Multi-Factor GLM Homework Answers

Crispin Jordan

26/10/2020

Question 1

Part (a)

Let's start by obtaining the data:

```
data(npk)
```

Let's look at the top of the dataframe (using the `head()` function):

```
head(npk)
```

```
##   block N P K yield
## 1     1 0 1 1  49.5
## 2     1 1 1 0  62.8
## 3     1 0 0 0  46.8
## 4     1 1 0 1  57.0
## 5     2 1 0 0  59.8
## 6     2 1 1 1  58.5
```

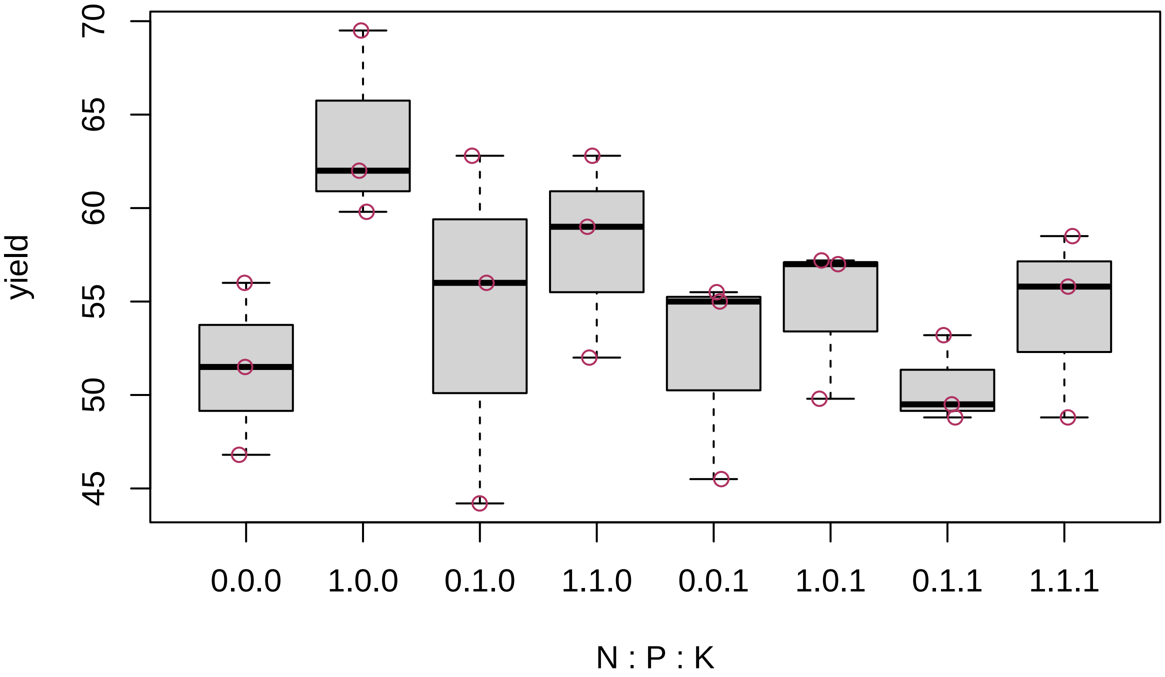
We see six columns. The first, `block`, the question told us to ignore for simplicity, which we will do (for now). Next, we have N, P and K, which indicate whether each of these elements were added to fields; a 0 and 1 indicates that the element was not or was added, respectively. Lastly, we have `yield`, which indicates the measure of yield for a given experimental unit. Our hypothesis is that the elements (N, P and K) would affect `yield`; therefore `yield` is the *dependent* variable.

NOTE that we should always start by asking whether the data meet the assumptions of randomization and independence. As we lack these details for these data, we'll assume the data meet these assumptions.

Part (b)

Let's plot the data. We'll use the `boxplot()` command. `yield` is the dependent variable, and so will go to the left of the tilda (`~`); we can plot different combinations of the treatments N, P and K by listing all three of these factors and place a '`*`' between them:

```
boxplot(yield ~ N*P*K, data= npk)
stripchart(yield ~ N*P*K, data= npk, add = TRUE, vertical = TRUE,
           method = "jitter", pch = 21, col = "maroon")
```



Let's look at the x-axis labels; we see three numbers (either 0's or 1's) separated by commas; the three numbers refer to N, P and K, in that order. A '1' indicates that an element was added.

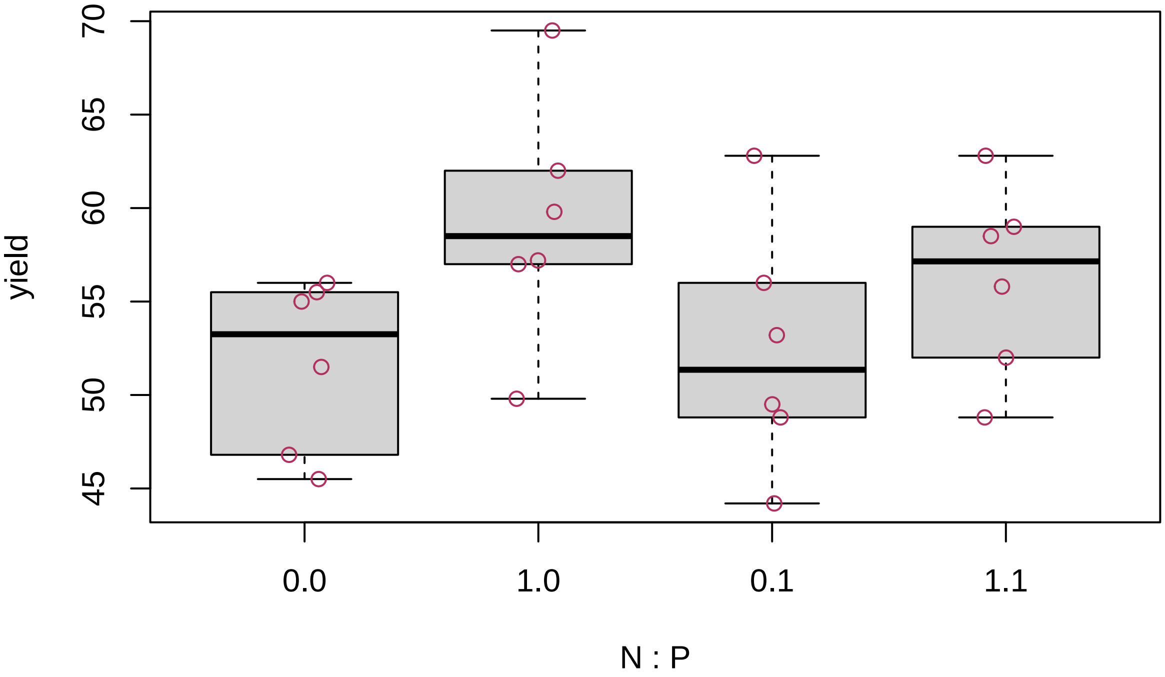
The widths of the boxplots are fairly consistent among the treatment combinations, suggesting that the data will meet the assumption of equal variance. (But also note that the sample size for each treatment combination is small, being 3; such a small sample size makes it hard to assess equal variance at this scale). Similarly, the boxplots are generally symmetrical, suggesting normally distributed data; but the sample sizes (when plotted like this) are ridiculously small to make any useful conclusions.

Finally, it is difficult to get a sense of any effects on yield when plotting the data like this.

Let's try plotting the data for combinations of 2 Factors, rather than combinations of 3 Factors (as we did, above).

