

Two-way randomized complete block design

Paul Schmidt

2023-11-07

Two-way ANOVA & pairwise comparison post hoc tests in a randomized complete block design.

Table of contents

1	Data	2
1.1	Import	2
1.2	Format	2
1.3	Explore	3
2	Model	9
3	ANOVA	10
4	Mean comparison	10

```
# (install &) load packages
pacman::p_load(
  conflicted,
  desplot,
  emmeans,
  ggtext,
  MetBrewer,
  multcomp,
  multcompView,
  tidyverse)

# handle function conflicts
conflicts_prefer(dplyr::filter)
conflicts_prefer(dplyr::select)
```

1 Data

This data is a slightly modified version of a dataset originally published in Gomez and Gomez (1984) from a yield (kg/ha) trial with 4 genotypes (**G**) and 6 nitrogen levels (**N**), leading to 24 treatment level combinations. The data set here has 3 complete replicates (**rep**) and is laid out as a randomized complete block design (RCBD).

1.1 Import

```
# data is available online:
path <- "https://raw.githubusercontent.com/SchmidtPaul/dsfair_quarto/master/data/riceRCBD."

dat <- read_csv(path) # use path from above
dat

# A tibble: 72 x 6
   row   col rep   N     G   yield
  <dbl> <dbl> <chr> <chr> <chr> <dbl>
1     2     6 rep1 Goomba Simba  4520
2     3     4 rep1 Koopa  Simba  5598
3     2     3 rep1 Toad   Simba  6192
4     1     1 rep1 Peach Simba  8542
5     2     1 rep1 Diddy Simba  5806
6     3     1 rep1 Yoshi Simba  7470
7     4     5 rep1 Goomba Nala   4034
8     4     1 rep1 Koopa  Nala   6682
9     3     2 rep1 Toad   Nala   6869
10    1     2 rep1 Peach  Nala   6318
# i 62 more rows
```

1.2 Format

Before anything, the columns **rep**, **N** and **G** should be encoded as factors, since R by default encoded them as character.

```
dat <- dat %>%
  mutate(across(c(rep, N, G), ~ as.factor(.x)))
```

1.3 Explore

We make use of `dlookr::describe()` to conveniently obtain descriptive summary tables. Here, we get can summarize per nitrogen level, per genotype and also per nitrogen-genotype-combination.

```
dat %>%
  group_by(N) %>%
  dlookr::describe(yield) %>%
  select(2:sd) %>%
  arrange(desc(mean))
```

```
# A tibble: 6 x 5
  N           n    na mean    sd
<fct> <int> <int> <dbl> <dbl>
1 Diddy     12     0 5866.  832.
2 Toad      12     0 5864. 1434.
3 Yoshi     12     0 5812  2349.
4 Peach     12     0 5797. 2660.
5 Koopa     12     0 5478.  657.
6 Goomba    12     0 4054.  672.
```

```
dat %>%
  group_by(G) %>%
  dlookr::describe(yield) %>%
  select(2:sd) %>%
  arrange(desc(mean))
```

```
# A tibble: 4 x 5
  G           n    na mean    sd
<fct> <int> <int> <dbl> <dbl>
1 Simba     18     0 6554. 1475.
2 Nala      18     0 6156. 1078.
3 Timon     18     0 5563. 1269.
4 Pumba     18     0 3642. 1434.
```

```
dat %>%
  group_by(N, G) %>%
  dlookr::describe(yield) %>%
  select(2:sd) %>%
```

```
arrange(desc(mean)) %>%
print(n=Inf)
```

```
# A tibble: 24 x 6
  N      G      n      na mean      sd
  <fct> <fct> <int> <int> <dbl> <dbl>
1 Peach Simba     3      0 8701.  270.
2 Yoshi Simba     3      0 7563.   86.9
3 Yoshi Nala      3      0 6951.  808.
4 Toad  Nala      3      0 6895  166.
5 Toad  Simba     3      0 6733.  490.
6 Yoshi Timon     3      0 6687.  496.
7 Peach Nala      3      0 6540.  936.
8 Diddy Simba     3      0 6400   523.
9 Diddy Nala      3      0 6259   499.
10 Peach Timon     3      0 6065. 1097.
11 Toad  Timon     3      0 6014   515.
12 Diddy Timon     3      0 5994   101.
13 Koopa Nala      3      0 5982   684.
14 Koopa Simba     3      0 5672   458.
15 Koopa Timon     3      0 5443.   589.
16 Koopa Pumba     3      0 4816   506.
17 Diddy Pumba     3      0 4812   963.
18 Goomba Pumba     3      0 4481.   463.
19 Goomba Nala      3      0 4306   646.
20 Goomba Simba     3      0 4253.   248.
21 Toad  Pumba     3      0 3816  1311.
22 Goomba Timon     3      0 3177.   453.
23 Yoshi Pumba     3      0 2047.   703.
24 Peach Pumba     3      0 1881.   407.
```

Additionally, we can decide to plot our data. One way to deal with the combination of two factors would be to use [panels/facets in ggplot2](#).

