

Augmented Field Trial Designs

An R tutorial on augmented field trial designs and how to adjust their values

Derek Michael Wright derek.wright@usask.ca www.dblogr.com/

2023-01-05

```
# devtools::install_github("derekmichaelwright/agData")
library(agData) # Loads: tidyverse, ggpibr, ggbeeswarm, ggrepel
library(GGally) # ggpairs()
library(latex2exp) # TeX()
```

Introduction to Augmented Designs

Augmented designs were developed as a way of controlling error in plant breeding trials which often have many genotypes that need to be tested, and limited seed or resources to do proper replications of all material (Federer, 1956). Therefore, in order to control for the heterogeneity that exists within a field, a set of check cultivars are replicated in each *block*. The block effects and error estimated from the replicated checks, is then used to adjust the values of each new genotype being tested.

Type I - Augmented RCBD

Federer, W. (1956) Augmented (or hoonuiaku) designs. *Hawaiian Planters Record*.

Type II - Modified

Lin, C.S. & Poushinsky, G. (1985) A modified augmented design (type 2) for rectangular plots. *Canadian journal of plant science*. 65(3): 743-749.

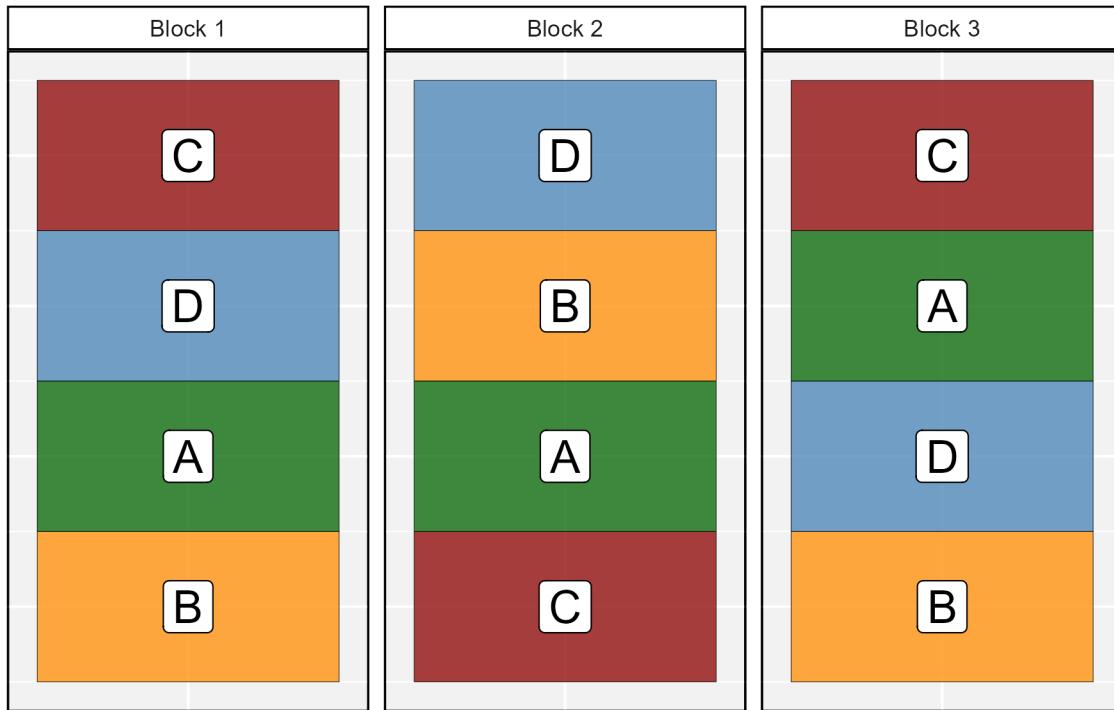
You, F.M., Song, Q., Jia, G., Cheng, Y., Duguid, S., Booker, H. and Cloutier, S. (2016) Estimation of genetic parameters and their sampling variances for quantitative traits in the type 2 modified augmented design. *The Crop Journal*. 4(2): 107-118.

Table S1 – The raw phenotypic data of a population with 243 RILs derived from a cross between ‘CDC Bethune’ and ‘Macbeth’ (BM) for the case study

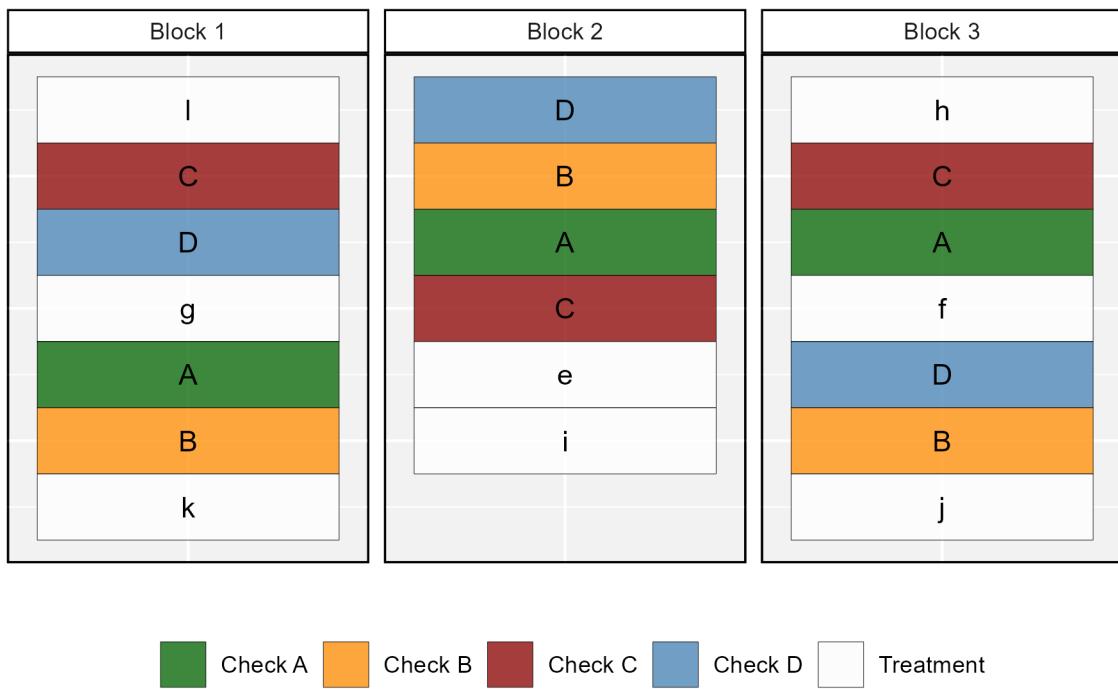
Type I - Augmented RCBD

RCBD

In a **Randomized Complete Block Design (RCBD)**, every genotype is present in each block, making each block a replicate with entries randomized within. As such, a field trial of 4 genotypes (A, B, C, D), with 4 replicates each (4 blocks) will look something like this:



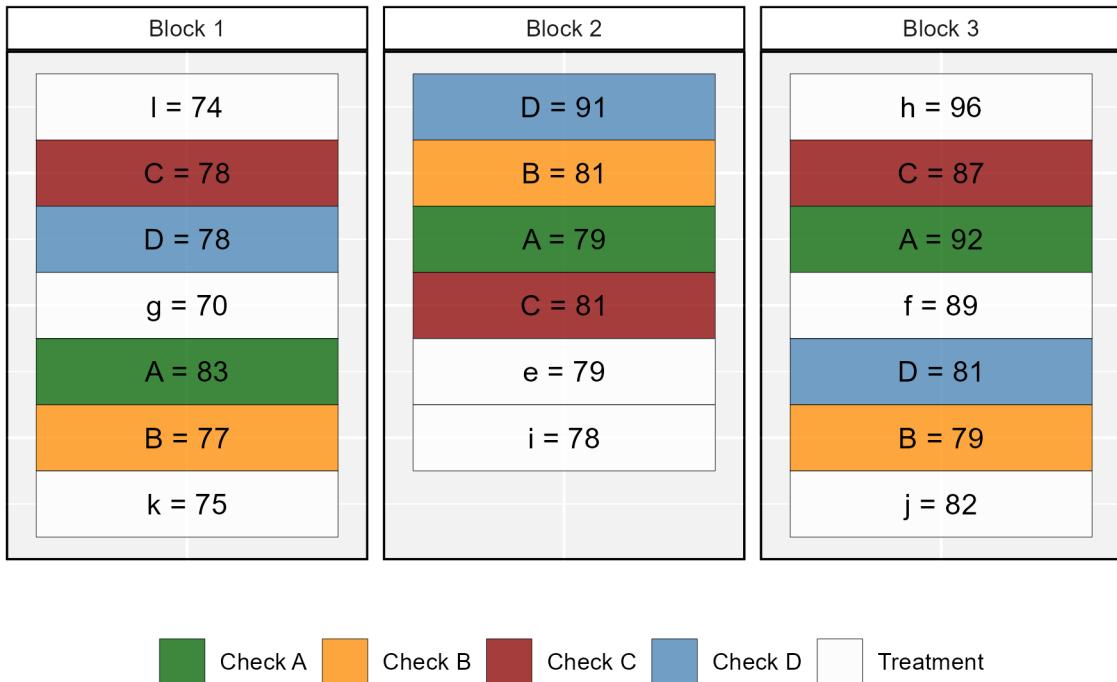
In an **Augmented RCBD**, the idea is to have a small set of check varieties which are present in each block, *i.e.*, an RCBD, augmented with unreplicated test varieties. *E.g.*, lets say we have 8 test varieties named: e, f, g, h, i, j, k, l. Our field trial might look like this:



Federer Example

Now lets explore the data analysis for such a trial. This data comes from a trial on Field 78 at Pioneer Mill Sugar Company, 1931 (**Federer, 1956**), which had 3 different level ditches (blocks), and the recorded data was in tons of sugar cane per acre (TCA).

