Three-Way ANOVA and Post-Hoc Comparisons

John Robert Torres

2025-04-30

1 Load required and new packages

```
if (!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

```
library(pacman)
pacman::p_load("here", "glue", "crayon","readxl", "writexl", "dplyr", "tidyr", "rstatix")
pacman::p_load("purrr", "car")

`%ni%` = Negate(`%in%`)
```

2 Set data paths and details

```
main.path = here::here()
data.path = file.path(main.path, "02 Data")
output.path = file.path(main.path, "04 Outputs")

file.name = "042925_Marco_Final_Data.xlsx"
sheet.name = "FINAL"
client.surname = "MARCO"
output.name = glue::glue(paste0(format(Sys.Date(), "%m%d%y"), "_{client.surname}.xlsx"))
```

3 Load dataset

```
df.part1 = readxl::read_excel(file.path(data.path, file.name),
                             sheet = "PART1")

df.part2 = readxl::read_excel(file.path(data.path, file.name),
                             sheet = "PART2")

scale  = 100000
```

4 Process data

```
df.proc1 = df.part1 %>%
  dplyr::select(-variety) %>%
  tidyr::pivot_longer(-c(digestion), names_to = "metric", values_to = "values") %>%
  dplyr::mutate(values = values / scale)

df.proc2 = df.part2 %>%
  tidyr::pivot_longer(c(cells, clumps), names_to = "metric", values_to = "values") %>%
  dplyr::mutate(values = values / scale)
```

5 Implement methodology

5.1 Week 0

```
w0.test = df.proc1 %>%
  dplyr::group_by(metric) %>%
  rstatix::t_test(values ~ digestion)

w0.means = df.proc1 %>%
  dplyr::group_by(metric, digestion) %>%
  dplyr::summarise(mean = mean(values),
                   sd = sd(values)) %>%
  dplyr::ungroup()
```

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

5.2 Three-Way ANOVA

```
test.3w = df.proc2 %>%
  dplyr::group_by(week, metric) %>%
 rstatix::anova_test(values ~ digestion * variety * treatment) %>%
  dplyr::ungroup()

test.means = df.proc2 %>%
  dplyr::group_by(week, digestion, variety, treatment, metric) %>%
  dplyr::summarise(mean = mean(values),
                   sd = sd(values)) %>%
  dplyr::ungroup()
```

```
## `summarise()` has grouped output by 'week', 'digestion', 'variety',
## 'treatment'. You can override using the `.groups` argument.
```

```r
# Load necessary libraries
library(dplyr)
library(car)
library(purrr)

# Split data by week
df_split <- df.part2 %>% group_split(week)

# Apply MANOVA separately for each week
test.manova <- map(df_split, ~ car::Manova(lm(cbind(cells, clumps) ~ digestion * variety * treat
ment, data = .)))

# View results for the first group
summary(test.manova[[1]], test = "Pillai")
```

```
## 
## Type II MANOVA Tests:
## 
## Sum of squares and products for error:
##               cells       clumps
## cells  1.099186e+14 2.801103e+13
## clumps 2.801103e+13 1.056840e+13
## 
## -------------------------------------------
## 
## Term: digestion
## 
## Sum of squares and products for the hypothesis:
##               cells       clumps
## cells  8336111342593 -267766842593
## clumps -267766842593    8601022593
## 
## Multivariate Test: digestion
##           Df test stat approx F num Df den Df      Pr(>F)
## digestion  1 0.2162702 12.27977      2     89 1.9517e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## -------------------------------------------
## 
## Term: variety
## 
## Sum of squares and products for the hypothesis:
##               cells       clumps
## cells  1247144907407 -482948425926
## clumps -482948425926  529453822963
## 
## Multivariate Test: variety
##         Df test stat approx F num Df den Df      Pr(>F)
## variety  2 0.2086278 5.240815      4    180 0.00051353 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## -------------------------------------------
## 
## Term: treatment
## 
## Sum of squares and products for the hypothesis:
##              cells      clumps
## cells  8.87670e+12 648001018519
## clumps 6.48001e+11 163467156296
## 
## Multivariate Test: treatment
##           Df test stat approx F num Df den Df   Pr(>F)
## treatment  2 0.1709618  4.20619      4    180 0.002796 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## 
## ------------------------------------------
## 
## Term: digestion:variety
## 
## Sum of squares and products for the hypothesis:
##               cells        clumps
## cells   4.339814e+12 849619592593
## clumps 8.496196e+11 194796711852
## 
## Multivariate Test: digestion:variety
##                    Df  test stat approx F num Df den Df  Pr(>F)
## digestion:variety   2 0.05021135 1.158849      4    180 0.33062
## 
## ------------------------------------------
## 
## Term: digestion:treatment
## 
## Sum of squares and products for the hypothesis:
##               cells       clumps
## cells   1.402583e+13 2.15206e+12
## clumps 2.152060e+12 3.66570e+11
## 
## Multivariate Test: digestion:treatment
##                      Df test stat approx F num Df den Df    Pr(>F)
## digestion:treatment   2 0.1548334 3.776083      4    180 0.0056471 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## ------------------------------------------
## 
## Term: variety:treatment
## 
## Sum of squares and products for the hypothesis:
##               cells        clumps
## cells   8.814291e+12 2.177632e+12
## clumps 2.177632e+12 7.958944e+11
## 
## Multivariate Test: variety:treatment
##                    Df test stat approx F num Df den Df    Pr(>F)
## variety:treatment   4 0.1442768 1.749306      8    180 0.089948 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## ------------------------------------------
## 
## Term: digestion:variety:treatment
## 
## Sum of squares and products for the hypothesis:
##               cells        clumps
## cells   4.999777e+12 1.386270e+12
## clumps 1.386270e+12 4.889804e+11
```