Note that we here define a custom set of colors for the Nitrogen levels that will be used throughout this chapter.

```
Ncolors <- met.brewer("VanGogh2", 6) %>%
  as.vector() %>%
  set_names(levels(dat$N))
```

```

ggplot(data = dat) +
  aes(y = yield, x = N, color = N) +
  facet_wrap(~G, labeller = label_both) +
  stat_summary(
    fun = mean,
    colour = "grey",
    geom = "line",
    linetype = "dotted",
    group = 1
  ) +
  geom_point() +
  scale_x_discrete(
    name = "Nitrogen"
  ) +
  scale_y_continuous(
    name = "Yield",
    limits = c(0, NA),
    expand = expansion(mult = c(0, 0.1))
  ) +
  scale_color_manual(
    values = Ncolors,
    guide = "none"
  ) +
  theme_bw() +
  theme(axis.text.x = element_text(
    angle = 45,
    hjust = 1,
    vjust = 1
  ))

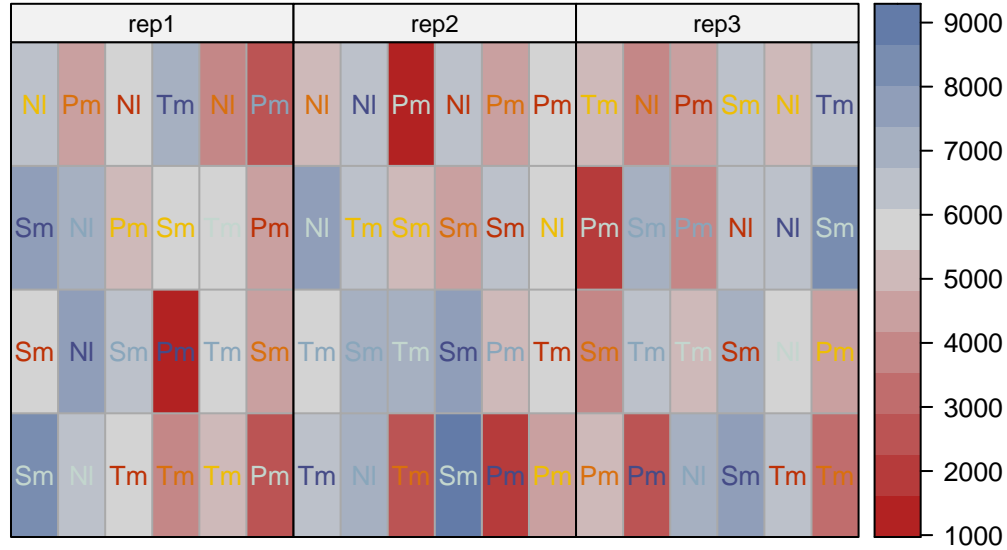
```



Finally, since this is an experiment that was laid with a certain experimental design (= a randomized complete block design; RCBD) - it makes sense to also get a field plan. This can be done via `desplot()` from `{desplot}`.

```
desplot(
  data = dat,
  form = rep ~ col + row | rep, # fill color per rep, headers per rep
  col.regions = c("white", "grey95", "grey90"),
  text = G, # genotype names per plot
  cex = 0.8, # genotype names: font size
  shorten = "abb", # genotype names: abbreviate
  col = N, # color of genotype names for each N-level
  col.text = Ncolors, # use custom colors from above
  out1 = col, out1.gpar = list(col = "darkgrey"), # lines between columns
  out2 = row, out2.gpar = list(col = "darkgrey"), # lines between rows
  main = "Field layout", # plot title
  show.key = TRUE, # show legend
  key.cex = 0.7 # legend font size
)
```


Yield per plot



```

repcolors <- c(met.brewer("VanGogh3", 1),
               met.brewer("Hokusai2", 1),
               met.brewer("OKeeffe2", 1)) %>%
  as.vector() %>%
  set_names(levels(dat$rep))

desplot(
  data = dat,
  form = rep ~ col + row | rep, # fill color per rep, headers per rep
  col.regions = repcolors,
  out1 = col, out1.gpar = list(col = "darkgrey"), # lines between columns
  out2 = row, out2.gpar = list(col = "darkgrey"), # lines between rows
  main = "Experimental design focus", # plot title
  show.key = FALSE # don't show legend
)

```


Experimental design focus

rep1						rep2						rep3					

2 Model

Finally, we can decide to fit a linear model with yield as the response variable. In this example it makes sense to mentally group the effects in our model as either *design effects* or *treatment effects*. The treatments here are the genotypes **G** and the nitrogen levels **N** which we will include in the model as main effects, but also via their interaction effect **N:G**. Regarding the design, the model needs to contain a block (**rep**) effect.

