

Stat 5200, Module 4: Factorial Design and Analysis

Jesse Wheeler

Contents

1 Overview	1
2 Factorial Treatment Structure	1
3 The Two-Way ANOVA Model	3
4 Understanding Interaction Effects	4
5 ANOVA Mechanics and Hypothesis Testing	5
6 Applied Examples in R	6
7 Factorial Contrasts	11
8 Advanced Topics in Factorials	12
8.1 Unbalanced Factorials	12
9 Multiple Comparisons?	15
10 Resource Constraints: Unreplicated & Fractional Designs	15

1 Overview

Objectives and overview

- This material is based on Chapters 8–10 of the textbook “A First Course in Design and Analysis of Experiments” (Oehlert, 2000).
- Module outcomes include:
 - You will understand complete and incomplete factorial treatment structures.
 - You will develop an intuitive and formal understanding of interaction effects.
 - You will be able to specify and fit additive and non-additive two-way ANOVA models.
 - You will learn how to construct and interpret main effect contrasts and interaction contrasts.
 - You will know how to fit and analyze factorial designs in R using `aov()` and `emmeans()`.

2 Factorial Treatment Structure

Introduction to Factorial Structures

	Beef ($j = 1$)	Cereal ($j = 2$)	Pork ($j = 3$)	Marginal Row Mean
High Intake ($i = 1$)	μ_{11}	μ_{12}	μ_{13}	$\bar{\mu}_{1\bullet}$
Low Intake ($i = 2$)	μ_{21}	μ_{22}	μ_{23}	$\bar{\mu}_{2\bullet}$
Marginal Col Mean	$\bar{\mu}_{\bullet 1}$	$\bar{\mu}_{\bullet 2}$	$\bar{\mu}_{\bullet 3}$	μ (Grand Mean)

- In many experiments, we want to evaluate the effect of two or more explanatory factors simultaneously.
- A *factorial treatment structure* defines treatments as combinations of levels of each factor.
- Suppose Factor A has a levels and Factor B has b levels.
- A *complete factorial design* includes all $g = a \times b$ possible treatment combinations.

Example 1: 2×3 Diet Experiment

- Factor 1: Protein Source (Beef, Cereal, Pork) $\implies b = 3$.
- Factor 2: Protein Intake (High, Low) $\implies a = 2$.
- Total treatment combinations: $g = 2 \times 3 = 6$ unique diets.

Example 2: 2×2 Fertilizer Experiment

- Factor A: Superphosphate (P) $\rightarrow$ Levels: {Not Added, Added}
- Factor B: Potash (K) $\rightarrow$ Levels: {Not Added, Added}
- $g = 2 \times 2 = 4$ treatments:
 1. None (P_1K_1)
 2. Superphosphate only (P_2K_1)
 3. Potash only (P_1K_2)
 4. Both Superphosphate and Potash (P_2K_2)

Notice the power of a factorial design: instead of running two separate experiments (one for Superphosphate and one for Potash), we run a single experiment that evaluates both factors simultaneously using the exact same experimental units. This yields higher statistical efficiency and allows us to study whether the two factors work together or interfere with each other.

Main Effects and Interactions

- We are generally interested in both the *main effects* of each treatment, but also the *interactions* between treatments. Consider the example below (Example 2):
- **Main Effects:** Evaluated by comparing the marginal row means ($\bar{\mu}_{1\bullet}$ vs. $\bar{\mu}_{2\bullet}$) or the marginal column means ($\bar{\mu}_{\bullet 1}$, $\bar{\mu}_{\bullet 2}$, $\bar{\mu}_{\bullet 3}$). For example, what is the difference in response between high and low intake treatments?

- **Interactions:** Evaluated by looking at differences between cells. For example, is the difference between High and Low intake the same for Beef as it is for Cereal?

$$(\mu_{B,H} - \mu_{B,L}) \stackrel{?}{=} (\mu_{C,H} - \mu_{C,L})$$

- While we *could* formalize and test all these specific comparisons using individual contrasts, this can be tedious and less interpretable in some cases.
- Instead, we will introduce the *Two-Way ANOVA* as an efficient way to test for overall main effects and interactions simultaneously, with highly interpretable results.
- Like one-way ANOVA, we will express the total sum of squares in terms of treatments and errors.

