Split Plot Designs

Jonathen, Melissa & Tiffany

5/24/2022

Introduction and Set-Up

Split plot designs are a variation of factorial ANOVA that account for the fact that one of the predictor variables is hard to randomize. In this dataset, we'll be looking at a split plot experiment by Durban et al. from a study published in 2003 of barley with fungicide treatments. Crops grown at the Scottish Crop Research Institute were broken up in to 4 main blocks with 2 fungal treatments and 70 barley genotypes randomized across those blocks with crop yield being the outcome variable.

yield - tonnes/ha block - 4 levels (4 groups) gen - genotype - 70 levels (70 genotypes) fung - fungicide - 2 levels (1/0 treated) row - row (of crop field) bed - column (of crop field)

Fitting with the standard experimental design of a split-plot: - whole-plot: block - sub-plot: fung - sub-sub-plot: gen

Loading Dataset

```r
# read in data from agridat library
data("durban.splitplot")
durban <- durban.splitplot
str(durban) # to double check our variables are of the right type
```

```
## 'data.frame':    560 obs. of  6 variables:
##  $ yield: num  5.89 6.17 5.68 5.85 5.8 6.01 5.89 4.53 5.32 5.36 ...
##  $ block: Factor w/ 4 levels "B1","B2","B3",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ gen  : Factor w/ 70 levels "G01","G02","G03",..: 54 44 68 59 61 67 45 10 27 60 ...
##  $ fung : Factor w/ 2 levels "F1","F2": 1 1 1 1 1 1 1 2 2 2 ...
##  $ row  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ bed  : int  1 2 3 4 5 6 7 8 9 10 ...
```

```r
head(durban)
```

```
##   yield block gen fung row bed
## 1  5.89    B1 G54   F1   1   1
## 2  6.17    B1 G44   F1   1   2
## 3  5.68    B1 G68   F1   1   3
## 4  5.85    B1 G59   F1   1   4
## 5  5.80    B1 G61   F1   1   5
## 6  6.01    B1 G67   F1   1   6
```

Exploring the Data

Now that we've read in our data, we want to explore the data to get a sense for what we have to work with. We know that yield is our outcome variable. So let's order the data in descending order with yield.

```
# set up agr order as a tibble (tbl_df vs df)
agr_order = as_tibble(durban)

# reorder levels from high to low yield
arrange(agr_order, desc(yield))
```