Let's first look at the data for combinations of N and P:

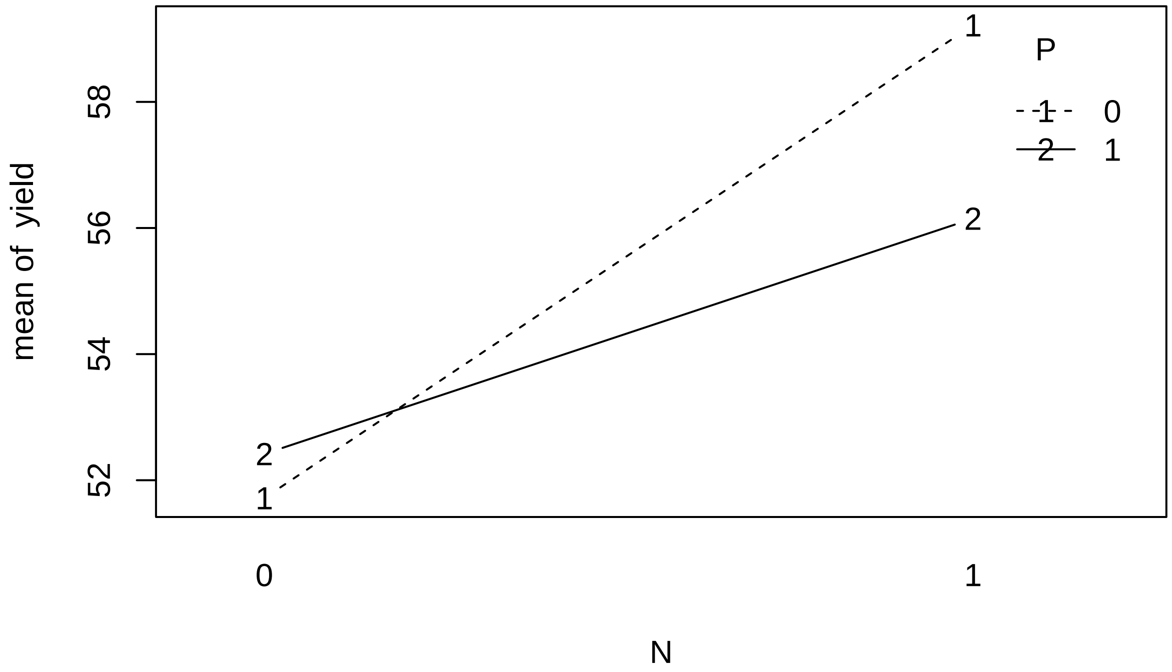
```
boxplot(yield ~ N*P, data= npk)
stripchart(yield ~ N*P, data= npk, add = TRUE, vertical = TRUE,
           method = "jitter", pch = 21, col = "maroon")
```



The boxplot suggests that the assumptions of equal variance (similar breadth among boxplots) and normality (fairly symmetrical boxplots) are met. The boxplot indicates that yield increases when N is present (i.e., the first value equal 1; see 1:0 and 1:1 (Nitrogen present) compared to 0:0 and 0:1 (Nitrogen absent)). It also suggests little evidence for an effect of Phosphorus (P), or for an interaction.

Let's also plot the data using an interaction plot. The first term (`N`) indicates the variable to be plotted along the x-axis; the effect of the second Factor listed (`P`) will be illustrated by presenting levels of this Factor within the plot; the third variable (`yield`) designates the y-axis; the option `type = "b"` indicates to plot both points and lines; `legend = TRUE` will cause a legend to be produced.

```
attach(npk)
interaction.plot(N, P, yield, type = "b", legend = TRUE)
```

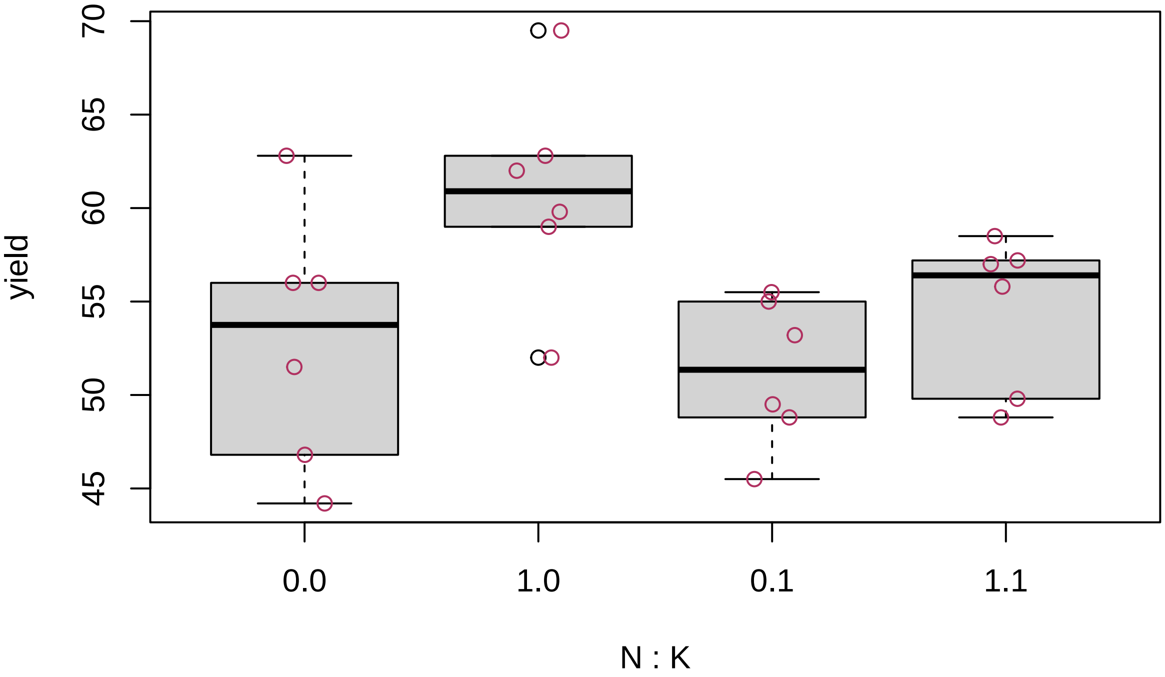


```
detach(npk)
```

The `interaction.plot` also suggests little evidence for an effect of Phosphorus (P), or for an interaction. Note that the `interaction.plot()` output displays slopes that appear unequal, which would suggest an interaction between N and P. But, the boxplot indicates that there's lots of variation around the means, so the fact that the lines do not look parallel might not indicate an interaction in this case (i.e., the mean values are likely not estimated with great certainty, so the slopes may also lack great certainty).

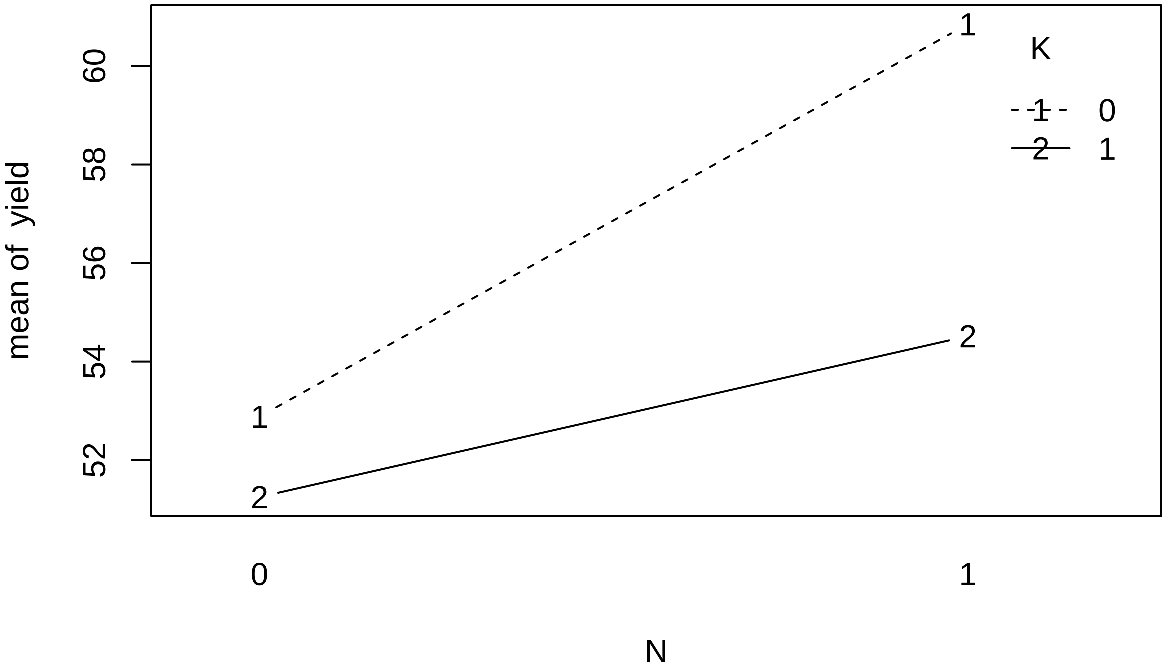
Now let's look at the data for combinations of N and K:

```
boxplot(yield ~ N*K, data= npk)
stripchart(yield ~ N*K, data= npk, add = TRUE, vertical = TRUE,
           method = "jitter", pch = 21, col = "maroon")
```