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

 Model assumptions met? (click to show)

It would be at this moment (i.e. after fitting the model and before running the ANOVA), that you should check whether the model assumptions are met. Find out more in the [summary article “Model Diagnostics”](#)

3 ANOVA

Based on our model, we can then conduct an ANOVA:

```
ANOVA <- anova(mod)
ANOVA

Analysis of Variance Table

Response: yield
          Df    Sum Sq   Mean Sq F value    Pr(>F)
N           5 30480453   6096091 15.4677 6.509e-09 ***
G           3 89885035 29961678 76.0221 < 2.2e-16 ***
rep         2 1084820    542410  1.3763  0.2627
N:G        15 69378044  4625203 11.7356 4.472e-11 ***
Residuals  46 18129432    394118
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Accordingly, the ANOVA's F-test found the nitrogen-genotype-interaction to be statistically different ($p < .001^{***}$).

4 Mean comparison

Besides an ANOVA, one may also want to compare adjusted yield means between cultivars via post hoc tests (t-test, Tukey test etc.). Especially because of the results of this ANOVA, we should compare means for all N:G interactions and **not** for the N and/or G main effects. When doing so, we still have multiple options to choose from. I here decide to compare all genotype means per nitrogen

```
mean_comp <- mod %>%
  emmeans(specs = ~ N|G) %>% # adj. mean per cultivar
  cld(adjust = "Tukey", Letters = letters) # compact letter display (CLD)