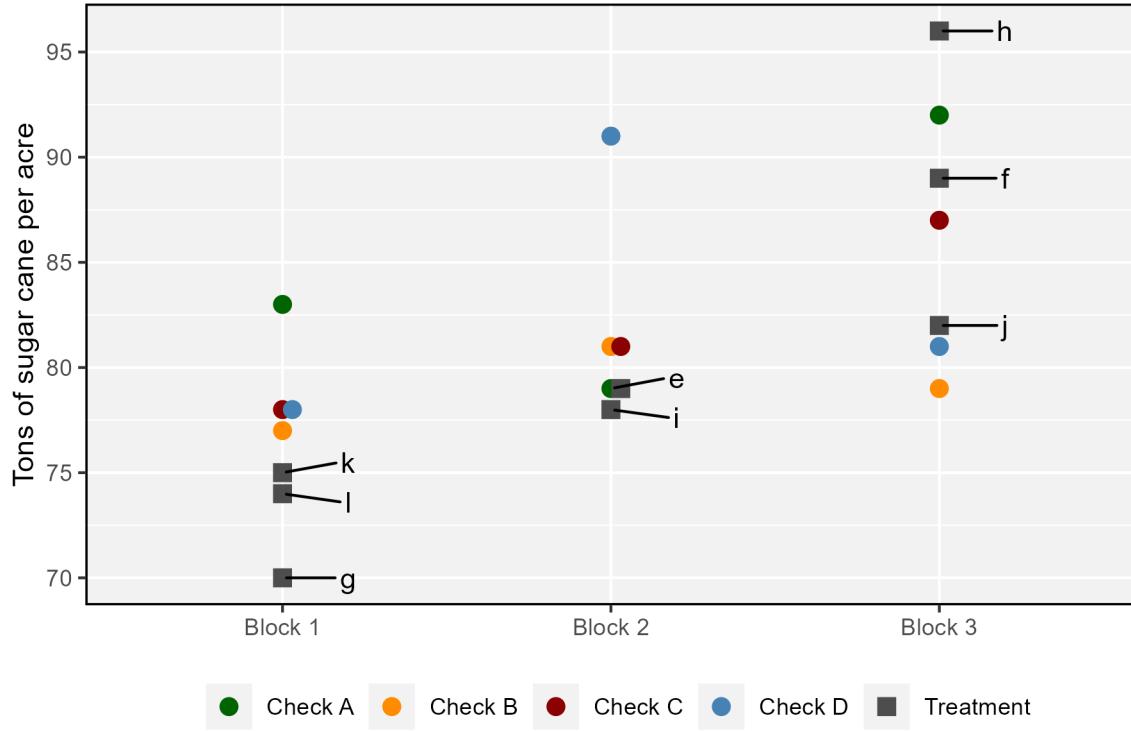
Tons of sugar cane per acre



█ Check A
 █ Check B
 █ Check C
 █ Check D
 █ Treatment

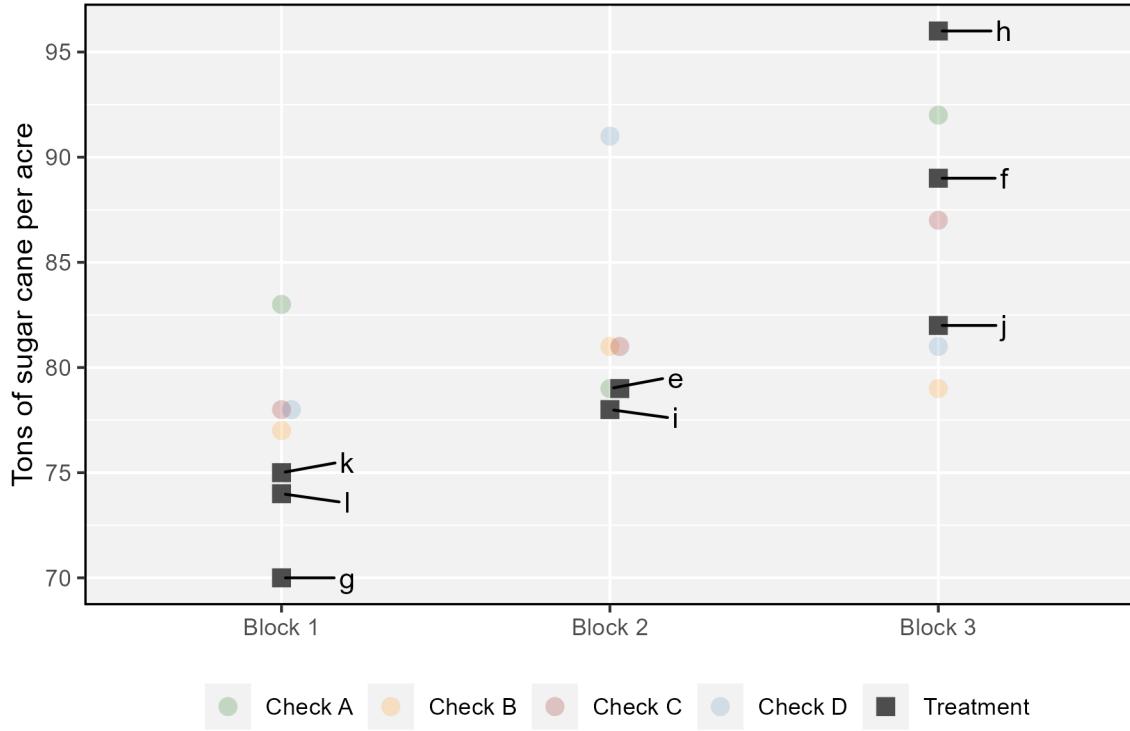
```

# Prep data
checks <- c("Check A", "Check B", "Check C", "Check D")
genotypes <- c("A", "B", "C", "D",
              "e", "f", "g", "h", "i", "j", "k", "l")
myColors <- c("darkgreen", "darkorange", "darkred", "steelblue", "white")
dd <- read.csv("data_augmented_designs_1.csv") %>%
  mutate(Type = ifelse(Genotype %in% genotypes[1:4], Genotype, "Treatment"),
         Type = plyr::mapvalues(Type, genotypes[1:4], checks),
         Type = factor(Type, levels = c(checks, "Treatment")),
         Block = factor(paste("Block", Block)),
         Genotype = factor(Genotype, levels = genotypes))
# Plot data
mp <- ggplot(dd, aes(x = "", y = Row)) +
  geom_tile(aes(fill = Type), color = "black", alpha = 0.75) +
  geom_text(aes(label = paste(Genotype, Yield, sep = " = "))) +
  facet_grid(. ~ Block) +
  scale_fill_manual(name = NULL, values = myColors) +
  scale_y_reverse() +
  theme_agData(legend.position = "bottom",
               axis.text.y = element_blank(),
               axis.ticks = element_blank()) +
  labs(title = "Tons of sugar cane per acre", y = NULL, x = NULL)
ggsave("aug_01_01.png", mp, width = 6, height = 4)
  
```



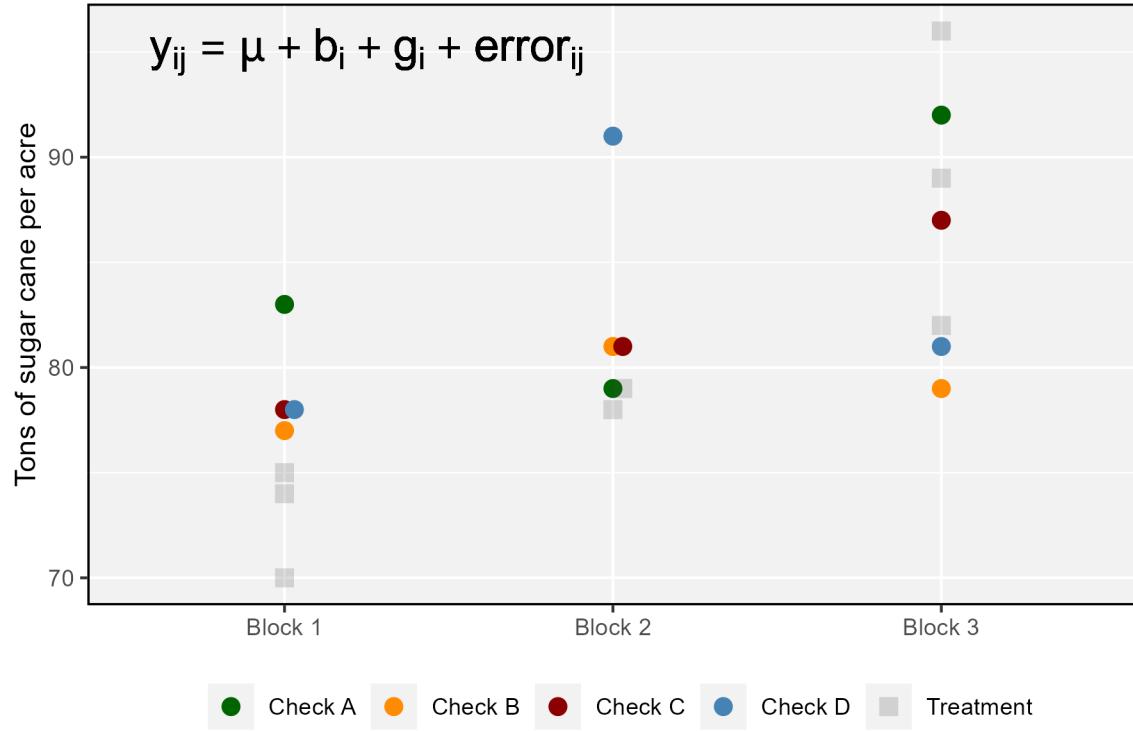
```
# Prep data
myColors <- c("darkgreen", "darkorange", "darkred", "steelblue", "grey30")
xt <- dd %>% filter(Type == "Treatment")
xc <- dd %>% filter(Type != "Treatment")
# Plot
mp <- ggplot(dd, aes(x = Block, y = Yield)) +
  geom_beeswarm(aes(color = Type, shape = Type), size = 3) +
  geom_text_repel(data = xt, aes(label = Genotype), nudge_x = 0.2) +
  scale_color_manual(name = NULL, values = myColors) +
  scale_shape_manual(name = NULL, values = c(16,16,16,16,15)) +
  scale_y_continuous(breaks = seq(70, 95, by = 5)) +
  theme_agData(legend.position = "bottom") +
  labs(y = "Tons of sugar cane per acre", x = NULL)
ggsave("aug_01_02.png", mp, width = 6, height = 4)
```

But first lets focus on the unreplicated test varieties.



```
# Plot
mp <- ggplot(dd, aes(x = Block, y = Yield)) +
  geom_beeswarm(aes(color = Type, shape = Type, alpha = Type), size = 3) +
  geom_text_repel(data = xt, aes(label = Genotype), nudge_x = 0.2) +
  scale_color_manual(name = NULL, values = myColors) +
  scale_shape_manual(name = NULL, values = c(16,16,16,16,15)) +
  scale_alpha_manual(name = NULL, values = c(0.2,0.2,0.2,0.2,1)) +
  scale_y_continuous(breaks = seq(70, 95, by = 5)) +
  theme_agData(legend.position = "bottom") +
  labs(y = "Tons of sugar cane per acre", x = NULL)
ggsave("aug_01_03.png", mp, width = 6, height = 4)
```

Clearly there are strong block effects, which is problematic since our test varieties are unreplicated, making comparisons among genotypes from different blocks potentially unreliable. *E.g.*, is genotype **j** a higher yielding genotype than **k**? Or, is it just in a block where all genotypes yield higher? By running an ANOVA on the replicated check varieties, we can get an estimate of the block effects and adjust our data accordingly.



```
# Plot
eq <- TeX("$y_{ij}=\mu+b_i+g_j+\text{error}_{ij}$")
mp <- ggplot(dd, aes(x = Block, y = Yield)) +
  geom_beeswarm(aes(color = Type, shape = Type, alpha = Type), size = 3) +
  geom_text(x = 1.25, y = 95, label = eq, parse = T, size = 6) +
  scale_color_manual(name = NULL, values = myColors) +
  scale_shape_manual(name = NULL, values = c(16,16,16,16,15)) +
  scale_alpha_manual(name = NULL, values = c(1,1,1,1,0.2)) +
  ylim(c(70,96)) +
  theme_agData(legend.position = "bottom") +
  labs(y = "Tons of sugar cane per acre", x = NULL)
ggsave("aug_01_04.png", mp, width = 6, height = 4)
```

$$y_{ij} = \mu + b_i + g_j + \text{error}_{ij}$$

where:

- μ = Mean
- b_i = Block effect
- g_j = Genotype effect
- error_{ij} = random error

Calculate Block Effects

```

# Run ANOVA on checks
fit <- lm(Yield ~ Block + Genotype, data = xc)
aov(fit)

## Call:
##   aov(formula = fit)
##
## Terms:
##           Block  Genotype Residuals
## Sum of Squares  69.50000  52.91667 161.83333
## Deg. of Freedom      2          3          6
##
## Residual standard error: 5.193479
## Estimated effects may be unbalanced

# Coefficients
coef(fit)

## (Intercept) BlockBlock 2 BlockBlock 3     GenotypeB     GenotypeC     GenotypeD
## 81.416667    4.000000    5.750000    -5.666667    -2.666667    -1.333333

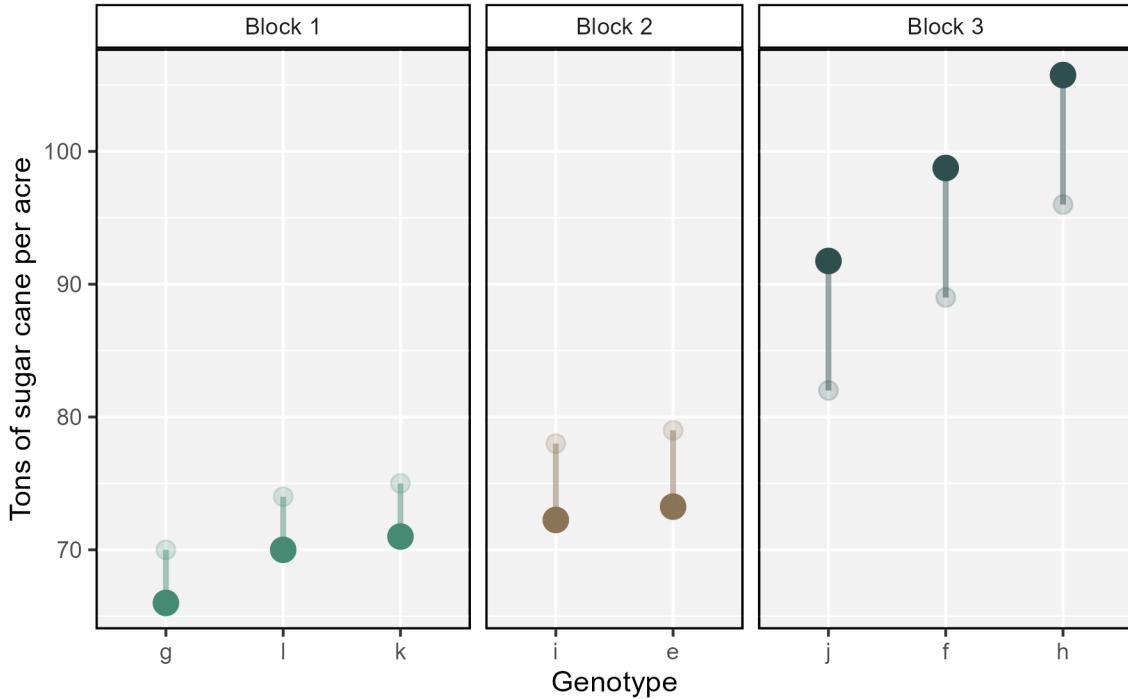
# Block effects
blockeffects <- c(coef(fit)[2:3], sum(-as.numeric(coef(fit)[2:3])))
names(blockeffects)[3] <- "Block3"
blockeffects

## BlockBlock 2 BlockBlock 3      Block3
##        4.00        5.75       -9.75

SSb = 69.5
SSt = 69.5 + 214.75 = 284.25

```

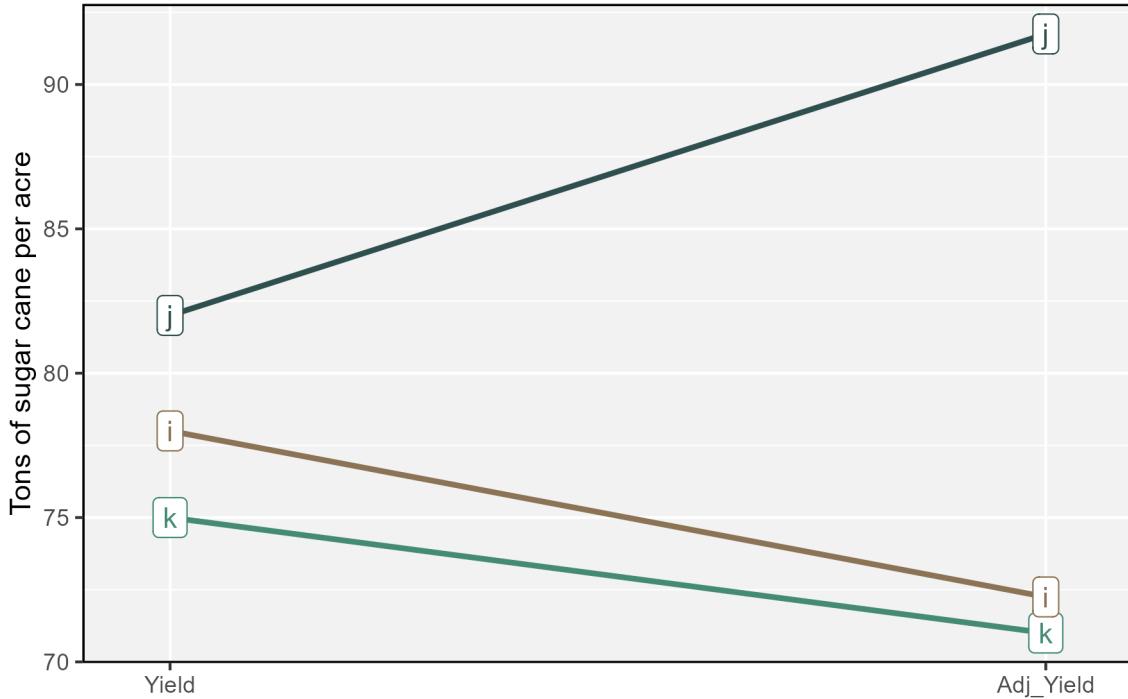
Block Effect Adjustments



```
# Prep data
myColors <- c("aquamarine4", "burlywood4", "darkslategrey")
blocks <- c("Block 1", "Block 2", "Block 3")
dd <- dd %>%
  mutate(BlockEffect = plyr::mapvalues(Block, blocks, blockeffects),
    BlockEffect = as.numeric(as.character(BlockEffect)),
    Adj_Yield = Yield - as.numeric(BlockEffect))
xt <- dd %>% filter(Type == "Treatment") %>% arrange(Yield) %>%
  mutate(Genotype = factor(Genotype, levels = unique(.\$Genotype)))
# Plot
mp <- ggplot(xt, aes(x = Genotype, color = Block)) +
  geom_point(aes(y = Yield), size = 3, alpha = 0.2) +
  geom_point(aes(y = Adj_Yield), size = 4) +
  geom_errorbar(aes(ymax = Yield, ymin = Adj_Yield),
    size = 1, width = 0, alpha = 0.5) +
  facet_grid(. ~ Block, scales = "free_x", space = "free_x") +
  scale_color_manual(values = myColors) +
  theme_agData(legend.position = "none") +
  labs(title = "Block Effect Adjustments", y = "Tons of sugar cane per acre")
ggsave("aug_01_05.png", mp, width = 6, height = 4)
```

This is important since the block effects can change the interpretation of the results. *E.g.*, lets compare a few genotypes before and after the adjustments.

