A3. ANOVA-Typen (I, II, III)

Warum die Quadratsummen bei unbalancierten Daten wichtig sind

Dr. Paul Schmidt

Um alle in diesem Kapitel verwendeten Pakete zu installieren und zu laden, kann man folgenden Code ausführen:

```
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)
```

Wenn man eine Varianzanalyse auf ein lineares Modell anwendet, bietet R tatsächlich drei verschiedene Varianten an, die traditionell als Typ I, Typ II und Typ III bezeichnet werden. Für perfekt balancierte Daten liefern alle drei dieselben Zahlen, weshalb der Unterschied in Einführungskursen oft übergangen wird. Sobald die Daten unbalanciert sind - und reale Experimente enden fast immer unbalanciert, sei es auch nur, weil eine Parzelle verloren ging oder eine Pflanze einging - können die drei Typen für denselben Effekt deutlich unterschiedliche p-Werte liefern. Dieses Kapitel erklärt, was die drei Typen tatsächlich berechnen, zeigt ein kleines Beispiel, in dem die Unterschiede sichtbar werden, und gibt praktische Empfehlungen, welcher Typ zu verwenden ist.

Ein schneller Überblick

Die drei Typen unterscheiden sich darin, *wie die Quadratsumme für jeden Term berechnet wird*, nicht im zugrunde liegenden Modell. Das Modell wird einmal mit `lm()` gefittet. Der “Typ” entscheidet lediglich, welche Vergleiche genesteter Modelle verwendet werden, um die Variation den einzelnen Termen zuzuordnen.

Typ	R-Aufruf	Wie die Quadratsummen berechnet werden	Typischer Einsatz
I (sequenziell)	`stats::anova(mod)`	Jeder Term wird auf die vorherigen aufgesetzt, in der Reihenfolge, in der sie in der Formel erscheinen.	Balancierte Daten oder echt hierarchische / genestete Modelle, bei denen die Termreihenfolge eine kausale Abfolge widerspiegelt.
II (hierarchisch)	`car::Anova(mod, type = "II")`	Jeder Haupteffekt wird für alle anderen	Unbalancierte Daten mit ausschließlich

Typ	R-Aufruf	Wie die Quadratsummen berechnet werden	Typischer Einsatz
		Haupteffekte adjustiert, aber nicht für Interaktionen, die ihn enthalten.	Haupteffekten, oder wenn Interaktionen vorhanden, aber nicht von primärem Interesse sind. Empfohlener Standard für die meisten angewandten Analysen [1].
III (marginal)	`car::Anova(mod, type = "III")`	Jeder Term wird für alle anderen Terme adjustiert, einschließlich höherer Interaktionen, die ihn enthalten.	Wenn Interaktionen vorhanden sind und Haupteffekte “am Rand” (marginal) getestet werden sollen. Erfordert Summen-Kontraste (sum-to-zero), um sinnvoll zu sein.

Zwei Punkte sind hervorzuheben, bevor wir uns Daten anschauen. Erstens berechnet `stats::anova()` immer sequenzielle Quadratsummen vom Typ I - die Funktion hat kein `type`-Argument, und das Ändern der Reihenfolge der Prädiktoren in der Formel verändert das Ergebnis. Zweitens ist `car::Anova()` (man beachte das großgeschriebene A) das Standardwerkzeug für Typ II und Typ III und ist für letzteren praktisch unverzichtbar, weil korrekte Typ-III-Tests eine bestimmte Art der Kontrastkodierung benötigen (mehr dazu weiter unten).

Warum Balance eine Rolle spielt

Wenn ein Datensatz balanciert ist - jede Faktorkombination hat dieselbe Anzahl an Beobachtungen - sind die Haupteffekte *orthogonal* zueinander. Orthogonalität bedeutet, dass die durch Faktor A erklärte Variation sich nicht mit der durch Faktor B erklärten Variation überschneidet, sodass es keine Rolle spielt, in welcher Reihenfolge man sie zuordnet. Alle drei ANOVA-Typen liefern dieselben Quadratsummen und dieselben p-Werte.

Unbalancierte Daten zerstören diese Orthogonalität. Die durch A und durch B erklärte Variation teilt sich nun einen gemeinsamen Anteil, und die drei Typen unterscheiden sich darin, wie sie mit diesem geteilten Anteil umgehen:

- **Typ I** weist die gesamte geteilte Variation demjenigen Term zu, der in der Formel zuerst steht.
- **Typ II** testet jeden Haupteffekt, nachdem der Beitrag der anderen Haupteffekte entfernt wurde, wobei Interaktionen ignoriert werden.

