FA9

Espiritu, Joseph Raphael, Harneyyer, Clores

2025-12-02

```
## Installing packages into 'C:/Users/josep/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)

## package 'tidyverse' successfully unpacked and MD5 sums checked
## package 'afex' successfully unpacked and MD5 sums checked
## package 'emmeans' successfully unpacked and MD5 sums checked
## package 'report' successfully unpacked and MD5 sums checked
## package 'effectsize' successfully unpacked and MD5 sums checked
## package 'broom' successfully unpacked and MD5 sums checked
## package 'scico' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\josep\AppData\Local\Temp\RtmpmKj5gw\downloaded_packages

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.6
## v forcats   1.0.1     v stringr   1.6.0
## v ggplot2   4.0.1     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.2.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## Loading required package: lme4
##
## Loading required package: Matrix
##
##
## Attaching package: 'Matrix'
##
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
##
## ************
## Welcome to afex. For support visit: http://afex.singmann.science/
##
## - Functions for ANOVAs: aov_car(), aov_ez(), and aov_4()
## - Methods for calculating p-values with mixed(): 'S', 'KR', 'LRT', and 'PB'
## - 'afex_aov' and 'mixed' objects can be passed to emmeans() for follow-up tests
```

```
## - Get and set global package options with: afex_options()
## - Set sum-to-zero contrasts globally: set_sum_contrasts()
## - For example analyses see: browseVignettes("afex")
## ************
##
##
## Attaching package: 'afex'
##
##
## The following object is masked from 'package:lme4':
##
##     lmer
##
##
## Welcome to emmeans.
## Caution: You lose important information if you filter this package's results.
## See '? untidy'
##
## Loading required package: carData
##
##
## Attaching package: 'car'
##
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
##
## The following object is masked from 'package:purrr':
##
##     some
```

```r
# --- 1. Enter your data (exactly as provided) -----------------
fertilizer <- rep(c("Blend X", "Blend Y", "Blend Z"), each = 20)
crop <- rep(rep(c("Wheat", "Corn", "Soy", "Rice"), each = 5), times = 3)

yield <- c(
  # Blend X
  123,156,112,100,168,
  128,150,174,116,109,
  166,178,187,153,195,
  151,125,117,158,155,
  # Blend Y
  135,130,176,120,155,
  175,132,120,187,184,
  140,145,159,131,126,
  167,183,142,167,168,
  # Blend Z
  156,180,147,146,193,
  186,138,178,176,190,
  185,206,188,165,188,
  175,173,154,191,169
)
```

```r
dat <- tibble(fertilizer = factor(fertilizer),
              crop = factor(crop),
              yield = yield)

# --- 2. Descriptives (cell means + SD) -----------------------
cell_stats <- dat %>%
  group_by(fertilizer, crop) %>%
  summarise(N = n(), mean = mean(yield), sd = sd(yield)) %>%
  arrange(fertilizer, crop)
```