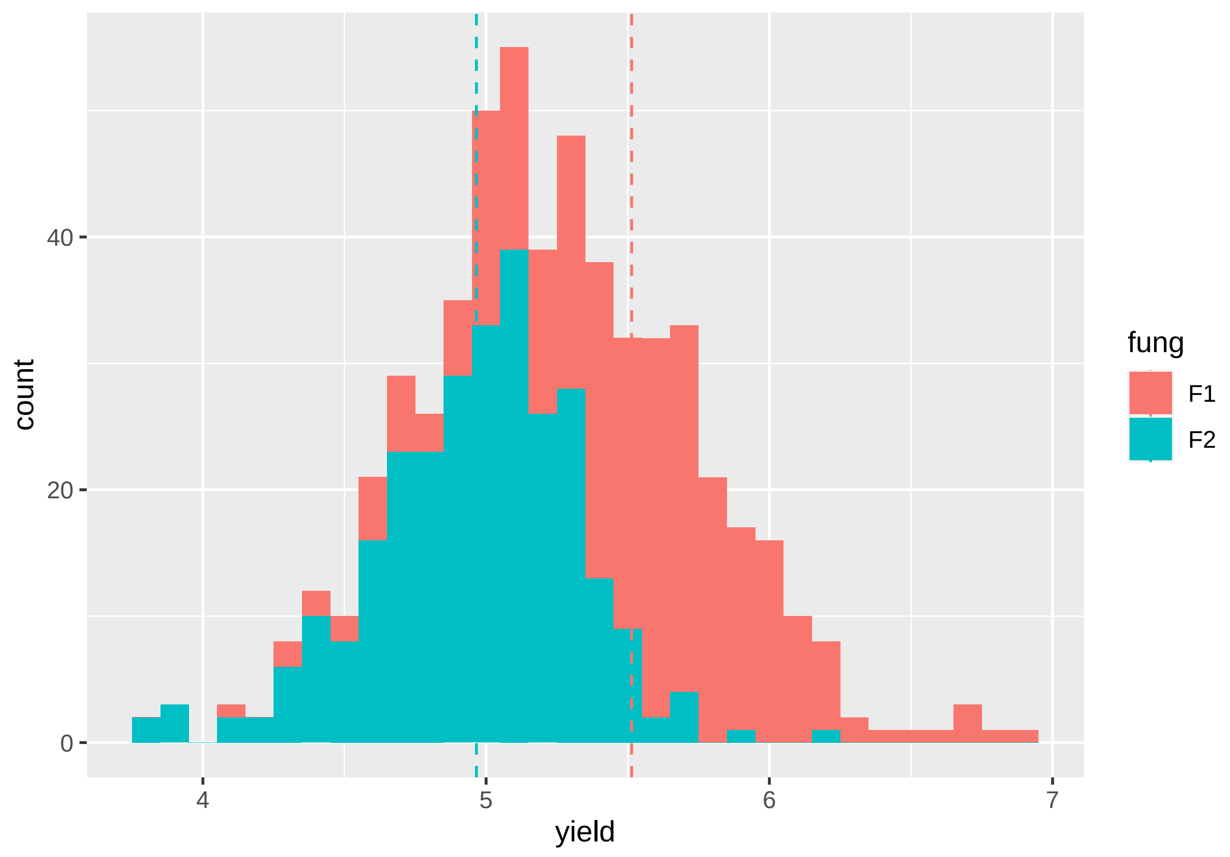
```
## # A tibble: 560 x 6
##    yield block gen   fung    row   bed
##    <dbl> <fct> <fct> <fct> <int> <int>
##  1  6.86 B1    G13   F1       10     7
##  2  6.8  B1    G03   F1        6     1
##  3  6.68 B1    G19   F1        5     5
##  4  6.66 B2    G03   F1        3    17
##  5  6.66 B1    G27   F1        4     6
##  6  6.6  B1    G36   F1        9     5
##  7  6.51 B1    G33   F1        9     6
##  8  6.45 B3    G03   F1        2    33
##  9  6.34 B1    G64   F1        4     2
## 10  6.3  B1    G62   F1        3     2
## # i 550 more rows
```

It seems we have a noticeable trend where fungicide treatment 'F1' has the higher crop yield. To make sure, let's do a group by and then average to see.

```
# checking averages of yield grouped by fungus treatment
by_yield <- agr_order %>% group_by(fung) %>%
  summarise_at(vars(yield), list(mean = mean))
by_yield
```

```
## # A tibble: 2 x 2
##   fung   mean
##   <fct> <dbl>
## 1 F1     5.51
## 2 F2     4.97
```