If we denote A as the first treatment type, B as the second, and AB as the interaction, then we will have a similar break down:

$$SS_T = SS_A + SS_B + SS_{AB} + SS_E$$

Notation for Two-Way Factorials

- To identify observations in a two-way design, we use three subscripts:

$$y_{ijk}$$

- $i = 1, \dots, a$: level of Factor A
- $j = 1, \dots, b$: level of Factor B
- $k = 1, \dots, n$: replication number within treatment combination (i, j)

- Summary Notation:

- Sample mean for treatment (i, j) : $\bar{y}_{ij\bullet} = \frac{1}{n} \sum_{k=1}^n y_{ijk}$
- Marginal mean for level i of Factor A: $\bar{y}_{i\bullet\bullet} = \frac{1}{bn} \sum_{j=1}^b \sum_{k=1}^n y_{ijk}$
- Marginal mean for level j of Factor B: $\bar{y}_{\bullet j\bullet} = \frac{1}{an} \sum_{i=1}^a \sum_{k=1}^n y_{ijk}$
- Overall grand mean: $\bar{y}_{\bullet\bullet\bullet} = \frac{1}{abn} \sum_{i=1}^a \sum_{j=1}^b \sum_{k=1}^n y_{ijk}$

3 The Two-Way ANOVA Model

Model Specification

- For completely randomized designs, we will start with *balanced factorial* treatments. All $n_{ij} = n$ for all i, j .
- We model the response y_{ijk} as:

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$$

where:

- μ : Overall grand mean.
- α_i : Main effect of level i of Factor A.
- β_j : Main effect of level j of Factor B.
- $(\alpha\beta)_{ij}$: Interaction effect between level i of A and level j of B.

– $\epsilon_{ijk} \stackrel{iid}{\sim} N(0, \sigma^2)$: Random errors.

- *Parameter Identification Restrictions (Sum-to-Zero Constraints):*

$$\sum_{i=1}^a \alpha_i = \sum_{j=1}^b \beta_j = \sum_{i=1}^a (\alpha\beta)_{ik} = \sum_{j=1}^b (\alpha\beta)_{kj} = 0 \text{ for all } k$$

- *Hierarchy of Models:*

1. **Full (Non-additive) Model:** $y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \epsilon_{ijk}$
2. **Additive Model:** $y_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ijk}$
3. **Main Effect Models:** $y_{ijk} = \mu + \alpha_i + \epsilon_{ijk}$, $y_{ijk} = \mu + \beta_j + \epsilon_{ijk}$
4. **Constant Model:** $y_{ijk} = \mu + \epsilon_{ijk}$

Degrees of Freedom

- When computing the degrees of freedom for each parameter, note that the complete factorial structure performs an ANOVA for treatment A and B simultaneously.

The degrees of freedom for each main effect is computed as we would in a one-way ANOVA:

- Main effect *A*: $(a - 1)$ where a is the number of treatment levels for factor *A*.
- Main effect *B*: $(b - 1)$ where b is the number of treatment levels for factor *A*.
- Interaction effect (*AB*): $(a - 1)(b - 1)$.

One way to think about this last degree of freedom calculation is to see how many degrees of freedom are unused if we just treated it as a one-way ANOVA for the individual treatment groups μ_{ij} : There are ab total treatment means and we lose one degree of freedom for the grand mean, so we have $ab - 1$ total degrees of freedom that must be accounted for. If there are $(a - 1)$ for the main effects of factor *A*, and $(b - 1)$ for the main effects of factor *B*, then there must be:

$$(ab - 1) - (a - 1) - (b - 1) = ab - a - b + 1 = (a - 1)(b - 1)$$

degrees of freedom left for the interaction effects.

4 Understanding Interaction Effects

What is an Interaction Effect?

- *Definition:* An interaction effect occurs when the effect of one factor on the response depends on the level of another factor.
- Ask yourself: "Does the difference between levels of Factor *A* stay the same regardless of Factor *B*?"
 - If **YES**: The factors are *additive* (No Interaction).
 - If **NO**: There is an *interaction* between factors.

Intuition: "Difference of Differences"

Consider a 2×2 design with population means μ_{ij} :

- Simple main effect of *A* when Factor *B* is at level 1: $\mu_{11} - \mu_{21}$

Source	DF	Sum of Squares (SS)	Mean Square	F-Statistic
Factor A	$a - 1$	$SS_A = bn \sum \hat{\alpha}_i^2$	$MS_A = \frac{SS_A}{a-1}$	$F_A = \frac{MS_A}{MS_E}$
Factor B	$b - 1$	$SS_B = an \sum \hat{\beta}_j^2$	$MS_B = \frac{SS_B}{b-1}$	$F_B = \frac{MS_B}{MS_E}$
Interaction AB	$(a - 1)(b - 1)$	$SS_{AB} = n \sum \sum \hat{\alpha}\hat{\beta}_{ij}^2$	$MS_{AB} = \frac{SS_{AB}}{(a-1)(b-1)}$	$F_{AB} = \frac{MS_{AB}}{MS_E}$
Error	$ab(n - 1)$	$SS_E = \sum \sum \sum (y_{ijk} - \bar{y}_{ij\bullet})^2$	$MS_E = \frac{SS_E}{ab(n-1)}$	
Total	$abn - 1$	$SS_T = SS_A + SS_B + SS_{AB} + SS_E$		

