Case Study : a multi-environment trial for yield using RCBD

Prof L. Gentzbittel Skoltech, Digital Agriculture Laboratory *

Prof C. Ben, Skoltech, Digital Agriculture Laboratory †

April, 5th 2021 - Skoltech

CASE STUDY PRESENTATION

Ten genotypes are assessed at five different locations.
Within-site variability is controlled (assessed) using RCBD in each site, with 4 blocks per site.

This is a first analysis to familiarize yourself with the method. YET it is not the state-of-the-art method that can be requested by reviewers or shareholders. A 'modern' and detailed analysis of this type of data will be carried out in the "Advanced Course"

PREPARATION OF THE WORKING INTERFACE IN R

```
### I. Set working directory
#On RStudio: tab 'Session'-> Set Working Directory -> Choose Directory.
#Choose the directory containing the datafile and the associated R script.

### II. Installation R packages needed for the analysis on RStudio:
#Click on the 'Packages' tab in the bottom-right window of R Studio interface->'Install Packages'
#Comment #1: R package installation requires a connection to internet
#Comment #2: Once packages have been installed, no need to re-install them again when you close-open again

### III. Initialisation of the working space
# To erase all graphs
graphics.off()
# To erase objects from the working space - Clean up of the memory
rm(list = ls())
# use of the constraint 'set-to-zero' for ANOVAs ## will see later in this script
options(contrasts=c('contr.treatment','contr.poly'))
#can also use 'contr.sum' for a 'sum-to-zero' constraint
```

LOADING REQUIRED METHODS FOR ANALYSIS

```
## Loading of the R packages needed for the analysis.
library(car)         # Levene's test
library(agricolae)   # Newman-Keuls & Tukeys tests
library(ggplot2)
library(dplyr)
library(openxlsx)    # to load excel files
```

*l.gentzbittel@skoltech.ru
†c.ben@skoltech.ru

STARTING THE ANALYSIS

```
## loading data file
Produc <- read.xlsx("07_MET_beginnerLevel.xlsx", sheet = 1)
str(Produc)
```

```
## 'data.frame':    200 obs. of  5 variables:
##  $ Env : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Rep : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Gen : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ Prod: chr  "101.847" "101.357" "99.896" "92.163" ...
##  $ Repb: chr  "1.1" "1.1" "1.1" "1.1" ...
```

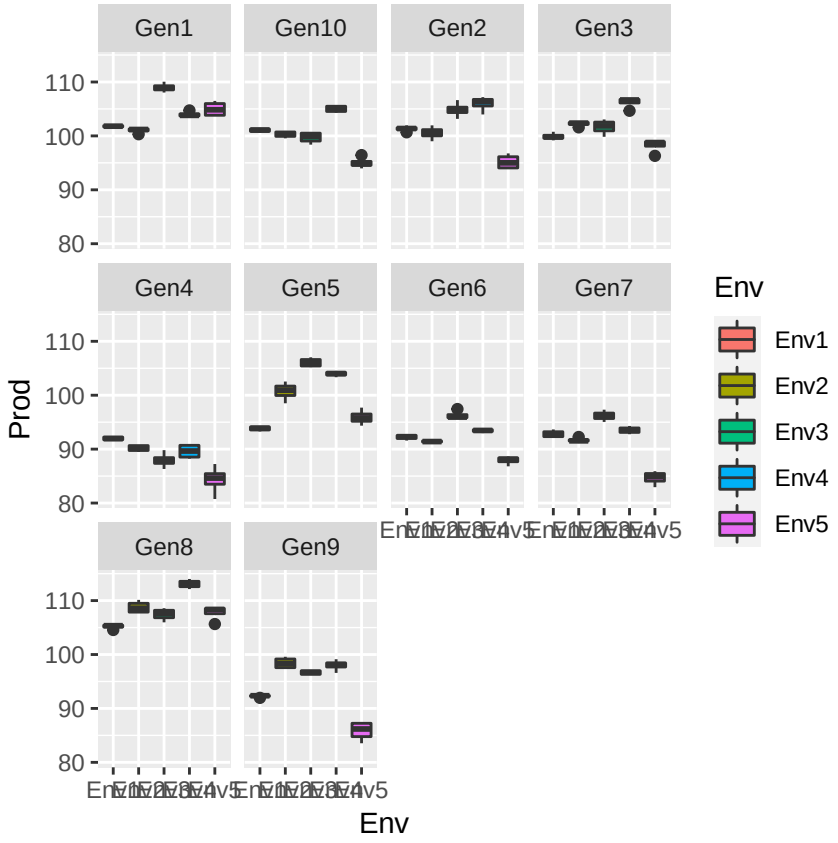
```
## The person who typed the data was lazy and has just indicated blocks and environments with numbers.
## need to transform numbers into factors, and modify names of levels in the same operation


Produc$Gen <- factor(paste("Gen", Produc$Gen, sep = ""))
Produc$Env <- factor(paste("Env", Produc$Env, sep = ""))
Produc$Rep <- factor(paste("Block", Produc$Rep, sep = ""))
Produc$Repb <- factor(paste("Block",Produc$Repb, sep = "/"))  ## To indicate that blocks are nested in env
Produc$Prod <- as.numeric(Produc$Prod)     ## may not be required on your computer. also a weakness of rea
str(Produc)
```

```
## 'data.frame':    200 obs. of  5 variables:
##  $ Env : Factor w/ 5 levels "Env1","Env2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Rep : Factor w/ 4 levels "Block1","Block2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Gen : Factor w/ 10 levels "Gen1","Gen10",..: 1 3 4 5 6 7 8 9 10 2 ...
##  $ Prod: num  101.8 101.4 99.9 92.2 94.1 ...
##  $ Repb: Factor w/ 20 levels "Block/1.1","Block/1.2",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
############ 1. visualisations

## Production per genotype, in each environment, using raw data from blocks
x11()
(graf1 <- ggplot(Produc, aes(x = Env, y = Prod, fill = Env)) +
    geom_boxplot() +
    facet_wrap( ~ Gen)
)
```

