Homework 6 - AGST 5014

Igor Kuivjogi Fernandes and Ashmita Upadhyay

2023-04-03

1. With the data set below, answer: (another resource on IBD: Chapter 14 - http://users.stat.umn.edu/~gary/book/fcdae.pdf) a) Is this a balanced or partially balanced design?

```
data("taste", package = "daewr")
str(taste)
```

```
## 'data.frame':    24 obs. of  3 variables:
##  $ panelist: Factor w/ 12 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ recipe  : Factor w/ 4 levels "A","B","C","D": 1 1 1 2 2 3 1 1 1 2 ...
##  $ score   : num  5 7 5 6 6 8 6 5 4 7 ...
```

```
table(taste$panelist)
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12
##  2  2  2  2  2  2  2  2  2  2  2  2
```

We have 12 different panelists, and each one tasted recipes 2 times.
Each panelist is a block.

```
table(taste$recipe)
```

```
##
## A B C D
## 6 6 6 6
```

We have 4 different recipes, and each one was tasted 6 times.

```
with(taste, table(panelist, recipe))
```

```
##         recipe
## panelist A B C D
##        1  1 1 0 0
##        2  1 0 1 0
##        3  1 0 0 1
##        4  0 1 1 0
##        5  0 1 0 1
##        6  0 0 1 1
##        7  1 1 0 0
```

```
##       8  1 0 1 0
##       9  1 0 0 1
##      10  0 1 1 0
##      11  0 1 0 1
##      12  0 0 1 1
```

All panelist just proved 2 (from 4) recipes, so this is a **incomplete block design**.
Now we have to check whether is balanced or partially balanced.

We already know each recipe occurred 6 times, but how many times each treatment pair within blocks occurred?
If we check all possible pair combination (AB, AC, AD, BC, BD, CD) within blocks, all of them occurred two times, $\lambda = 2$. This means we have a Balanced Incomplete Block (BIB) design.