- Simple main effect of A when Factor B is at level 2: $\mu_{12} - \mu_{22}$

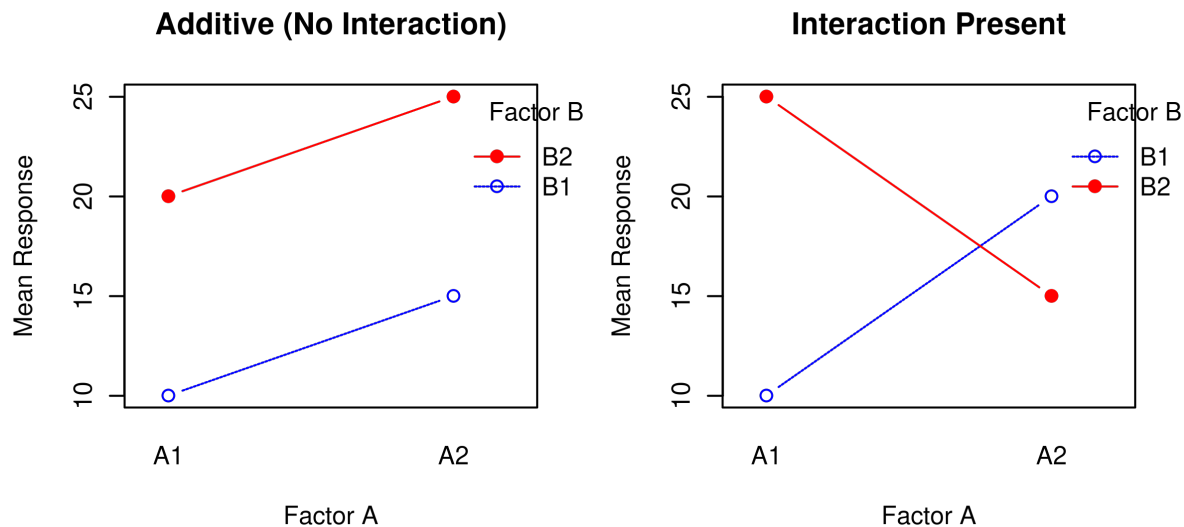
The *interaction effect* is the difference between these two simple effects:

$$(\mu_{11} - \mu_{21}) - (\mu_{12} - \mu_{22})$$

If this difference equals zero, the effect of A is constant across levels of B. If non-zero, an interaction exists.

Graphical Diagnosis: Interaction Plots

- An *interaction plot* displays treatment group means across levels of one factor, with lines connecting levels of the second factor.
- **Parallel lines** $\implies$ No Interaction (Additive).
- **Non-parallel or crossing lines** $\implies$ Interaction present.



5 ANOVA Mechanics and Hypothesis Testing

Two-Way ANOVA Table (Balanced Design)

- For a balanced design (n observations per cell, $N = abn$ total):

Sequence of Testing: Top-Down Rule

1. **Step 1: Test Interaction** ($H_0 : (\alpha\beta)_{ij} = 0$ for all i, j)

- If **Significant**: Stop! The model is non-additive. Do not interpret main effect F -tests in isolation. Examine simple main effects or interaction contrasts.

2. **Step 2: Test Main Effects**

- If interaction is non-significant, proceed to test $H_0 : \alpha_i = 0$ and $H_0 : \beta_j = 0$ independently using the main effect F -tests.

6 Applied Examples in R

Example 1: Significant Interaction (ToothGrowth)

