Case Study : Alpha Lattice design

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

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

Feb 2022 - Skoltech

CASE STUDY PRESENTATION

A spring oats trial. There were 24 varieties in 3 replicates, each consisting of 6 incomplete blocks of 4 plots. Planted in a alpha design. 72 observations on the following 5 variables: PLOT Plot number; REP Replicate code; BLOCK Incomplete block code; GEN Genotype code; YIELD Observed dry matter yield (tonnes/ha)

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 datafile
main_dir <- dirname(rstudioapi::getSourceEditorContext()$path)
setwd(main_dir)
```

LOADING REQUIRED METHODS FOR ANALYSIS

```
library(agricolae)
library(tidyverse)
library(ggplot2)
library(openxlsx)  ## to import/export Excel files
```

ANALYSIS OF THE CASE STUDY

```
################ generate an alpha-lattice using agricolae
```

[*]l.gentzbittel@skoltech.ru

[†]c.ben@skoltech.ru

```
# Treatments. Here: num. of genotypes
Genotype <- c(paste("gen0", 1:9, sep = ""), paste("gen", 10:24, sep = ""))

g <- length(Genotype)
g
```

```
## [1] 24
```

```
r <- 3   # two replicates
k <- 4   # block size , num. of plots per block
s <- 6   # num of blocks per replicate

## we have the condition for a alpha-lattice : g = s * k

(obs <- s * k * r)   # total num. of plots
```

```
## [1] 72
```

```
(b <- s * r)   # total num. of blocks
```

```
## [1] 18
```

```
book <- design.alpha(Genotype, k, r, seed = 3264)
```

```
##
## Alpha Design (0,1) - Serie  III
##
## Parameters Alpha Design
## ========================
## Treatmeans : 24
## Block size : 4
## Blocks     : 6
## Replication: 3
##
## Efficiency factor
## (E ) 0.7540984
##
## <<< Book >>>
```

```
## the Efficiency factor is 75% of a RCBD.
```

The different quantities associated with this design:

```
ls(book)
```

```
## [1] "book"       "parameters" "sketch"     "statistics"
```

```
book$parameters
```

```
## $design
## [1] "alpha"
##
## $trt
##  [1] "gen01" "gen02" "gen03" "gen04" "gen05" "gen06" "gen07" "gen08" "gen09"
## [10] "gen10" "gen11" "gen12" "gen13" "gen14" "gen15" "gen16" "gen17" "gen18"
## [19] "gen19" "gen20" "gen21" "gen22" "gen23" "gen24"
##
## $k
## [1] 4
##
## $r
## [1] 3
##
```

```
## $serie
## [1] "III"
##
## $seed
## [1] 3264
##
## $kinds
## [1] "Super-Duper"
```
```
# map of field plots
book$sketch
```
```
## $rep1
##      [,1]    [,2]    [,3]    [,4]
## [1,] "gen19" "gen22" "gen11" "gen18"
## [2,] "gen06" "gen24" "gen07" "gen23"
## [3,] "gen15" "gen01" "gen09" "gen20"
## [4,] "gen21" "gen03" "gen12" "gen05"
## [5,] "gen13" "gen04" "gen02" "gen16"
## [6,] "gen14" "gen10" "gen17" "gen08"
##
## $rep2
##      [,1]    [,2]    [,3]    [,4]
## [1,] "gen18" "gen20" "gen14" "gen05"
## [2,] "gen08" "gen02" "gen06" "gen21"
## [3,] "gen19" "gen01" "gen16" "gen24"
## [4,] "gen07" "gen04" "gen11" "gen17"
## [5,] "gen09" "gen13" "gen22" "gen12"
## [6,] "gen23" "gen15" "gen03" "gen10"
##
## $rep3
##      [,1]    [,2]    [,3]    [,4]
## [1,] "gen01" "gen06" "gen17" "gen22"
## [2,] "gen18" "gen09" "gen23" "gen08"
## [3,] "gen15" "gen05" "gen07" "gen02"
## [4,] "gen13" "gen03" "gen19" "gen14"
## [5,] "gen04" "gen20" "gen24" "gen12"
## [6,] "gen16" "gen21" "gen11" "gen10"
```

After implementation in the field and the harvest, the yield data are gathered and typed in:

```
dbook <- read.table("AlphaLatticeOat.csv", sep = " ", header = TRUE)
str(dbook)
```
```
## 'data.frame':    72 obs. of  5 variables:
##  $ PLOT : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ REP  : chr  "R1" "R1" "R1" "R1" ...
##  $ BLOCK: chr  "B1" "B1" "B1" "B1" ...
##  $ GEN  : chr  "G11" "G04" "G05" "G22" ...
##  $ YIELD: num  4.12 4.45 5.88 4.58 4.65 ...
```
```
## Function for lattice analysis in agricolae:
str(PBIB.test)
```
```
## function (block, trt, replication, y, k, method = c("REML", "ML", "VC"),
##     test = c("lsd", "tukey"), alpha = 0.05, console = FALSE, group = TRUE)
```
```
## function (block, trt, replication, y, k, method = c('REML', 'ML', 'VC'),
## test = c('lsd', 'tukey'), alpha = 0.05, console = FALSE, group = TRUE)
```

the details of this analysis rely on Mixed Linear Models (MLM). We do not go into the details here. This will be covered in 'Advanced Level'

```
# require(nlme) if method = REML or LM in PBIB.test; and require(MASS) if
# method=VC
model <- PBIB.test(dbook$BLOCK, dbook$GEN, dbook$REP, dbook$YIELD, k = 4, method = c("REML"),
    test = "tukey", alpha = 0.05, group = TRUE)
```

```
##
## <<< to see the objects: means, comparison and groups. >>>
```

`replication` and `block in replications` are considered as random factors. `Genotypes` are considered as fixed factors. This is the *intrablock analysis*.

```
ls(model)
```

```
## [1] "ANOVA"      "comparison" "Fstat"      "groups"     "means"
## [6] "method"     "model"      "parameters" "statistics" "vartau"
```

Test for differences among genotypes:

```
model$ANOVA
```

```
## Analysis of Variance Table
##
## Response: dbook$YIELD
##            Df Sum Sq Mean Sq F value    Pr(>F)
## dbook$GEN 23 10.679 0.46429  5.4478 9.749e-06 ***
## Residuals 31  2.642 0.08523
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Adjusted means:

```
model$means
```

```
##       dbook$YIELD dbook$YIELD.adj       SE r       std    Min    Max      Q25
## G01      5.162500        5.107700 0.2760760 3 0.53370869 4.6512 5.7161 4.88570
## G02      4.514200        4.478532 0.2760760 3 0.57417098 4.0510 5.1566 4.19300
## G03      3.343067        3.499200 0.2760760 3 0.45630094 2.8873 3.7999 3.11465
## G04      4.400667        4.490095 0.2760760 3 0.04328907 4.3599 4.4461 4.37795
## G05      5.064367        5.037210 0.2760159 3 0.84064177 4.1972 5.8757 4.65870
## G06      4.710333        4.536662 0.2760159 3 0.46385272 4.2474 5.1751 4.47795
## G07      4.129933        4.111136 0.2760760 3 0.51036090 3.6096 4.6297 3.88005
## G08      4.324533        4.527634 0.2760760 3 0.58404226 3.9821 4.9989 3.98735
## G09      3.611600        3.502181 0.2760159 3 0.60614974 3.1407 4.2955 3.26965
## G10      4.388933        4.373200 0.2760760 3 0.44959887 4.0875 4.9057 4.13055
## G11      4.384667        4.283264 0.2760760 3 0.64120061 3.9205 5.1163 4.01885
## G12      4.914433        4.755276 0.2760760 3 0.64134136 4.1746 5.3127 4.71530
## G13      4.829400        4.757913 0.2760159 3 0.61921457 4.2530 5.4840 4.50210
## G14      4.558433        4.775662 0.2760159 3 0.18595769 4.3887 4.7572 4.45905
## G15      4.893300        4.969111 0.2760159 3 0.20654566 4.6783 5.0902 4.79485
## G16      4.726067        4.730131 0.2760760 3 0.50189379 4.3852 5.3024 4.43790
## G17      4.728467        4.602612 0.2760159 3 0.37897601 4.3234 5.0744 4.55550
## G18      4.440367        4.361692 0.2760760 3 0.58655208 3.9737 5.0988 4.11115
## G19      4.865633        4.840328 0.2760159 3 0.39835065 4.5552 5.3148 4.64105
## G20      3.784867        4.039985 0.2760159 3 0.20878353 3.6056 4.0141 3.67025
## G21      4.815733        4.795007 0.2760159 3 0.50347514 4.4130 5.3802 4.53350
## G22      4.638333        4.527545 0.2760159 3 0.43173136 4.2397 5.0969 4.40905
## G23      4.136200        4.252449 0.2760159 3 0.23152540 3.8721 4.3042 4.05220
## G24      4.142367        4.153874 0.2760760 3 0.72608882 3.5655 4.9577 3.73470
##        Q50     Q75
## G01 5.1202 5.41815
## G02 4.3350 4.74580
## G03 3.3420 3.57095
## G04 4.3960 4.42105
```

```
## G05 5.1202 5.49795
## G06 4.7085 4.94180
## G07 4.1505 4.39010
## G08 3.9926 4.49575
## G09 3.3986 3.84705
## G10 4.1736 4.53965
## G11 4.1172 4.61675
## G12 5.2560 5.28435
## G13 4.7512 5.11760
## G14 4.5294 4.64330
## G15 4.9114 5.00080
## G16 4.4906 4.89650
## G17 4.7876 4.93100
## G18 4.2486 4.67370
## G19 4.7269 5.02085
## G20 3.7349 3.87450
## G21 4.6540 5.01710
## G22 4.5784 4.83765
## G23 4.2323 4.26825
## G24 3.9039 4.43080
```