    b) Run both, the intra-block and inter-block analysis on it.
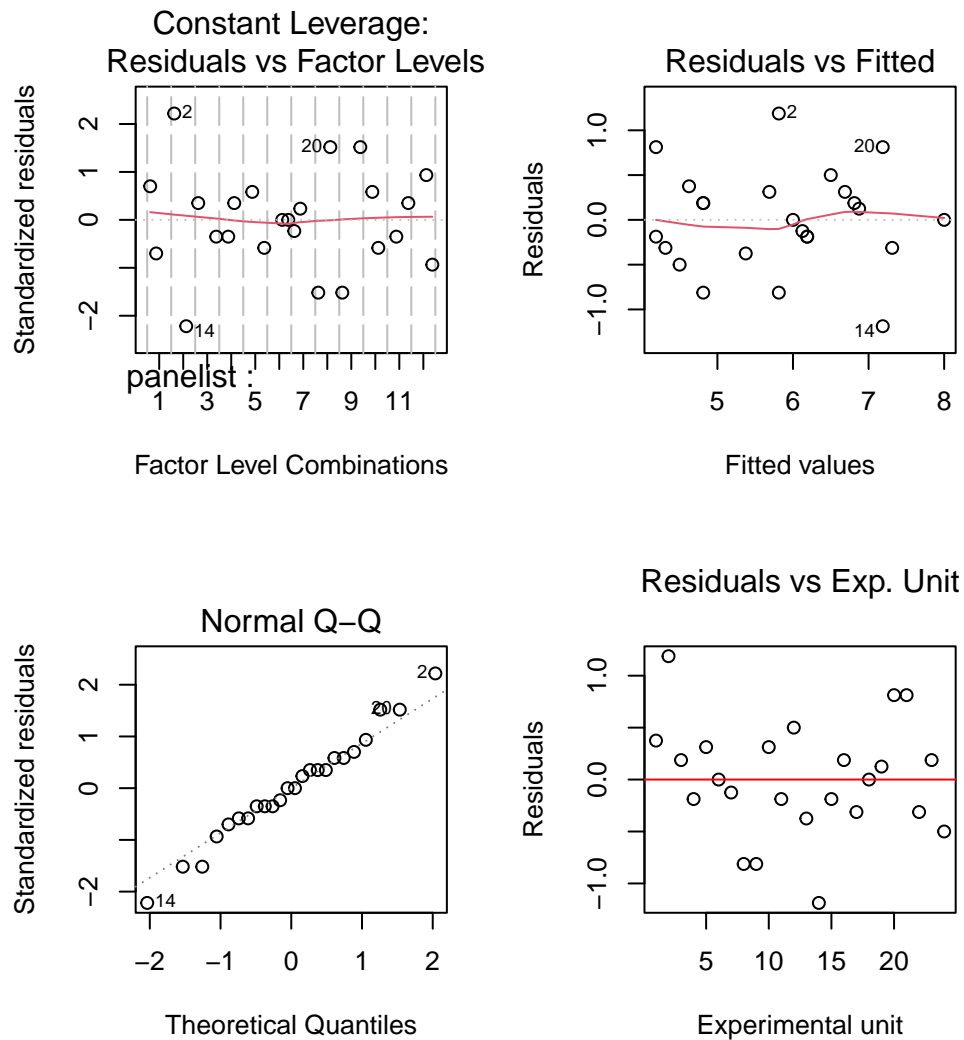
For intra-block analysis, we treat block as fixed.
As we have an incomplete block design, better to use Type III sum of squares.

```r
fit1 <- lm(score ~ panelist + recipe, data = taste, contrasts = list(recipe = contr.sum))
car::Anova(fit1, type = 'III')
```

```
## Anova Table (Type III tests)
##
## Response: score
##              Sum Sq Df F value    Pr(>F)
## (Intercept) 44.444  1 58.1818 3.232e-05 ***
## panelist     7.458 11  0.8876   0.58110
## recipe       9.125  3  3.9818   0.04649 *
## Residuals    6.875  9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The recipe is significant, using a significance level of $\alpha = 0.05$.

```r
par(mfrow = c(2, 2))
plot(fit1, which = 5)
plot(fit1, which = 1)
plot(fit1, which = 2)
plot(residuals(fit1) ~ rownames(taste), main = 'Residuals vs Exp. Unit',
     font.main = 1, data = taste, xlab = 'Experimental unit', ylab = 'Residuals')
abline(h = 0, col = 'red')
```

Constant Leverage:
Residuals vs Factor Levels

Residuals vs Fitted

Standardized residuals

panelist :
1   3   5   7   9   11

Factor Level Combinations

Residuals

Fitted values

Normal Q–Q

Residuals vs Exp. Unit

Standardized residuals

Theoretical Quantiles

Residuals

Experimental unit

Seems we have homogeneity of variance across levels and also constant variability in residuals vs fitted plot. The residuals are normally distributed. The residuals vs exp. unit plot seems to have an horizontal pattern too.

```
emmeans::emmeans(fit1, pairwise ~ recipe, adjust = 'Tukey')
```

```
## $emmeans
##  recipe emmean    SE df lower.CL upper.CL
##  A        5.46 0.418  9     4.51     6.40
##  B        6.21 0.418  9     5.26     7.15
##  C        6.83 0.418  9     5.89     7.78
##  D        4.83 0.418  9     3.89     5.78
##
## Results are averaged over the levels of: panelist
## Confidence level used: 0.95
##
## $contrasts
```

```
##  contrast estimate    SE df t.ratio p.value
##   A - B     -0.750 0.618  9  -1.214  0.6342
##   A - C     -1.375 0.618  9  -2.225  0.1882
##   A - D      0.625 0.618  9   1.011  0.7472
##   B - C     -0.625 0.618  9  -1.011  0.7472
##   B - D      1.375 0.618  9   2.225  0.1882
##   C - D      2.000 0.618  9   3.236  0.0421
##
## Results are averaged over the levels of: panelist
## P value adjustment: tukey method for comparing a family of 4 estimates
```

The C recipe has slightly higher score than D (p-value < 0.05), but all the other pairwise comparisons aren't significant.

For the inter-block analysis, we can just treat block as random.