And, an interaction plot for the same data:

```
attach(npk)
interaction.plot(N, K, yield, type = "b", legend = TRUE)
```

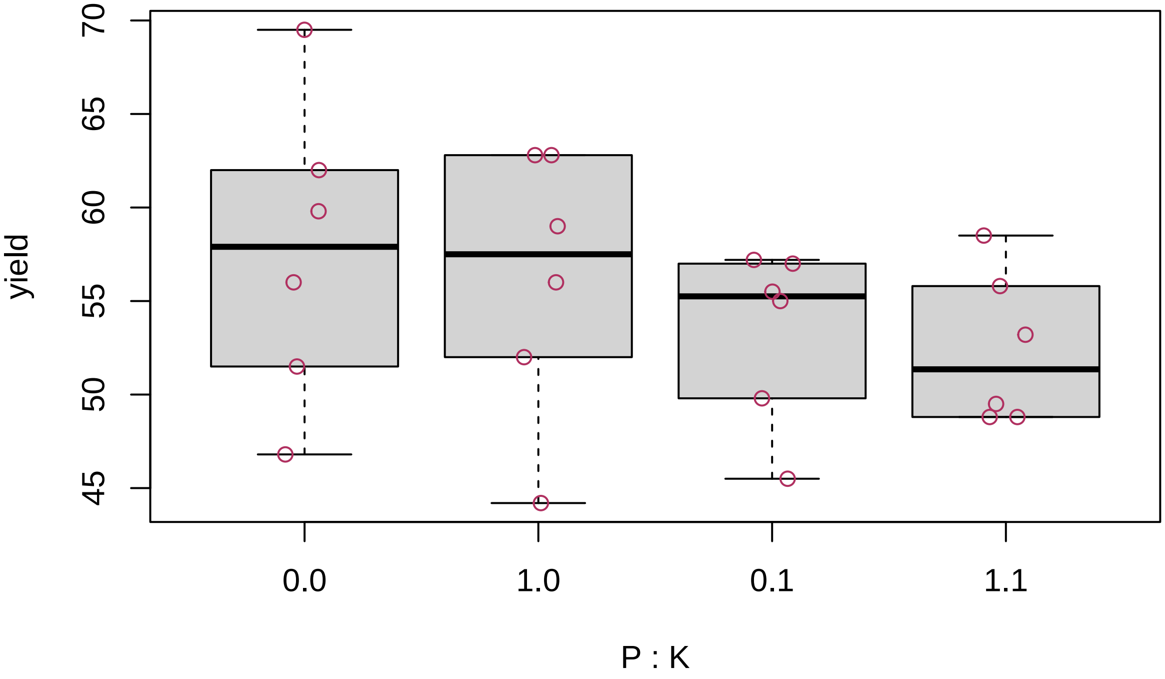


```
detach(npk)
```

It is hard to assess with so few data, but my first impression is that `yield` increases with N but decreases with K (and there's little evidence for an interaction; the lines are relatively parallel). The plots suggest relatively equal variance and normally distributed residuals.

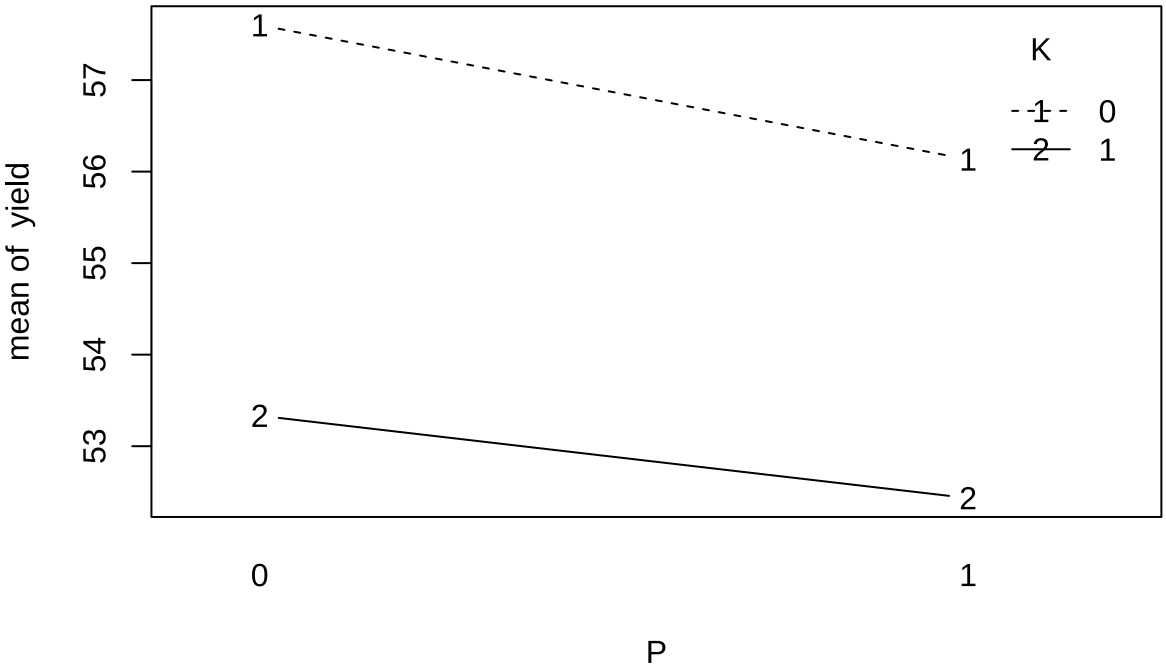
Let's look at the last combination of P and K:

```
boxplot(yield ~ P*K, data= npk)
stripchart(yield ~ P*K, data= npk, add = TRUE, vertical = TRUE,
           method = "jitter", pch = 21, col = "maroon")
```



And, an interaction plot for the same data:

```
attach(npk)
interaction.plot(P, K, yield, type = "b", legend = TRUE)
```



```
detach(npk)
```

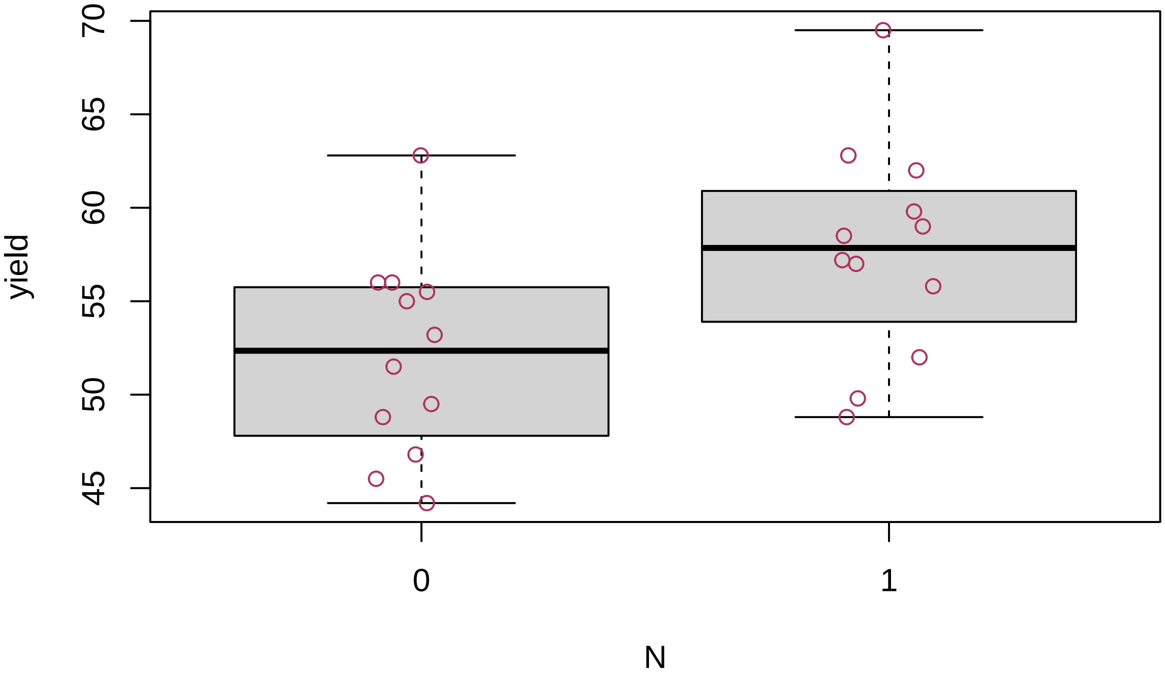
These plots suggests a suggests that there is no effect of P, but that the addition of K *reduces* yield (compare

0:0 and 1:0 (K absent) with 0:1 and 1:1 (K present)). Again, the shapes of the boxplots suggest that the data will meet the assumptions of equal variance and normality.

