Case Study : Simple Square Lattice design

Prof L. Gentzbittel Skoltech, Project Center for Agro Technologies *

Prof C. Ben, Skoltech, Project Center for Agro Technologies †

Oct. 20th , 2021 - Skoltech

CASE STUDY PRESENTATION

25 soybean varieties are tested in a Simple Square Lattice with block size = 5 and 2 replicates.

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 Alpha_latticefile and the associated R script.

### II. Possibly, installation of new 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 RStudio.

### 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())


# this is a trick to detect which folder contains the R script and the Alpha_lattice
main_dir <- dirname(rstudioapi::getSourceEditorContext()$path)
setwd(main_dir)
```

LOADING REQUIRED METHODS FOR ANALYSIS

```
library(agricolae)
```

ANALYSIS OF THE CASE STUDY

```
##################
##################   Analysis of simple square lattice using agricolae
##################

lattice5x5 <- read.table("soy.txt", header=T)
str(lattice5x5)
```

*l.gentzbittel@skoltech.ru

†c.ben@skoltech.ru

```
## 'data.frame':    50 obs. of  4 variables:
##  $ Group   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Block   : int  1 1 1 1 1 2 2 2 2 2 ...
##  $ Treatmnt: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Yield   : int  6 7 5 8 6 16 12 12 13 8 ...
```

```
lattice5x5$Group <- factor(lattice5x5$Group)
lattice5x5$Block <- factor(lattice5x5$Block)
lattice5x5$Treatmnt <- factor(lattice5x5$Treatmnt)
str(lattice5x5)
```

```
## 'data.frame':    50 obs. of  4 variables:
##  $ Group   : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Block   : Factor w/ 10 levels "1","2","3","4",..: 1 1 1 1 1 2 2 2 2 2 ...
##  $ Treatmnt: Factor w/ 25 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Yield   : int  6 7 5 8 6 16 12 12 13 8 ...
```

```
## see the field experiment:
lattice5x5
```

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

```
## 41     2     9        4    17
## 42     2     9        9    10
## 43     2     9       14    30
## 44     2     9       19     9
## 45     2     9       24    23
## 46     2    10        5    15
## 47     2    10       10    15
## 48     2    10       15    22
## 49     2    10       20    16
## 50     2    10       25    19
```

```
## what are the parameters of this design ?

attach(lattice5x5)
model1 <-PBIB.test(Block, Treatmnt, Group, Yield, k = 5,
                   method = "VC", group = TRUE, console = TRUE)
```

```
##
## ANALYSIS PBIB:  Yield
##
## Class level information
## Block : 10
## Treatmnt : 25
##
## Number of observations:  50
##
## Estimation Method:  Variances component model
##
##     Fit Statistics
## AIC       285.6274
## BIC       352.5482
##
## Analysis of Variance Table
##
## Response: Yield
##                 Df Sum Sq Mean Sq F value   Pr(>F)
## Group            1 212.18 212.180 15.5386 0.001166 **
## Treatmnt.unadj 24 559.28  23.303  1.7066 0.135789
## Block/Group     8 501.84  62.730  4.5939 0.004629 **
## Residual       16 218.48  13.655
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Coefficient of variation: 27.1 %
## Yield Means: 13.62
##
## Parameters PBIB
##              .
## Treatmnt    25
## Block size   5
## Block/Group  5
## Group        2
##
## Efficiency factor 0.75
##
## Comparison test lsd
##
## Treatments with the same letter are not significantly different.
##
##    Yield.adj groups
```