```
library(nlme)
fit2 <- lme(
  score ~ recipe,
  random = ~ 1 | panelist,
  data = taste
)
anova(fit2)
```

```
##             numDF denDF   F-value p-value
## (Intercept)     1    11 1139.5349  <.0001
## recipe          3     9    9.7674  0.0034
```

Again, now using a mixed model, the recipe is significant using a significance level of $\alpha = 0.05$.

```
par(mfrow = c(2, 2))
plot(resid(fit2) ~ fitted(fit2))
abline(h = 0, lty = 2)
qqnorm(fit2$residuals)
qqline(fit2$residuals, lty = 2)
plot(residuals(fit2) ~ rownames(taste), main = 'Residuals vs Exp. Unit',
     font.main = 1, data = taste, xlab = 'Experimental unit', ylab = 'Residuals')
abline(h = 0, lty = 2)
```

Normal Q–Q Plot

Residuals vs Exp. Unit
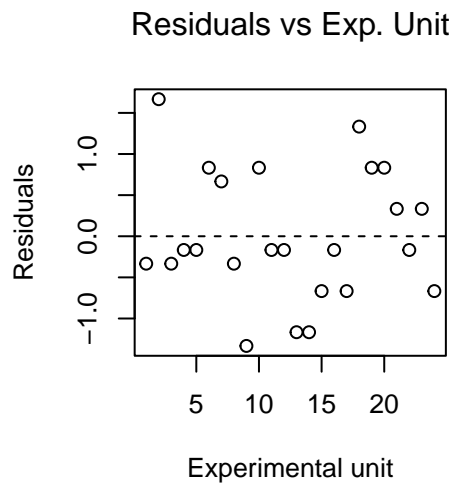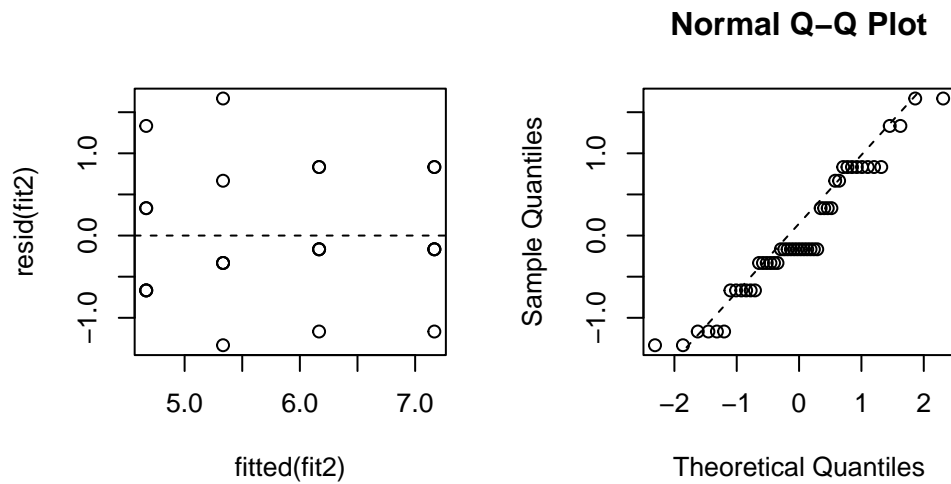
Seems the residuals are still good and the model is adequate, although the QQ-normal plot was better in the fixed model.