Block Effect Adjustments



```
# Prep data
xt <- xt %>%
  filter(Genotype %in% c("i", "j", "k")) %>%
  select(Genotype, Yield, Adj_Yield) %>%
  gather(Trait, Value, Yield, Adj_Yield) %>%
  mutate(Trait = factor(Trait, levels = c("Yield", "Adj_Yield")))
# Plot
mp <- ggplot(xt, aes(x = Trait, y = Value,
                      group = Genotype, color = Genotype)) +
  geom_line(size = 1) +
  geom_label(aes(label = Genotype)) +
  scale_color_manual(values = myColors) +
  coord_cartesian(xlim = c(1.5, 1.5)) +
  theme_agData(legend.position = "none") +
  labs(title = "Block Effect Adjustments",
       y = "Tons of sugar cane per acre", x = NULL)
ggsave("aug_01_06.png", mp, width = 6, height = 4)
```

augmentedRCBD Package

We can also do this with the R package `augmentedRCBD` which contains a function `augmentedRCBD` that can carry make these adjustments.

```
# devtools::install_github("aravind-j/augmentedRCBD")
library(augmentedRCBD)
```

```

out <- augmentedRCBD(block = dd$Block, treatment = dd$Genotype, y = dd$Yield,
                      checks = c("A", "B", "C", "D"), group = F)

## 
## Augmented Design Details
## =====
## 
## Number of blocks      "3"
## Number of treatments   "12"
## Number of check treatments "4"
## Number of test treatments "8"
## Check treatments       "A, B, C, D"
## 
## ANOVA, Treatment Adjusted
## =====
##                                     Df Sum Sq Mean Sq F value Pr(>F)
## Block (ignoring Treatments)        2 360.1 180.04  6.675 0.0298 *
## Treatment (eliminating Blocks)    11 285.1 25.92   0.961 0.5499
## Treatment: Check                 3   52.9 17.64   0.654 0.6092
## Treatment: Test and Test vs. Check 8 232.2 29.02   1.076 0.4779
## Residuals                         6 161.8 26.97
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## ANOVA, Block Adjusted
## =====
##                                     Df Sum Sq Mean Sq F value Pr(>F)
## Treatment (ignoring Blocks)     11 575.7 52.33   1.940  0.215
## Treatment: Check                3   52.9 17.64   0.654  0.609
## Treatment: Test                 7 505.9 72.27   2.679  0.125
## Treatment: Test vs. Check      1   16.9 16.88   0.626  0.459
## Block (eliminating Treatments)  2   69.5 34.75   1.288  0.342
## Residuals                         6 161.8 26.97
## 
## Treatment Means
## =====
##      Treatment Block   Means      SE r Min Max Adjusted Means
## 1          A      84.66667 3.844188 3 79 92      84.66667
## 2          B      79.00000 1.154701 3 77 81      79.00000
## 3          C      82.00000 2.645751 3 78 87      82.00000
## 4          D      83.33333 3.929942 3 78 91      83.33333
## 5          e Block 2 79.00000      NA 1 79 79      78.25000
## 6          f Block 3 89.00000      NA 1 89 89      86.50000
## 7          g Block 1 70.00000      NA 1 70 70      73.25000
## 8          h Block 3 96.00000      NA 1 96 96      93.50000
## 9          i Block 2 78.00000      NA 1 78 78      77.25000
## 10         j Block 3 82.00000      NA 1 82 82      79.50000
## 11         k Block 1 75.00000      NA 1 75 75      78.25000
## 12         l Block 1 74.00000      NA 1 74 74      77.25000
## 
## Coefficient of Variation
## =====
## 6.372367

```

```

## 
## Overall Adjusted Mean
## =====
## 81.0625
##
## Standard Errors
## =====
##                                     Std. Error of Diff. CD (5%)
## Control Treatment Means           4.240458 10.37603
## Two Test Treatments (Same Block) 7.344688 17.97180
## Two Test Treatments (Different Blocks) 8.211611 20.09309
## A Test Treatment and a Control Treatment 6.704752 16.40594

```

```
out[[2]] # Adjusted Means
```

	Treatment	Block	Means	SE	r	Min	Max	Adjusted Means
## 1	A		84.66667	3.844188	3	79	92	84.66667
## 2	B		79.00000	1.154701	3	77	81	79.00000
## 3	C		82.00000	2.645751	3	78	87	82.00000
## 4	D		83.33333	3.929942	3	78	91	83.33333
## 5	e	Block 2	79.00000	NA	1	79	79	78.25000
## 6	f	Block 3	89.00000	NA	1	89	89	86.50000
## 7	g	Block 1	70.00000	NA	1	70	70	73.25000
## 8	h	Block 3	96.00000	NA	1	96	96	93.50000
## 9	i	Block 2	78.00000	NA	1	78	78	77.25000
## 10	j	Block 3	82.00000	NA	1	82	82	79.50000
## 11	k	Block 1	75.00000	NA	1	75	75	78.25000
## 12	l	Block 1	74.00000	NA	1	74	74	77.25000

```
out[[5]] # Block Effects
```

```

## Block 1 Block 2 Block 3
## -3.25    0.75    2.50

```

```
out[[6]] # Genotype Effects
```

	A	B	C	D	e	f	g	h
##	3.604167	-2.062500	0.937500	2.270833	-2.812500	5.437500	-7.812500	12.437500
##	i	j	k		l			
##	-3.812500	-1.562500	-2.812500	-3.812500				

```
out[[7]] # Standard Errors
```

	Std. Error of Diff.	CD (5%)
## Control Treatment Means	4.240458	10.37603
## Two Test Treatments (Same Block)	7.344688	17.97180
## Two Test Treatments (Different Blocks)	8.211611	20.09309
## A Test Treatment and a Control Treatment	6.704752	16.40594

```
out[[8]] # Overall Mean
```

```
## [1] 81.0625
```

```
out[[9]] # CV
```

```
## [1] 6.372367
```

To help better understand, lets do these calculations ourselves.

Block Effects:

$$b_i = \frac{1}{n_c} * \left(\sum y_{bi} - \sum \bar{y}_c - \sum y_{ti} \right)$$

where

- b_i = block effect
- n_c = number of check varieties
- $\sum y_{bi}$ = sum of all measurements within block
- $\sum \bar{y}_c$ = sum of check means
- $\sum y_{ti}$ = measurement for individual treatment within block

Calculate Block 1 effect: $i = 1$

- $n_c = 4$
- $\sum y_{b1} = 74 + 78 + 78 + 70 + 83 + 77 + 75 = 535$
- $\sum \bar{y}_c = 84.67 + 79.00 + 82.00 + 83.33 = 329$
- $\sum y_{t1} = 74 + 70 + 75 = 219$
- $b_1 = \frac{1}{4} * (535 - 329 - 219) = -3.25$

Calculate Block 2 effect: $i = 2$

- $\sum y_{b2} = 91 + 81 + 79 + 81 + 79 + 78 = 489$
- $\sum y_{t2} = 79 + 78 = 157$
- $b_2 = \frac{1}{4} * (489 - 329 - 157) = 0.75$

Calculate Block 3 effect: $i = 3$

- $\sum y_{b3} = 96 + 87 + 92 + 89 + 81 + 79 + 82 = 606$
- $\sum y_{t3} = 96 + 89 + 82 = 267$
- $b_3 = \frac{1}{4} * (606 - 329 - 267) = 2.5$

```
# Data prep
y_c <- dd %>% filter(Genotype %in% c("A", "B", "C", "D")) %>%
  group_by(Genotype) %>% summarise(Mean = mean(Yield))
y_c
```

```

## # A tibble: 4 x 2
##   Genotype Mean
##   <fct>    <dbl>
## 1 A         84.7
## 2 B         79.0
## 3 C         82.0
## 4 D         83.3

# Block1 effects
y_t1 <- dd$Yield[dd$Block=="Block 1" & dd>Type=="Treatment"]
y_b1 <- dd$Yield[dd$Block=="Block 1"]
b1 <- ( 1/4 ) * ( sum(y_b1, -y_c$Mean, -y_t1) )

# Block2 effects
y_t2 <- dd$Yield[dd$Block=="Block 2" & dd>Type=="Treatment"]
y_b2 <- sum(dd$Yield[dd$Block=="Block 2"])
b2 <- ( 1/4 ) * ( sum(y_b2, -y_c$Mean, -y_t2) )

# Block3 effects
y_t3 <- dd$Yield[dd$Block=="Block 3" & dd>Type=="Treatment"]
y_b3 <- sum(dd$Yield[dd$Block=="Block 3"])
b3 <- ( 1/4 ) * ( sum(y_b3, -y_c$Mean, -y_t3) )
bb <- data.frame(Block = c("Block 1","Block 2","Block 3"),
                  Block_Effect = c(b1, b2, b3))
bb

```

```

##      Block Block_Effect
## 1 Block 1     -3.25
## 2 Block 2      0.75
## 3 Block 3      2.50

```

Mean Effect:

$$m = \frac{1}{n_c + n_t} * \left(\sum y - (n_b - 1) * \sum \bar{y}_c - \sum (n_{ti} * b_i) \right)$$

- m = mean effect
- n_c = number of check varieties
- n_t = number of treatments (test varieties)
- $\sum y$ = sum of all measurements
- n_b = number of blocks
- $\sum \bar{y}_c$ = sum of check means
- n_{ti} = number of treatments (test varieties) within block
- b_i = block effect

Calculate mean effect:

- $n_c = 4$
- $n_t = 8$
- $\sum y = 74 + 78 + 78...81 + 79 + 82 = 1630$
- $n_b = 3$
- $\sum \bar{y}_c = 84.67 + 79.00 + 82.00 + 83.33 = 329$
- $\sum (b_i * n_{ti}) = (3 * (-3.25) + 2 * (0.75) + 3 * (2.5)) = -0.75$
- $m = \frac{1}{4+8} * (1630 - (3 - 1) * 329 - (-0.75)) = 81.0625$

```
m <- ( 1/(4+8) ) * ( sum(dd$Yield) - (3 - 1) * sum(y_c$Mean) - sum(3 * b1 + 2 * b2 + 3 * b3) )  
m
```

```
## [1] 81.0625
```

Calculate genotype effects

```
dd <- dd %>%  
  left_join(y_c, by = "Genotype") %>%  
  left_join(bb, by = "Block") %>%  
  mutate(AdjMean = ifelse(Type == "Treatment", Yield - Block_Effect, Mean),  
        Genotype_Effect = AdjMean - m)
```

The sum of block effects should add to zero, along with the sum of check effects and treatment effects

```
sum(bb$Block_Effect)
```

```
## [1] 0
```

```
check_Effects <- unique(dd$Genotype_Effect[dd>Type == "Check"])  
treatment_Effects <- dd$Genotype_Effect[dd>Type == "Treatment"]  
sum(check_Effects, treatment_Effects)
```

```
## [1] -4.75
```

Calculate total sum of squares

$$SS_t = \sum y^2 - \frac{(\sum y)^2}{n} = 133652 - \frac{1630^2}{20} = 807$$

```
sum(dd$Yield^2) - (sum(dd$Yield)^2 / 20)
```

```
## [1] 807
```

n_g = number of genotypes

$$SS_t = 1630^2$$

Type II - Modified

Modified augmented design (type II) was developed to account for row and column and heterogeneity. A limitation of the Augmented RCBD method. The Type II design involves the use of a common “control plot” and “subcontrol plots” to attempt to control for row and column and heterogeneity.

```

# Prep data
checks <- data.frame(Check = c("Check1", "Check2", "Check3"),
                      Name = c("CDC Bethune", "Hanley", "Macbeth"))
dd <- read.csv("data_augmented_designs_2.csv") %>%
  arrange(Row, Col) %>%
  rename(MainCheck=Cp..plot.control., SubCheck=Csp..sub.plot.control.) %>%
  mutate(Type = ifelse(MainCheck == 1, "Check1", "Treatment"),
         Type = ifelse(SubCheck == 1, "Check2", Type),
         Type = ifelse(SubCheck == 2, "Check3", Type),
         Type = factor(Type, levels = c("Treatment", "Check1", "Check2", "Check3")),
         Block = paste0(Row, Col),
         Block = plyr::mapvalues(Block, unique(Block), 1:49))

