type III adjusts each main effect for the interaction, whereas Type II does not.

Side-by-side Comparison

To make the differences plain, we collect the p-values for all three types into a single tidy table:

```
get_p <- function(model, term, type) {
  tbl <- if (type == "I") {
    broom::tidy(anova(model))
  } else {
    broom::tidy(Anova(model, type = type))
  }
  tbl %>% filter(term == !!term) %>% pull(p.value)
}

terms <- c("diet", "supp", "diet:supp")
types <- c("I", "II", "III")

crossing(term = terms, type = types) %>%
  rowwise() %>%
  mutate(p = get_p(mod_ds, term, type)) %>%
  ungroup() %>%
  pivot_wider(names_from = type, values_from = p,
              names_prefix = "Type ") %>%
  mutate(across(starts_with("Type"), \ (x) round(x, 4)))

# A tibble: 3 × 4
  term      `Type I` `Type II` `Type III`
<chr>    <dbl>    <dbl>    <dbl>
1 diet      0         0.001    0.0145
2 diet:supp 0.98      0.98     0.98
3 supp     0.0363    0.0363   0.165
```

Note that the interaction row (`diet:supp`) is identical across all three types — the highest-order interaction is always tested the same way. The differences are in the main effects.

The Contrast Coding Pitfall for Type III

There is one more point that trips up almost everyone who encounters Type III ANOVA for the first time. A correct Type III test requires **sum-to-zero contrasts** for the factors in the model. The R default is `contr.treatment` (reference coding), and

`car::Anova(mod, type = "III")` with this default will happily produce a table that looks fine but whose main-effect p-values are not what most users think they are: they depend on which level happens to be the reference.

The fix is either to set sum contrasts globally before fitting the model, or to set them directly on the factors. The global switch is:

```
options(contrasts = c("contr.sum", "contr.poly"))
mod_sum <- lm(response ~ diet * supp, data = dat)
Anova(mod_sum, type = "III")
```

Anova Table (Type III tests)

```
Response: response
      Sum Sq Df F value    Pr(>F)
(Intercept) 3479.7  1 998.1784 < 2.2e-16 ***
diet          51.7  1  14.8209 0.0009994 ***
supp          17.1  1   4.9035 0.0385802 *
diet:supp      0.0  1   0.0006 0.9800473
Residuals    69.7 20
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To compare, here is the same call with R's default treatment contrasts (we temporarily switch back):

```
options(contrasts = c("contr.treatment", "contr.poly"))
mod_trt <- lm(response ~ diet * supp, data = dat)
Anova(mod_trt, type = "III")
```

Anova Table (Type III tests)

```
Response: response
      Sum Sq Df F value    Pr(>F)
(Intercept) 597.21  1 171.3119 2.888e-11 ***
diet         24.95  1   7.1576  0.01454 *
supp          7.25  1   2.0785  0.16486
diet:supp     0.00  1   0.0006  0.98005
Residuals    69.72 20
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# reset to sum contrasts for the rest of the chapter
options(contrasts = c("contr.sum", "contr.poly"))
```

The main-effect rows for `diet` and `supp` look very different between the two calls. The version with treatment contrasts tests whether the *reference cell* equals zero, which is almost never a scientifically meaningful question. Only the sum-contrast version produces the “average main effect” that Type III is supposed to deliver. The `?car::Anova` help page is explicit about this requirement.

! Type III without sum contrasts is almost always wrong

If you need Type III sums of squares, set

`options(contrasts = c("contr.sum", "contr.poly"))` *before* fitting the model, or use `contrasts = list(factor1 = contr.sum, factor2 = contr.sum)` *inside* `lm()`. Refitting the model after only changing the option is essential — the contrasts are baked into the model matrix at fit time.

Which Type Should One Use?

There is no single answer that satisfies every situation, but the following guidelines cover the overwhelming majority of applied analyses:

- For **balanced data**, the choice does not matter numerically. Type I is the simplest report and is perfectly adequate.
- For **unbalanced data with main effects only**, Type II is the statistically most powerful option and avoids the arbitrary order dependence of Type I. Ø. Langsrud [1] gives a careful argument for preferring Type II over Type III in this setting; the core point is that Type II exploits the assumption of no interaction (which one is making anyway, if no interaction is in the model) to recover more degrees of freedom.
- For **unbalanced data with interactions that are of scientific interest**, Type III is the conventional choice in many fields (and the SAS default, which is why it is so widespread). It tests each main effect adjusted for the interaction, which matches the “what is the average effect of A, averaging over levels of B” interpretation — but only with sum-to-zero contrasts.
- For **mixed models** (e.g. `lmerTest::lmer`), the question is more subtle and is usually handled via Satterthwaite or Kenward-Roger approximations rather than by choosing an SS type. Those are covered in the mixed-models material of this course.

A practical workflow is to fit the model, check assumptions (see A1. Model Diagnostics), then report Type II by default, and switch to Type III only when the interaction is both present and scientifically meaningful. Whatever is reported, the chosen type and the contrast coding should be stated explicitly in the methods section, because the same model can produce three different ANOVA tables.

💡 Additional Resources

- Ø. Langsrud [1] — clear statistical argument for preferring Type II over Type III for unbalanced data.
- Fox, J. and Weisberg, S. (2019), *An R Companion to Applied Regression*, 3rd ed. — discussion of Type II and Type III in the `car` package context. See also `?car::Anova`.
- Anova - Type I/II/III SS explained
- How to interpret Type I, II, and III ANOVA? (CrossValidated)

i Key Takeaways

1. **Balanced data: all three types agree.** Worrying about the type only becomes necessary once the design is unbalanced.
2. `stats::anova()` is **Type I** and depends on the order of terms in the formula. Two scientifically equivalent models can produce different p-values.
3. `car::Anova()` provides **Type II and Type III** and is order-independent.
4. **Type III needs sum-to-zero contrasts.** Use `options(contrasts = c("contr.sum", "contr.poly"))` *before* fitting, otherwise the main-effect tests are not what they appear to be.
5. **Default recommendation: Type II** for most unbalanced designs without a scientifically central interaction, Type III when interactions are central. Always report which type and which contrasts were used.

Bibliography

- [1] Ø. Langsrud, "ANOVA for unbalanced data: Use Type II instead of Type III sums of squares," *Statistics and Computing*, vol. 13, no. 2, pp. 163–167, 2003, doi: 10.1023/A:1023260610025.