```
## 11 23.551052      a
## 14 20.751714     ab
## 15 19.329946    abc
## 1  19.068070    abc
## 22 18.530607    abc
## 24 17.326518   abcd
## 2  16.972820  abcde
## 25 15.404751 abcdef
## 4  14.768731 abcdef
## 3  14.646302 abcdef
## 6  13.170054  bcdef
## 5  12.846963  bcdef
## 13 12.629284  bcdef
## 16 12.622429  bcdef
## 12 12.455803  bcdef
## 23 12.204089  bcdef
## 21 11.625857   cdef
## 20 11.401323   cdef
## 18 10.700662   cdef
## 17 10.527180   cdef
## 7   9.074805    def
## 10  8.448948    def
## 9   8.370716     ef
## 19  7.323091      f
## 8   6.748286      f
##
## <<< to see the objects: means, comparison and groups. >>>
```

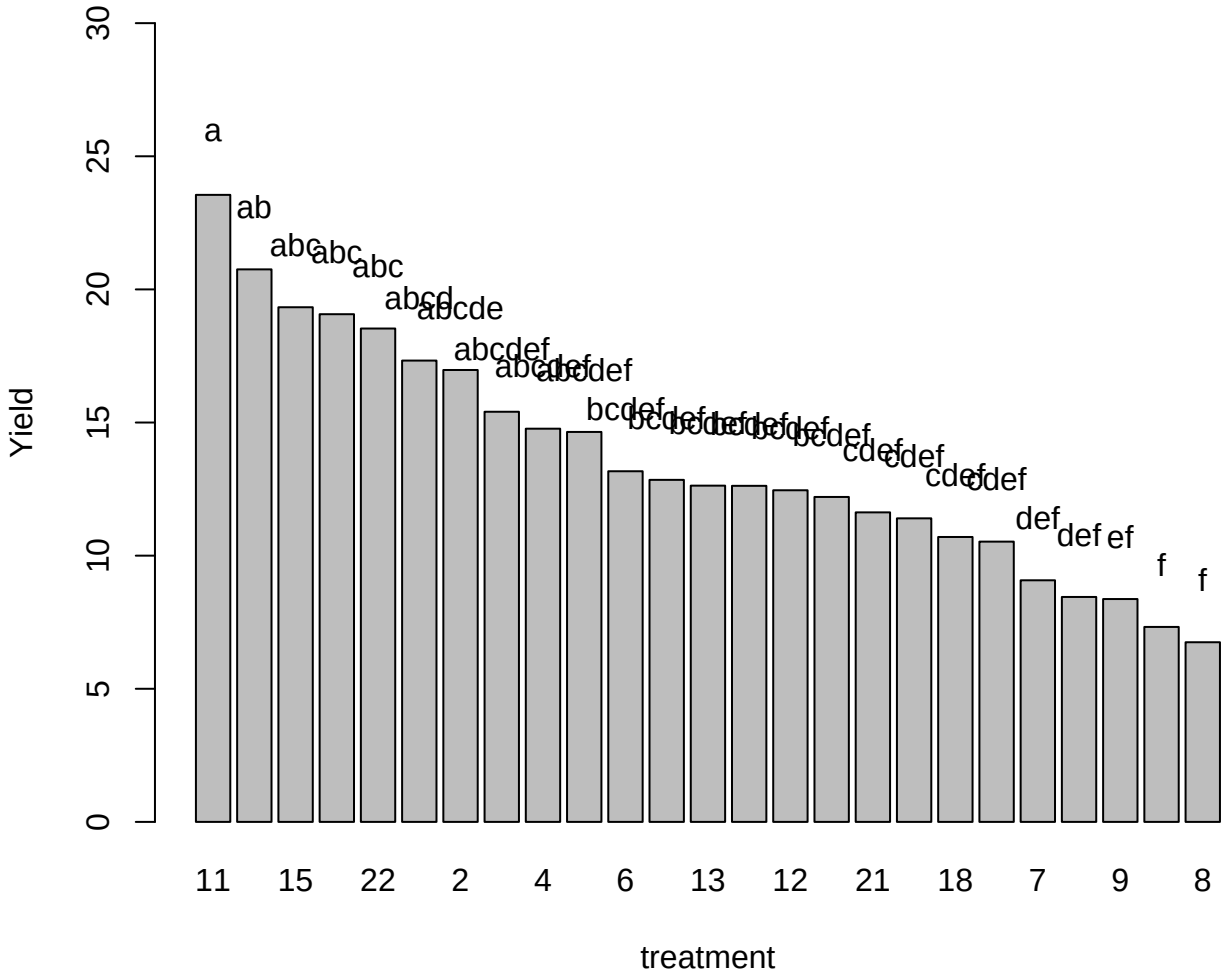
```
model1$means
```