```
## 
## Multivariate Test: digestion:variety:treatment
##                                 Df  test stat  approx F num Df den Df  Pr(>F)
## digestion:variety:treatment  4 0.07373306 0.8612481        8     180 0.55037
```

5.2.1 Week 1 Cells

```
df = df.proc2 %>%
  dplyr::filter(week == 1 & metric == "cells")

w1.cells.dt = df %>%
  rstatix::anova_test(values ~ digestion * treatment)

w1.cells.dxt = df %>%
  dplyr::select(digestion, treatment, values) %>%
  dplyr::group_by(digestion) %>%
  rstatix::pairwise_t_test(values ~ treatment,
                           p.adjust.method = "bonferroni")

w1.cells.txd = df %>%
  dplyr::select(digestion, treatment, values) %>%
  dplyr::group_by(treatment) %>%
  rstatix::pairwise_t_test(values ~ digestion,
                           p.adjust.method = "bonferroni")
```

5.2.3 Week 2 Clumps

```
df = df.proc2 %>%
  dplyr::filter(week == 2 & metric == "clumps")

w2.clumps.dt = df %>%
  rstatix::anova_test(values ~ digestion * treatment)

w2.clumps.dxt = df %>%
  dplyr::group_by(digestion) %>%
  rstatix::pairwise_t_test(values ~ treatment,
                           p.adjust.method = "bonferroni")

w2.clumps.txd = df %>%
  dplyr::group_by(treatment) %>%
  rstatix::pairwise_t_test(values ~ digestion,
                           p.adjust.method = "bonferroni")

w2.clumps.d = df %>%
  rstatix::pairwise_t_test(values ~ digestion,
                           p.adjust.method = "bonferroni")

w2.clumps.dv = df %>%
  rstatix::anova_test(values ~ digestion * variety)

w2.clumps.dxv = df %>%
  dplyr::group_by(digestion) %>%
  rstatix::pairwise_t_test(values ~ variety,
                           p.adjust.method = "bonferroni")

w2.clumps.vxd = df %>%
  dplyr::group_by(variety) %>%
  rstatix::pairwise_t_test(values ~ digestion,
                           p.adjust.method = "bonferroni")
```

5.2.4 Week 3 Cells

```
df = df.proc2 %>%
  dplyr::filter(week == 3 & metric == "cells")

w3.cells.dt = df %>%
  rstatix::anova_test(values ~ digestion * treatment)

w3.cells.dxt = df %>%
  dplyr::group_by(digestion) %>%
  rstatix::pairwise_t_test(values ~ treatment,
                           p.adjust.method = "bonferroni")

w3.cells.txd = df %>%
  dplyr::group_by(treatment) %>%
  rstatix::pairwise_t_test(values ~ digestion,
                           p.adjust.method = "bonferroni")
```

5.2.5 Week 3 Clumps

```
df = df.proc2 %>%
  dplyr::filter(week == 3 & metric == "clumps")

w3.clumps.dv = df %>%
  rstatix::anova_test(values ~ digestion * variety)

w3.clumps.dxv = df %>%
  dplyr::group_by(digestion) %>%
  rstatix::pairwise_t_test(values ~ variety,
                           p.adjust.method = "bonferroni")

w3.clumps.vxd = df %>%
  dplyr::group_by(variety) %>%
  rstatix::pairwise_t_test(values ~ digestion,
                           p.adjust.method = "bonferroni")
```

5.2.6 Week 4 Cells