```

Field Plans

This data set includes includes 8 field trials from 2 locations over 4 years.

```

unique(dd$Environment)

## [1] "M2009" "M2010" "M2011" "M2012" "S2009" "S2010" "S2011" "S2012"

# Plotting function
gg_FieldPlan <- function(env) {
  myColors <- c("white", "darkgreen", "darkorange", "darkred")
  xx <- dd %>% filter(Environment == env)
  mp <- ggplot(xx, aes(x = 1, y = SubPlotNum)) +
    geom_tile(aes(fill = Type), color = "black", alpha = 0.5) +
    geom_text(aes(label = Genotype)) +
    facet_grid(Row ~ Col) +
    scale_fill_manual(values = myColors) +
    scale_y_reverse() +
    theme_agData(axis.text = element_blank(),
                axis.ticks = element_blank(),
                legend.position = "none") +
    labs(title = paste("Field plan - ", env), x = NULL, y = NULL)
  ggsave(paste0("aug_02_01_", env, ".png"), mp, width = 10, height = 10)
}
gg_FieldPlan(env = "M2009")
gg_FieldPlan(env = "M2010")
gg_FieldPlan(env = "M2011")
gg_FieldPlan(env = "M2012")
gg_FieldPlan(env = "S2009")
gg_FieldPlan(env = "S2010")
gg_FieldPlan(env = "S2011")
gg_FieldPlan(env = "S2012")

## [1] TRUE

## [1] FALSE

```

Field plan - M2009

1	2	3	4	5	6	7	
BM61 BM136 BM101 CDC Bethune BM79 BM53 BM47	BM59 BM110 BM220 CDC Bethune BM209 BM98 BM240	BM22 Hanley BM130 CDC Bethune Macbeth BM234 BM33	BM111 BM57 BM233 CDC Bethune Hanley Macbeth BM173	Macbeth Hanley BM189 CDC Bethune BM32 BM16 BM41	BM92 BM222 Hanley CDC Bethune BM164 BM152 Macbeth	BM129 BM249 Macbeth CDC Bethune BM131 Hanley BM149	1
BM19 BM187 BM44 CDC Bethune BM225 BM214 BM43	BM70 BM224 CDC Bethune BM231 BM84 BM203	Hanley BM216 BM161 CDC Bethune BM108 Macbeth BM142	BM116 BM99 BM191 CDC Bethune BM193 BM4 BM27	BM172 BM144 BM134 CDC Bethune BM48 BM26 BM14	BM109 BM242 BM107 CDC Bethune BM113	Hanley BM162 BM115 CDC Bethune BM171 Macbeth BM105	2
Hanley BM199 BM186 CDC Bethune BM179 Macbeth BM60	Hanley BM235 BM73 CDC Bethune BM245 BM97 Macbeth	BM5 BM91 CDC Bethune BM30 BM206 BM126	BM112 BM52 BM37 CDC Bethune BM184 BM95 BM226	BM9 BM210 BM36 CDC Bethune BM72 BM154 BM122	BM200 BM62 BM176 CDC Bethune Hanley Macbeth CDC Bethune	BM218 BM38 BM124 CDC Bethune BM185 BM94 BM66	3
BM28 BM248 BM64 CDC Bethune BM198 BM250 BM237	BM181 BM202 CDC Bethune BM212 BM228 BM81	BM34 BM6 BM178 CDC Bethune Hanley BM182 BM12	BM195 BM167 Hanley CDC Bethune BM201 Macbeth BM82	BM133 BM119 Macbeth CDC Bethune BM35 Hanley BM49	BM246 BM1 BM23 CDC Bethune BM90 BM114	Macbeth Hanley BM158 CDC Bethune BM83 BM45 BM85	4
BM17 BM55 BM63 CDC Bethune BM11 Hanley Macbeth	BM65 BM180 BM145 CDC Bethune BM2 BM183 BM241	BM213 BM252 BM88 CDC Bethune BM106 BM20 BM8	BM58 BM227 BM18 CDC Bethune BM232 BM190 BM51	BM196 BM239 BM100 CDC Bethune BM74 BM25 BM211	BM244 BM7 BM137 CDC Bethune BM223 BM118 BM160	BM215 BM128 BM135 CDC Bethune BM56 BM40 BM153	5
BM236 Macbeth BM76 CDC Bethune BM140 BM156	BM13 BM143 BM50 CDC Bethune Macbeth BM31 Hanley	BM104 BM77 BM96 CDC Bethune BM42 BM120 BM54	BM117 BM254 BM9 CDC Bethune BM150 BM238 BM243	BM194 Macbeth CDC Bethune BM69 BM229 Hanley	BM251 BM170 BM188 CDC Bethune BM29 BM3 BM78	BM157 BM86 Macbeth CDC Bethune Hanley BM87 BM192	6
BM80 BM68 BM168 CDC Bethune BM121 BM205 BM166	BM217 BM102 CDC Bethune BM123 BM197 BM146	BM151 BM46 BM89 CDC Bethune Hanley BM253 BM165	Macbeth BM127 BM155 CDC Bethune BM24 BM177	BM175 BM247 BM204 CDC Bethune BM207 Macbeth Hanley	BM138 BM221 BM208 CDC Bethune BM139 BM174 BM75	BM15 BM141 BM132 CDC Bethune BM10 BM230 BM39	7

Field plan - S2012

1	2	3	4	5	6	7	
BM1 BM2 BM3 CDC Bethune Hanley BM4 Macbeth	BM5 BM6 BM7 CDC Bethune CDC Bethune BM8 BM9 BM10	BM11 BM12 BM13 CDC Bethune BM14 BM15 BM16	BM17 BM18 BM19 CDC Bethune Hanley Macbeth BM20	BM22 BM23 CDC Bethune BM24 BM25 BM26	BM27 BM28 BM29 CDC Bethune BM30 BM31 BM32	Macbeth Hanley BM33 CDC Bethune BM34 BM35 BM36	1
BM37 BM38 BM39 CDC Bethune BM40 BM41 BM42	Macbeth BM43 BM44 CDC Bethune BM45 BM46 Hanley	BM47 BM48 BM49 CDC Bethune BM50 BM51 BM52	BM53 BM54 BM55 CDC Bethune BM56 BM57 BM58	BM59 BM60 BM61 CDC Bethune BM62 BM63 BM64	BM65 BM66 CDC Bethune BM68 BM69 BM70	BM72 BM73 CDC Bethune BM74 BM75 BM76	2
Hanley BM77 BM78 CDC Bethune Macbeth BM79 BM80	BM81 Hanley BM82 CDC Bethune Macbeth BM83 BM84	BM85 BM86 BM87 CDC Bethune BM88 BM89 BM90	BM91 BM92 CDC Bethune BM94 BM95 BM96	BM97 BM98 CDC Bethune BM99 BM100 Hanley	BM101 BM102 CDC Bethune BM104 BM105 BM106	Hanley BM107 BM108 CDC Bethune BM109 Macbeth BM110	3
BM111 BM112 BM113 CDC Bethune BM114 BM115 BM116	BM117 BM118 BM119 CDC Bethune Macbeth BM120 Hanley	BM121 BM122 BM123 CDC Bethune BM124 BM126	BM127 BM128 BM129 CDC Bethune BM130 Macbeth Hanley	BM131 BM132 BM133 CDC Bethune BM134 BM135 BM136	BM137 BM138 BM139 CDC Bethune Hanley BM140 Macbeth	BM141 BM142 BM143 CDC Bethune BM144 BM145 BM146	4
BM147 BM149 CDC Bethune BM150 BM151 BM152	BM153 BM154 BM155 CDC Bethune BM156 BM157 BM158	Hanley CDC Bethune BM160 BM161 BM162	BM164 BM165 CDC Bethune BM166 BM167 BM168	BM170 BM171 CDC Bethune BM172 BM173 BM174	BM175 BM176 BM177 CDC Bethune BM178 BM179 BM180	BM181 BM182 BM183 CDC Bethune BM184 BM185 BM186	5
BM187 BM188 BM189 CDC Bethune BM190 BM191 BM192	BM193 BM194 BM195 CDC Bethune BM196 BM197 BM198	BM199 BM200 BM201 CDC Bethune BM202 Hanley Macbeth	BM203 BM204 Macbeth CDC Bethune Hanley BM205 BM206	BM207 Macbeth BM208 CDC Bethune BM209 BM210 Hanley	BM211 BM212 Macbeth CDC Bethune BM213 BM214 Hanley	Macbeth BM215 BM216 CDC Bethune BM217 Hanley BM218	6
BM220 BM221 CDC Bethune BM222 BM223 BM224	BM225 BM226 BM227 CDC Bethune BM228 BM229 BM230	BM231 BM232 BM233 CDC Bethune BM234 BM235 BM236	BM237 BM238 BM239 CDC Bethune BM240 BM241 BM242	Macbeth Hanley BM243 CDC Bethune BM244 BM245 BM246	BM247 BM248 Macbeth CDC Bethune BM249 Hanley BM250	Macbeth Hanley BM251 CDC Bethune BM252 BM253 BM254	7

Method I (adjustment by design structure)

$$Y'_{ij(k)} = Y_{ij(k)} - R_i - C_j$$

Where:

- $Y'_{ij(k)}$ = adjusted value of the k th test line in the ij th block
- $Y_{ij(k)}$ = observed value of the k th test line in the ij th block
- $X_{ij(A)}$ = observed value for the control plot in the ij th block (CDC Bethune)
- $R_i = \frac{\sum_{j=1}^c X_{ij(A)}}{c} - \bar{X}_A$
- $C_j = \frac{\sum_{ij} X_{ij(A)}}{r} - \bar{X}_A$

- $\bar{X}_A = \frac{\sum_i \sum_j X_{ij(A)}}{(r*c)}$

or

$$Y'_{ij(k)} = Y_{ij(k)} - (\bar{X}_i - \bar{X}) - (\bar{X}_j - \bar{X})$$

where:

- $Y'_{ij(k)}$ = Adjusted mean
- $Y_{ij(k)}$ = Raw data of plot in row r and column c
- \bar{X} = Mean of all check1
- \bar{X}_i = Mean of check1 in row i
- \bar{X}_j = Mean of check1 in column j

```
method_I <- function(env = "M2009") {
  traits <- c("Yield", "Oil.content", "Iodine", "Linolenic")
  xx <- dd %>%
    filter(Environment == env) %>%
    gather(Trait, Value, traits) %>%
    mutate(AdjustedValue = NA,
          BlockEffect = NA )
  # Adjust Values
  i<-1
  for(i in 1:nrow(xx)) {
    x1 <- xx %>% filter(MainCheck == 1, Trait == xx$Trait[i])
    x1_bar <- mean(x1$Value, na.rm = T)
    Ri <- mean(x1$Value[x1$Row==xx$Row[i]]) - x1_bar
    Cj <- mean(x1$Value[x1$Col==xx$Col[i]]) - x1_bar
    xx$AdjustedValue[i] <- xx$Value[i] - Ri - Cj
    xx$BlockEffect[i] <- Ri + Cj
  }
  # Plot
  i <- "Yield"
  for(i in traits) {
    xi <- xx %>% filter(Trait == i)
    myMin1 <- min(xi$Value, na.rm = T)
    myMax1 <- max(xi$Value, na.rm = T)
    xA <- xi %>% filter(MainCheck == 1)
    myMin2 <- min(xA$Value, na.rm = T)
    myMax2 <- max(xA$Value, na.rm = T)
    myColors <- c("black", "darkgreen", "darkorange", "darkred")
    xc <- xi %>% filter(MainCheck == 1)
    mp1 <- ggplot(xc, aes(x = Col, y = Row, fill = Value)) +
      geom_tile() +
      geom_text(aes(label = round(Value, 1))) +
      scale_fill_continuous(name = NULL, low = "white", high = "darkgreen") +
      scale_x_continuous(breaks = 1:7) +
      scale_y_continuous(breaks = 1:7) +
      theme_agData(legend.position = "none") +
      labs(title = paste(env, "- Main Check", i))
    mp2 <- ggplot(xc, aes(x = Col, y = Row, fill = BlockEffect)) +
      geom_tile() +
      geom_text(aes(label = round(BlockEffect, 3))) +
      scale_fill_continuous(name = NULL, low = "white", high = "darkgreen") +
      scale_x_continuous(breaks = 1:7) +
      scale_y_continuous(breaks = 1:7) +
      theme_agData(legend.position = "none") +
      labs(title = paste(env, "- Block Effect", i))
  }
}
```

```

scale_fill_continuous(name = NULL, low = "white", high = "darkgreen") +
scale_x_continuous(breaks = 1:7) +
scale_y_continuous(breaks = 1:7) +
theme_agData(legend.position = "none") +
labs(title = "Block Effects")
mp3 <- ggplot(xi, aes(x = Value, y = AdjustedValue,
                      color = Type, size = Type)) +
  geom_point(alpha = 0.7) +
  geom_abline(alpha = 0.5) +
  scale_color_manual(name = NULL, values = myColors) +
  scale_size_manual(name = NULL, values = c(1,2,2,2)) +
  scale_x_continuous(limits = c(min(myMin1,myMin2), max(myMax1, myMax2))) +
  scale_y_continuous(limits = c(min(myMin1,myMin2), max(myMax1, myMax2))) +
  theme_agData(legend.position = "none") +
  labs(x = i, y = paste("Adjusted", i),
       title = paste("Main Check Range =", myMin2, "-", myMax2))
xi <- xi %>% rename(RawValue=Value) %>%
  gather(Trait, Value, RawValue, AdjustedValue) %>%
  mutate(Trait = factor(Trait, levels = c("RawValue", "AdjustedValue")))
mp4 <- ggplot(xi, aes(x = Type, y = Value, color = Type, shape = Trait)) +
  geom_quasirandom(dodge.width = 0.8) +
  scale_color_manual(name = NULL, values = myColors) +
  scale_shape_manual(name = NULL, values = c(16,17)) +
  theme_agData(legend.position = "none", legend.box="vertical") +
  labs(y = i, x = "unadjusted vs adjusted",
       title = paste("Treatment Range =", myMin1, "-", myMax1))
mp <- ggarrange(mp1, mp2, mp3, mp4, ncol = 2, nrow = 2)
ggsave(paste0("aug_02_02_", i, "_", env, "_MI.png"), mp,
       width = 10, height = 8)
}
# Output
xx
}
M2009_MI <- method_I(env = "M2009")
#M2010_MI <- method_I(env = "M2010")
#M2011_MI <- method_I(env = "M2011")
#M2012_MI <- method_I(env = "M2012")