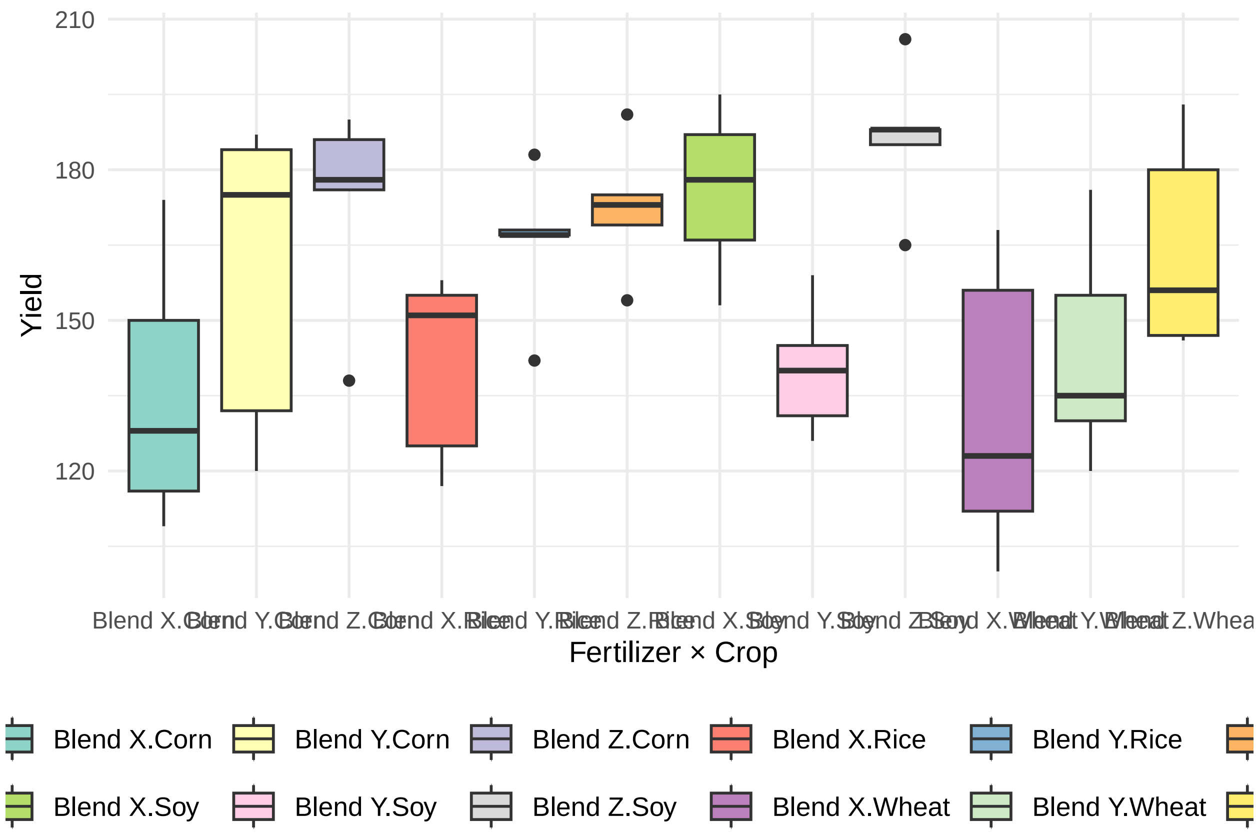
```
## `summarise()` has grouped output by 'fertilizer'. You can override using the
## `.groups` argument.
```

```r
print(cell_stats)
```

```
## # A tibble: 12 x 5
## # Groups:   fertilizer [3]
##    fertilizer crop      N  mean    sd
##    <fct>      <fct> <int> <dbl> <dbl>
##  1 Blend X    Corn      5  135.  26.6
##  2 Blend X    Rice      5  141.  18.8
##  3 Blend X    Soy       5  176.  16.7
##  4 Blend X    Wheat     5  132.  29.1
##  5 Blend Y    Corn      5  160.  31.3
##  6 Blend Y    Rice      5  165.  14.7
##  7 Blend Y    Soy       5  140.  12.9
##  8 Blend Y    Wheat     5  143.  22.3
##  9 Blend Z    Corn      5  174.  20.7
## 10 Blend Z    Rice      5  172.  13.3
## 11 Blend Z    Soy       5  186.  14.6
## 12 Blend Z    Wheat     5  164.  21.1
```

```r
# --- 3. Assumption checks -----------------------------------
# Boxplots (visual outlier check)
ggplot(dat, aes(
  x = interaction(fertilizer, crop),
  y = yield,
  fill = interaction(fertilizer, crop)
)) +
  geom_boxplot() +
  scale_fill_brewer(palette = "Set3") +
  theme_minimal() +
  labs(
    x = "Fertilizer × Crop",
    y = "Yield",
    fill = "Group"
  ) +
  theme(
    legend.position = "bottom",
    legend.box = "horizontal",
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10),
    legend.key.size = unit(0.7, "cm"),
    legend.spacing.x = unit(0.4, "cm"),
```

```
    legend.text.align = 0
  ) +
  guides(fill = guide_legend(nrow = 2, byrow = TRUE))
```




```