```
df = df.proc2 %>%
  dplyr::filter(week == 4 & metric == "cells")

w4.cells.vt = df %>%
  rstatix::anova_test(values ~ variety * treatment)

w4.cells.vxt = df %>%
  dplyr::group_by(variety) %>%
  rstatix::pairwise_t_test(values ~ treatment,
                           p.adjust.method = "bonferroni")

w4.cells.txv = df %>%
  dplyr::group_by(treatment) %>%
  rstatix::pairwise_t_test(values ~ variety,
                           p.adjust.method = "bonferroni")

w4.cells.dt = df %>%
  rstatix::anova_test(values ~ digestion * treatment)

w4.cells.dxt = df %>%
  dplyr::group_by(digestion) %>%
  rstatix::pairwise_t_test(values ~ treatment,
                           p.adjust.method = "bonferroni")

w4.cells.txd = df %>%
  dplyr::group_by(treatment) %>%
  rstatix::pairwise_t_test(values ~ digestion,
                           p.adjust.method = "bonferroni")

w4.cells.dv = df %>%
  rstatix::anova_test(values ~ digestion * variety)

w4.cells.dxv = df %>%
  dplyr::group_by(digestion) %>%
  rstatix::pairwise_t_test(values ~ variety,
                           p.adjust.method = "bonferroni")

w4.cells.vxd = df %>%
  dplyr::group_by(variety) %>%
  rstatix::pairwise_t_test(values ~ digestion,
                           p.adjust.method = "bonferroni")
```

5.2.7 Week 4 Clumps

```
df = df.proc2 %>%
  dplyr::filter(week == 4 & metric == "clumps")

w4.clumps.dv = df %>%
  rstatix::anova_test(values ~ digestion * variety)

w4.clumps.dxv = df %>%
  dplyr::group_by(digestion) %>%
  rstatix::pairwise_t_test(values ~ variety,
                           p.adjust.method = "bonferroni")

w4.clumps.vxd = df %>%
  dplyr::group_by(variety) %>%
  rstatix::pairwise_t_test(values ~ digestion,
                           p.adjust.method = "bonferroni")
```

6 Export necessary data

```r
export.list = list(part1 = df.proc1,
                   part2 = df.proc2,
                   w0.test = w0.test,
                   w0.means = w0.means,
                   `3w.test` = test.3w,
                   w1.cells.dt = w1.cells.dt,
                   w1.cells.dxt = w1.cells.dxt,
                   w1.cells.txd = w1.cells.txd,
                   w2.clumps.dt = w2.clumps.dt,
                   w2.clumps.dxt = w2.clumps.dxt,
                   w2.clumps.txd = w2.clumps.txd,
                   w2.clumps.dv = w2.clumps.dv,
                   w2.clumps.dxv = w2.clumps.dxv,
                   w2.clumps.vxd = w2.clumps.vxd,
                   w3.cells.dt = w3.cells.dt,
                   w3.cells.dxt = w3.cells.dxt,
                   w3.cells.txd = w3.cells.txd,
                   w3.clumps.dv = w3.clumps.dv,
                   w3.clumps.dxv = w3.clumps.dxv,
                   w3.clumps.vxd = w3.clumps.vxd,
                   w4.cells.dt = w4.cells.dt,
                   w4.cells.dxt = w4.cells.dxt,
                   w4.cells.txd = w4.cells.txd,
                   w4.cells.dv = w4.cells.dv,
                   w4.cells.dxv = w4.cells.dxv,
                   w4.cells.vxd = w4.cells.vxd,
                   w4.cells.vt = w4.cells.vt,
                   w4.cells.vxt = w4.cells.vxt,
                   w4.cells.txv = w4.cells.txv,
                   w4.clumps.dv = w4.clumps.dv,
                   w4.clumps.dxv = w4.clumps.dxv,
                   w4.clumps.vxd = w4.clumps.vxd)

if(length(export.list) != 0){
  if (!file.exists(file.path(output.path, output.name))) {
    writexl::write_xlsx(export.list, file.path(output.path, output.name))
    cat(crayon::green("File successfully written."))
  } else {
    cat(crayon::red(glue::glue("Filename already used: {output.name}")))
    overwrite = readline(prompt = "Overwrite (1 for Yes, 0 for No): ")
    if (overwrite == "1") {
      writexl::write_xlsx(export.list, file.path(output.path, output.name))
      cat(crayon::green("File successfully overwritten"))
    } else {
      cat(crayon::red("File not overwritten"))
    }
  }
}
```

```
## File successfully written.
```