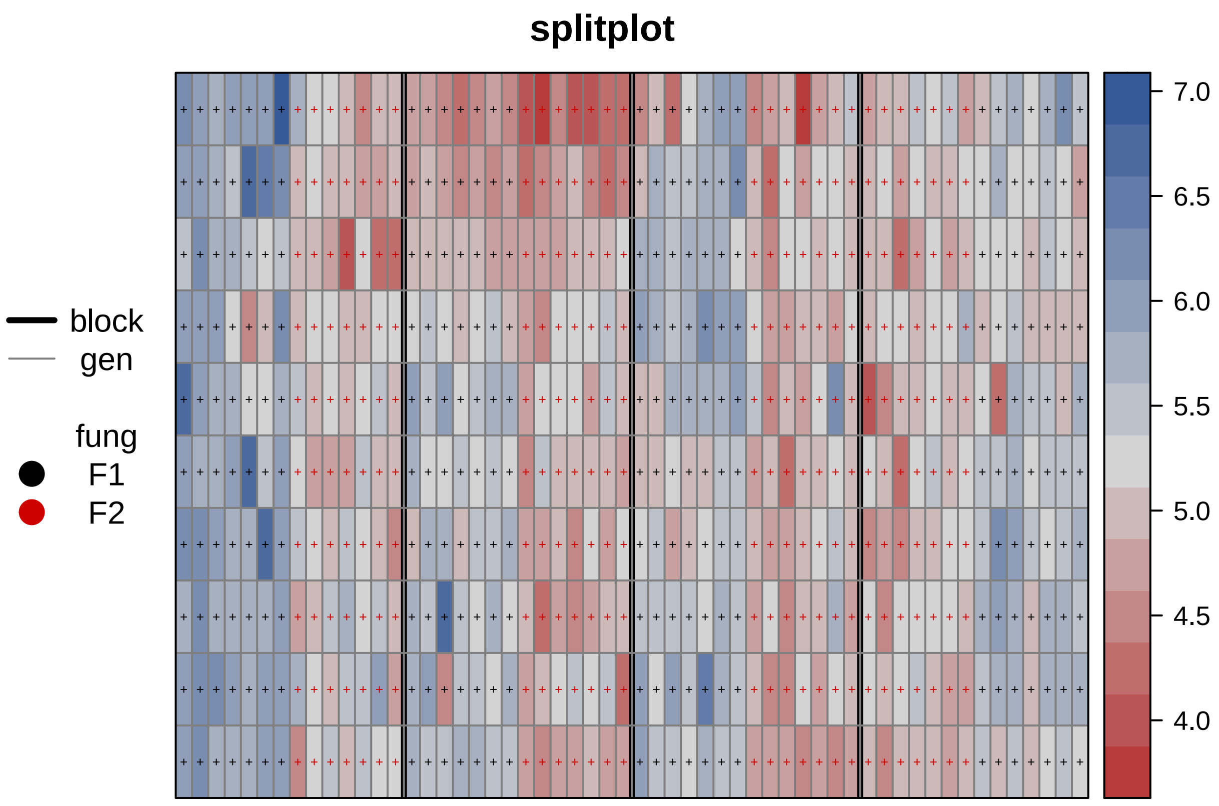
```
mean.plot <- ggplot(data=agr_order,aes(x=yield,fill=fung)) + geom_histogram(binwidth=0.1) + geom_vline(
mean.plot
```



Visualizing Our Data

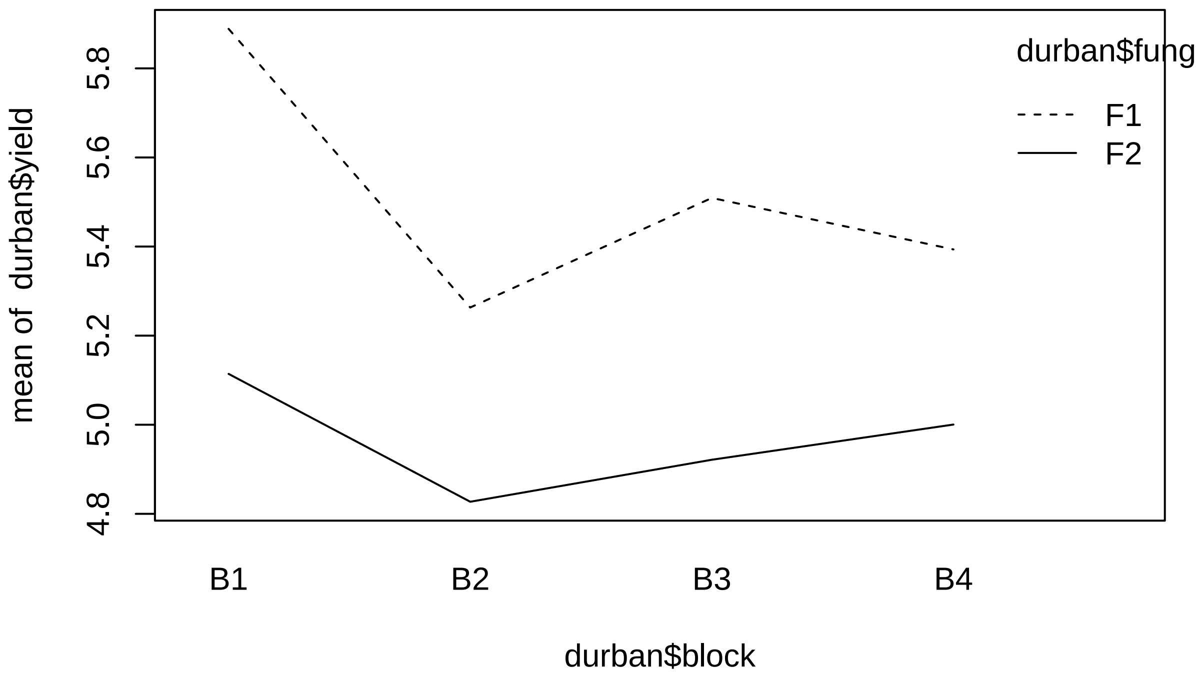
Here's a general sense of how the plots are set-up

```
desplot(durban, yield~bed*row,col=fung,
        out1=block,out2=gen,out2.gpar=list(col="gray50",lwd=1,lty=1),main="splitplot")
```