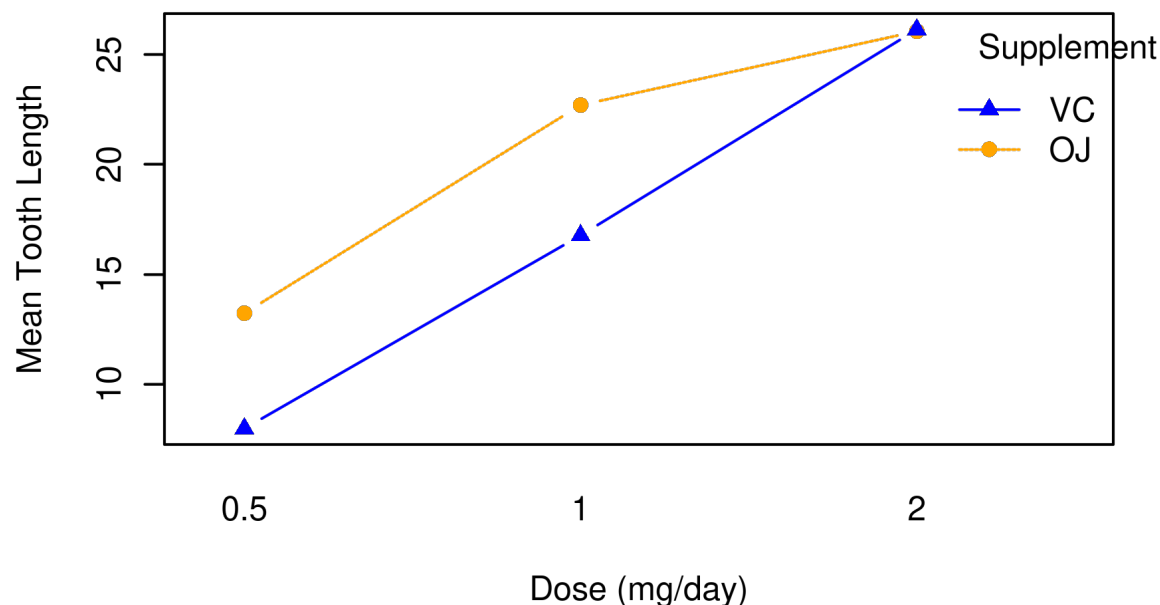
- The built-in `ToothGrowth` dataset measures tooth length (`len`) in guinea pigs under two factors:
 - **Supplement** (`supp`): Orange Juice (OJ) or Ascorbic Acid (VC).
 - **Dose** (`dose`): 0.5, 1.0, or 2.0 mg/day.
- First, we must convert `dose` from a numeric variable into a factor so R treats it as categorical.

```
# Load data and convert dose to a factor
data(ToothGrowth)
ToothGrowth$dose <- as.factor(ToothGrowth$dose)
```

Visualizing the Interaction

We start with an interaction plot. Crossing or converging lines suggest an interaction effect.

```
interaction.plot(
  x.factor = ToothGrowth$dose,
  trace.factor = ToothGrowth$supp,
  response = ToothGrowth$len,
  type = "b", pch = c(16, 17),
  col = c("orange", "blue"),
  xlab = "Dose (mg/day)",
  ylab = "Mean Tooth Length",
  trace.label = "Supplement"
)
```



Fitting the Model and Testing (Top-Down Rule)

We fit the full model with the `*` operator and check the ANOVA table.

```
mod_tooth <- aov(len ~ supp * dose, data = ToothGrowth)
summary(mod_tooth)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
supp	1	205.4	205.4	15.572	0.000231 ***
dose	2	2426.4	1213.2	92.000	< 2e-16 ***
supp:dose	2	108.3	54.2	4.107	0.021860 *
Residuals	54	712.1	13.2		

 Signif. codes:
 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

- **Interpretation:** The `supp:dose` interaction is significant ($p = 0.02$). Following our Top-Down Rule, we **stop** and do not interpret the main effects blindly. The effect of the supplement depends entirely on the dose.

Analyzing Simple Effects with `emmeans`

Because the interaction is significant, we should look at the *simple effects* – pairwise differences that vary just a single factor. In this case, the effect of the supplement *at each specific level* of the dose.

```
library(emmeans)
# The "/" operator tells emmeans to compare 'supp' AT each level of 'dose'
emm_tooth <- emmeans(mod_tooth, ~ supp | dose)
pairs(emm_tooth)
```

```
dose = 0.5:
  contrast estimate    SE df t.ratio p.value
OJ - VC      5.25 1.62 54   3.233  0.0021

dose = 1:
  contrast estimate    SE df t.ratio p.value
OJ - VC      5.93 1.62 54   3.651  0.0006

dose = 2:
  contrast estimate    SE df t.ratio p.value
OJ - VC     -0.08 1.62 54  -0.049  0.9609
```

- **Conclusion:** Orange juice (OJ) results in significantly longer teeth at doses of 0.5 and 1.0, but there is no significant difference between supplements at a dose of 2.0.

Demystifying the emmeans Syntax

- In a one-way ANOVA, we could use `emmeans(model, ~ Factor)` to get the estimated means for each group.
- In a two-way ANOVA with a significant interaction, we must examine *simple main effects*. We need to know the effect of one factor *at specific levels* of the other factor.

Breaking down: `~ supp | dose`

- `~ supp`: This tells R to calculate the differences between the levels of the Supplement factor (OJ vs. VC).
- `| (The Pipe/Bar)`: In R formulas, the vertical bar means “given” or “conditioned on.”
- `dose`: This is the grouping variable. It tells R to split the analysis up by the levels of Dose.
- **Plain English Translation:** “Compare the mean tooth length of OJ vs. VC, and do this comparison three separate times: once for the 0.5 dose, once for the 1.0 dose, and once for the 2.0 dose.”
- From our figure of the data (interaction plot), the Tooth Growth data is an example we may want to consider *polynomial contrasts* for factorial data. **There will be a 6000-level HW question** assignment related to this.

Example 2: No Interaction (Additive Model)

- The `npk` dataset in Canvas records pea yields based on applying Nitrogen (N) and Phosphate (P).
- Let’s evaluate if Nitrogen and Phosphate interact to affect yield.