```

Method III (adjustment by regression)

Method III

$$Y'_{ij(k)} = Y_{ij(k)} - b(X_{ij(A)} - \bar{X}_A)$$

where

- $Y_{ij(k)}$ = observed value of the k th test line in the ij th block
- $X_{ij(A)}$ = the observed value for the control plot in the ij th block (CDC Bethune)
- b = regression coefficient of the mean of the control subplots by the main control plot
- \$\$

$$m_a = y_{rc} - \text{slope} * (\bar{c}_r - \bar{c}) - (\bar{c}_c - \bar{c}) = y_{rc} - \bar{c}_r - \bar{c}_c - 2\bar{c}$$

```

method_III <- function(env = "M2009") {
  traits <- c("Yield", "Oil.content", "Iodine", "Linolenic")
  xx <- dd %>%
    filter(Environment == env) %>%
    gather(Trait, Value, traits) %>%
    mutate(AdjustedValue = NA,
          BlockEffect = NA)
  # Adjust Values
  i<-1
  for(i in 1:nrow(xx)) {
    x2 <- xx %>% filter(MainCheck == 1, Trait == xx$Trait[i]) %>%
      select(Row, Col, Type, Value) %>%
      spread(Type, Value)
    x3 <- xx %>% filter(SubCheck %in% 1:2, Trait == xx$Trait[i]) %>%
      select(Row, Col, Type, Value) %>%
      spread(Type, Value)
    x123 <- left_join(x2, x3, by = c("Row", "Col")) %>%
      filter(!is.na(Check2), !is.na(Check3)) %>%
      mutate(Check23 = (Check2 + Check3) / 2)
    bb <- as.vector(coefficients(lm(Check1 ~ Check23, data = x123))[2])
    #
    x1_bar <- mean(x123$Check1, na.rm = T)
    # not the mean of all main checks, but only ones with subchecks in the block
    x1ij <- xx %>%
      filter(Row == xx$Row[i], Col == xx$Col[i],
             Type == "Check1", Trait == xx$Trait[i]) %>%
      pull(Value)
    xx$AdjustedValue[i] <- xx$Value[i] - bb * (x1ij - x1_bar)
    xx$BlockEffect[i] <- bb * (x1ij - x1_bar)
    #
    #mp <- ggplot(x123, aes(y = Check1, x = Check23)) +
    #  geom_point() + geom_smooth(method = "lm", se = F)
    #ggsave(paste0("aug_02_03_MIII_", xx$Trait[i], "_", env, ".png"), mp, width = 6, height = 4)
  }
  # Plot
  i <- "Yield"
  for(i in traits) {
    xi <- xx %>% filter(Trait == i)
    myMin1 <- min(xi$Value, na.rm = T)
    myMax1 <- max(xi$Value, na.rm = T)
    xA <- xi %>% filter(MainCheck == 1)
    myMin2 <- min(xA$Value, na.rm = T)
    myMax2 <- max(xA$Value, na.rm = T)
    myColors <- c("black", "darkgreen", "darkorange", "darkred")
    xc <- xi %>% filter(MainCheck == 1)
    mp1 <- ggplot(xc, aes(x = Col, y = Row, fill = Value)) +
      geom_tile() +
      geom_text(aes(label = round(Value, 1))) +
      scale_fill_continuous(name = NULL, low = "white", high = "darkgreen") +
      scale_x_continuous(breaks = 1:7) +
      scale_y_continuous(breaks = 1:7) +

```

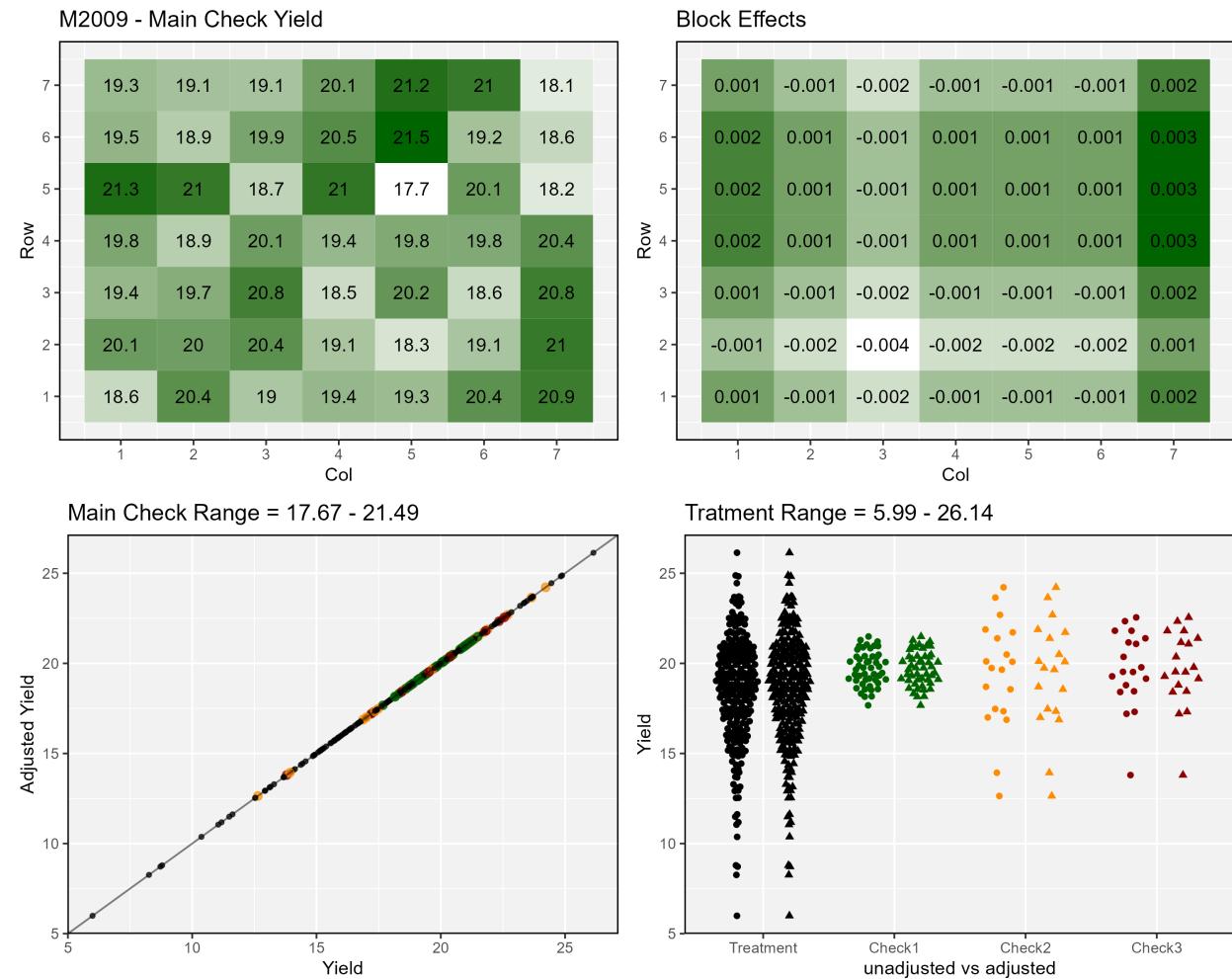
```

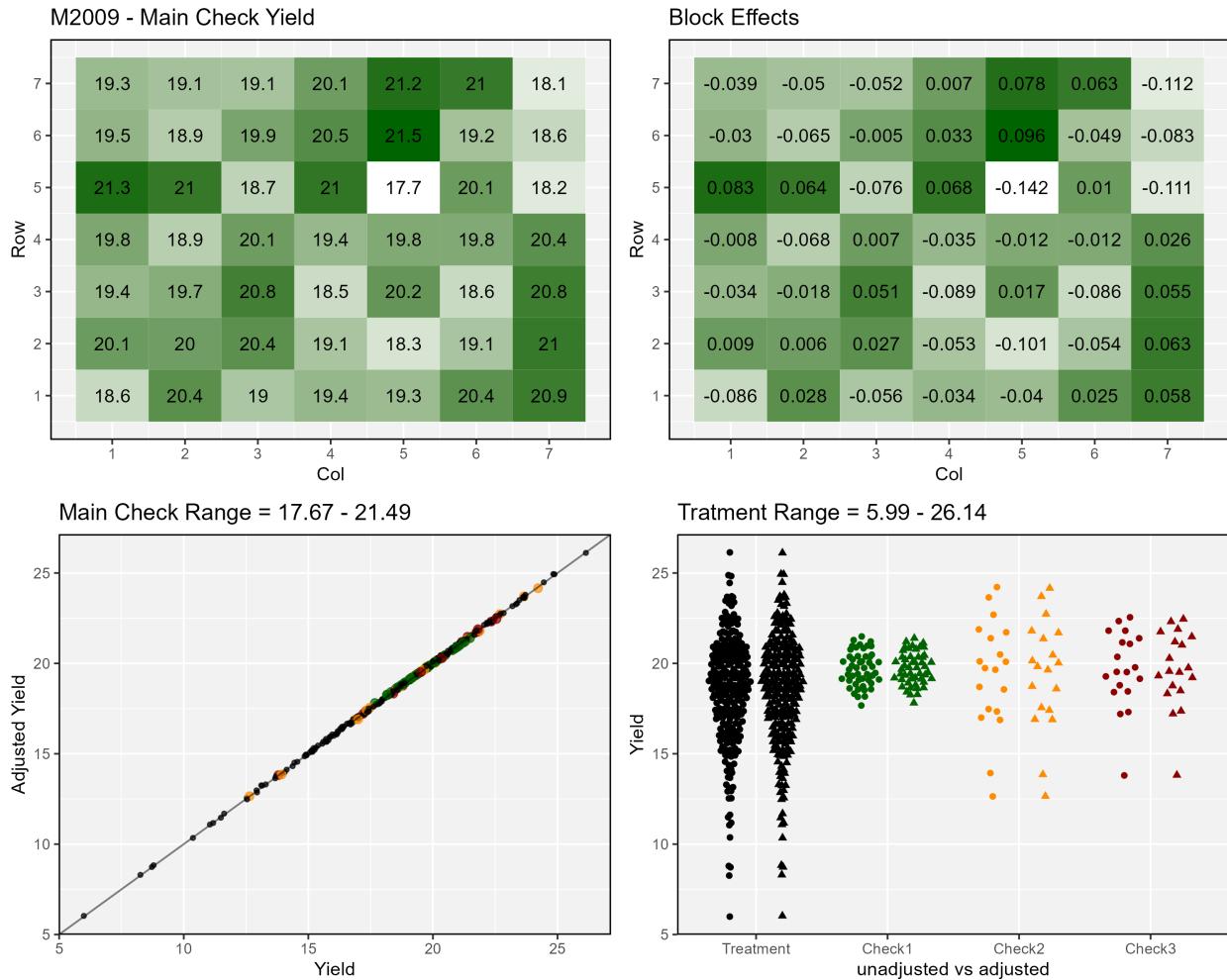
theme_agData(legend.position = "none") +
  labs(title = paste(env, "- Main Check", i))
mp2 <- ggplot(xc, aes(x = Col, y = Row, fill = BlockEffect)) +
  geom_tile() +
  geom_text(aes(label = round(BlockEffect, 3))) +
  scale_fill_continuous(name = NULL, low = "white", high = "darkgreen") +
  scale_x_continuous(breaks = 1:7) +
  scale_y_continuous(breaks = 1:7) +
  theme_agData(legend.position = "none") +
  labs(title = "Block Effects")
mp3 <- ggplot(xi, aes(x = Value, y = AdjustedValue,
                       color = Type, size = Type)) +
  geom_point(alpha = 0.7) +
  geom_abline(alpha = 0.5) +
  scale_color_manual(name = NULL, values = myColors) +
  scale_size_manual(name = NULL, values = c(1,2,2,2)) +
  scale_x_continuous(limits = c(min(myMin1,myMin2), max(myMax1, myMax2))) +
  scale_y_continuous(limits = c(min(myMin1,myMin2), max(myMax1, myMax2))) +
  theme_agData(legend.position = "none") +
  labs(x = i, y = paste("Adjusted", i),
       title = paste("Main Check Range =", myMin2, "-", myMax2))
xi <- xi %>% rename(RawValue=Value) %>%
  gather(Trait, Value, RawValue, AdjustedValue) %>%
  mutate(Trait = factor(Trait, levels = c("RawValue", "AdjustedValue")))
mp4 <- ggplot(xi, aes(x = Type, y = Value, color = Type, shape = Trait)) +
  geom_quasirandom(dodge.width = 0.8) +
  scale_color_manual(name = NULL, values = myColors) +
  scale_shape_manual(name = NULL, values = c(16,17)) +
  theme_agData(legend.position = "none", legend.box="vertical") +
  labs(y = i, x = "unadjusted vs adjusted",
       title = paste("Treatment Range =", myMin1, "-", myMax1))
mp <- ggarrange(mp1, mp2, mp3, mp4, ncol = 2, nrow = 2)
ggsave(paste0("aug_02_02_", i, "_", env, "_MIII.png"), mp,
       width = 10, height = 8)
}
# Output
xx
}
M2009_MIII <- method_III(env = "M2009")
#M2010_MIII <- method_III(env = "M2010")
#M2011_MIII <- method_III(env = "M2011")
#M2012_MIII <- method_III(env = "M2012")