Finally, as we've not seen evidence for any strong interactions, let's plot the data for each Factor, separately.

For N:

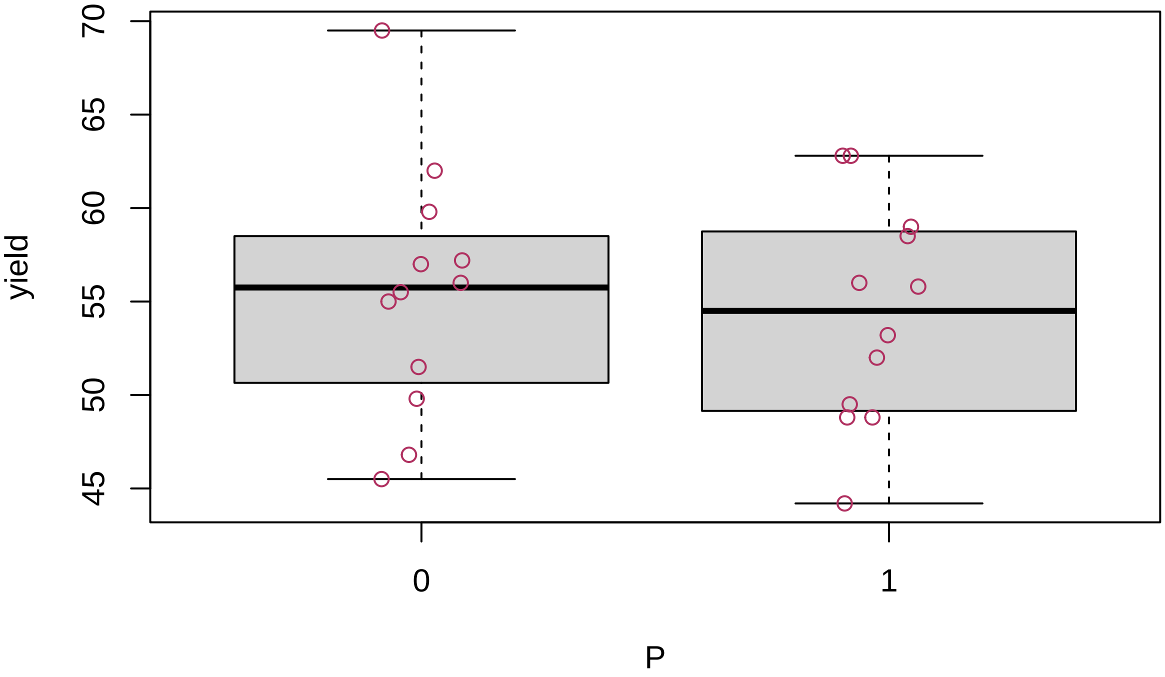
```
boxplot(yield ~ N, data= npk)
stripchart(yield ~ N, data= npk, add = TRUE, vertical = TRUE,
           method = "jitter", pch = 21, col = "maroon")
```



It looks like `yield` tends to increase with the addition of Nitrogen (and the assumptions of equal variance and normality are met).

Now compare P treatments:

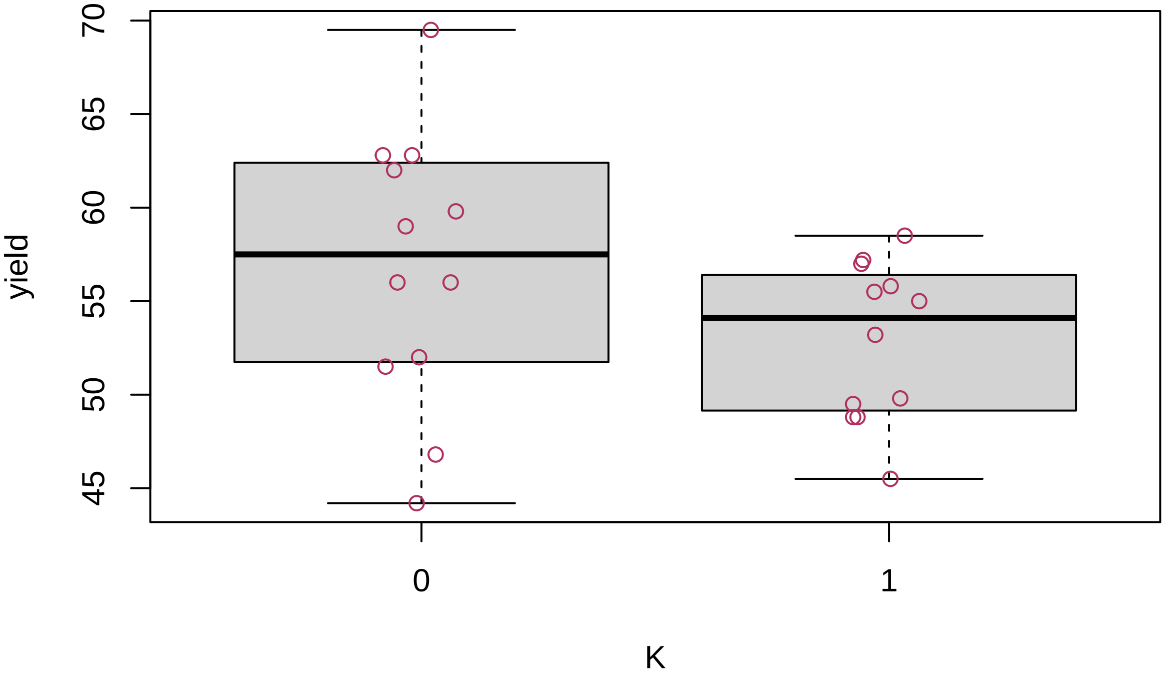
```
boxplot(yield ~ P, data= npk)
stripchart(yield ~ P, data= npk, add = TRUE, vertical = TRUE,
           method = "jitter", pch = 21, col = "maroon")
```



It looks like adding Phosphorus does not affect `yield` (but, again, assumptions are likely met).

Finally, compare K treatments:

```
boxplot(yield ~ K, data= npk)
stripchart(yield ~ K, data= npk, add = TRUE, vertical = TRUE,
           method = "jitter", pch = 21, col = "maroon")
```



These data suggest that adding potassium (K) tends to decrease `yield`.

Part (c)

Let's model the data.