# 1. Create ID column BEFORE using aov_ez
dat <- dat %>% mutate(row = row_number())

# 2. Run the two-way ANOVA
mod <- aov_ez(
  id = "row",
  dv = "yield",
  between = c("fertilizer", "crop"),
  data = dat,
  anova_table = list(correction = "none", es = "pes")
)
```

```
## Contrasts set to contr.sum for the following variables: fertilizer, crop
```

```
# 3. Extract residuals
resid_vals <- residuals(mod$lm)

# 4. Shapiro test
shapiro.test(resid_vals)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  resid_vals
## W = 0.98027, p-value = 0.4402
```

```r
# Levene's test for homogeneity

leveneTest(yield ~ fertilizer * crop, data = dat)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group 11  0.6746 0.7551
##       48
```

```r
# --- 4. Two-way ANOVA (Type II table similar to JASP) ----------
# Use aov() + car::Anova (Type II) or afex. We'll print Type II.

lm1 <- lm(yield ~ fertilizer * crop, data = dat)
anova2 <- Anova(lm1, type = "II")   # Type II table
anova2_tidy <- broom::tidy(anova2)
anova2_tidy <- anova2_tidy %>%
  mutate(sum_sq = sumsq, df = df, statistic = statistic, p.value = p.value) %>%
  select(term, sum_sq, df, statistic, p.value)

print(anova2_tidy)
```

```
## # A tibble: 4 x 5
##   term            sum_sq    df statistic   p.value
##   <chr>            <dbl> <dbl>     <dbl>     <dbl>
## 1 fertilizer       8783.     2      9.93  0.000245
## 2 crop             3412.     3      2.57  0.0649
## 3 fertilizer:crop  6226.     6      2.35  0.0456
## 4 Residuals       21220.    48     NA    NA
```