```
NP <- read.csv('NP.csv')
str(NP)

'data.frame': 24 obs. of 3 variables:
 $ yield: num  49.5 62.8 46.8 57 59.8 58.5 55.5 56 62..
 $ P : int  1 1 0 0 0 1 0 1 1 1 ...
 $ N : int  0 1 0 1 1 1 0 0 0 1 ...
```

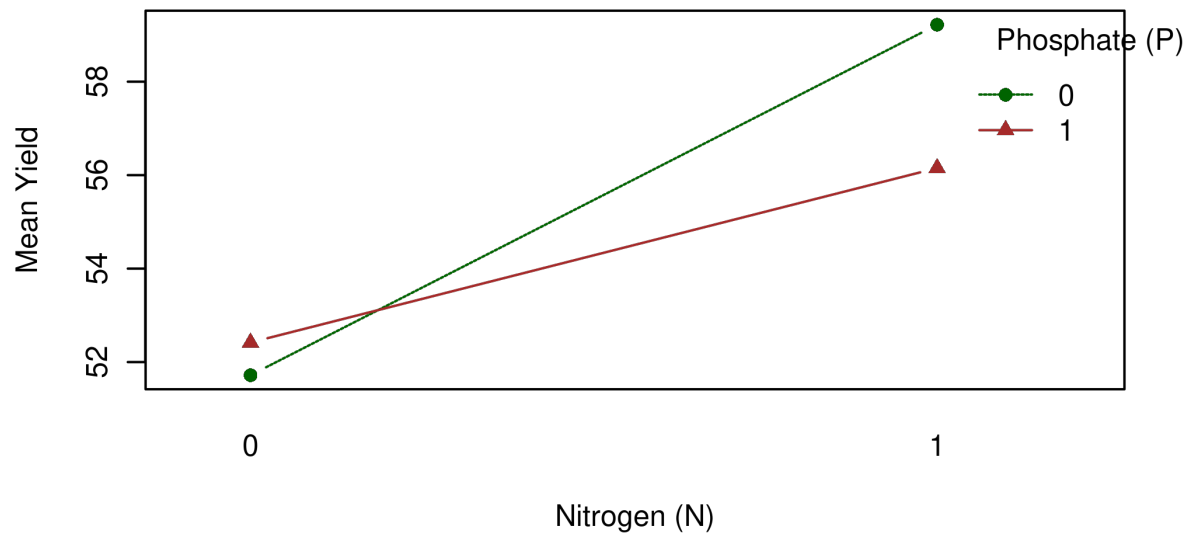
```
mod_NP <- aov(yield ~ N * P, data = NP)
summary(mod_NP)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
N	1	189.3	189.28	5.758	0.0263 *
P	1	8.4	8.40	0.256	0.6187
N:P	1	21.3	21.28	0.647	0.4305
Residuals	20	657.4	32.87		

Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

- **Interpretation:** The N:P interaction is not significant ($P = 0.4305$). That is, there is not strong evidence in the data that the effect of Nitrogen inclusions on Mean Yield depends on the inclusion of Phosphate.
- We can examine this further with the interaction plot

```
# Visualize the data
interaction.plot(
  x.factor = NP$N,
  trace.factor = NP$P,
  response = NP$yield,
  type = "b", pch = c(16, 17),
  col = c("darkgreen", "brown"),
  xlab = "Nitrogen (N)", ylab = "Mean Yield",
  trace.label = "Phosphate (P)"
)
```



- Not significant interaction. Now what?

- Historically, we might want to re-fit an *additive model* to conserve degrees of freedom. Doing so allows us to pool degrees of freedom from the interaction term, leading to higher statistical power to detect main effects.

Problem: This is called *post-selection inference*, and can mean we have inflated type-I error rates.

The reason why is because we are doing some type of “double dipping”. We should not use the data to decide *which* hypothesis to test, and then use that exact same data to test the hypothesis. Doing so leads to additional Type-I errors.

- Current recommended practice: we are more concerned about the increased Type-I error rate than we are of improving statistical power. We will interpret P-values in the context of accounting for interactions, even if they are non-significant.

(Historic approach): Fitting the Additive Model

We use the `+` operator to fit a model assuming the effects of N and P are strictly independent.

```
# Fit additive model (no interaction)
mod_NP_add <- aov(yield ~ N + P, data = NP)
summary(mod_NP_add)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
N	1	189.3	189.28	5.857	0.0247 *
P	1	8.4	8.40	0.260	0.6155
Residuals	21	678.7	32.32		

```
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- **Conclusion:** Nitrogen (N) has a significant main effect on yield ($p = 0.0247$), while Phosphate (P) does not ($p = 0.6155$). Because the model is additive, we conclude Nitrogen increases yield regardless of whether Phosphate is applied or not.
- We could continue to extract treatment group means and treatment effects using the `emmeans` package as needed (see below).