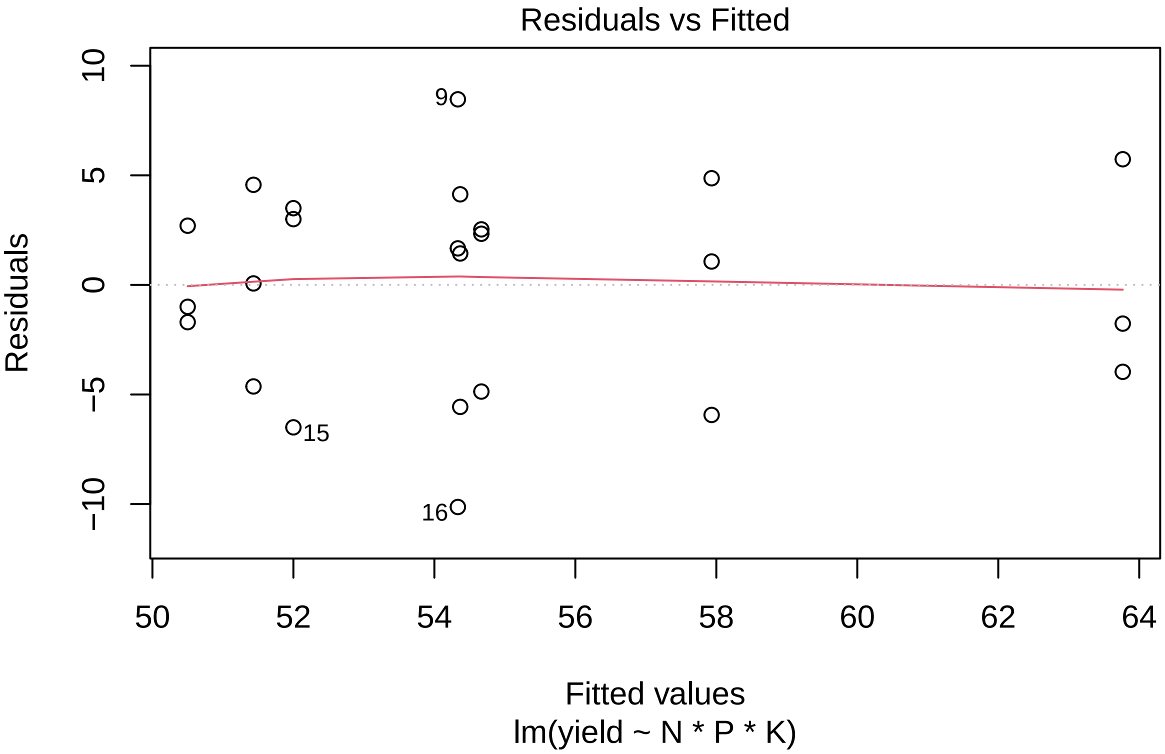
We know from both the question's description of the dataset that we have a balanced experimental design, with 3 replicates in all combinations of the levels of the three Factors. The replication is sufficient to allow us to model interactions; the fact that the data are balanced means that we can obtain p-values using the `anova()` function.

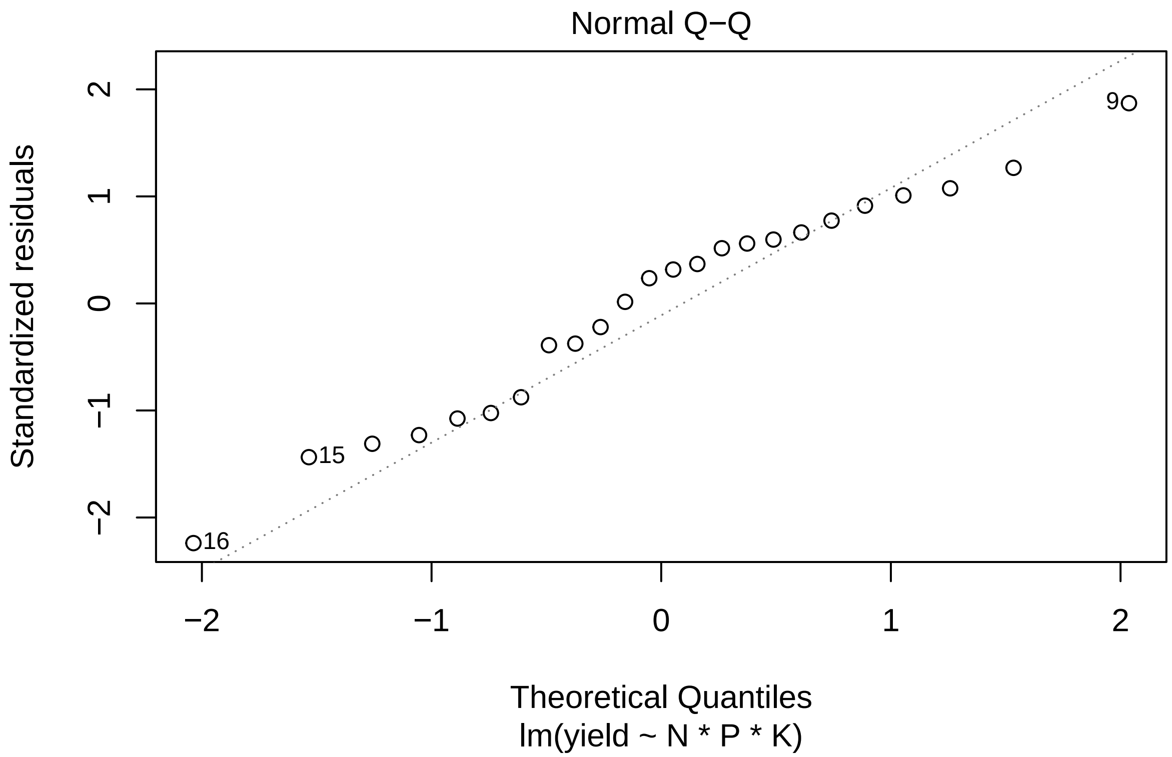
Let's create the model. Remember that our *dependent* variable is `yield`. Also note this **shortcut**. Instead of entering each factor (N, P and K) and all interactions (N:P, N:K, K:P, and the 3-way interaction, N:P:K), we can simply enter `N*P*K`. Nice, eh?

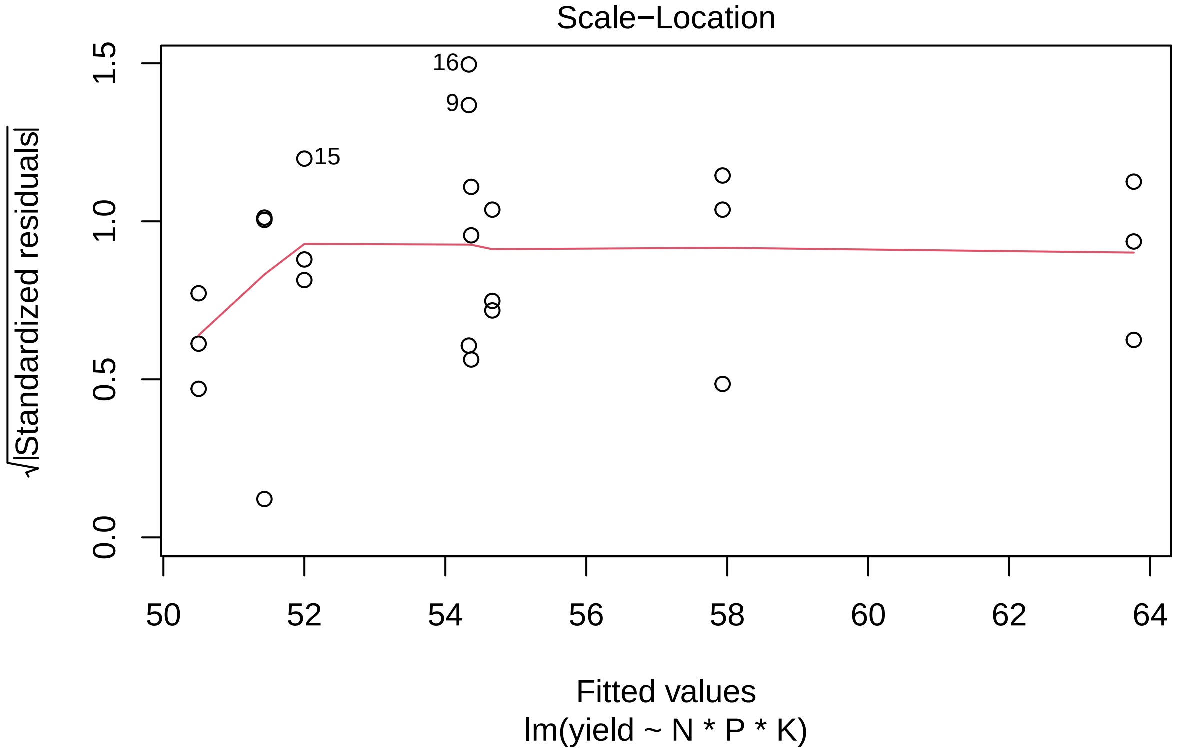
```
npk.lm <- lm(yield ~ N*P*K, data= npk)
```