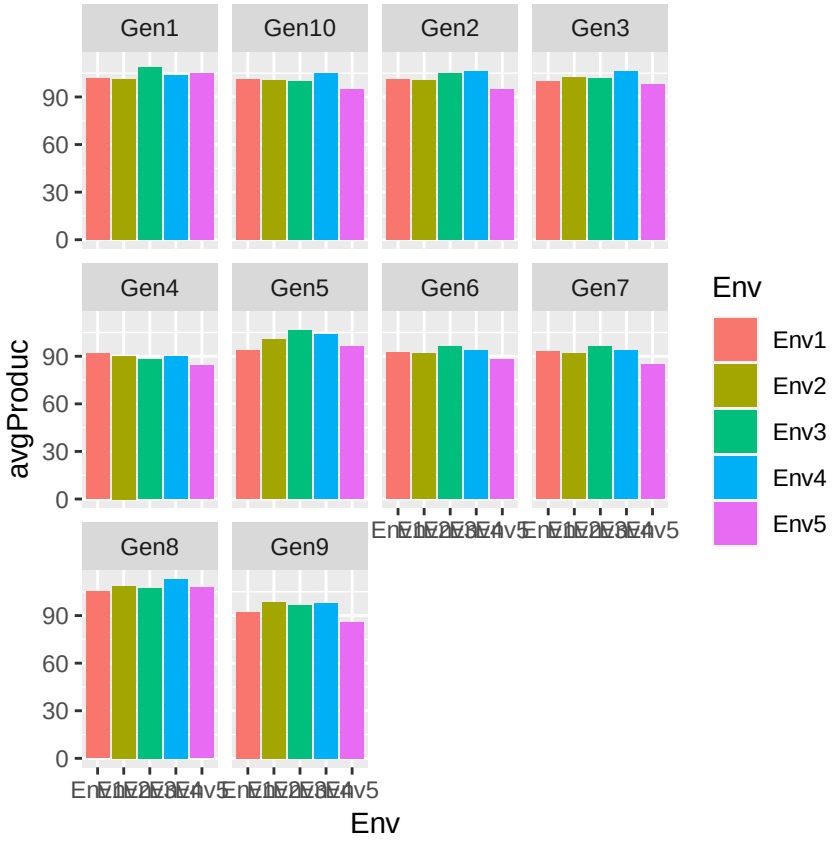



```
## A summary : average production per site and per genotype, by creasing order (add sd and n )
Summaries <- Produc %>%
    group_by(Env, Gen) %>%
     summarise(avgProduc = mean(Prod, na.rm = TRUE),
               sdProduc = sd(Prod, na.rm = TRUE),
               nbData = n()
               ) %>%
     arrange(desc(avgProduc)) %>%
    print(n = Inf)  # to seee all data
```

```
## # A tibble: 50 x 5
## # Groups:   Env [5]
##    Env   Gen   avgProduc sdProduc nbData
##    <fct> <fct>     <dbl>    <dbl>  <int>
##  1 Env4  Gen8      113.     0.816      4
##  2 Env3  Gen1      109.     0.836      4
##  3 Env2  Gen8      109.     1.12       4
##  4 Env5  Gen8      108.     1.41       4
##  5 Env3  Gen8      107.     1.15       4
##  6 Env4  Gen3      106.     1.05       4
##  7 Env3  Gen5      106.     0.849      4
##  8 Env4  Gen2      106.     1.40       4
##  9 Env1  Gen8      105.     0.438      4
## 10 Env4  Gen10     105.     0.667      4
## 11 Env5  Gen1      105.     1.39       4
## 12 Env3  Gen2      105.     1.42       4
## 13 Env4  Gen5      104.     0.493      4
## 14 Env4  Gen1      104.     0.521      4
## 15 Env2  Gen3      102.     0.495      4
## 16 Env1  Gen1      102.     0.282      4
## 17 Env3  Gen3      102.     1.40       4
```