```
##    Yield Yield.adj       SE r        std Min Max   Q25  Q50   Q75
## 1   15.0 19.068070 2.993997 2 12.7279221   6  24 10.50 15.0 19.50
## 2   14.0 16.972820 2.993997 2  9.8994949   8  15  9.75 11.5 13.25
## 3   10.5 14.646302 2.993997 2  7.7781746  17  24 18.75 20.5 22.25
## 4   12.5 14.768731 2.993997 2  6.3639610   7  14  8.75 10.5 12.25
## 5   10.5 12.846963 2.993997 2  6.3639610   7  12  8.25  9.5 10.75
## 6   14.5 13.170054 2.993997 2  2.1213203   9  30 14.25 19.5 24.75
## 7   11.5  9.074805 2.993997 2  0.7071068  14  22 16.00 18.0 20.00
## 8    8.0  6.748286 2.993997 2  5.6568542  11  18 12.75 14.5 16.25
## 9   11.5  8.370716 2.993997 2  2.1213203  11  16 12.25 13.5 14.75
## 10  11.5  8.448948 2.993997 2  4.9497475  12  13 12.25 12.5 12.75
## 11  20.5 23.551052 2.993997 2  4.9497475   9  13 10.00 11.0 12.00
## 12  10.5 12.455803 2.993997 2  4.9497475   7  21 10.50 14.0 17.50
## 13   9.5 12.629284 2.993997 2  3.5355339  14  16 14.50 15.0 15.50
## 14  19.5 20.751714 2.993997 2 14.8492424   8  14  9.50 11.0 12.50
## 15  18.0 19.329946 2.993997 2  5.6568542  15  23 17.00 19.0 21.00
## 16  14.5 12.622429 2.993997 2  4.9497475  11  12 11.25 11.5 11.75
## 17  13.5 10.527180 2.993997 2  3.5355339  14  23 16.25 18.5 20.75
## 18  12.5 10.700662 2.993997 2  0.7071068  14  19 15.25 16.5 17.75
## 19  11.0  7.323091 2.993997 2  2.8284271   5  16  7.75 10.5 13.25
## 20  15.0 11.401323 2.993997 2  1.4142136   8  17 10.25 12.5 14.75
## 21  11.0 11.625857 2.993997 2  4.2426407   6  15  8.25 10.5 12.75
## 22  19.0 18.530607 2.993997 2  5.6568542  13  16 13.75 14.5 15.25
## 23  11.5 12.204089 2.993997 2  0.7071068  11  12 11.25 11.5 11.75
## 24  18.5 17.326518 2.993997 2  6.3639610   4  12  6.00  8.0 10.00
## 25  16.5 15.404751 2.993997 2  3.5355339  10  13 10.75 11.5 12.25
```

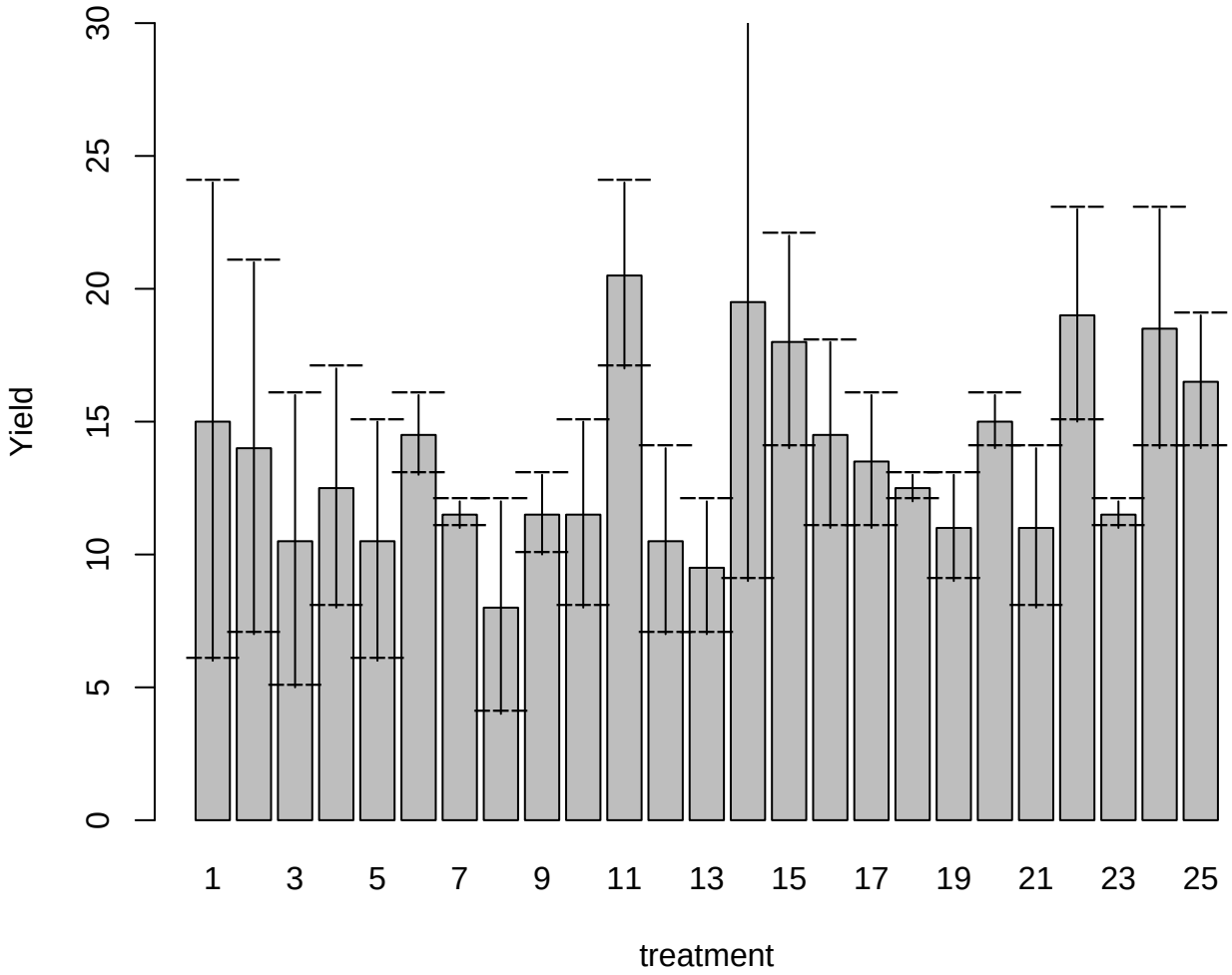
```
#model1$comparison
model1$groups
```