- **Typ III** testet jeden Haupteffekt, nachdem der Beitrag aller anderen Terme entfernt wurde, einschließlich Interaktionen.

Ein konkretes 2-mal-2-Beispiel

Um dies sichtbar zu machen, konstruieren wir einen kleinen unbalancierten Zwei-Faktor-Datensatz. Die beiden Faktoren `diet` und `supp` haben jeweils zwei Stufen, und die Zellbesetzungen sind bewusst ungleich:

```
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
```

Die Designmatrix ist deutlich unbalanciert: Die Kombination *low diet* + *supplement A* hat 10 Beobachtungen, während *high diet* + *supplement A* nur 3 hat. Wir fitten nun das Zwei-Faktor-Modell mit Interaktion:

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

Typ I hängt von der Termreihenfolge ab

Schauen wir zuerst, was `stats::anova()` tut. Wir fitten dasselbe Modell, vertauschen aber die Reihenfolge der beiden Faktoren in der Formel und vergleichen die beiden Typ-I-Tabellen nebeneinander:

```
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
```

Die Quadratsummen und p-Werte für `diet` und `supp` unterscheiden sich zwischen den beiden Tabellen, einzig aufgrund der Termreihenfolge. Welcher Faktor auch immer zuerst aufgeführt wird, “absorbiert” die geteilte Variation. Genau dieses Verhalten macht Typ I als Standard unattraktiv: Eine wissenschaftlich bedeutungslose Entscheidung (welcher Faktor zuerst geschrieben wird) verändert die berichtete Teststatistik.

Typ II und Typ III nicht

Im Gegensatz dazu erzeugt `car::Anova()` reihenfolgeunabhängige Tabellen:

```
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
```

Die Typ-II- und Typ-III-Tabellen sind gleich, unabhängig davon, ob die Formel `diet * supp` oder `supp * diet` lautet. Die beiden Tabellen unterscheiden sich allerdings voneinander, weil Typ III jeden Haupteffekt für die Interaktion adjustiert, während Typ II das nicht tut.

Vergleich nebeneinander

Um die Unterschiede deutlich zu machen, sammeln wir die p-Werte für alle drei Typen in einer einzigen aufgeräumten Tabelle:

```
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
```

Man beachte, dass die Interaktionszeile (`diet:supp`) über alle drei Typen identisch ist - die Interaktion höchster Ordnung wird immer auf dieselbe Weise getestet. Die Unterschiede liegen in den Haupteffekten.

Die Falle der Kontrastkodierung bei Typ III

Es gibt noch einen Punkt, über den fast jeder stolpert, der zum ersten Mal auf die Typ-III-ANOVA trifft. Ein korrekter Typ-III-Test erfordert **Summen-Kontraste (sum-to-zero)** für die Faktoren im Modell. Der R-Standard ist `contr.treatment` (Referenzkodierung), und `car::Anova(mod, type = "III")` wird mit diesem Standard bereitwillig eine Tabelle erzeugen, die in Ordnung aussieht, deren Haupteffekt-p-Werte aber nicht das sind, was die meisten Anwender denken: Sie hängen davon ab, welche Stufe zufällig die Referenz ist.

Die Lösung besteht entweder darin, Summen-Kontraste global vor dem Fitten des Modells zu setzen, oder sie direkt an den Faktoren zu setzen. Der globale Umschalter ist:

```
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
```

Zum Vergleich hier derselbe Aufruf mit R's Standard-Treatment-Kontrasten (wir schalten vorübergehend zurück):

```r
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
```

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

Die Haupteffekt-Zeilen für `diet` und `supp` sehen zwischen den beiden Aufrufen sehr unterschiedlich aus. Die Version mit Treatment-Kontrasten testet, ob die *Referenzzelle* gleich null ist, was fast nie eine wissenschaftlich sinnvolle Frage ist. Nur die Version mit Summen-Kontrasten erzeugt den "durchschnittlichen Haupteffekt", den Typ III liefern soll. Die Hilfeseite `?car::Anova` ist bezüglich dieser Anforderung explizit.