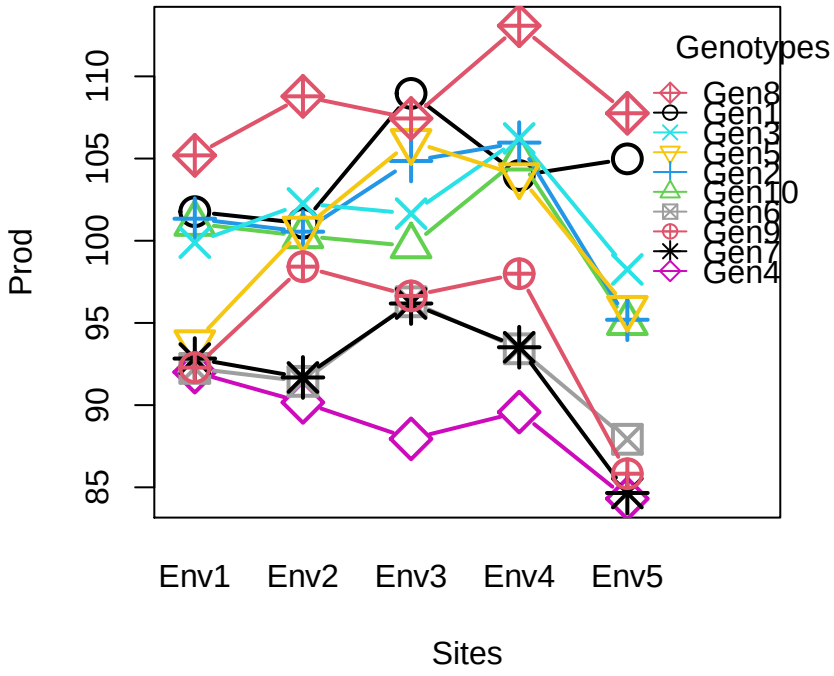
```
## 18 Env1  Gen2      101.     0.528      4
## 19 Env2  Gen1      101.     0.533      4
## 20 Env1  Gen10     101.     0.368      4
## 21 Env2  Gen5      101.     1.71       4
## 22 Env2  Gen2      101.     1.23       4
## 23 Env2  Gen10     100.     0.619      4
## 24 Env1  Gen3       99.9    0.661      4
## 25 Env3  Gen10      99.7    1.05       4
## 26 Env2  Gen9       98.4    1.05       4
## 27 Env5  Gen3       98.3    1.32       4
## 28 Env4  Gen9       98.0    1.05       4
## 29 Env3  Gen9       96.6    0.400      4
## 30 Env3  Gen6       96.3    0.771      4
## 31 Env3  Gen7       96.2    0.956      4
## 32 Env5  Gen5       95.9    1.40       4
## 33 Env5  Gen2       95.2    1.36       4
## 34 Env5  Gen10      95.0    1.03       4
## 35 Env1  Gen5       93.8    0.425      4
## 36 Env4  Gen7       93.5    0.669      4
## 37 Env4  Gen6       93.4    0.352      4
## 38 Env1  Gen7       92.8    0.659      4
## 39 Env1  Gen9       92.3    0.216      4
## 40 Env1  Gen6       92.2    0.502      4
## 41 Env1  Gen4       92.0    0.346      4
## 42 Env2  Gen7       91.7    0.387      4
## 43 Env2  Gen6       91.4    0.179      4
## 44 Env2  Gen4       90.2    0.610      4
## 45 Env4  Gen4       89.6    1.34       4
## 46 Env3  Gen4       87.9    1.43       4
## 47 Env5  Gen6       87.9    0.793      4
## 48 Env5  Gen9       85.8    1.82       4
## 49 Env5  Gen7       84.7    1.29       4
## 50 Env5  Gen4       84.3    2.68       4
```

```r
## The standard - and UNinformative -- barplot.
## Evidence for information loss when compared to boxplots.
x11()
(graf2 <- ggplot( Summaries, aes( x = Env, y = avgProduc, fill = Env)) +
    geom_bar(stat = "identity") +
    facet_wrap( ~ Gen))
```




```
## the ratio information/ink is very low for this figure

## A classical interaction plot:
x11()
interaction.plot(x.factor = Produc$Env, trace.factor = Produc$Gen, response = Produc$Prod,
                 type = "b",  # lines and points
                 xlab = "Sites", ylab = "Prod", trace.label = "Genotypes",
                 pch=c(1:10), cex=2, lty=1, lwd=2, col = as.numeric(Produc$Gen))
```




```