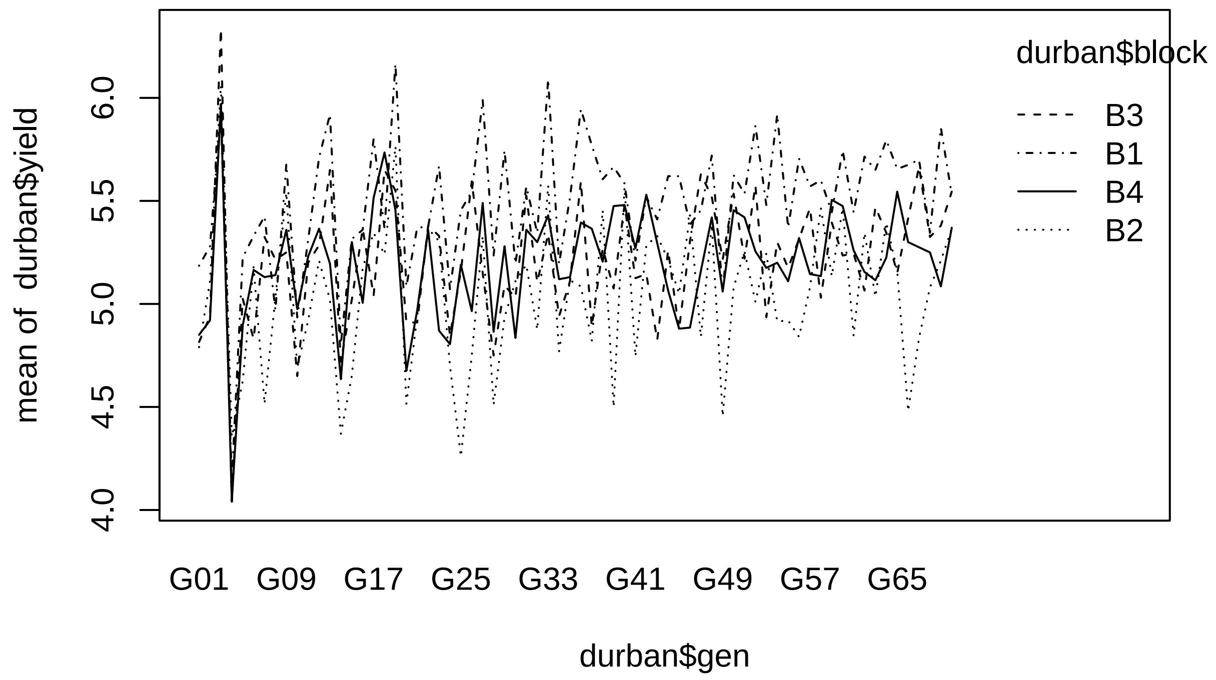


We can also visualize the differences in means based on different groupings: block vs. fungal treatment, block vs. barley genotype, fungal treatment vs. barley genotype