Extracting Treatment Means and Effects

- Following modern practice, we retain the full model (`mod_NP`) even though the interaction was not significant.
- We can use `emmeans` to extract the estimated marginal means (treatment means) for Nitrogen, averaging over the levels of Phosphate.

```
library(emmeans)

# Extract Estimated Marginal Means for Nitrogen (N)
# mod_NP is FULL model, not additive
emm_N <- emmeans(mod_NP, ~ N)
emm_N
```

N	emmean	SE	df	lower.CL	upper.CL
0	52.1	1.66	20	48.6	55.5
1	57.7	1.66	20	54.2	61.1

```
Results are averaged over the levels of: P
Confidence level used: 0.95
```

Calculating the Treatment Effect

To find the actual size of the treatment effect (the difference between adding Nitrogen vs. not adding Nitrogen), we pass our means into the `pairs()` function.

```
# Test the pairwise difference (Treatment Effect)
pairs(emm_N)
```

```
contrast estimate    SE df t.ratio p.value
NO - N1      -5.62 2.34 20  -2.400  0.0263
```

Results are averaged over the levels of: P

- **Interpretation:** The estimated mean yield without Nitrogen is 52.1, and with Nitrogen is 57.7. The `pairs()` output shows that applying Nitrogen increases the mean yield by an estimated 5.62 units, and this main effect is statistically significant.
- Notice the P-value is unchanged from our original ANOVA test.

7 Factorial Contrasts

Contrasts in Factorial Designs

- Contrasts in a factorial design operate exactly the same way as in a one-way ANOVA. We are simply picking weights to compare means.
- Depending on our research question, we apply these contrasts to two different types of means:
 1. *Main Effect Contrasts:* We compare the **marginal means** of one factor (averaging over the levels of the other factor).
 2. *Cell / Simple Effect Contrasts:* We compare the specific **treatment combination means** (μ_{ij}).
- While possible, we do not need to build complex weight matrices by hand. We use software to apply standard adjustments (like Tukey HSD) directly to the means we care about.

Main Effect Contrasts (No Interaction)

If there is no significant interaction, we want to know which levels of Factor A differ from each other overall. We extract the marginal means and apply a Tukey adjustment.

```
library(emmeans)

# Fit a Three-Way ANOVA
mod <- aov(response ~ Factor_A*Factor_B*Factor_C, data = DATA)

# Extract marginal means for Factor A
emm_A <- emmeans(mod, ~ Factor_A)

# Perform all pairwise comparisons with Tukey adjustment
pairs(emm_A, adjust = "tukey")
```

Cell Means & Simple Effect Contrasts (Interaction Present)

If an interaction is present, marginal means are misleading. We must compare the specific combinations (cells), or compare the levels of Factor A strictly within a single level of Factor B.

Effect	Type I (Sequential)	Type II (Hierarchical)	Type III (Marginal)
A	$SS(A)$	$SS(A \mid B, C)$	$SS(A \mid B, C, AB, AC, BC, ABC)$
B	$SS(B \mid A)$	$SS(B \mid A, C)$	$SS(B \mid A, C, AB, AC, BC, ABC)$
C	$SS(C \mid A, B)$	$SS(C \mid A, B)$	$SS(C \mid A, B, AB, AC, BC, ABC)$
AB	$SS(AB \mid A, B, C)$	$SS(AB \mid A, B, C, AC, BC)$	$SS(AB \mid A, B, C, AC, BC, ABC)$
AC	$SS(AC \mid A, B, C, AB)$	$SS(AC \mid A, B, C, AB, BC)$	$SS(AC \mid A, B, C, AB, BC, ABC)$
BC	$SS(BC \mid A, B, C, AB, AC)$	$SS(BC \mid A, B, C, AB, AC)$	$SS(BC \mid A, B, C, AB, AC, ABC)$
ABC	$SS(ABC \mid A, B, C, AB, AC, BC)$	$SS(ABC \mid A, B, C, AB, AC, BC)$	$SS(ABC \mid A, B, C, AB, AC, BC)$

Table 1: Comparison of Sums of Squares conditional notation for factors A, B, and C.

```
# Extract the means for every specific treatment combination
emm_cells <- emmeans(mod, ~ Factor_A * Factor_B)

# Compare Factor A ONLY within specific levels of Factor B
# (This tests the simple main effects)
contrast(emm_cells, method = "pairwise", by = "Factor_B")

# OR:
# emm_cells <- emmeans(mod, ~ Factor_A | Factor_B)
# contrast(emm_cells, method = "pairwise")

# Alternative: Compare all cells against a specific Control cell
# using Dunnett's adjustment
contrast(emm_cells, method = "trt.vs.ctrl", ref = 1)
```

Custom Contrasts

- You can still compute any custom contrasts, just as we did with the one-way ANOVA.
- To do so, print the output of a `emmeans(...)` function call, to see the order of the means; then you can use a custom list of contrasts, using the same order for your vector of contrast weights.