```

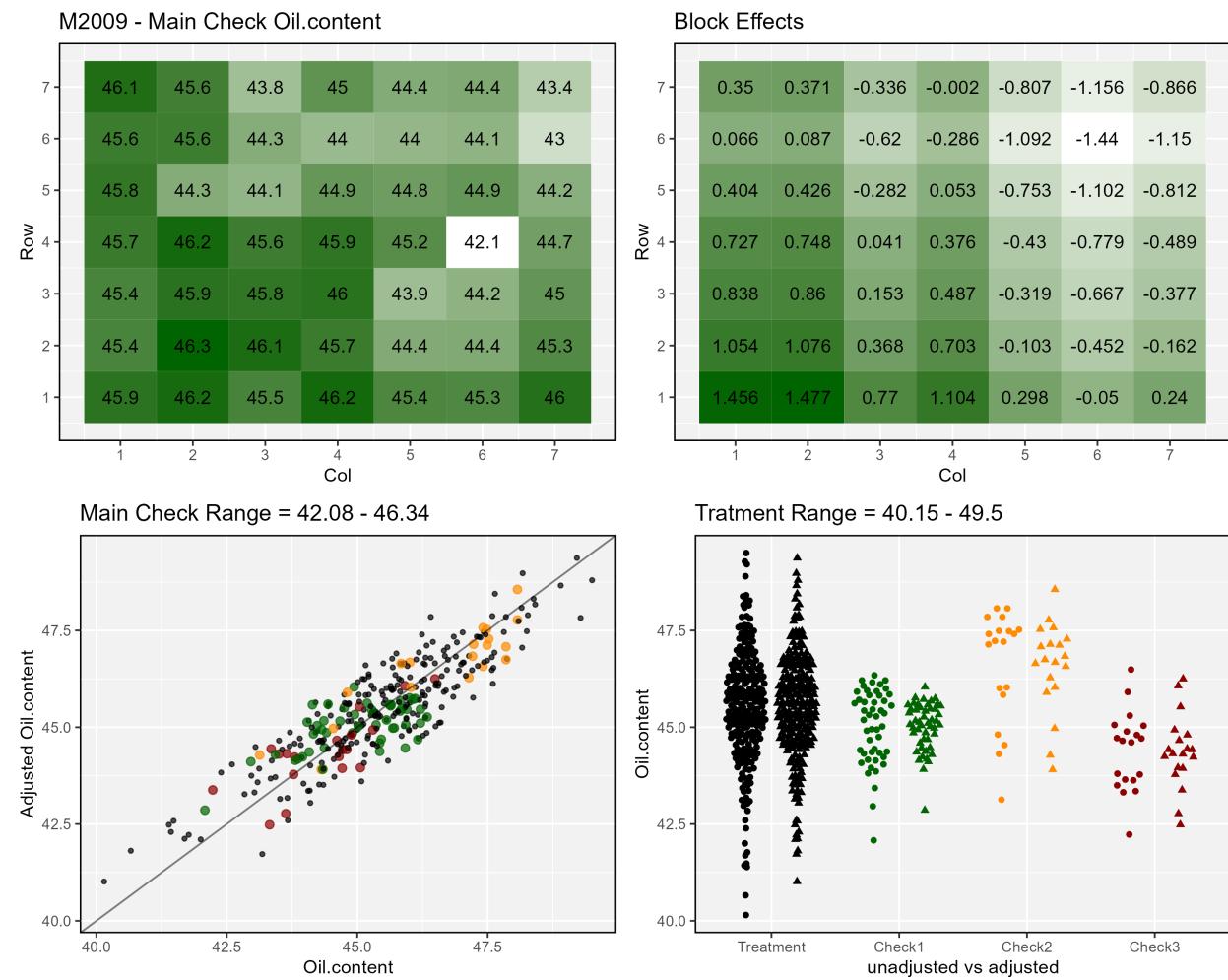
Adjustments

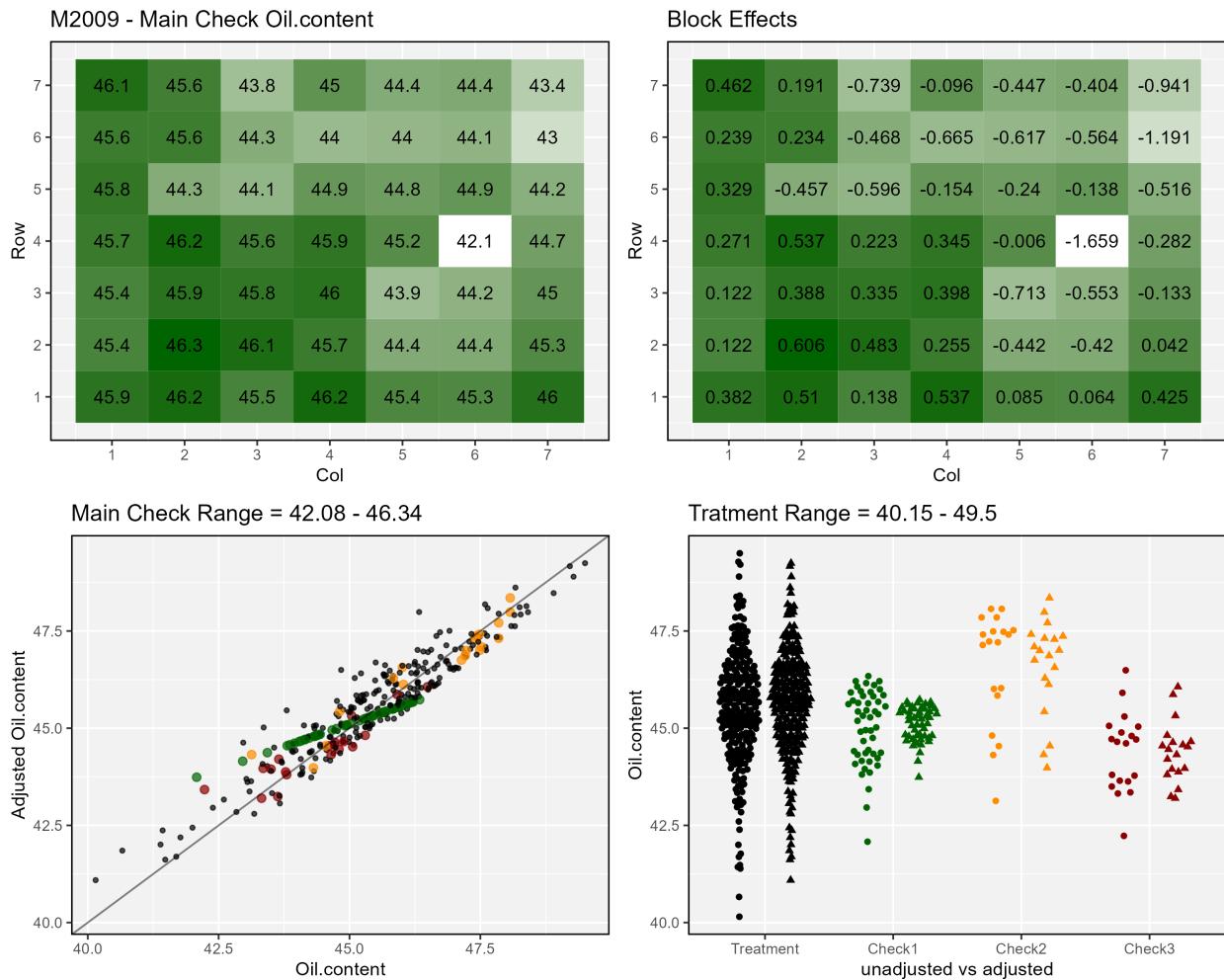
Yield



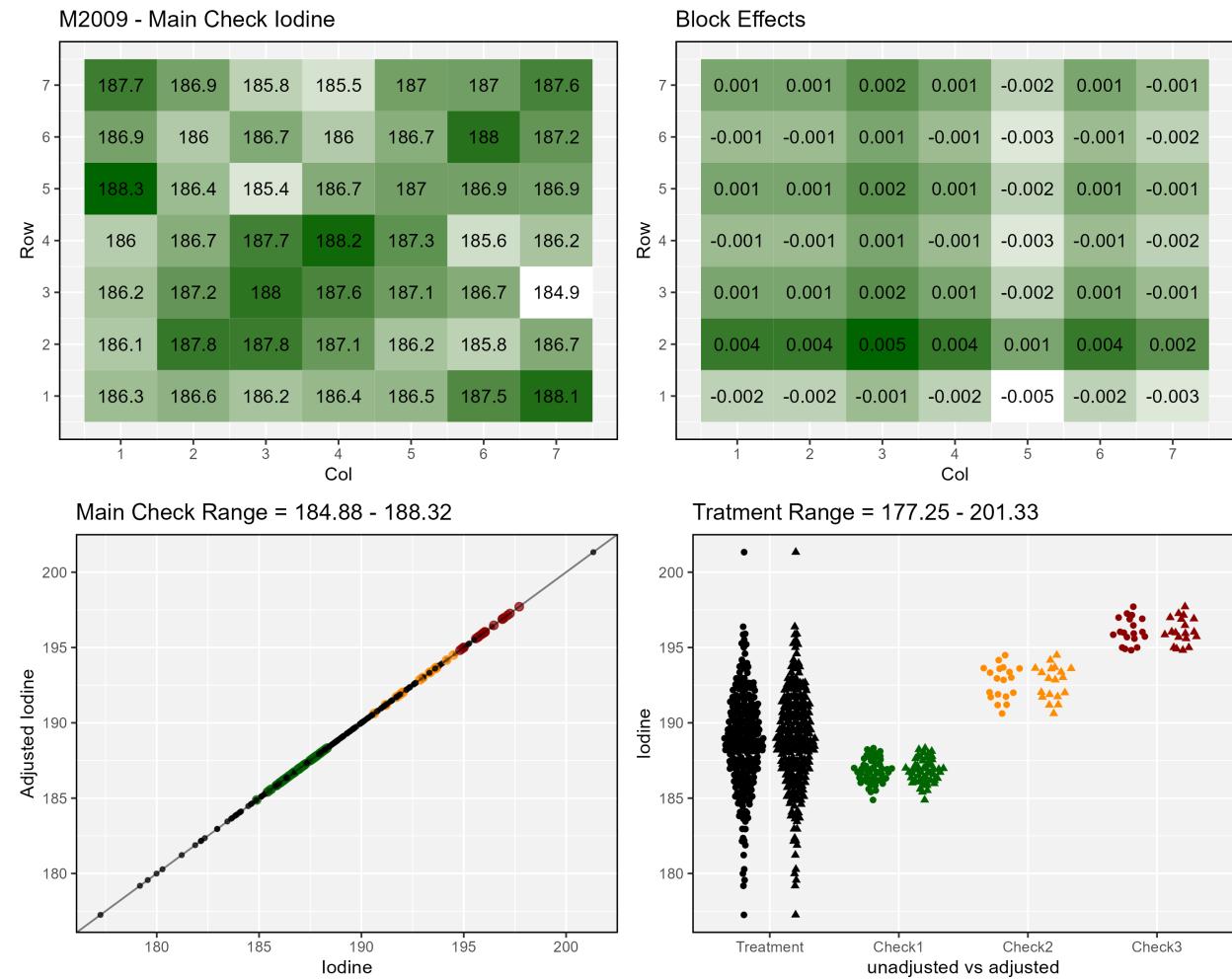


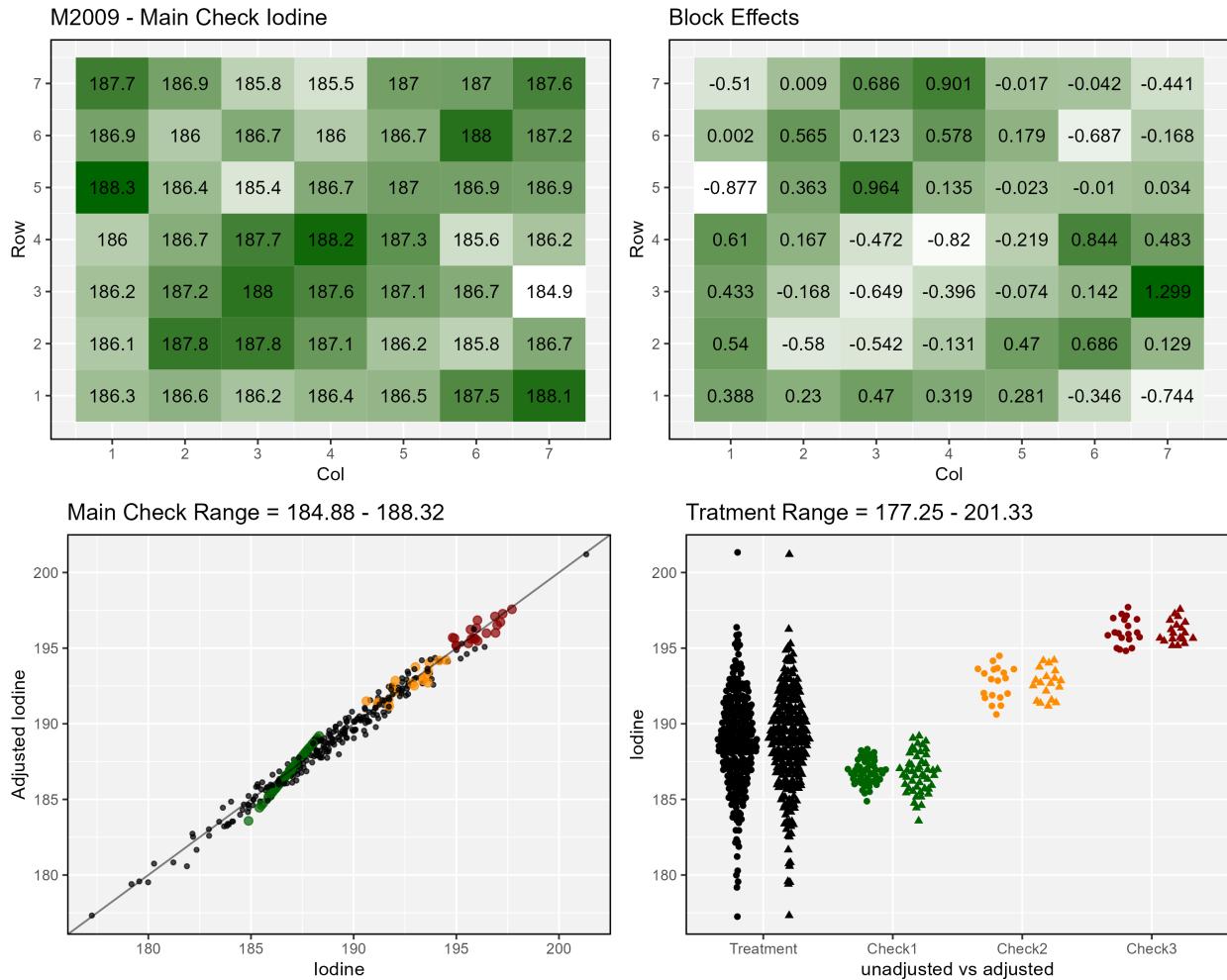
Oil Content



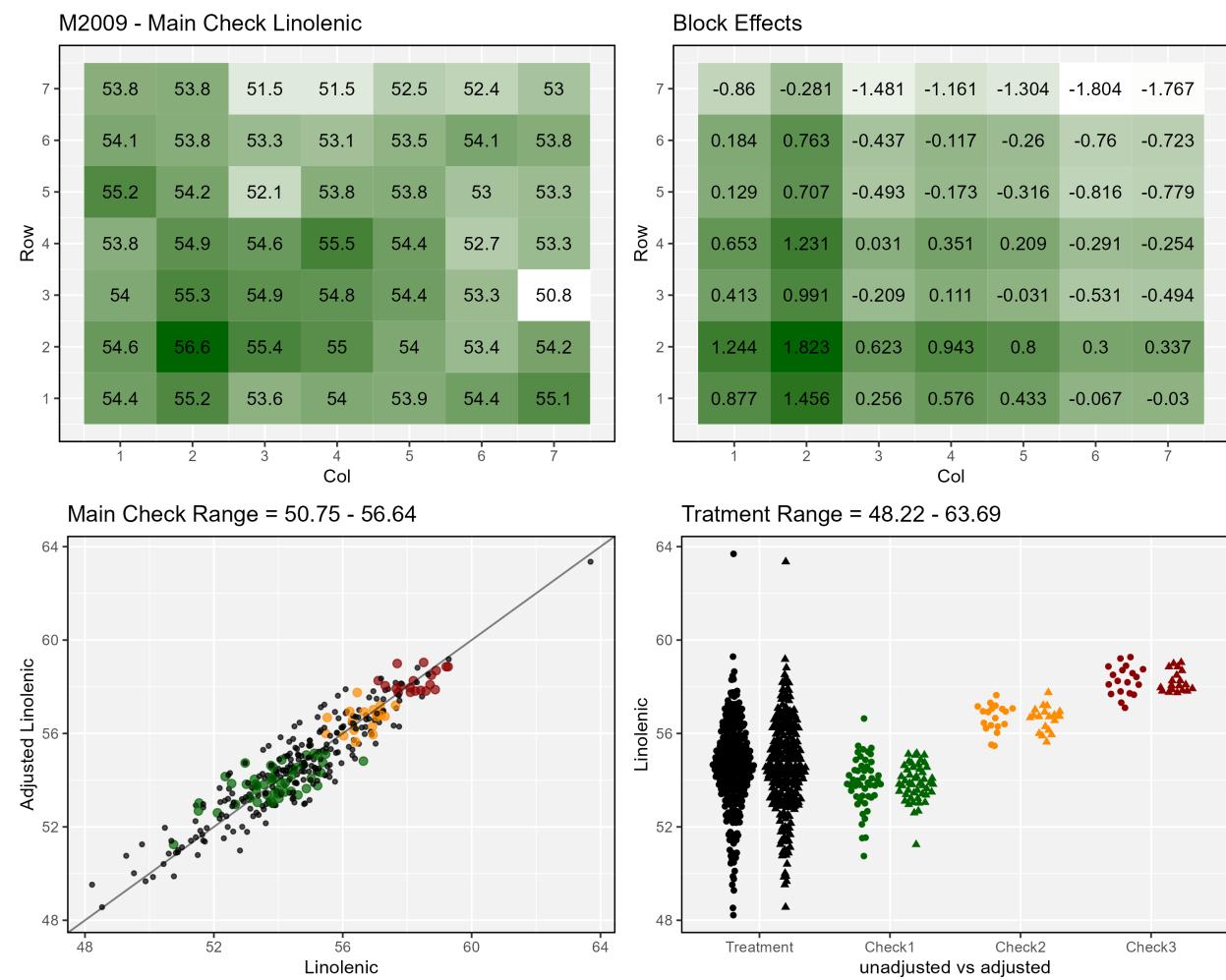


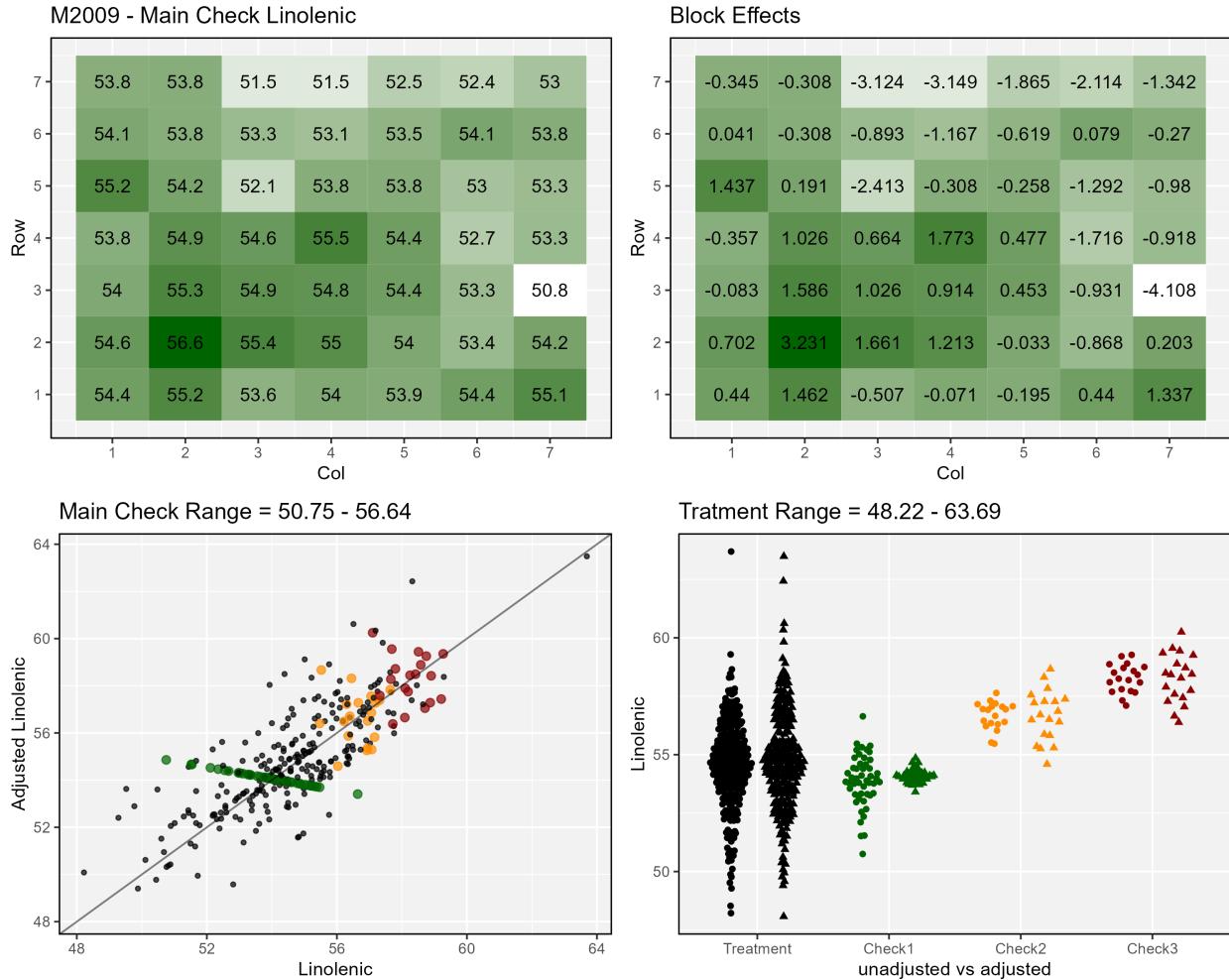
Iodine





Linolenic





Relative Efficiency

$$RE = \frac{IWPE_{unadj}}{IWPE_{adj}} * 100$$

Where:

- $RE = Relative Efficiency$
- $IWPE_{unadj} = IntraWholePlotError_{unadjusted} = \sum_{j=1}^c X_{ij(A)}$
- $IWPE_{adj} = IntraWholePlotError_{adjusted} = \sum_{j=1}^c X_{ij(A)}$

```
#  
RE <- function(xx = M2009_MI, myEnv = "M2009", myTrait = "Yield") {  
  xx <- xx %>% filter(Environment == myEnv, Trait == myTrait)  
  sc <- unique(xx$Genotype[xx$SubCheck != 0])  
  i <- 1 #Hanley  
  myRE1 <- NULL
```

```

myRE2 <- NULL
for(i in 1:length(sc)) {
  xi <- xx %>% filter(Genotype == sc[i])
  myRE1[i] <- sum((xi$value - mean(xi$value))^2)
  myRE2[i] <- sum((xi$AdjustedValue - mean(xi$AdjustedValue))^2)
}
myRE1 <- sum(myRE1) / (sum(xx$Genotype%in%sc) - length(sc))
myRE2 <- sum(myRE2) / (sum(xx$Genotype%in%sc) - length(sc))
100 * myRE1 / myRE2
}