```
emmeans::emmeans(fit2, pairwise ~ recipe, adjust = 'Tukey')
```

```
## $emmeans
##  recipe emmean    SE df lower.CL upper.CL
##  A        5.33 0.346 11     4.57     6.09
##  B        6.17 0.346  9     5.38     6.95
##  C        7.17 0.346  9     6.38     7.95
##  D        4.67 0.346  9     3.88     5.45
##
## Degrees-of-freedom method: containment
## Confidence level used: 0.95
##
## $contrasts
##  contrast estimate    SE df t.ratio p.value
```

```
##  A - B      -0.833 0.489  9  -1.705  0.3750
##  A - C      -1.833 0.489  9  -3.751  0.0195
##  A - D       0.667 0.489  9   1.364  0.5494
##  B - C      -1.000 0.489  9  -2.046  0.2412
##  B - D       1.500 0.489  9   3.069  0.0541
##  C - D       2.500 0.489  9   5.115  0.0029
##
## Degrees-of-freedom method: containment
## P value adjustment: tukey method for comparing a family of 4 estimates
```

The expected marginal means are slightly different from the fixed block model. For the C recipe, in this mixed model the EMM is 7.17, whereas for the fixed model it was 6.83.

For this mixed model, the recipe C is still significantly higher than D, but now the difference is 2.5, whereas in the fixed model it was 2.0. Now, the recipe C is also significantly higher than A, using a significance level of $\alpha = 0.05$.

2. With the data set below, answer:.

```
data(yates.oats, package = "agridat")

yates.oats$nitro <- as.factor(yates.oats$nitro)

?agridat::yates.oats
str(yates.oats)
```

```
## 'data.frame':    72 obs. of  8 variables:
##  $ row  : int  16 12 3 14 8 5 15 11 3 14 ...
##  $ col  : int  3 4 3 1 2 2 4 4 4 2 ...
##  $ yield: int  80 60 89 117 64 70 82 102 82 114 ...
##  $ nitro: Factor w/ 4 levels "0","0.2","0.4",..: 1 1 1 1 1 1 2 2 2 2 ...
##  $ gen  : Factor w/ 3 levels "GoldenRain","Marvellous",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ block: Factor w/ 6 levels "B1","B2","B3",..: 1 2 3 4 5 6 1 2 3 4 ...
##  $ grain: num  20 15 22.2 29.2 16 ...
##  $ straw: num  28 25 40.5 28.8 32 ...
```
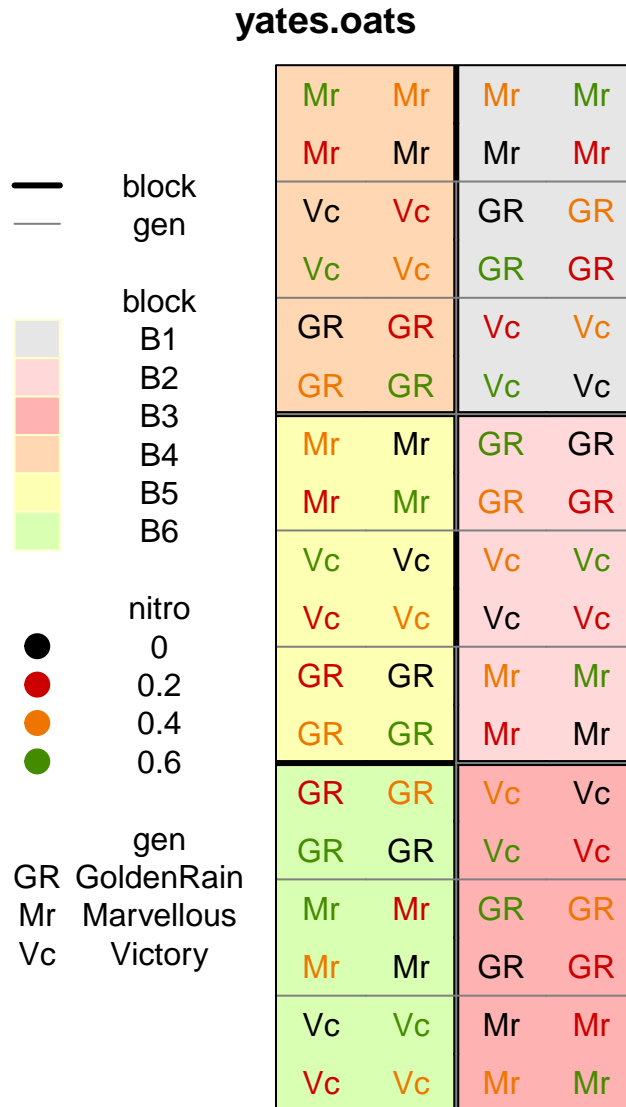
a) What design was used in the following experiment?
   It's a split-plot design.
   The varieties were applied to the **main** plots, whereas manurial treatments were applied in the **sub-plots**.

Checking the field layout:

```
desplot::desplot(
  yates.oats,
  block ~ col + row,
  col = nitro,
  text = gen,
  cex = 1,
  aspect = 511 / 176,
  out1 = block,
  out2 = gen,
  out2.gpar = list(col = "gray50", lwd = 1, lty = 1)
)
```

yates.oats

The main plot is the genotype, and within this main plot we have randomized sub-plots comprising all the 4 possible levels of nitrogen level. The blocks form 3 main plots within block, and we have 6 blocks.

b) Run the appropriate analysis using both ANOVA and REML.

```
with(yates.oats, interaction.plot(x.factor = gen,
                                  trace.factor = nitro,
                                  response = yield))
```
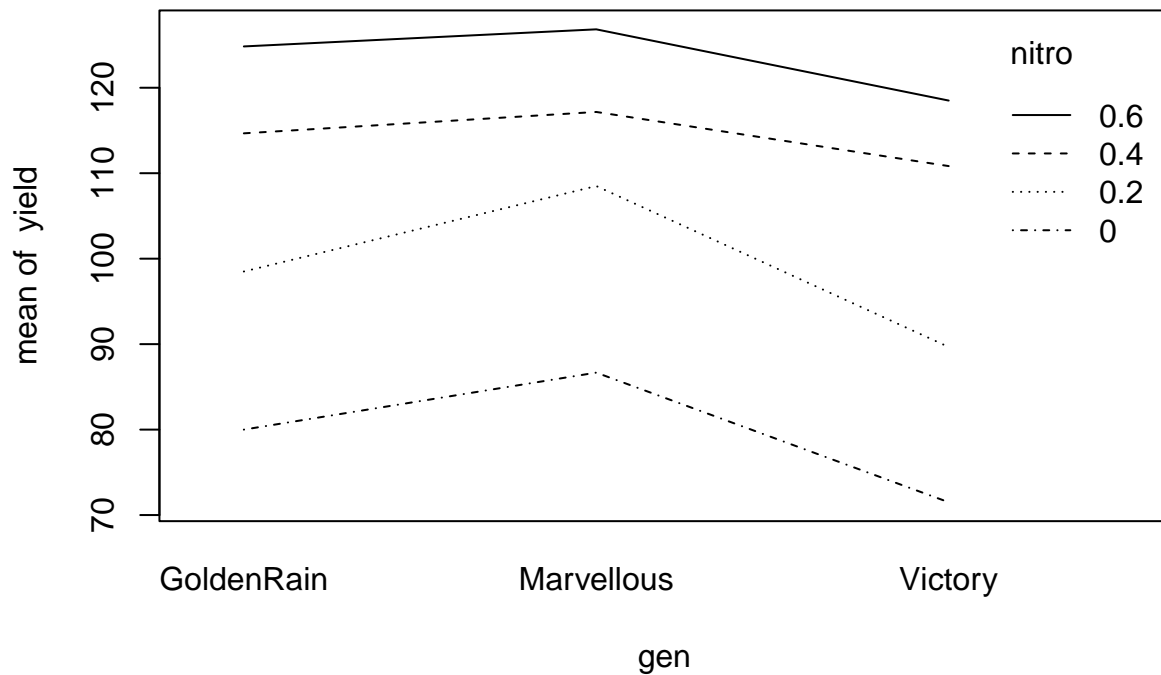
Seems we don't have interaction between genotypes and nitrogen doses because the lines are quite parallel.

Let's do ANOVA.
We have to add a error strata because in split-plot designs we have two error terms: one for the main-plot and another for the sub-plots.
When using block as random, the whole-plot error is `B + B:V` (not the Residual term as usual).