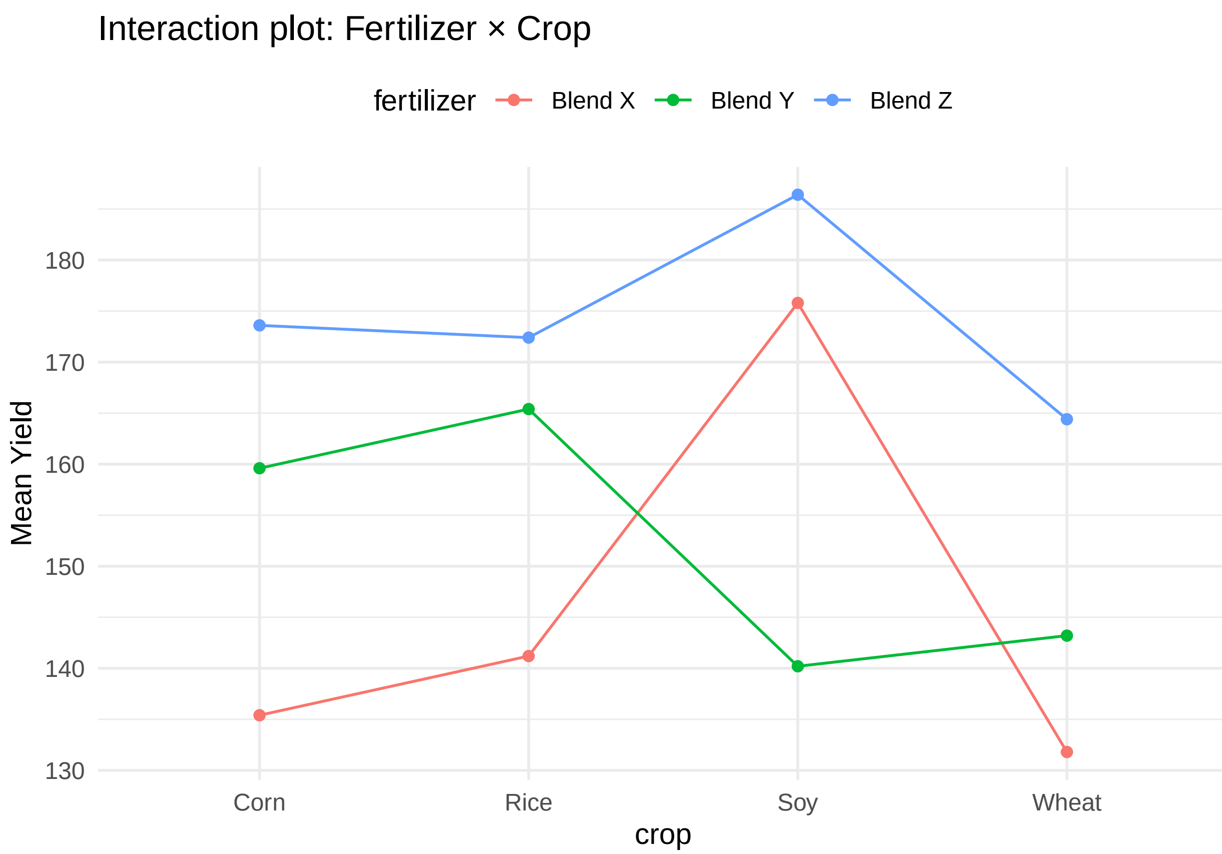
```r
# partial eta2
eta2_table <- effectsize::eta_squared(lm1, partial = TRUE)
print(eta2_table)
```

```
## # Effect Size for ANOVA (Type I)
##
## Parameter       | Eta2 (partial) |       95% CI
## -----------------------------------------------
## fertilizer      |           0.29 | [0.11, 1.00]
## crop            |           0.14 | [0.00, 1.00]
## fertilizer:crop |           0.23 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].
```

```r
# --- 5. Interaction plot (JASP-style) --------------------------
means <- dat %>% group_by(crop, fertilizer) %>% summarise(m = mean(yield))
```

```
## `summarise()` has grouped output by 'crop'. You can override using the
## `.groups` argument.
```

```r
ggplot(means, aes(x = crop, y = m, group = fertilizer, color = fertilizer)) +
  geom_point() + geom_line() +
  labs(title = "Interaction plot: Fertilizer × Crop", y = "Mean Yield") +
  theme_minimal() + theme(legend.position = "top")
```




```r
# --- 6. If significant interaction -> simple main effects & post-hocs
# We'll test interaction and then run emmeans simple effects
anova_results <- anova(lm1)
print(anova_results)
```

```
## Analysis of Variance Table
##
## Response: yield
##                 Df  Sum Sq Mean Sq F value    Pr(>F)
## fertilizer       2  8782.9  4391.5  9.9333 0.0002455 ***
## crop             3  3411.7  1137.2  2.5724 0.0649438 .
## fertilizer:crop  6  6225.9  1037.7  2.3471 0.0455555 *
## Residuals       48 21220.4   442.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Use emmeans to get simple effects
emm <- emmeans(lm1, ~ fertilizer * crop)
# Simple effects: fertilizer within each crop
simple_fert <- contrast(emm, interaction = "pairwise", by = "crop", adjust = "bonferroni")
print(simple_fert)
```

```
## crop = Corn:
##  fertilizer_pairwise estimate   SE df t.ratio p.value
##  Blend X - Blend Y      -24.2 13.3 48  -1.820  0.2251
##  Blend X - Blend Z      -38.2 13.3 48  -2.873  0.0181
##  Blend Y - Blend Z      -14.0 13.3 48  -1.053  0.8931
```

```
##
## crop = Rice:
##  fertilizer_pairwise estimate   SE df t.ratio p.value
##  Blend X - Blend Y      -24.2 13.3 48  -1.820  0.2251
##  Blend X - Blend Z      -31.2 13.3 48  -2.346  0.0694
##  Blend Y - Blend Z       -7.0 13.3 48  -0.526  1.0000
##
## crop = Soy:
##  fertilizer_pairwise estimate   SE df t.ratio p.value
##  Blend X - Blend Y       35.6 13.3 48   2.677  0.0304
##  Blend X - Blend Z      -10.6 13.3 48  -0.797  1.0000
##  Blend Y - Blend Z      -46.2 13.3 48  -3.474  0.0033
##
## crop = Wheat:
##  fertilizer_pairwise estimate   SE df t.ratio p.value
##  Blend X - Blend Y      -11.4 13.3 48  -0.857  1.0000
##  Blend X - Blend Z      -32.6 13.3 48  -2.451  0.0538
##  Blend Y - Blend Z      -21.2 13.3 48  -1.594  0.3523
##
## P value adjustment: bonferroni method for 3 tests
```

```r
# Simple effects: crop within fertilizer
simple_crop <- contrast(emm, interaction = "pairwise", by = "fertilizer", adjust = "bonferroni")
print(simple_crop)
```

```