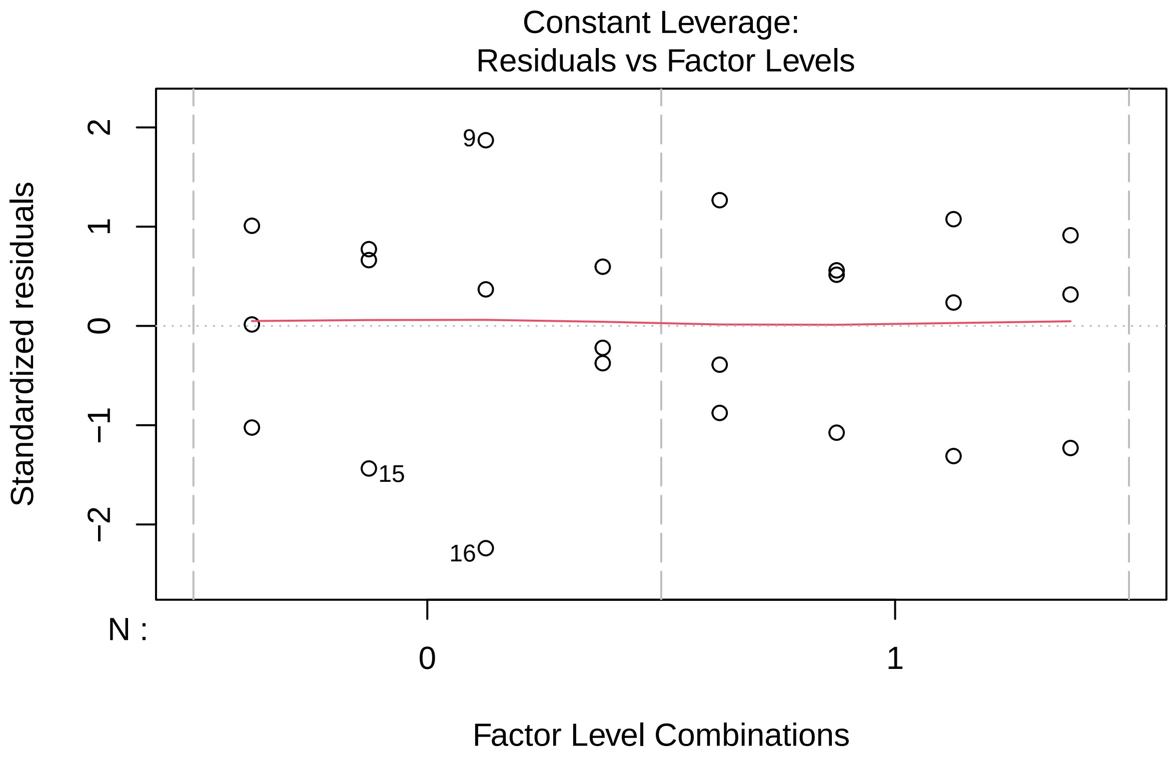
Now, let's plot the residuals to check teh assumptions of equal variance and normally distributed residuals:

```
plot(npk.lm)
```




Normal Q−Q
Standardized residuals
2
1
0
−1
−2
9
15
16
−2
−1
0
1
2
Theoretical Quantiles
lm(yield ~ N * P * K)


Scale−Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
16
9
15
50
52
54
56
58
60
62
64
Fitted values
lm(yield ~ N * P * K)



The first plot suggests that the data meet the assumption of equal variance (notice that the 'spread' of residuals is relatively equal as we move along the fitted values (i.e., along the x-axis), whereas the second plot indicates that the data are normally distributes (the points are generally close to the dotted line). The third plot also suggests the data likely meet the assumption of equal variance: The red line is almost perfectly flat, except where it dips downward on the far left. This downward dip is due to the two treatment combinations with the lowest mean values. Note however, that we have a very small sample size if each of these two groups; for this reason, I'm not too worried about unequal variance, as it is very easy for fluctuations to occur with small sample sizes. Moreover, we learned that violations of equal variance have the lowest impact on conclusions when they arise due to one group having a notably lower variance: *if* variance is unequal with these data, then it appears that they'd be unequal because one group (the group with the smallest mean) had an unusually small variance. Again, we have reason to not worry much about equal variance.

As we don't know much about how the data were collected, we'll assume that the data meet the assumption of random sampling and independence.

As the data meet the assumptions of our analysis, let's calculate p-values:

```
anova(npk.lm)
```

```
## Analysis of Variance Table
##
## Response: yield
##           Df Sum Sq Mean Sq F value  Pr(>F)
## N          1 189.28 189.282  6.1608 0.02454 *
## P          1   8.40   8.402  0.2735 0.60819
## K          1  95.20  95.202  3.0986 0.09746 .
## N:P        1  21.28  21.282  0.6927 0.41750
## N:K        1  33.14  33.135  1.0785 0.31448
## P:K        1   0.48   0.482  0.0157 0.90192
## N:P:K      1  37.00  37.002  1.2043 0.28870
## Residuals 16 491.58  30.724
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This next step is not necessary, but let's also use the `Anova()` function to calculate p-values (assuming type 3 sums of squares). Notice that we obtain the same p-values, *because the data are balanced*:

```
npk.lm.ss3 <- lm(yield ~ N*P*K, data= npk,
                 contrasts = list(N = contr.sum, P = contr.sum, K = contr.sum))

library(car)
```

```
## Loading required package: carData
```

```
Anova(npk.lm.ss3, type = 3)
```

```
## Anova Table (Type III tests)
##
## Response: yield
##              Sum Sq Df   F value  Pr(>F)
## (Intercept)   72270  1 2352.2641 < 2e-16 ***
## N               189  1    6.1608 0.02454 *
## P                 8  1    0.2735 0.60819
## K                95  1    3.0986 0.09746 .
## N:P              21  1    0.6927 0.41750
## N:K              33  1    1.0785 0.31448
## P:K               0  1    0.0157 0.90192
## N:P:K            37  1    1.2043 0.28870
## Residuals       492 16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Yes, we confirm that the two sets of output are identical.

The p-values indicate what we'd already suspected: there's no strong evidence for any interactions (either 2- or 3-way interactions). So, let's focus on the main effects, which reflect our expectations from plotting the data. The p-value for N (0.02) provides 'suggestive' or 'moderate' evidence for an effect of N on `yield`; the p-value for K (0.09) is not too far from 0.05; we might say that this p-value provides between 'weak' to 'suggestive' evidence for an effect of P on `yield`. Finally, the p-value for P is 0.6 (very weak evidence for any possible effect of P on yield).

Our next job is to understand our results in greater depth. Exactly what we do should depend on our original motivation for the experiment. For example, if our motivation had been to determine whether combinations of N, P and K interact, then we could focus next on characterizing interactions. In the present case, let's keep it simple: let's focus on understanding the 'main effects' of N, P and K on `yield` on their own (i.e., when they are not involved in interactions). This is straight-forward in our case because there is no evidence for interactions, either from plots or p-values. Therefore, we can characterize the effect size of N by averaging over the effects of P and K; we can use a similar approach to understand the effect sizes for P (average over N and K) and K (average over N and P).

We'll use `emmeans()` for this next step:

```
library(emmeans)
```

Let's start by determining the effect size of N on 'yield':

```
n.emmeans <- emmeans(npk.lm, "N")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
n.emmeans
```

```
##  N emmean  SE df lower.CL upper.CL
##  0   52.1 1.6 16     48.7     55.5
##  1   57.7 1.6 16     54.3     61.1
##
## Results are averaged over the levels of: P, K
## Confidence level used: 0.95
```

This first output provides the mean `yield` when N was present vs. absent, averaged over P and K (notice that `emmeans` reminds of this fact in the output!). We could report these emmeans (i.e., "estimated marginal means") for the two N treatments: their means, SE and 95% CI's. It looks like mean yield was about 5 units higher when N was added, which represents an approximate increase of 10%. Let's quantify that better with the `pairs()` function:

```
n.pairs <- pairs(n.emmeans)
n.pairs
```

```
##  contrast estimate   SE df t.ratio p.value
##  0 - 1       -5.62 2.26 16 -2.482  0.0245
##
## Results are averaged over the levels of: P, K
```

This output indicates that mean yield was 5.62 units higher when N was added (notice that this estimate was negative when the mean when N was present was subtracted from the mean when N was absent; this implies that the mean is higher when N is present, as we saw in the figures we created above); the SE for this effect size equals 2.26.