8 Advanced Topics in Factorials

8.1 Unbalanced Factorials

Types of Sums of Squares

- When cell sizes are unequal ($n_{ijk} \neq n$), factors lose orthogonality.
- *Why this matters:* The sum-of-squares (SS) for a particular term depends on what other terms are included in the model! See Oehlert (Example 10.1 from 2000) for an example.
- This gives rise to three different “types” of SS to consider. In a *balanced design*, all three give the same answer, so we don’t have to worry about the differences.
- The choice of *Sums of Squares (SS)* dictates which model terms are held constant in the reduced vs. full model comparison ($SS_{\text{Effect}} = SS_{\text{Error, reduced}} - SS_{\text{Error, full}}$).
- Historically, this was very problematic, and caused people to avoid analyzing data from unbalanced factorial designs. Today, software can readily solve this issue for us.

Type I Sum of Squares

Type I (Sequential)

Summary: The SS for any term is adjusted *only* for the terms that precede it in the model.

- For historic reasons, this is the output of the ‘aov()’ function.
- The hypothesis being tested is whether adding a specific term improves the model, *given* the terms already present.
- **Disadvantage:** The SS depends on the order in which you specify model terms!

See the Type I column of Table 1 for an example with a 3-way ANOVA.

```
# Create unbalanced data by dropping the first 8 observations
unbal_tooth <- ToothGrowth[-(1:8), ]

# Model 1: supp first
mod_I_A <- aov(len ~ supp * dose, data = unbal_tooth)
summary(mod_I_A)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
supp	1	4.6	4.6	0.330	0.5684
dose	2	1637.4	818.7	58.312	2.43e-13 ***
supp:dose	2	115.6	57.8	4.115	0.0227 *
Residuals	46	645.8	14.0		

```
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Type I Sum of Squares III

Doing the other order, we get completely different results!

```
# Model 2: dose first (Notice the SS for the main effects change!)
mod_I_B <- aov(len ~ dose * supp, data = unbal_tooth)
summary(mod_I_B)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
dose	2	1497.0	748.5	53.312	1.05e-12 ***
supp	1	145.0	145.0	10.330	0.00239 **
dose:supp	2	115.6	57.8	4.115	0.02269 *
Residuals	46	645.8	14.0		

```
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Type II Sum of Squares

Type II (Hierarchical)

Summary: The SS for any term is adjusted for the largest *hierarchical* model that does not contain that term.

- Useful for “building” hierarchical models. That is, helps find a hierarchical model that includes all necessary terms without retaining unnecessary ones.

- Directly tests if higher-order interactions are needed.
- Doesn't depend on order that terms are specified.

See the Type II column of Table 1 for an example with a 3-way ANOVA.

```
library(car) # standard package for Type-II

# Type II SS adjusts for all terms at the same or lower level
mod_II <- aov(len ~ supp * dose, data = unbal_tooth)
Anova(mod_II, type = 2)

# Reversing the order of terms will yield the exact same result here
```

Anova Table (Type II tests)

```
Response: len
      Sum Sq Df F value    Pr(>F)
supp    145.02  1 10.3295 0.002394 **
dose   1637.37  2 58.3119 2.433e-13 ***
supp:dose 115.56  2  4.1155 0.022686 *
Residuals 645.83 46
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Type III Sum of Squares

Type III (fully adjusted)

Summary: The SS for any term is adjusted for *all other terms*.

- Tests equally weighted hypotheses about treatment effects, placing no restriction on the cell means. This directly answer the question: “Does Factor A have an effect on average response across all levels of factor B”?
- Standard for factorial experiments.
- Like Type II, doesn't depend on model order.

See the Type III column of Table 1 for an example with a 3-way ANOVA.

