

# ANOVA Types

Paul Schmidt

2023-12-01

What are ANOVA type I, II and III and why you should care

## Table of contents

<b>1</b>	<b>Background &amp; Implementation</b>	<b>1</b>
<b>2</b>	<b>When Results Differ</b>	<b>2</b>
2.1	Examples . . . . .	2
2.1.1	Unbalanced data & order of terms . . . . .	2
2.1.2	Interaction . . . . .	4
<b>3</b>	<b>Conclusion</b>	<b>6</b>

In the realm of statistical analysis, particularly when dealing with linear models, understanding the nuances between different types of ANOVA (Analysis of Variance) is crucial. Specifically, we will delve into the distinctions between Type I, Type II, and Type III ANOVA, their theoretical underpinnings, and practical implications, especially in the context of unbalanced data.

## 1 Background & Implementation

- **Type I ANOVA** (Sequential):
  - `stats::anova(model)`
  - This method evaluates the significance of each predictor sequentially. For example, in a model with two predictors A and B, Type I ANOVA first assesses the effect of A, then the effect of B given A. The order of variables in your model is critical here, as it can alter the results, particularly in unbalanced datasets.
- **Type II ANOVA** (Hierarchical):

- `car::Anova(model, type = "II")`
- Type II looks at each main effect in the model, but unlike Type I, it does not depend on the order of terms. It's particularly useful for models with main effects only, both in balanced and unbalanced datasets. Here, each main effect is tested after accounting for other main effects, but without considering interactions.
- **Type III ANOVA (Marginal):**
  - `car::Anova(model, type = "III")`
  - This type tests each main effect after considering all other terms, including interactions. Type III is the go-to method when your model includes interactions, particularly in unbalanced datasets. It's robust against the order of terms in your model.

## 2 When Results Differ

Results from these ANOVA types differ notably in unbalanced datasets. In balanced datasets, all three types usually yield similar results. However, in unbalanced scenarios:

- Type I results depend on the order of terms. Type II and III remain consistent regardless of term order but differ from Type I.
- Type III adjusts main effects for interactions, leading to different outcomes than Type II in models with interaction terms.

### 2.1 Examples

#### 2.1.1 Unbalanced data & order of terms

This example is based on the augmented design chapter. The data is unbalanced so that for a Type I ANOVA there is a difference between the following two models, while for Type II and Type III it does not make a difference.

```
mod1 <- lm(yield ~ gen + block, data = dat)
mod2 <- lm(yield ~ block + gen, data = dat)
```

Here are the p-values summarized in a single table:

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

```

dat <- data.frame(
  gen = c(
    "st", "14", "26", "ci", "17", "wa", "22", "13", "st", "ci", "04", "15", "30",
    "03", "wa", "24", "st", "18", "27", "ci", "25", "28", "05", "wa", "st", "09",
    "06", "ci", "wa", "20", "11", "23", "st", "02", "21", "wa", "ci", "10", "08",
    "16", "st", "29", "07", "ci", "01", "wa", "12", "19"
  ),
  yield = c(
    2972L, 2405L, 2855L, 2592L, 2572L, 2608L, 2705L, 2391L, 3122L,
    3023L, 3018L, 2477L, 2955L, 3055L, 2477L, 2783L, 2260L, 2603L,
    2857L, 2918L, 2825L, 1903L, 2065L, 3107L, 3348L, 2268L, 2148L,
    2940L, 2850L, 2670L, 3380L, 2770L, 1315L, 1055L, 1688L, 1625L,
    1398L, 1293L, 1253L, 1495L, 3538L, 2915L, 3265L, 3483L, 3013L,
    3400L, 2385L, 3643L),
  block = rep(c("I", "II", "III", "IV", "V", "VI"), each = 8L),
  row = c(
    1L, 2L, 3L, 4L, 5L, 6L, 7L, 8L, 1L, 2L, 3L, 4L, 5L, 6L, 7L,
    8L, 1L, 2L, 3L, 4L, 5L, 6L, 7L, 8L, 9L, 10L, 11L, 12L, 13L, 14L,
    15L, 16L, 9L, 10L, 11L, 12L, 13L, 14L, 15L, 16L, 9L, 10L, 11L,
    12L, 13L, 14L, 15L, 16L),
  col = rep(rep(1:3, 2), each = 8L)
)

# Define models
mod1 <- lm(yield ~ gen + block, data = dat)
mod2 <- lm(yield ~ block + gen, data = dat)

# Your get_p_values function
get_p_values <- function(mymodel, myterm, mytype) {
  if (mytype == "I") {
    x <- broom::tidy(stats::anova(mymodel))
  } else {
    x <- broom::tidy(car::Anova(mymodel, type = mytype))
  }
  x %>% filter(term == myterm) %>% pull(p.value)
}

# Define terms and types
terms <- c("gen", "block")
types <- c("I", "II", "III")
models <- list(mod1 = mod1, mod2 = mod2)