Groups of adj. means based on Tukey test:

```
model$groups
```

```
##       dbook$YIELD.adj groups
## G01          5.107700      a
## G05          5.037210      a
## G15          4.969111      a
## G19          4.840328     ab
## G21          4.795007     ab
## G14          4.775662     ab
## G13          4.757913     ab
## G12          4.755276     ab
## G16          4.730131     ab
## G17          4.602612     ab
## G06          4.536662     ab
## G08          4.527634     ab
## G22          4.527545     ab
## G04          4.490095     ab
## G02          4.478532     ab
## G10          4.373200     ab
## G18          4.361692     ab
## G11          4.283264     ab
## G23          4.252449     ab
## G24          4.153874     ab
## G07          4.111136     ab
## G20          4.039985     ab
## G09          3.502181      b
## G03          3.499200      b
```

```
plot(model, las = 2)
```
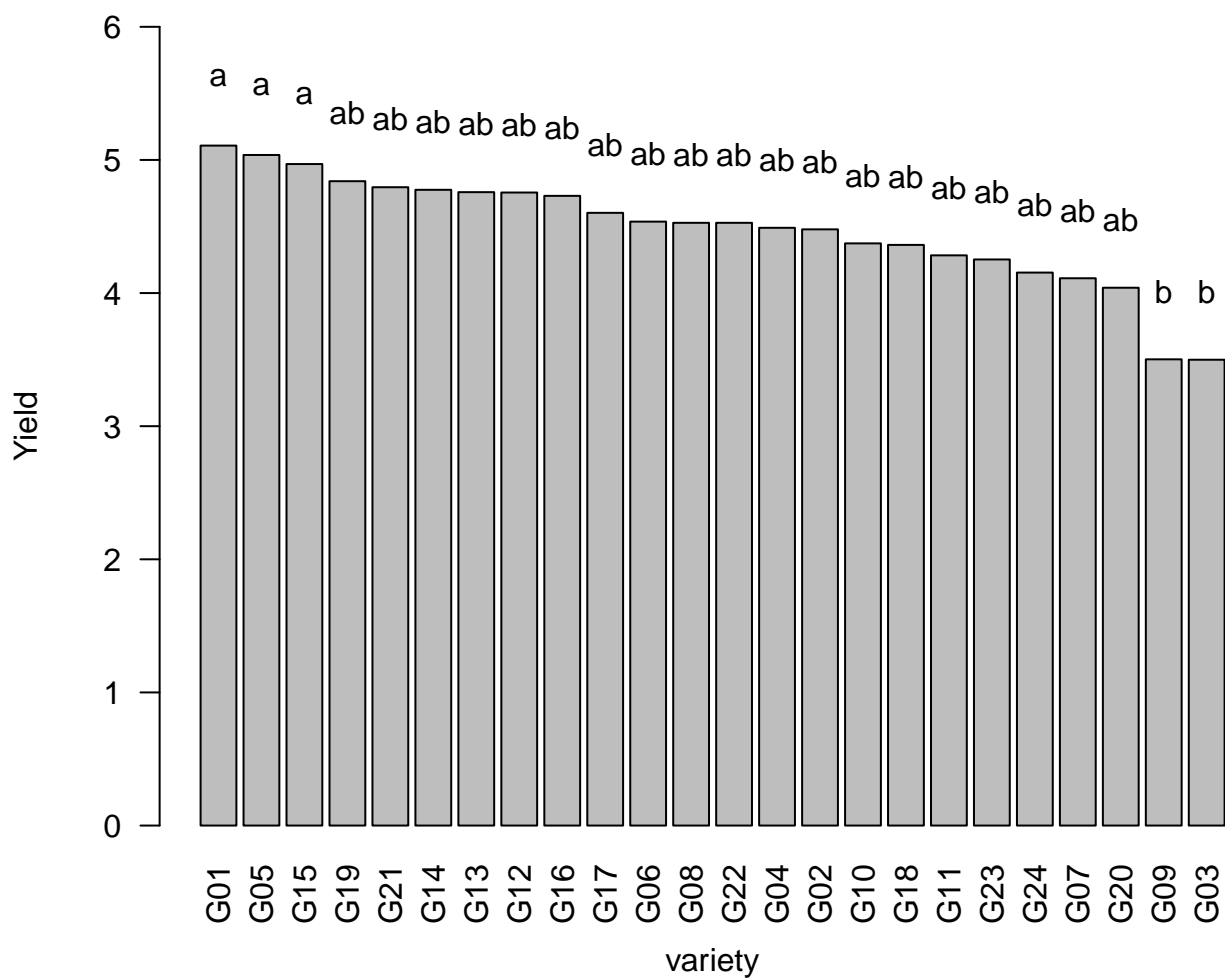
```
##
## Warning  values plot is not adjusted
```