## fertilizer = Blend X:
##  crop_pairwise estimate   SE df t.ratio p.value
##  Corn - Rice       -5.8 13.3 48  -0.436  1.0000
##  Corn - Soy       -40.4 13.3 48  -3.038  0.0231
##  Corn - Wheat       3.6 13.3 48   0.271  1.0000
##  Rice - Soy       -34.6 13.3 48  -2.602  0.0738
##  Rice - Wheat       9.4 13.3 48   0.707  1.0000
##  Soy - Wheat       44.0 13.3 48   3.309  0.0107
##
## fertilizer = Blend Y:
##  crop_pairwise estimate   SE df t.ratio p.value
##  Corn - Rice       -5.8 13.3 48  -0.436  1.0000
##  Corn - Soy        19.4 13.3 48   1.459  0.9067
##  Corn - Wheat      16.4 13.3 48   1.233  1.0000
##  Rice - Soy        25.2 13.3 48   1.895  0.3847
##  Rice - Wheat      22.2 13.3 48   1.669  0.6093
##  Soy - Wheat       -3.0 13.3 48  -0.226  1.0000
##
## fertilizer = Blend Z:
##  crop_pairwise estimate   SE df t.ratio p.value
##  Corn - Rice        1.2 13.3 48   0.090  1.0000
##  Corn - Soy       -12.8 13.3 48  -0.963  1.0000
##  Corn - Wheat       9.2 13.3 48   0.692  1.0000
##  Rice - Soy       -14.0 13.3 48  -1.053  1.0000
##  Rice - Wheat       8.0 13.3 48   0.602  1.0000
##  Soy - Wheat       22.0 13.3 48   1.654  0.6274
##
## P value adjustment: bonferroni method for 6 tests
```

```r
# Also full Tukey HSD for each main factor

tuk_fert <- emmeans(lm1, pairwise ~ fertilizer, adjust = "tukey")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```r
tuk_crop <- emmeans(lm1, pairwise ~ crop, adjust = "tukey")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```r
print(tuk_fert)
```

```
## $emmeans
##  fertilizer emmean  SE df lower.CL upper.CL
##  Blend X       146 4.7 48      137      156
##  Blend Y       152 4.7 48      143      162
##  Blend Z       174 4.7 48      165      184
##
## Results are averaged over the levels of: crop
## Confidence level used: 0.95
##
## $contrasts
##  contrast          estimate   SE df t.ratio p.value
##  Blend X - Blend Y    -6.05 6.65 48  -0.910  0.6367
##  Blend X - Blend Z   -28.15 6.65 48  -4.234  0.0003
##  Blend Y - Blend Z   -22.10 6.65 48  -3.324  0.0048
##
## Results are averaged over the levels of: crop
## P value adjustment: tukey method for comparing a family of 3 estimates
```

```r
print(tuk_crop)
```

```
## $emmeans
##  crop  emmean   SE df lower.CL upper.CL
##  Corn     156 5.43 48      145      167
##  Rice     160 5.43 48      149      171
##  Soy      167 5.43 48      157      178
##  Wheat    146 5.43 48      136      157
##
## Results are averaged over the levels of: fertilizer
## Confidence level used: 0.95
##
## $contrasts
##  contrast     estimate   SE df t.ratio p.value
##  Corn - Rice     -3.47 7.68 48  -0.452  0.9690
##  Corn - Soy     -11.27 7.68 48  -1.467  0.4647
##  Corn - Wheat     9.73 7.68 48   1.268  0.5876
##  Rice - Soy      -7.80 7.68 48  -1.016  0.7410
##  Rice - Wheat    13.20 7.68 48   1.719  0.3251
##  Soy - Wheat     21.00 7.68 48   2.735  0.0419
##
## Results are averaged over the levels of: fertilizer
## P value adjustment: tukey method for comparing a family of 4 estimates
```