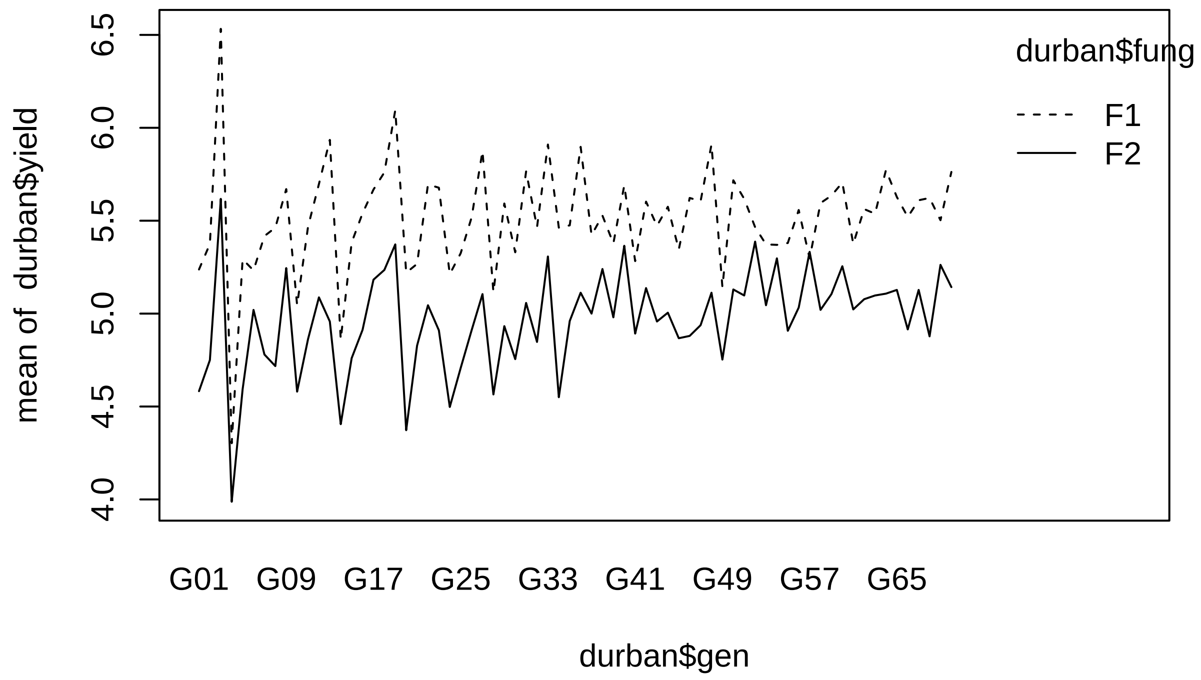
```
interaction.plot(durban$block, durban$fung, durban$yield)
```



```r
interaction.plot(durban$gen, durban$block, durban$yield)
```



```r
interaction.plot(durban$gen, durban$fung, durban$yield)
```