#
RE(xx = M2009_MI, myEnv = "M2009", myTrait = "Yield")

## [1] 100.0216

RE(xx = M2009_MIII, myEnv = "M2009", myTrait = "Yield")

## [1] 100.3977

#
RE(xx = M2009_MI, myEnv = "M2009", myTrait = "Oil.content")

## [1] 139.0951

RE(xx = M2009_MIII, myEnv = "M2009", myTrait = "Oil.content")

## [1] 149.9792

#
RE(xx = M2009_MI, myEnv = "M2009", myTrait = "Iodine")

## [1] 99.92228

RE(xx = M2009_MIII, myEnv = "M2009", myTrait = "Iodine")

## [1] 132.9257

#
RE(xx = M2009_MI, myEnv = "M2009", myTrait = "Linolenic")

## [1] 150.8236

RE(xx = M2009_MIII, myEnv = "M2009", myTrait = "Linolenic")

## [1] 31.32183

```

It is important to note that the F-tests from the ANOVA results apply to Row and Column effects, which can be accounted for by Method 1 adjustments. The Row and Column effects generally are best at accounting for gradients that stretch across a substantial portion of the field.

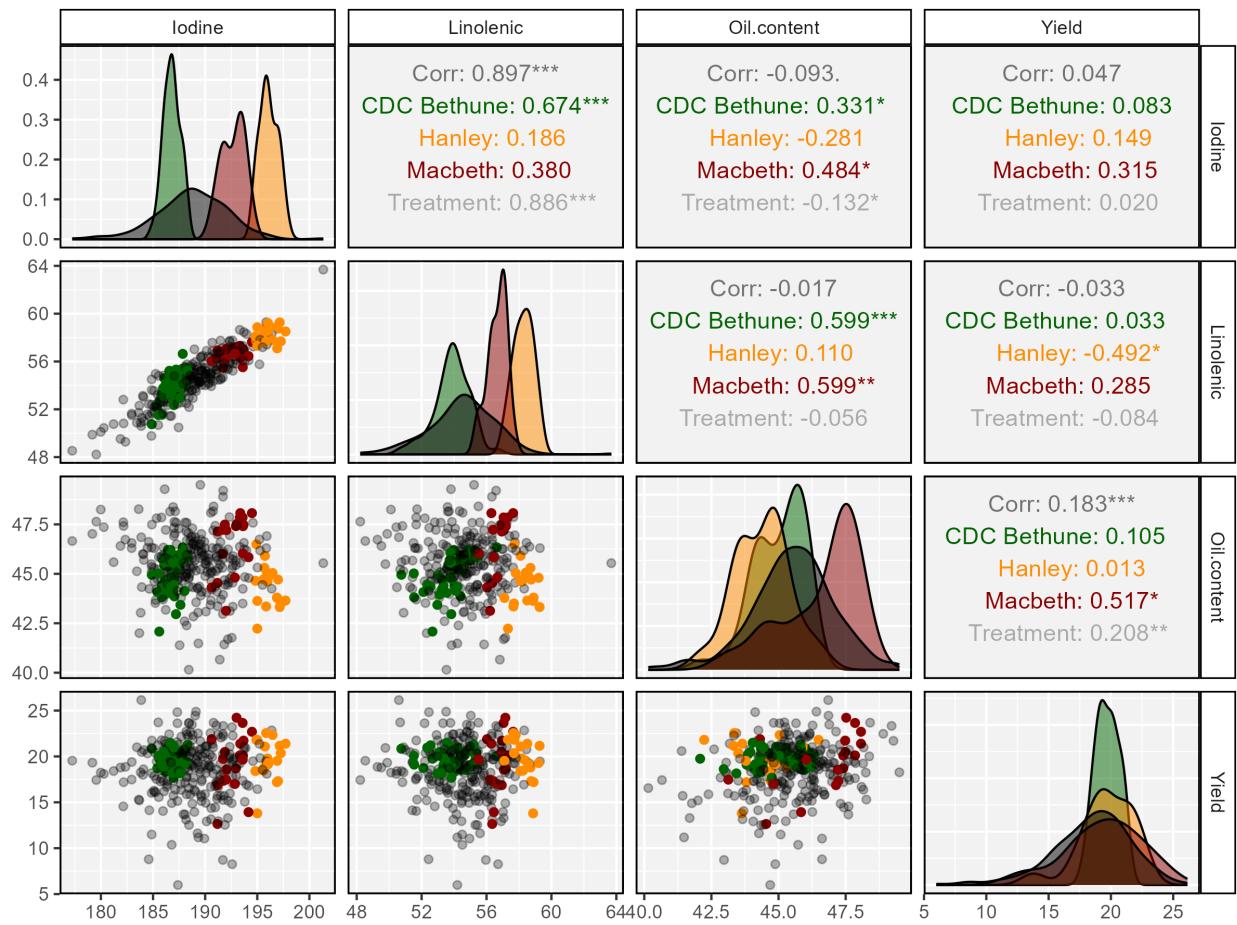
Method 3 does not necessarily require gradients or any other type of pattern in the field to account for field effects; it only requires that the secondary checks are affected by the field in a way similar to how the primary checks are affected.