> ! Typ III ohne Summen-Kontraste ist fast immer falsch
>
> Wenn man Quadratsummen vom Typ III benötigt, setzt man `options(contrasts = c("contr.sum", "contr.poly"))` *vor* dem Fitten des Modells, oder man verwendet `contrasts = list(factor1 = contr.sum, factor2 = contr.sum)` innerhalb von `lm()`. Das erneute Fitten des Modells nach dem bloßen Ändern der Option ist unverzichtbar - die Kontraste werden zum Zeitpunkt des Fittens in die Modellmatrix eingebacken.

Welchen Typ sollte man verwenden?

Es gibt keine einzelne Antwort, die jeder Situation gerecht wird, aber die folgenden Leitlinien decken die überwältigende Mehrheit angewandter Analysen ab:

- Für **balancierte Daten** spielt die Wahl numerisch keine Rolle. Typ I ist die einfachste Darstellung und vollkommen ausreichend.
- Für **unbalancierte Daten mit ausschließlich Haupteffekten** ist Typ II die statistisch trennschärfste Option und vermeidet die willkürliche Reihenfolgeabhängigkeit von Typ I. Ø. Langsrud [1] gibt eine sorgfältige Begründung dafür, in diesem Fall Typ II gegenüber Typ III zu bevorzugen; der Kernpunkt ist, dass Typ II die Annahme keiner Interaktion ausnutzt (die man ohnehin trifft, wenn keine Interaktion im Modell ist), um mehr Freiheitsgrade zurückzugewinnen.
- Für **unbalancierte Daten mit Interaktionen, die von wissenschaftlichem Interesse sind**, ist Typ III in vielen Disziplinen die konventionelle Wahl (und der SAS-Standard, weshalb er so verbreitet ist). Er testet jeden Haupteffekt adjustiert für die Interaktion, was zur Interpretation "was ist der durchschnittliche Effekt von A, gemittelt über die Stufen von B" passt - aber nur mit Summen-Kontrasten (sum-to-zero).

- Für **gemischte Modelle** (z.B. `lmerTest::lmer`) ist die Frage subtiler und wird üblicherweise über Satterthwaite- oder Kenward-Roger-Approximationen behandelt statt über die Wahl eines Quadratsummen-Typs. Diese werden im Mixed-Models-Material dieses Kurses behandelt.

Ein praktischer Workflow besteht darin, das Modell zu fitten, die Annahmen zu prüfen (siehe A1. Modelldiagnostik), dann standardmäßig Typ II zu berichten und nur dann zu Typ III zu wechseln, wenn die Interaktion sowohl vorhanden als auch wissenschaftlich bedeutsam ist. Was auch immer berichtet wird, der gewählte Typ und die Kontrastkodierung sollten im Methodenteil explizit angegeben werden, denn dasselbe Modell kann drei verschiedene ANOVA-Tabellen erzeugen.

Weiterführende Ressourcen

- Ø. Langsrud [1] - klare statistische Begründung dafür, Typ II gegenüber Typ III für unbalancierte Daten zu bevorzugen.
- Fox, J. and Weisberg, S. (2019), *An R Companion to Applied Regression*, 3rd ed. - Diskussion von Typ II und Typ III im Kontext des Pakets `car`. Siehe auch `?car::Anova`.
- Anova - Type I/II/III SS explained
- How to interpret Type I, II, and III ANOVA? (CrossValidated)

Wichtigste Erkenntnisse

1. **Balancierte Daten: Alle drei Typen stimmen überein.** Sich um den Typ Gedanken zu machen, wird erst notwendig, sobald das Design unbalanciert ist.
2. **`stats::anova()` ist Typ I** und hängt von der Reihenfolge der Terme in der Formel ab. Zwei wissenschaftlich äquivalente Modelle können unterschiedliche p-Werte erzeugen.
3. **`car::Anova()` liefert Typ II und Typ III** und ist reihenfolgeunabhängig.
4. **Typ III benötigt Summen-Kontraste (sum-to-zero).** Man verwende `options(contrasts = c("contr.sum", "contr.poly"))` *vor* dem Fitten, andernfalls sind die Haupteffekt-Tests nicht das, wonach sie aussehen.
5. **Standardempfehlung: Typ II** für die meisten unbalancierten Designs ohne wissenschaftlich zentrale Interaktion, Typ III, wenn Interaktionen zentral sind. Stets angeben, welcher Typ und welche Kontraste verwendet wurden.

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.