```
library(emmeans)

# Type III evaluates equally weighted cell means
mod_III <- aov(len ~ supp * dose, data = unbal_tooth)

# joint_tests() from emmeans computes Type III SS.
joint_tests(mod_III)

model term df1 df2 F.ratio p.value
supp      1  46 12.000 0.0012
dose      2  46 50.169 <0.0001
supp:dose 2  46  4.115 0.0227
```

Which Type Should You Use in Practice?

- **Type I:** Avoid unless testing a strict hierarchical sequence. Order of terms in `aov()` changes the output!
- **Type II:** Best default when you are confident higher-order interactions are zero, or you're only interested in hierarchical models.
- **Type III:** Standard for factorial experiments. Tests equal weighting across all treatment cells regardless of sample size differences.

Contrasts for Unbalanced Data

A final note: contrasts for unbalanced data will generally follow the Type-III SS.

9 Multiple Comparisons?

Multiple Comparisons in Factorial Designs

- When doing a full factorial analysis, with all higher order interactions, we end up doing *many* F-tests for individual effects.
- For instance, in a three-way factorial analysis, there are $\binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 7$ total tests.
- Despite this, we have largely not discussed adjusting p-values for multiple tests!
- Common statistical practice does *not* adjust for these multiple F-tests. **Why?**
- There's not a satisfying answer other than: "that's what we do".
- Oehlert (2000) does argue, however, that we do multiple-testing adjustments for a "family" of tests.
- Common practice treats each main effect and interaction as a separate "family" of hypotheses and make multiple comparisons adjustments within a family, but not between them.

10 Resource Constraints: Unreplicated & Fractional Designs

Unreplicated Factorials ($n = 1$)

- Experiments are expensive. Sometimes we can only afford a single replicate per treatment combination ($n = 1$).
- *The Problem:* Fitting a full factorial model uses up all available degrees of freedom. The error degrees of freedom (df_E) becomes 0, so R returns NA for all standard errors and p-values.

```
unrep_data <- read.csv('unrepData.csv')

# Fits main effects, 2-way, 3-way, and 4-way interactions
mod_unrep <- aov(yield ~ A * B * C * D, data = unrep_data)

# F-values and p-values will not be returned
summary(mod_unrep)
```

	Df	Sum Sq	Mean Sq
A	1	1193.7	1193.7
B	1	9.2	9.2
C	2	525.3	262.7
D	1	309.9	309.9
A:B	1	24.5	24.5
A:C	2	6.7	3.4
B:C	2	43.4	21.7
A:D	1	0.4	0.4
B:D	1	0.0	0.0
C:D	2	5.2	2.6
A:B:C	2	2.4	1.2
A:B:D	1	15.4	15.4
A:C:D	2	12.5	6.2
B:C:D	2	15.9	7.9
A:B:C:D	2	5.3	2.6

The Fix: Pooling into Error

- Relying on the *Sparsity of Effects Principle*, we assume high-order interactions (e.g., 3-way and 4-way) are effectively zero (just random noise).
- We restrict the model, pushing those negligible higher-order terms into the residual error to mathematically generate degrees of freedom.

```
# Restrict the model to main effects and 2-way interactions only
# The "^2" syntax drops 3-way and 4-way terms into the error
mod_pooled <- aov(yield ~ (A + B + C + D)^2, data = unrep_data)

# Now R can calculate p-values for main and 2-way effects!
summary(mod_pooled)
```

Fractional Factorials: A High-Level Overview

- **The Problem:** Full factorial designs grow exponentially ($2^5 = 32$ treatment combinations). We often lack the budget or material to run every combination.
- **The Solution:** A *fractional factorial design* runs a carefully structured subset (e.g., a "half-fraction" of 16 combinations) to estimate main effects and low-level interactions efficiently.
- We are only going to briefly mention the concept here. For more details, see Oehlert (Chapter 18 of 2000).
- **The Cost (Aliasing):** By skipping some combinations, certain effects become perfectly tangled together. For instance, we may not tell if observed change in yield is caused by Factor C, or by the $A \times B$ interaction.
- **The Strategy:** We rely on the *sparsity of effects principle*—the assumption that higher-order interactions (3-way, 4-way) are usually zero. If C is aliased with $A \times B \times D$, we assume all the observed variation belongs to C.

Practical Advice for Software

Never arbitrarily drop runs from an experiment! Use specialized software to generate optimal fractions, and use the `alias()` function to see exactly what is entangled.

```
library(FrF2)

# Generate a half-fraction for 3 factors (4 runs instead of 8)
frac_data <- FrF2(nruns = 4, nfactors = 3, factor.names = c("A", "B", "C"))

# Fit the full model (adding mock response data for the formula)
frac_mod <- aov(rnorm(4) ~ A * B * C, data = frac_data)

# R will return NA in summary(frac_mod) for terms it cannot estimate.
# Use alias() to reveal exactly which effects are confounded:
alias(frac_mod)
```

Acknowledgments

- Compiled on October 13, 2026 using R version 4.5.2.
- Licensed under the [Creative Commons Attribution-NonCommercial license](#).  Please share and remix non-commercially, mentioning its origin.
- We acknowledge [students and instructors for previous versions of this course / slides](#).

References

Oehlert GW (2000). *A First Course in Design and Analysis of Experiments*. W. H. Freeman. ISBN 0-7167-3510-5. [1](#), [16](#), [23](#), [25](#)