mean_comp

G = Nala:
      N      emmean   SE df lower.CL upper.CL .group
Goomba  4306 362 46      3310      5302    a
```

Koopa	5982	362	46	4986	6978	b
Diddy	6259	362	46	5263	7255	b
Peach	6540	362	46	5544	7537	b
Toad	6895	362	46	5899	7891	b
Yoshi	6951	362	46	5954	7947	b

G = Pumba:

N	emmean	SE	df	lower.CL	upper.CL	.group
Peach	1881	362	46	884	2877	a
Yoshi	2047	362	46	1050	3043	a
Toad	3816	362	46	2820	4812	b
Goomba	4481	362	46	3485	5478	b
Diddy	4812	362	46	3816	5808	b
Koopa	4816	362	46	3820	5812	b

G = Simba:

N	emmean	SE	df	lower.CL	upper.CL	.group
Goomba	4253	362	46	3256	5249	a
Koopa	5672	362	46	4676	6668	ab
Diddy	6400	362	46	5404	7396	bc
Toad	6733	362	46	5736	7729	bc
Yoshi	7563	362	46	6567	8560	cd
Peach	8701	362	46	7704	9697	d

G = Timon:

N	emmean	SE	df	lower.CL	upper.CL	.group
Goomba	3177	362	46	2181	4174	a
Koopa	5443	362	46	4446	6439	b
Diddy	5994	362	46	4998	6990	b
Toad	6014	362	46	5018	7010	b
Peach	6065	362	46	5069	7062	b
Yoshi	6687	362	46	5691	7684	b

Results are averaged over the levels of: rep

Confidence level used: 0.95

Conf-level adjustment: sidak method for 6 estimates

P value adjustment: tukey method for comparing a family of 6 estimates

significance level used: alpha = 0.05

NOTE: If two or more means share the same grouping symbol,

then we cannot show them to be different.

But we also did not show them to be the same.

Note that if you would like to see the underlying individual contrasts/differences between adjusted means, simply add `details = TRUE` to the `cld()` statement. Furthermore, check out the [Summary Article “Compact Letter Display”](#).

Finally, we can create a plot that displays both the raw data and the results, *i.e.* the comparisons of the adjusted means that are based on the linear model.

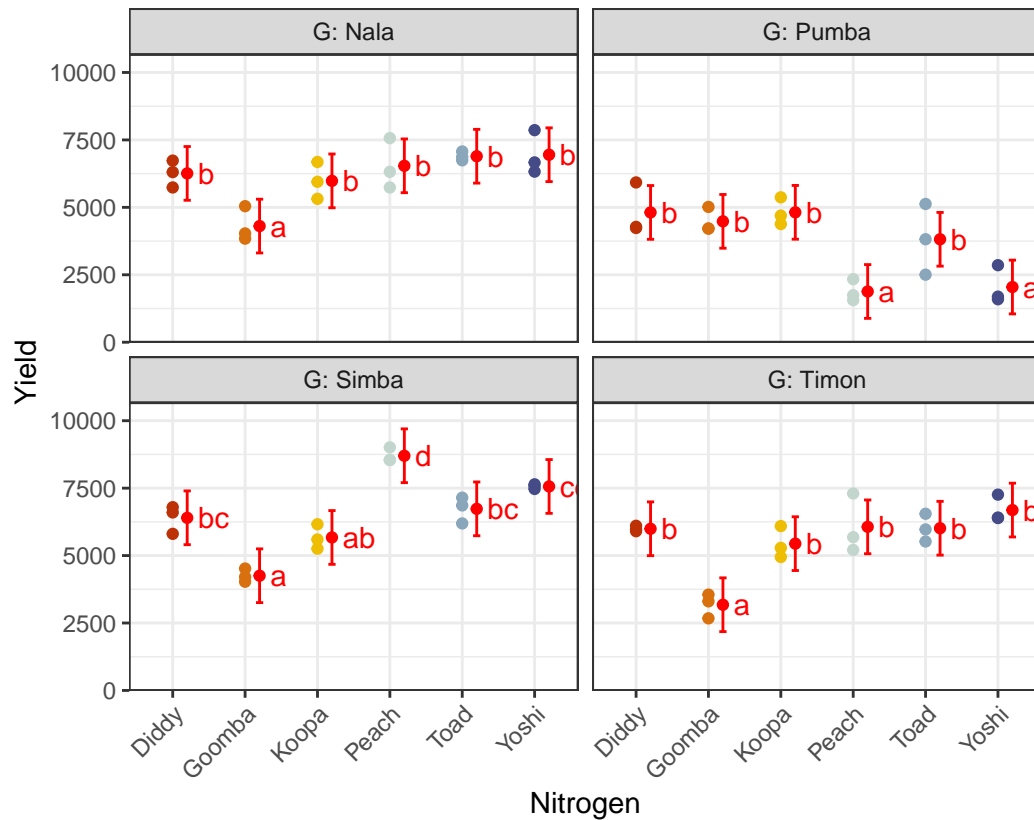
```
my_caption <- "The four facettes represent genotypes Simba, Nala, Timon and Pumba. Black d

ggplot() +
  facet_wrap(~G, labeller = label_both) + # facette per G level
  aes(x = N) +
  # black dots representing the raw data
  geom_point(
    data = dat,
    aes(y = yield, color = N)
  ) +
  # red dots representing the adjusted means
  geom_point(
    data = mean_comp,
    aes(y = emmean),
    color = "red",
    position = position_nudge(x = 0.2)
  ) +
  # red error bars representing the confidence limits of the adjusted means
  geom_errorbar(
    data = mean_comp,
    aes(ymin = lower.CL, ymax = upper.CL),
    color = "red",
    width = 0.1,
    position = position_nudge(x = 0.2)
  ) +
  # red letters
  geom_text(
    data = mean_comp,
    aes(y = emmean, label = str_trim(.group)),
    color = "red",
    position = position_nudge(x = 0.35),
    hjust = 0
  ) +
  scale_x_discrete(
    name = "Nitrogen"
```

```

) +
scale_y_continuous(
  name = "Yield",
  limits = c(0, NA),
  expand = expansion(mult = c(0, 0.1))
) +
scale_color_manual(
  values = Ncolors,
  guide = "none"
) +
theme_bw() +
labs(caption = my_caption) +
theme(
  plot.caption = element_textbox_simple(margin = margin(t = 5)),
  plot.caption.position = "plot",
  axis.text.x = element_text(
    angle = 45,
    hjust = 1,
    vjust = 1
  )
)

```



The four facettes represent genotypes Simba, Nala, Timon and Pumba. Black dots represent raw data. Red dots and error bars represent adjusted means with 95% confidence limits per cultivar. For each genotype separately, means followed by a common letter are not significantly different according to the Tukey-test.

Gomez, Kwanchai A, and Arturo A Gomez. 1984. *Statistical Procedures for Agricultural Research*. 2nd ed. An International Rice Research Institute Book. Nashville, TN: John Wiley & Sons.