```

```
# Creating the combined tibble
crossing(model = names(models), term = terms, type = types) %>%
  rowwise() %>%
  mutate(
    p = get_p_values(models[[model]], term, type)
  ) %>%
  ungroup() %>%
  pivot_wider(
    names_from = term,
    values_from = p
  )
```

```
# A tibble: 6 x 4
  model type      block      gen
  <chr> <chr>      <dbl>    <dbl>
1 mod1  I          0.000208  0.00911
2 mod1  II         0.000208  0.293
3 mod1  III        0.000208  0.293
4 mod2  I          0.00000576 0.293
5 mod2  II         0.000208  0.293
6 mod2  III        0.000208  0.293
```

### 2.1.2 Interaction

In this example, our model includes an interaction term between two factors, ‘N’ and ‘G’. As said before, Type III adjusts main effects for interactions, leading to different outcomes than Type II.

```
mod <- lm(yield ~ N + G + N:G, data = dat)
```

Here are the p-values summarized in a single table:

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

dat <- data.frame(
  N = c(
    "Koopa", "Peach", "Yoshi", "Koopa", "Peach", "Toad", "Toad", "Koopa", "Yoshi",
    "Goomba", "Toad", "Yoshi", "Yoshi", "Toad", "Koopa", "Peach", "Yoshi",
```

```

"Koopa", "Goomba", "Goomba", "Diddy", "Koopa", "Goomba", "Goomba", "Toad",
"Peach", "Yoshi", "Diddy", "Peach", "Diddy", "Peach", "Peach", "Koopa",
"Diddy", "Koopa", "Goomba", "Toad", "Toad", "Yoshi", "Yoshi", "Diddy",
"Yoshi", "Koopa", "Peach", "Diddy", "Diddy", "Diddy", "Goomba"
),
G = c(
  "Simba", "Nala", "Pumba", "Timon", "Nala", "Nala", "Simba", "Simba", "Nala",
  "Nala", "Simba", "Simba", "Simba", "Pumba", "Nala", "Pumba", "Nala", "Nala",
  "Nala", "Simba", "Simba", "Pumba", "Pumba", "Timon", "Timon", "Timon", "Nala",
  "Nala", "Simba", "Pumba", "Timon", "Pumba", "Timon", "Timon", "Pumba",
  "Simba", "Pumba", "Simba", "Pumba", "Timon", "Nala", "Timon", "Simba",
  "Simba", "Timon", "Simba", "Simba", "Pumba"
),
yield = c(
  6162, 6318, 1690, 4948, 5736, 7072, 6192, 5598, 6666, 5044, 7146, 7578, 7642,
  2504, 6682, 2338, 6324, 5316, 4034, 4030, 6600, 5372, 4216, 3554, 5970, 5684,
  7862, 6732, 9012, 5924, 7302, 1560, 6094, 5904, 4694, 4520, 5126, 6860, 1594,
  6392, 5738, 7260, 5256, 8548, 5974, 6794, 5806, 4212
)
)
)

mod <- lm(yield ~ N*G, data = dat)

anova_type_I <- broom::tidy(stats::anova(mod)) %>%
  select(term, typeI = p.value)

anova_type_II <- broom::tidy(car::Anova(mod, type = "II")) %>%
  select(term, typeII = p.value)

anova_type_III <- broom::tidy(car::Anova(mod, type = "III")) %>%
  select(term, typeIII = p.value)

list(anova_type_I, anova_type_II, anova_type_III) %>%
  reduce(full_join, by = "term") %>%
  filter(term %in% c("N", "G", "N:G")) %>%
  mutate(across(starts_with("type"), ~insight::format_p(., name = NULL)))

# A tibble: 3 x 4
  term typeI typeII typeIII
<chr> <chr> <chr> <chr>
1 N      < .001 0.001 0.019

```

```
2 G      < .001 < .001 0.872
3 N:G    < .001 < .001 < .001
```

### 3 Conclusion

In most cases it's probably best to conduct a Type III ANOVA, e.g. via `car::Anova(model, type = "III")`.

#### Additional Resources

- [Anova – Type I/II/III SS explained](#)
- StackExchange Discussion on [How to interpret type I, type II, and type III ANOVA and MANOVA?](#)