2B ANOVA REPORTING

A two-way ANOVA showed a statistically significant interaction between fertilizer blend and crop type, indicating that the effectiveness of each fertilizer depends on the crop.

A two-way ANOVA was performed to examine the effects of fertilizer blend (Blend X, Blend Y, Blend Z) and crop type (Wheat, Corn, Soy, Rice) on crop yield. Visual inspection of boxplots indicated no extreme outliers among the 12 fertilizer × crop combinations (see boxplot on page 4) . Residual analysis confirmed that the assumption of normality was met, as shown by the Shapiro–Wilk test, $W = 0.980$, $p = .440$ (page 5) . Homogeneity of variance was assessed using Levene's test (median centered), which was non-significant, $F(11, 48) = 0.675$, $p = .755$, indicating equal variances across groups (page 5). Thus, ANOVA assumptions were satisfied.

The two-way ANOVA (Type II) revealed a significant main effect of fertilizer blend, $F(2, 48) = 9.93$, $p < .001$, partial $^2 = .29$ (pages 5–6), indicating that average yields differed across fertilizer types. The main effect of crop was not statistically significant, $F(3, 48) = 2.57$, $p = .065$, partial $^2 = .14$, suggesting that when averaged across fertilizer, crops did not significantly differ in yield.

Importantly, there was a significant fertilizer × crop interaction, $F(6, 48) = 2.35$, $p = .046$, partial $^2 = .23$ (pages 5–6) . This indicates that the effect of fertilizer blend on yield differed depending on the crop type. Because the interaction was significant, simple main effects were examined using Bonferroni-adjusted pairwise comparisons.

Simple Main Effects of Fertilizer Within Each Crop

(From page 7 of the PDF) Corn: Blend Z produced significantly higher yields than Blend X ($p = .018$), whereas the differences between Blend X vs Blend Y and Blend Y vs Blend Z were not significant.

Rice: No fertilizer pair reached significance after adjustment.

Soy: Blend X produced significantly higher yields than Blend Y ($p = .030$), and Blend Z produced significantly higher yields than Blend Y ($p = .003$).

Wheat: No pairwise fertilizer differences were statistically significant.

Simple Main Effects of Crop Within Each Fertilizer

(From page 7 of the PDF)

Blend X: Soy yielded significantly more than Wheat ($p = .011$) and Corn ($p = .023$).

Blend Y: No crop comparisons reached significance.

Blend Z: No crop comparisons reached significance after adjustment.

Tukey HSD Main-Effects Post Hocs

(From page 8 of the PDF)

Fertilizer:

Blend Z > Blend X (mean difference = 28.15, $p < .001$)

Blend Z > Blend Y (mean difference = 22.10, $p = .005$)

Blend X vs Blend Y not significant

Crop: Only Soy > Wheat reached significance ($p = .042$); all other crop pairs were non-significant.