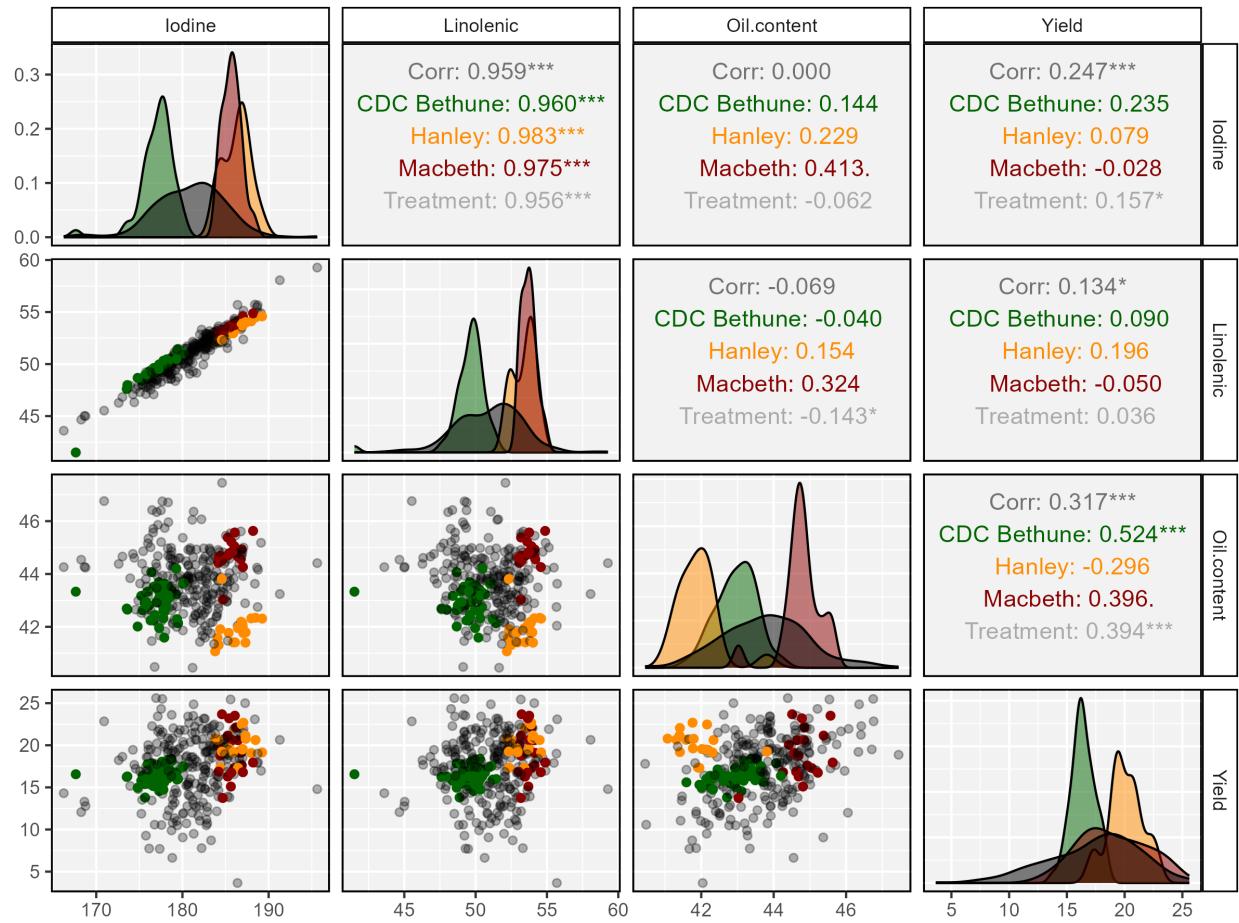
The type-2 modified augmented design intentionally makes the final selection of an adjustment method to be somewhat subjective, based on the user's understanding of the biological system. Evidence on the appropriateness of each method includes:

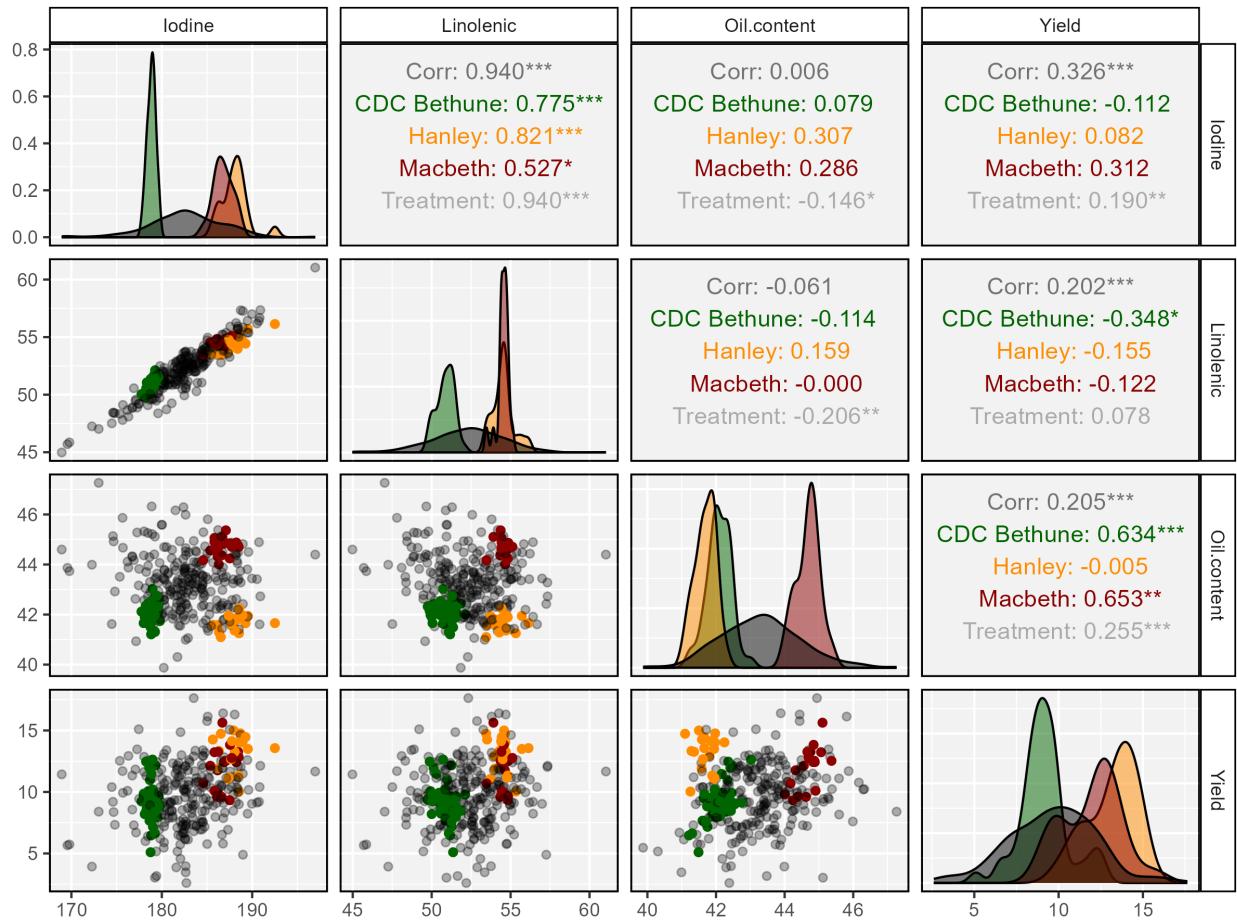
- Relative efficiency of Method 1 vs Method 3
- ANOVA results for Row and Column effects
- Biological meaning of analysis parameters
- I once had a relative efficiency of 112% for Method 3 but a negative Method 3 regression coefficient
- Heat maps or other semi-quantitative/qualitative evaluations of field effects
- Knowledge of the field or fieldbook notes from the experiment
- Knowledge of the lines used as checks
- e.g., you may have more confidence in a marginal Method 1 relative efficiency when you know that your primary check used to calculate Method 1 adjustments is much less sensitive to field effects than most of the other lines in the experiment.

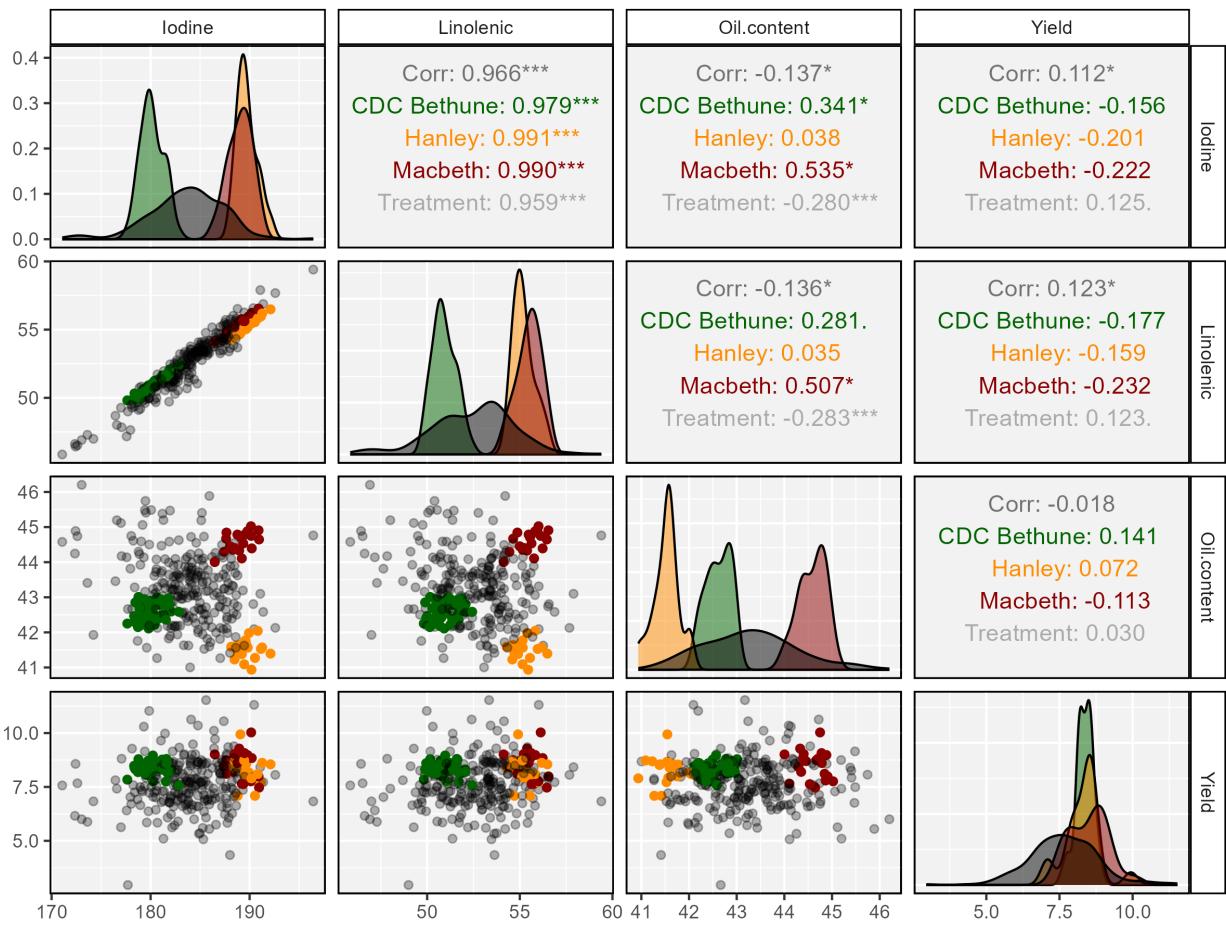
Correlation Plots

```
gg_Corr <- function(env = "M2009") {
  # Prep data
  xx <- dd %>% filter(Environment == env) %>%
    mutate(Genotype = ifelse(MainCheck > 0 | SubCheck > 0, Genotype, "Treatment"))
  myColors <- c("darkgreen", "darkorange", "darkred", alpha("black",0.3))
  # Plot
  mp <- ggpairs(xx, columns = 13:16, aes(color = Genotype)) +
    scale_color_manual(values = myColors) +
    scale_fill_manual(values = alpha(myColors,0.5)) +
    theme_agData()
  ggsave(paste0("aug_03_01_", env, ".png"), mp, width = 8, height = 6)
}
gg_Corr(env = "M2009")
gg_Corr(env = "M2010")
gg_Corr(env = "M2011")
gg_Corr(env = "M2012")
gg_Corr(env = "S2009")
gg_Corr(env = "S2010")
gg_Corr(env = "S2011")
gg_Corr(env = "S2012")
```









Derek Michael Wright www.dblogr.com/