```
oats_aov_rdm_block <- aov(
  yield ~ nitro * gen + Error(block/gen),  # or Error(block + block:gen)
  data = yates.oats
)
summary(oats_aov_rdm_block)
```

```
##
## Error: block
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  5  15875    3175
##
## Error: block:gen
##           Df Sum Sq Mean Sq F value Pr(>F)
## gen        2   1786   893.2   1.485  0.272
## Residuals 10   6013   601.3
##
## Error: Within
##           Df Sum Sq Mean Sq F value   Pr(>F)
## nitro      3  20020    6673  37.686 2.46e-12 ***
```

8

```
## nitro:gen  6     322       54    0.303     0.932
## Residuals 45   7969       177
## ---
## Signif. codes:   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The nitrogen dose (allocated in the sub-plot) effect is significant using a significance level of $\alpha = 0.05$, but the variety (allocated in the main plot) effect is not, nor the interaction "nitro:gen", which agrees with the interaction plot.

We could approach the problem using a mixed model using the REML method.

```
oats_rmle <- nlme::lme(
  fixed = yield ~ nitro * gen,
  random = ~ 1 | block/gen,
  method = 'REML',
  data = yates.oats
)
anova(oats_rmle)
```

```
##               numDF denDF   F-value p-value
## (Intercept)      1    45 245.14299  <.0001
## nitro            3    45  37.68562  <.0001
## gen              2    10   1.48534  0.2724
## nitro:gen        6    45   0.30282  0.9322
```

We got the same results as the previous ANOVA model, where the nitrogen dose effect is significance, but genotype is not, nor the interaction "nitro:gen".

3. Design an experiment and present both the design and the layout (field map) for the following experiments:

a) To evaluate the effect of fertilization scheme and strawberry variety on fruit mass. Assume you have a 32 EU's available. The experiment consists of 8 different plots of land, 2 fertilization schemes, and 4 varieties.

We can use a split-plot design, where the 8 different plots of land are the blocks, the 2 fertilizers levels will be allocated in the main plot, and the 4 varieties levels will be allocated in the sub-plot.
First, we randomize the 2 fertilization levels (1|. and 2|.) within each block.
Second, we randomize the 4 varieties levels (.|1, .|2, .|3, .|4) within each main plot.

```
straw_sp <- FielDHub::split_plot(wp = 2, sp = 4, reps = 4, seed = 2023)
straw_sp$fieldBook
```

```
##     ID LOCATION PLOT REP WHOLE_PLOT SUB_PLOT TRT_COMB
## 1    1        1  101   1          1        2      1|2
## 2    2        1  101   1          1        1      1|1
## 3    3        1  101   1          1        3      1|3
## 4    4        1  101   1          1        4      1|4
## 5    5        1  102   1          2        1      2|1
## 6    6        1  102   1          2        4      2|4
## 7    7        1  102   1          2        3      2|3
## 8    8        1  102   1          2        2      2|2
## 9    9        1  201   2          1        4      1|4
## 10  10        1  201   2          1        3      1|3
## 11  11        1  201   2          1        2      1|2
## 12  12        1  201   2          1        1      1|1
```

```
## 13 13      1   202    2        2           3       2|3
## 14 14      1   202    2        2           1       2|1
## 15 15      1   202    2        2           2       2|2
## 16 16      1   202    2        2           4       2|4
## 17 17      1   301    3        2           1       2|1
## 18 18      1   301    3        2           4       2|4
## 19 19      1   301    3        2           3       2|3
## 20 20      1   301    3        2           2       2|2
## 21 21      1   302    3        1           2       1|2
## 22 22      1   302    3        1           4       1|4
## 23 23      1   302    3        1           3       1|3
## 24 24      1   302    3        1           1       1|1
## 25 25      1   401    4        2           2       2|2
## 26 26      1   401    4        2           1       2|1
## 27 27      1   401    4        2           3       2|3
## 28 28      1   401    4        2           4       2|4
## 29 29      1   402    4        1           3       1|3
## 30 30      1   402    4        1           2       1|2
## 31 31      1   402    4        1           4       1|4
## 32 32      1   402    4        1           1       1|1
```

```
plot(straw_sp)
```