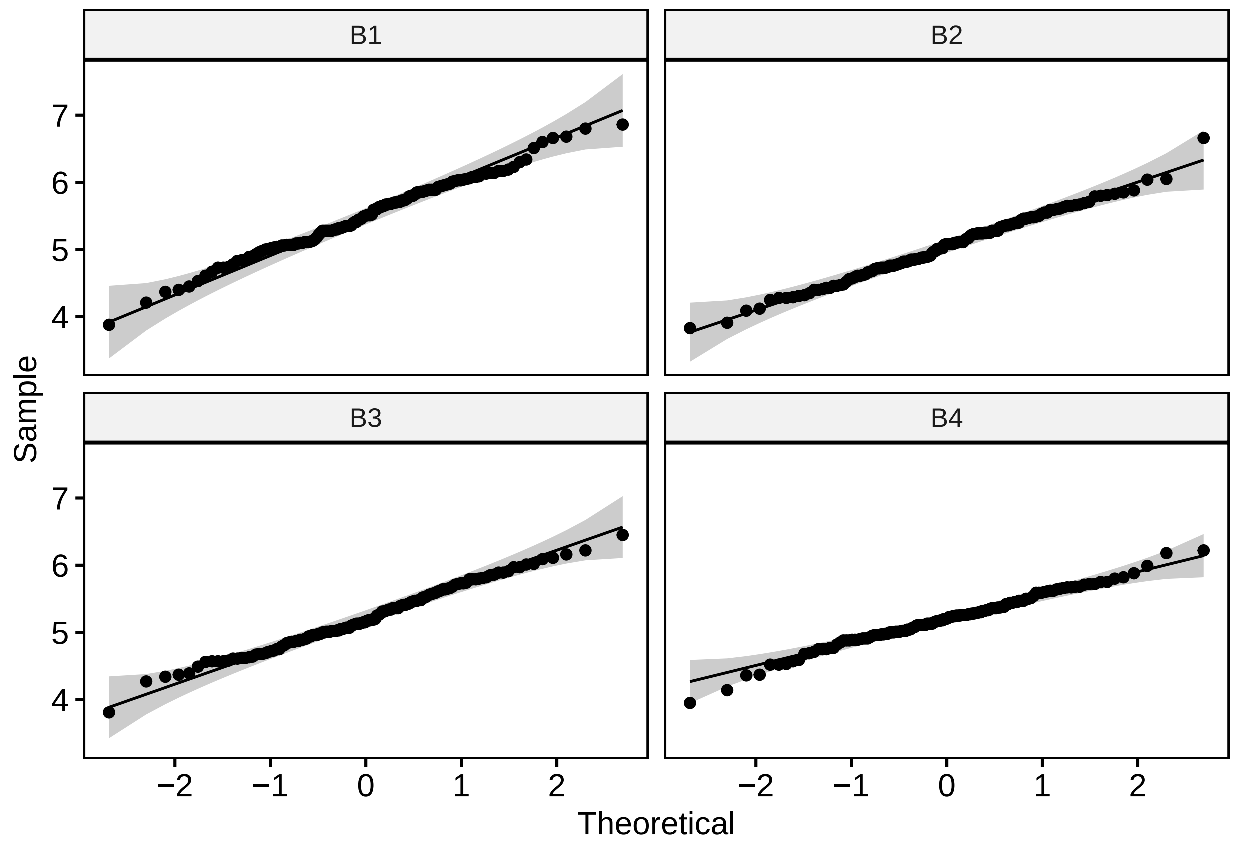


Normality and Variance

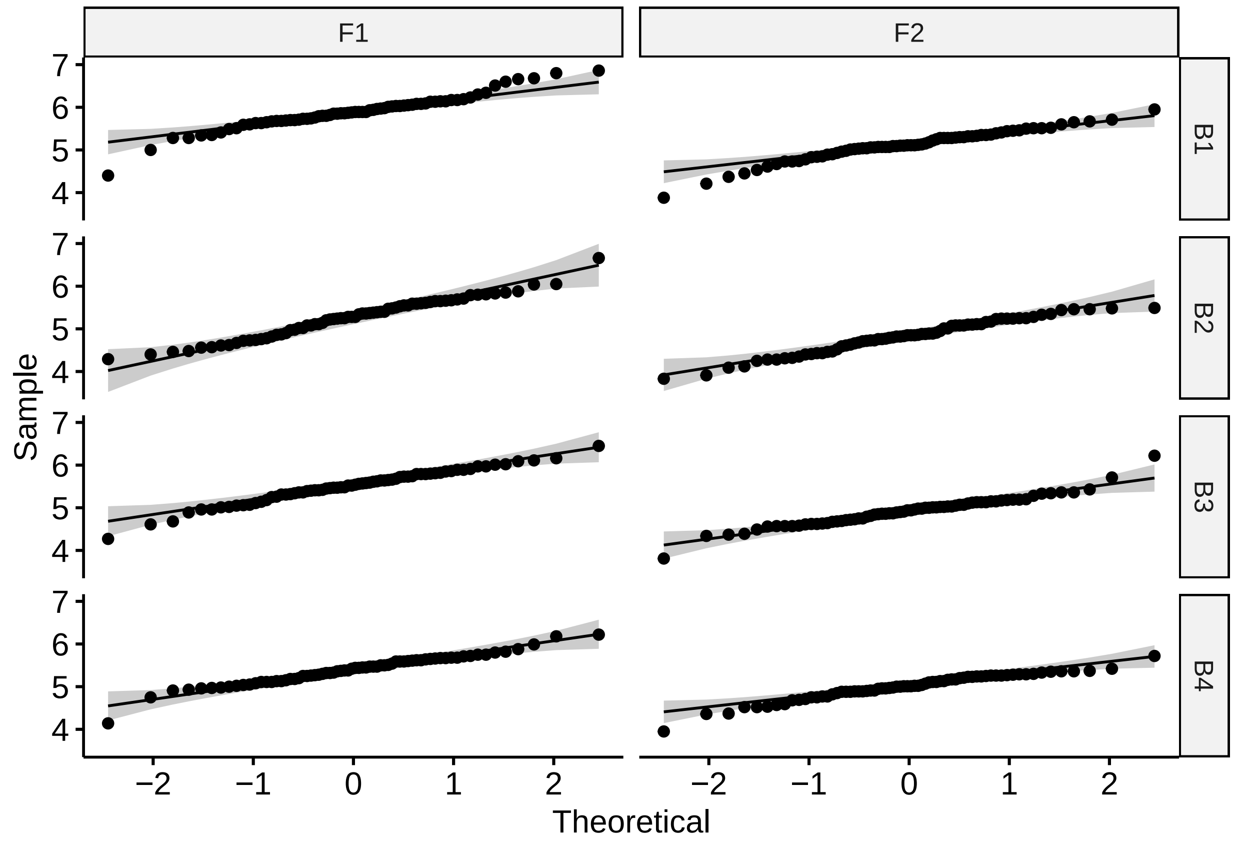
Like a standard factorial ANOVA, there is an assumption that the data is normally distributed within groups (assessed visually with a qq-plot).

```
# normality tests

# whole plot level
ggqqplot(data=durban,x='yield',facet.by='block')
```



```
# sub-plot level
ggqqplot(data=durban,x='yield') + facet_grid(block~fung)
```