## your conclusions ?
## "Egular" and "Advanced" training sessions will provide you with methodes to go further in the analysis
## and understand the pattern of variation


################
################  old school analysis : use of aov()
################


## model w/accounting for blocks in each site.
## this goes down to a CRD in a each site

model1 <- aov( Prod ~ Gen * Env , data = Produc)
summary(model1)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Gen           9   7038   782.0  731.16 <2e-16 ***
## Env           4   1353   338.3  316.33 <2e-16 ***
## Gen:Env      36   1033    28.7   26.84 <2e-16 ***
## Residuals   150    160     1.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Model with blocks nested in sites
model2 <- aov( Prod ~ Rep %in% Env + Gen * Env , data = Produc)
summary(model2)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Gen           9   7038   782.0 741.457 <2e-16 ***
## Env           4   1353   338.3 320.789 <2e-16 ***
## Rep:Env      15     18     1.2   1.141  0.327
```

```
## Env:Gen      36   1033    28.7  27.218 <2e-16 ***
## Residuals   135    142     1.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Check and understand Df. Why 15Df for blocks within sites ?


## INCORRECT MODEL : the breeder forget that blocks are nested within sites
## Model with blocks not nested in sites
model2BAD <- aov( Prod ~ Rep + Gen * Env , data = Produc)
summary(model2BAD)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Rep           3      2     0.6   0.569  0.636
## Gen           9   7038   782.0 724.857 <2e-16 ***
## Env           4   1353   338.3 313.607 <2e-16 ***
## Gen:Env      36   1033    28.7  26.608 <2e-16 ***
## Residuals   147    159     1.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### HORRIBLE MODEL - TOTALLY WRONG - why ?
model2HORROR <- aov( Prod ~ Rep * Gen * Env , data = Produc)
summary(model2HORROR)
```

```
##              Df Sum Sq Mean Sq
## Rep           3      2     0.6
## Gen           9   7038   782.0
## Env           4   1353   338.3
## Rep:Gen      27     37     1.4
## Rep:Env      12     16     1.4
## Gen:Env      36   1033    28.7
## Rep:Gen:Env 108    105     1.0
```