Split Plot Design (RCBD) 16X2
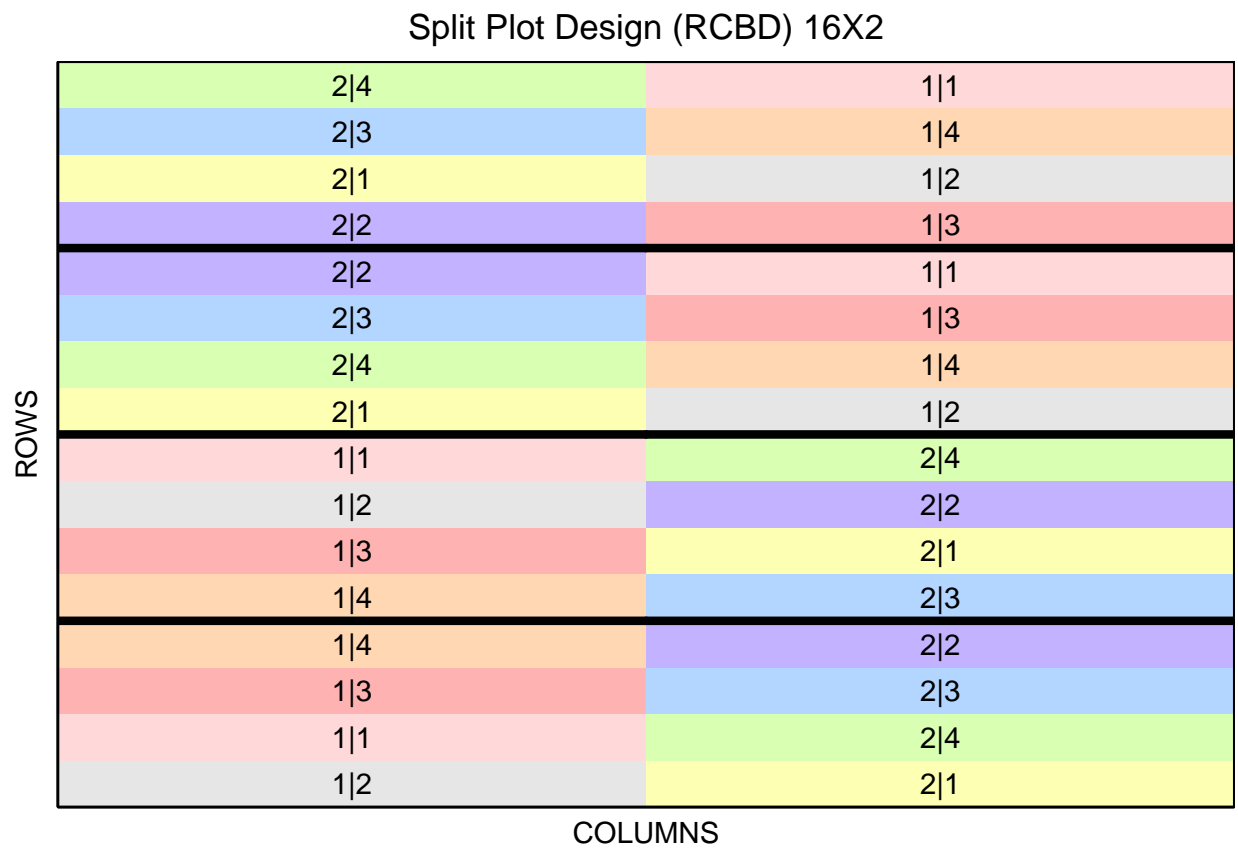
b) An experiment to evaluate wine flavor. There are 42 brands of wine and 14 panelists.