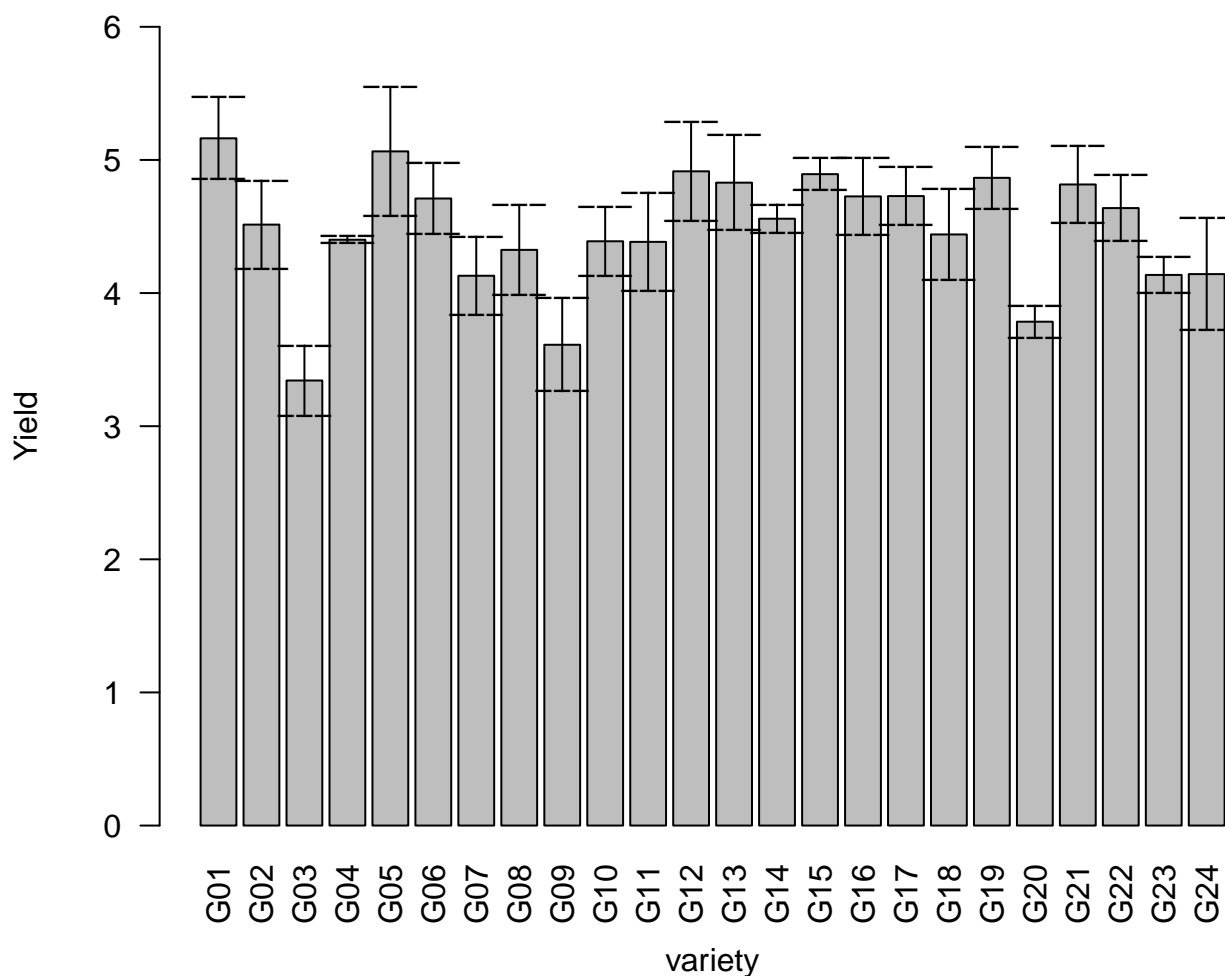
Groups and Range
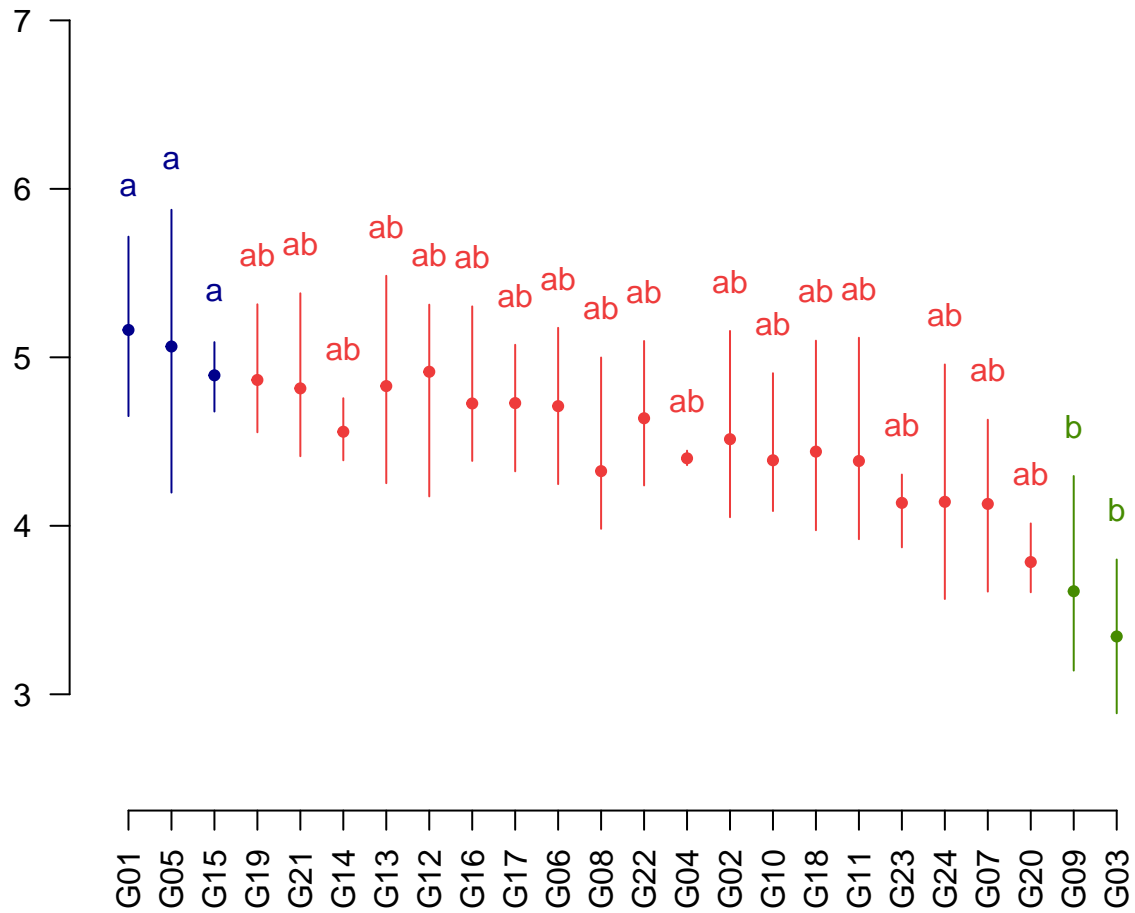

```
## How to plot the Alpha_lattice
x11()
bar.group(model$groups, ylim = c(0, 6), xlab = "variety", ylab = "Yield", las = 2)
```

```
x11()
bar.err(model$means, ylim = c(0, 6), xlab = "variety", ylab = "Yield", las = 2)
```

```
# Save the means in a excel file:
write.xlsx(as.data.frame(model$means %>%
    mutate(Genotype = row.names(.))), "AdjustedMeans.xlsx", overwrite = TRUE)