Next, we can obtain 95% CI's for this effect size:

```
confint(n.pairs)
```

```
##  contrast estimate   SE df lower.CL upper.CL
##  0 - 1       -5.62 2.26 16    -10.4    -0.82
##
## Results are averaged over the levels of: P, K
## Confidence level used: 0.95
```

We'll discuss these results for N below, in *Part (d)*.

Now, let's repeat this process for P:

```
p.emmeans <- emmeans(npk.lm, "P")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
p.emmeans
```

```
##  P emmean  SE df lower.CL upper.CL
##  0   55.5 1.6 16     52.1     58.9
##  1   54.3 1.6 16     50.9     57.7
##
## Results are averaged over the levels of: N, K
## Confidence level used: 0.95
```

```
p.pairs <- pairs(p.emmeans)
p.pairs
```

```
##  contrast estimate   SE df t.ratio p.value
##  0 - 1        1.18 2.26 16 0.523   0.6082
##
## Results are averaged over the levels of: N, K
```

```
confint(p.pairs)
```

```
##  contrast estimate   SE df lower.CL upper.CL
##  0 - 1        1.18 2.26 16    -3.61     5.98
##
## Results are averaged over the levels of: N, K
## Confidence level used: 0.95
```

Note that the 95% CI's for the effect size of P ranges from -3.61 to 5.98; these suggest a range of possible effects of increasing yield by about 6.5% (3.61 / 55.5) to decreasing yield by about 10.8% (5.98 / 55.5). This implies a lot of uncertainty regarding the effect of P on yield.

. . . and we can also repeat this process for K:

```
k.emmeans <- emmeans(npk.lm, "K")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
k.emmeans
```

```
##  K emmean  SE df lower.CL upper.CL
##  0   56.9 1.6 16     53.5     60.3
##  1   52.9 1.6 16     49.5     56.3
##
## Results are averaged over the levels of: N, P
## Confidence level used: 0.95
```

```
k.pairs <- pairs(k.emmeans)
k.pairs
```

```
##  contrast estimate   SE df t.ratio p.value
##  0 - 1        3.98 2.26 16 1.760   0.0975
##
## Results are averaged over the levels of: N, P
```

```
confint(k.pairs)
```

```
##  contrast estimate   SE df lower.CL upper.CL
##  0 - 1        3.98 2.26 16   -0.814     8.78
##
## Results are averaged over the levels of: N, P
## Confidence level used: 0.95
```

Note that, even though the p-value for K was relatively large, the range of reasonable effect sizes for K (see the 95% confidence intervals for the effect size in the output of `confint(k.pairs)`: -0.814 to 8.78) are very similar to those for N; , see `confint(n.pairs)` output for 95% CI's for effect size of N. This highlights the importance of considering effect sizes and not only p-values (the effect sizes indicate that it is reasonable to expect that effect sizes of N and K could be similar, but in opposite directions (adding K decreased yield)).

Part (d)

Let's use the results from the N treatment as an example of how we might report these results.

First, we'd refer to a nice plot of our data (not provided here). Next, we say something like. . . Both out displayed plots and p-values provide no evidence for either 2- or 3-way interactions between N, P and K on yield (2-Factor GLM; N:P, $F_{1,16} = 0.6927$, p = 0.41750; N:K . . . etc (fill the rest in, yourself)). Therefore, we focused on characterizing the main effects of N, P and K on `yield`. Adding N increased yield from (estimated marginal mean (SE)) 52.1 (1.6) to 57.7 (1.6) (main effect of N: $F_{1,16} = 6.1608$, p = 0.02454). This effect size (estimated contrast = 5.62, SE = 2.26) represents an approximate 10% increase in yield due to the addition of Nitrogen; 95% confidence intervals for this effect size (0.82 to 10.4) indicate, however that this effect size could reasonable range from roughly 1.5% (0.82 / 52.1) to 20% (10.4 / 52.1), reflecting high uncertainty in this estimated effect size.

. . . You could follow a similar approach for report results from P or K.

Question 2

Part (a)

Let's import the data:

```
amp <- read.table("Amplitude.csv", header = TRUE, sep = ",")
```

Now, let's inspect the data:

```
summary(amp)
```

```
##    Genotype          RecordingCondition   Amplitude
##  Length:117         Length:117         Min.   :  5.70
##  Class :character   Class :character   1st Qu.: 43.01
##  Mode  :character   Mode  :character   Median : 78.06
##                                        Mean   :102.51
##                                        3rd Qu.:141.10
##                                        Max.   :457.32
```

```
head(amp)
```

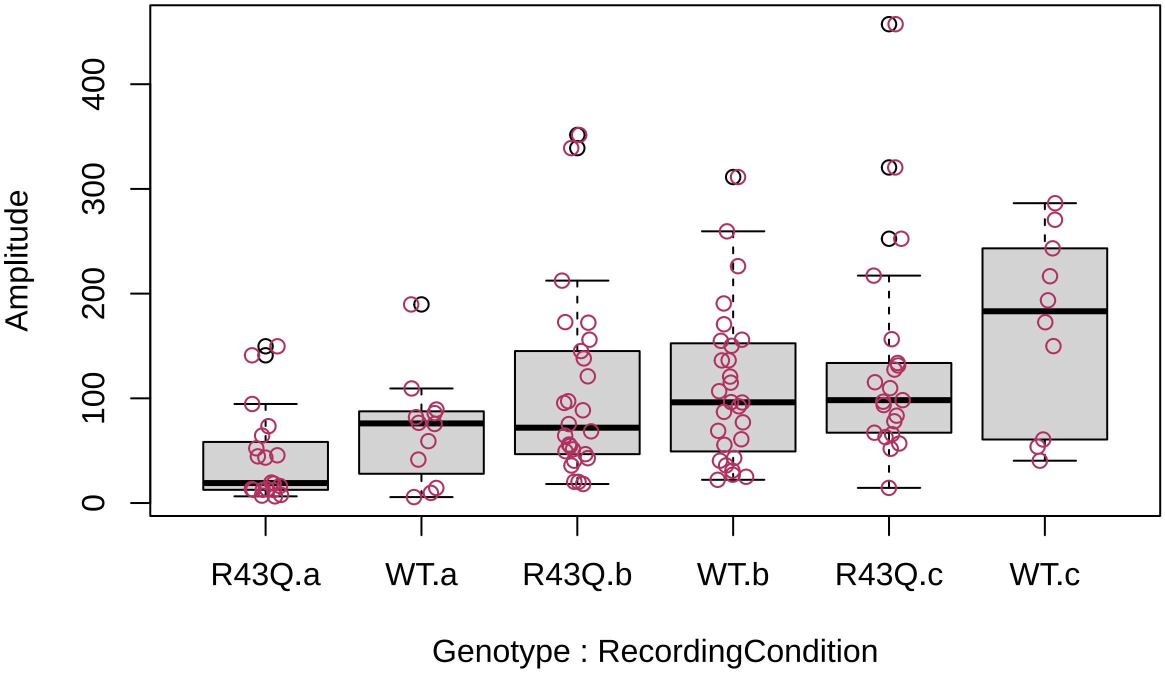
```
##   Genotype RecordingCondition Amplitude
## 1       WT                  a     14.40
## 2       WT                  a      5.70
## 3       WT                  a     89.30
## 4       WT                  a    109.42
## 5       WT                  a      9.70
## 6       WT                  a     85.90
```

We see that the dataframe, `amp`, holds 3 columns: `Genotype`, `RecordingCondition`, and `Amplitude`. The hypothesis behind this research is that the `Genotype` and conditions (`RecordingCondition`) affect the extent of FS cell activation (`Amplitude`). Therefore, `Amplitude` is the *dependent* variable, and `Genotype` and `RecordingCondition`are the *independent* variables.