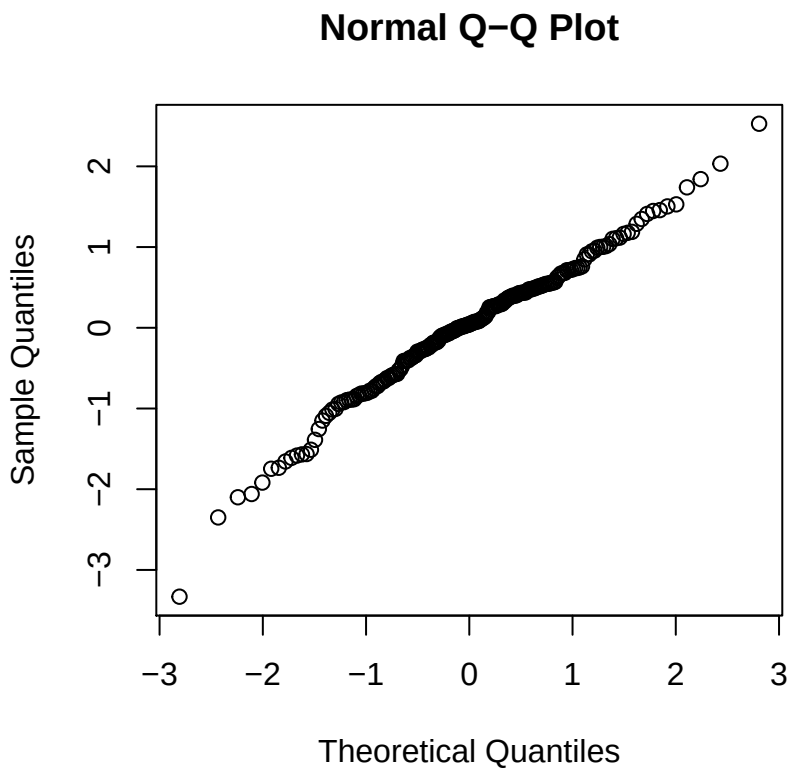
```
## This syntax using alternative coding of blocks is also acceptable.
## you will understand why the computation of Df is correct during the "Regular Course"
modele3 <- aov( Prod ~ Gen * Env + Repb, data = Produc)
summary(modele3)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Gen           9   7038   782.0 741.457 <2e-16 ***
## Env           4   1353   338.3 320.789 <2e-16 ***
## Repb         15     18     1.2   1.141  0.327
## Gen:Env      36   1033    28.7  27.218 <2e-16 ***
## Residuals   135    142     1.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# test Gaussian distribution of residuals - use model2.
shapiro.test(model2$residuals)  # explore using graphics to decide if it is a concern or not really
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model2$residuals
## W = 0.98306, p-value = 0.01644
```

```
x11()
qqnorm(model2$residuals)
```




```
# test for variance homogeneity. Hand-made Levene's test !
## Caution: there is only one numerical value par combination Genotype * site * block thus NO variance ;-)
## eg:
summary(aov( abs(model2$residuals) ~ Produc$Gen:Produc$Env:Produc$Repb) )  # no test !
```

```
##                                   Df Sum Sq Mean Sq
## Produc$Gen:Produc$Env:Produc$Repb 199  60.04  0.3017
```

```
## if me make the asumption the the variability within blocks is homogeneous, we can test such as :
summary( aov(abs(model2$residuals) ~ Produc$Gen:Produc$Env) )
```

```
##                       Df Sum Sq Mean Sq F value   Pr(>F)
## Produc$Gen:Produc$Env  49  25.25  0.5154   2.222 0.000123 ***
## Residuals             150  34.79  0.2319
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## NOT very good. We need to explore why.

## Let's have a look at residuals variances per site:
Produc$Residuals <- model2$residuals
head(Produc) ## to see beginining of dataframe
```

```
##    Env    Rep  Gen    Prod      Repb Residuals
## 1 Env1 Block1 Gen1 101.847 Block/1.1  -0.00570
## 2 Env1 Block1 Gen2 101.357 Block/1.1  -0.06645
## 3 Env1 Block1 Gen3  99.896 Block/1.1  -0.07245
## 4 Env1 Block1 Gen4  92.163 Block/1.1   0.07155
## 5 Env1 Block1 Gen5  94.080 Block/1.1   0.18680
## 6 Env1 Block1 Gen6  92.746 Block/1.1   0.42955
```

```
tail(Produc) ## to see the end
```

```
##      Env    Rep   Gen    Prod      Repb Residuals
```

```
## 195 Env5 Block4  Gen5  97.699 Block/5.4  2.033175
## 196 Env5 Block4  Gen6  88.643 Block/5.4  0.955925
## 197 Env5 Block4  Gen7  85.330 Block/5.4  0.909175
## 198 Env5 Block4  Gen8 108.514 Block/5.4  0.992175
## 199 Env5 Block4  Gen9  85.182 Block/5.4 -0.410575
## 200 Env5 Block4 Gen10  94.731 Block/5.4 -0.040075
```

```
## A summary of residuals:
Produc %>%
    group_by(Env) %>%
     summarise(avgResiduals = mean(Residuals, na.rm = TRUE),  ## to check one property of residuals
               varResiduals = sd(Residuals, na.rm = TRUE)^2,
               nbData = n()
               )
```

```
## # A tibble: 5 x 4
##   Env   avgResiduals varResiduals nbData
##   <fct>        <dbl>        <dbl>  <int>
## 1 Env1     -3.94e-17        0.149     40
## 2 Env2     -1.89e-17        0.624     40
## 3 Env3      4.11e-17        0.568     40
## 4 Env4      3.96e-17        0.578     40
## 5 Env5      4.58e-17        1.73      40
```

```
### Does our hypothesis make sense ?
## Sure we would need to use a ANOVA method that authorize unequal variances
## please attend the "Regular" and "Advanced" training sessions :-)
```