There is also an assumption that there is an equality of variance between groups

```
# equality of variances

# whole plot level
levene.test(durban$yield,durban$block)
```

```
##
##  Modified robust Brown-Forsythe Levene-type test based on the absolute
##  deviations from the median
##
## data:  durban$yield
## Test Statistic = 7.0954, p-value = 0.00011
```

```
#sub plot level - block & fung

durban.var <- durban[order(durban$block,durban$fung),]
rownames(durban.var) <- 1:560
# copy of data set with a grouping variable for block AND fungal treatment

durban.var$fung.cell <- rep(seq(1,8,by=1),each=70)
levene.test(durban.var$yield,durban.var$fung.cell)
```

```
##
##  Modified robust Brown-Forsythe Levene-type test based on the absolute
##  deviations from the median
##
## data:  durban.var$yield
## Test Statistic = 2.327, p-value = 0.02399
```

Model Fitting

First, we want to run an ANOVA ignoring the experimental design.

```
model.bad <- aov(yield ~ fung*gen,data=durban)
summary(model.bad)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## fung          1  42.02   42.02 345.432 <2e-16 ***
## gen          69  39.28    0.57   4.680 <2e-16 ***
## fung:gen     69   5.09    0.07   0.606  0.994
## Residuals   420  51.09    0.12
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The residuals/error degrees of freedom is higher, especially in comparison to the degrees of freedom that can be used to test for differences in the sub-plot factor fungal treatment.

In order to account for our experimental design, an additional term needs to be added into the formula specifying an error term for fungal treatment - in the form of Error(whole plot:spilt plot).

```
model.good <- aov(yield ~ fung*gen + Error(block:fung),data=durban)
```

```
## Warning in aov(yield ~ fung * gen + Error(block:fung), data = durban): Error()
## model is singular
```

```
summary(model.good)
```

```
##
## Error: block:fung
##           Df Sum Sq Mean Sq F value Pr(>F)
## fung       1  42.02   42.02   13.73   0.01 *
## Residuals  6  18.36    3.06
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Within
##            Df Sum Sq Mean Sq F value Pr(>F)
## gen        69  39.28  0.5693   7.201 <2e-16 ***
## fung:gen   69   5.09  0.0738   0.933  0.629
## Residuals 414  32.73  0.0791
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is an additional way to run an analysis that accounts for a split-plot design which is the lmer() function from the lme4 package. Instead of the Error(whole plot:sub-plot) term added to the aov() function, the additional term specified would be indicated with (1|whole plot:sub-plot).

Also we want to use the anova() function rather than summary() to display our results in a readable manner.

```
model.lmer <- lmer(yield ~ fung*gen + (1|block:fung),data=durban)
anova(model.lmer)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##          Sum Sq Mean Sq NumDF DenDF F value  Pr(>F)
## fung      1.086 1.08583     1     6 13.7333 0.01002 *
## gen      39.284 0.56933    69   414  7.2008 < 2e-16 ***
## fung:gen  5.090 0.07377    69   414  0.9331 0.62901
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

If you compare both models, you can see that our p-values are just about the same, as well the degrees of freedom both for predictors/interactions and residuals

Interpreting Results

In reviewing the results of our analyses, we found significant main effects for both fungal treatment type and barley genotype, but no significant interaction for fungal treatments & barley genotype together.