x11()
model$means %>%
    select("dbook$YIELD.adj", "dbook$YIELD") %>%
    rename(Adj = "dbook$YIELD.adj", Obs = "dbook$YIELD") %>%
    ggplot() + aes(y = Adj, x = Obs) + geom_point(size = 2) + geom_abline(slope = 1,
    intercept = 0) + xlim(3.4, 5.3) + ylim(3.4, 5.3) + labs(x = "Obs. yield (tons/ha)",
    y = "Adjust. yield (tons/ha)") + theme_minimal()
```
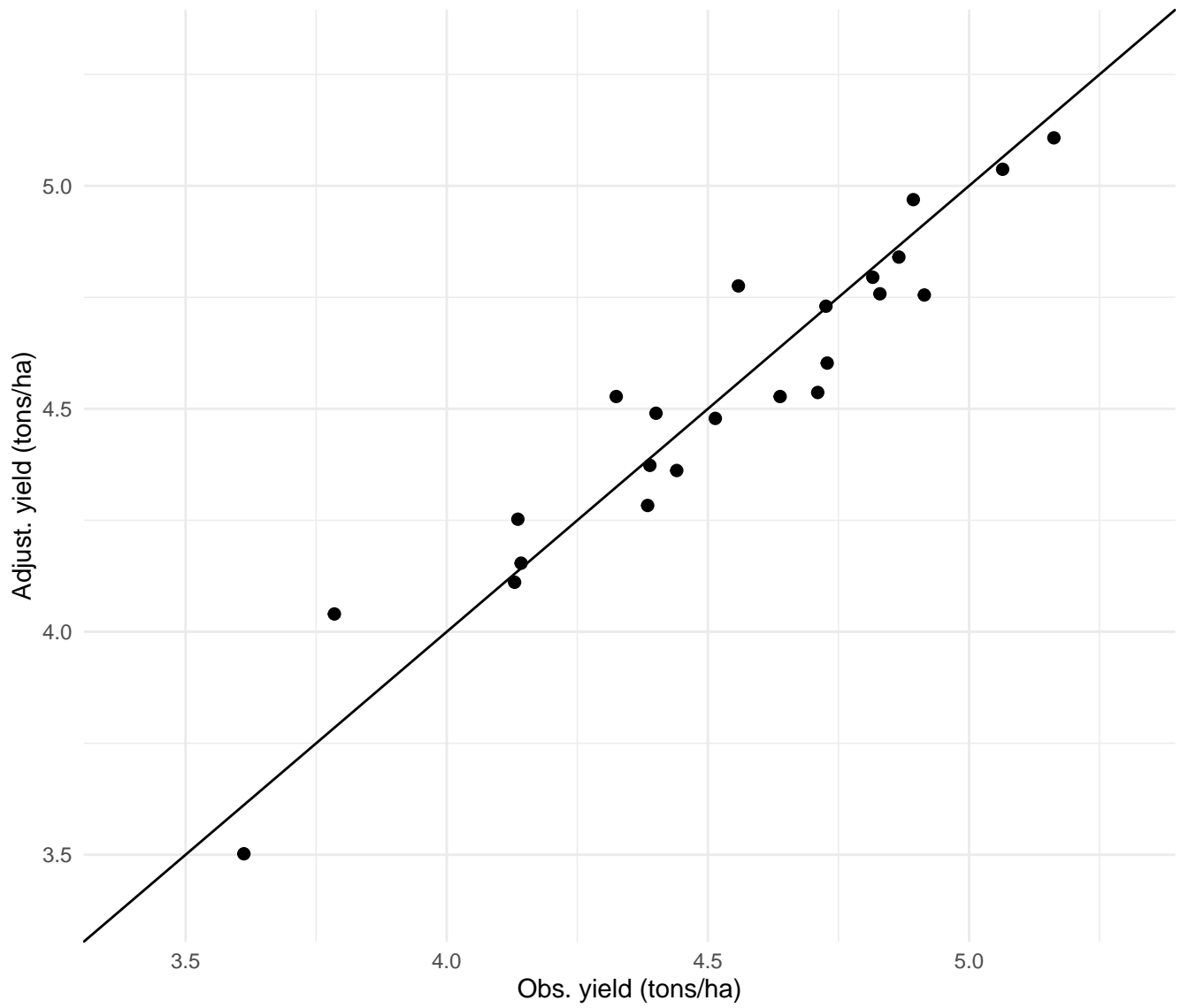
What is your opinion when comparing observed means/raw data and adjusted means ?.