```
##    Yield.adj groups
## 11 23.551052      a
## 14 20.751714     ab
## 15 19.329946    abc
## 1  19.068070    abc
## 22 18.530607    abc
## 24 17.326518   abcd
## 2  16.972820  abcde
## 25 15.404751 abcdef
## 4  14.768731 abcdef
## 3  14.646302 abcdef
## 6  13.170054  bcdef
## 5  12.846963  bcdef
## 13 12.629284  bcdef
## 16 12.622429  bcdef
## 12 12.455803  bcdef
## 23 12.204089  bcdef
## 21 11.625857   cdef
## 20 11.401323   cdef
## 18 10.700662   cdef
## 17 10.527180   cdef
## 7   9.074805    def
## 10  8.448948    def
## 9   8.370716     ef
## 19  7.323091      f
## 8   6.748286      f
```

```r
## How to plot results of the simple square lattice
x11()
bar.group(model1$groups, ylim = c(0, 30),
          xlab = "treatment", ylab = "Yield")
```




```
x11()
bar.err(model1$means, ylim = c(0, 30), xlab = "treatment", ylab = "Yield")
```




```
## alternative method to estimate variances
model2 <-PBIB.test(Block, Treatmnt, Group, Yield, k = 5,
                   method="REML", group = TRUE, console = TRUE)
```

```
##
## ANALYSIS PBIB:  Yield
##
## Class level information
## Block : 10
## Treatmnt : 25
##
## Number of observations:  50
##
## Estimation Method:  Residual (restricted) maximum likelihood
##
## Parameter Estimates
##               Variance
## Block:Group 19.629999
## Group        4.014993
## Residual    13.655001
##
##                       Fit Statistics
## AIC                        224.56945
## BIC                        258.69797
## -2 Res Log Likelihood      -84.28473
##
## Analysis of Variance Table
##
```

```
## Response: Yield
##           Df Sum Sq Mean Sq F value  Pr(>F)
## Treatmnt  24 644.63  26.859   1.967 0.08244 .
## Residuals 16 218.48  13.655
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Coefficient of variation: 27.1 %
## Yield Means: 13.62
##
## Parameters PBIB
##              .
## Treatmnt    25
## Block size   5
## Block/Group  5
## Group        2
##
## Efficiency factor 0.75
##
## Comparison test lsd
##
## Treatments with the same letter are not significantly different.
##
##    Yield.adj groups
## 11 23.551052      a
## 14 20.751714     ab
## 15 19.329946    abc
## 1  19.068069    abc
## 22 18.530607    abc
## 24 17.326518   abcd
## 2  16.972820  abcde
## 25 15.404751 abcdef
## 4  14.768731 abcdef
## 3  14.646301 abcdef
## 6  13.170054  bcdef
## 5  12.846963  bcdef
## 13 12.629284  bcdef
## 16 12.622430  bcdef
## 12 12.455803  bcdef
## 23 12.204089  bcdef
## 21 11.625857   cdef
## 20 11.401323   cdef
## 18 10.700662   cdef
## 17 10.527180   cdef
## 7   9.074805    def
## 10  8.448948    def
## 9   8.370716     ef
## 19  7.323091      f
## 8   6.748286      f
##
## <<< to see the objects: means, comparison and groups. >>>
```

What are your conclusions ?