We can use a (partially) balanced incomplete block design. In this case, each panelist taste a smaller subset of brands (not all the 42). It's not feasible to make each panelist tastes 42 different wine brands because they would potentially lose their ability to give proper scores to each brand.

```
wine <- FielDHub::incomplete_blocks(t = 42, k = 3, r = 2, seed = 2023)
df_wine <- wine$fieldBook
df_wine$TREATMENT <- as.integer(substr(df_wine$TREATMENT, start = 3, stop = 4))

# contingency table
with(df_wine[order(df_wine$TREATMENT), ], table(IBLOCK, TREATMENT))
```

```
##       TREATMENT
## IBLOCK 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
##     1  1 0 0 1 0 0 1 0 0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
##     2  0 0 1 0 0 0 0 0 0  0  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  1  0
##     3  0 0 0 0 0 1 0 0 0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  1  1
##     4  0 0 0 0 0 0 0 1 1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  0  0
##     5  0 0 0 0 1 0 0 0 0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  1
##     6  0 2 0 0 1 0 0 0 0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0
##     7  0 0 0 0 0 1 0 0 0  0  0  0  0  0  0  0  0  1  0  0  1  0  0  1  0  0  0
##     8  0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  0
##     9  0 0 0 0 0 0 0 0 1  0  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
##     10 0 0 1 1 0 0 0 0 0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  0
##     11 0 0 0 0 0 0 0 0 0  0  0  0  1  0  0  0  1  0  1  0  0  1  0  0  0  0  0
##     12 0 0 0 0 0 0 1 1 0  0  0  0  0  1  0  0  0  1  0  0  0  0  1  0  0  0  0
##     13 1 0 0 0 0 0 0 0 0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##     14 0 0 0 0 0 0 0 0 0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
##       TREATMENT
## IBLOCK 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
##     1   0  1  0  0  0  0  0  0  1  0  0  0  0  0  0
##     2   0  0  0  1  0  0  0  0  0  1  0  0  0  0  0
##     3   0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
##     4   0  1  0  0  0  0  0  0  0  1  0  0  0  0  0
##     5   0  0  1  1  0  0  0  0  0  0  0  0  0  0  0
##     6   0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
##     7   0  0  0  0  0  0  0  0  1  0  1  0  0  0  0
##     8   1  0  0  0  1  0  0  0  0  0  0  1  0  1  0
##     9   0  0  1  0  1  0  1  0  0  0  0  0  0  0  0
##     10  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##     11  0  0  0  0  0  0  1  0  0  0  0  0  0  1  0
##     12  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
##     13  0  0  0  0  0  2  0  1  0  0  0  1  0  0  0
##     14  1  0  0  0  0  0  0  0  0  0  1  0  1  0  1
```

Each person (block) tasted 6 wines.
Every wine brand was tasted twice.
If we count every treatment pairs combinations, we have $\lambda_1 = 2$, $\lambda_2 = 1$, $\lambda_3 = 0$, i.e., some pairs occur together two times, one time, or never. Hence, we have a Partially Incomplete Block (PBIB) design.

```
plot(wine)
```

Incomplete Blocks Design Field Layout 6X14

7	31	13	37	14	16	6	19	20	4	34	18	35	28
36	21	27	29	5	2	38	32	9	25	22	23	33	11
1	3	26	8	12	42	17	39	30	24	41	40	10	15
4	15	24	25	27	2	36	41	32	3	19	8	33	42
16	37	6	22	30	5	23	28	12	10	13	7	1	38
29	26	35	9	31	21	20	18	34	11	17	14	39	40

ROWS

COLUMNS

The columns are the people, whereas rows are the wines tasted by the person, so every person tasted 6 brands.