Part (b)

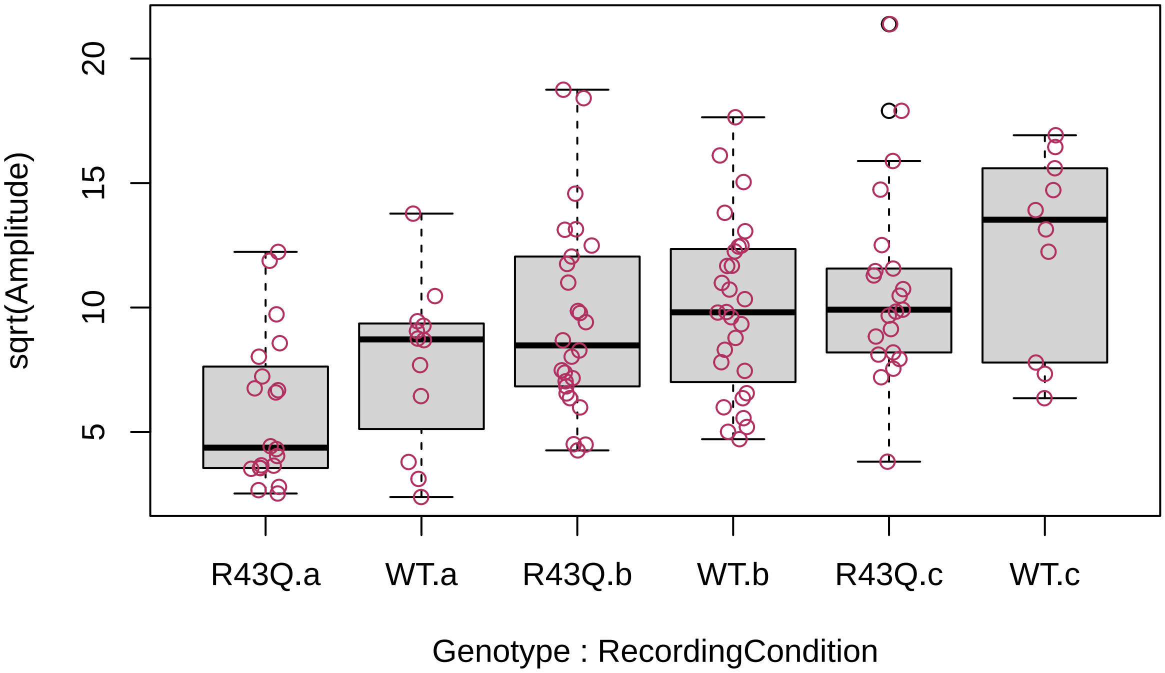
Given that `Amplitude` is the dependent variable, we can plot the data like this:

```
boxplot(Amplitude ~ Genotype*RecordingCondition, data = amp)
stripchart(Amplitude ~ Genotype*RecordingCondition, data = amp, add = TRUE,
           vertical = TRUE, method = "jitter", pch = 21, col = "maroon")
```



The first thing we notice is that the the boxplots are asymmetrical; and more importantly, the breadth of the boxplots seems to increase as the mean values increase (suggesting that the data will violate the assumption of equal variance). These observations suggest that we may need to transform the data for the analysis. Let's try that now, with a square-root transformation (I also tried a log-transformation and thought that a square-root transformation was marginally better - you should try them both and decide for yourself):

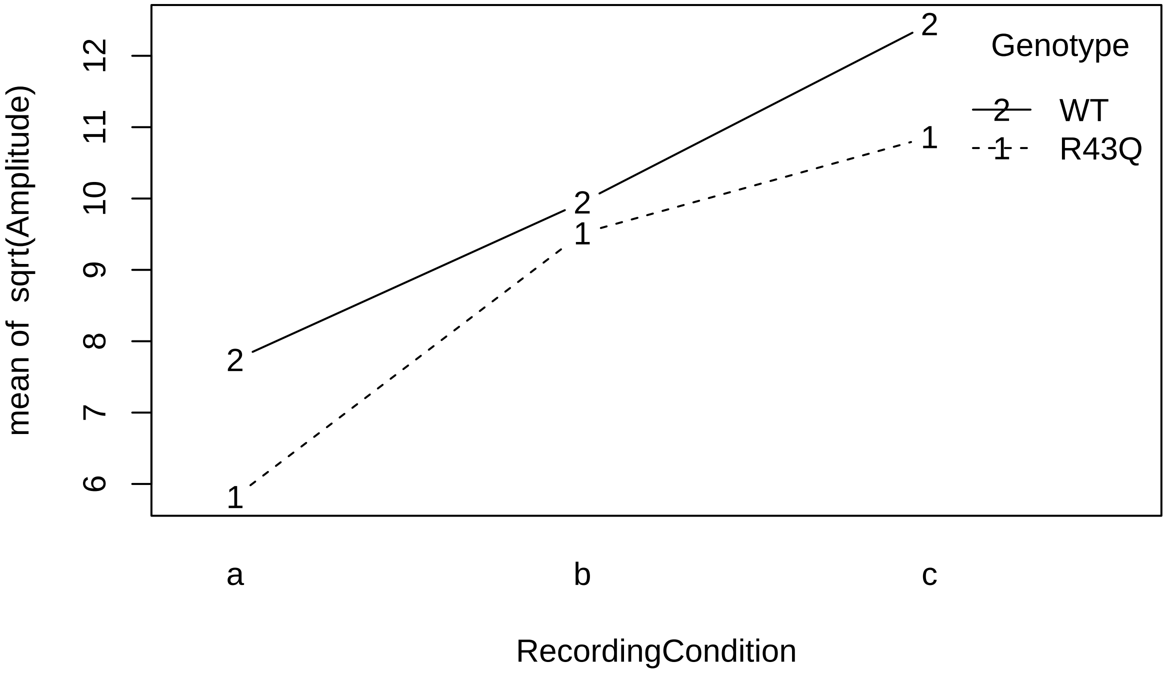
```
boxplot(sqrt(Amplitude) ~ Genotype*RecordingCondition, data = amp)
stripchart(sqrt(Amplitude) ~ Genotype*RecordingCondition, data = amp,
           add = TRUE, vertical = TRUE, method = "jitter", pch = 21,
           col = "maroon")
```



That looks a bit better. Now, the boxplots are relatively symmetrical, suggesting that the residuals will be normally distributed; and more important, the breadth of each boxplot is relatively similar, suggesting that the data meet the assumtpion of equal variance. We also see that `log(Amplitude)` seems to differ among the level of `RecordingCondition` (a, b and c), increasing from `a` to `c`. Also note that in condition `a`, the value of `log(Amplitude)` seems to be higher for the wild-type (WT) than the mutant (R43.Q) than we see for other levels of `RecordingCondition`; this *might* suggest a possible interaction between `Genotype` and `RecordingCondition`.

Let's also look at the data using an interaction plot (we'll plot the data again on the square-root-scale):

```
attach(amp)
interaction.plot(RecordingCondition, Genotype, sqrt(Amplitude), type = "b",
                 legend = TRUE)
```



```
detach(amp)
```

This interaction plot provides a slightly different view than the boxplot, above. The interaction plot suggests that the difference between genotypes is relatively similar in conditions 'a' and 'c', but might be slightly less in 'b'. Clearly, it is difficult to tell whether we expect an interaction; we'll need to wait for the results from our model to clear this up.

Part (c)

Before we model the data, let's think about the experimental design and the assumptions. Let's start by looking at the levels of replication within each treatment combination:

```
library(doBy)
summaryBy(Amplitude ~ Genotype + RecordingCondition, data = amp,
          FUN = c(length, mean, sd))
```

```
##   Genotype RecordingCondition Amplitude.length Amplitude.mean Amplitude.sd
## 1     R43Q                  a               20       42.46250     43.16827
## 2     R43Q                  b               26      105.14827     88.32765
## 3     R43Q                  c               21      132.89200    103.16408
## 4       WT                  a               12       69.90167     50.91706
## 5       WT                   b                28       110.48571      73.55984
## 6       WT                   c                10       168.78238      90.91207
```

The code, above, caused the function summaryBy to look at the values of Amplitude for all combinations of the levels of Genotype and RecordingCondition; for each combination, it will calculate the length, mean and sd. The length is what interests us here, because length refers to the number of data points (for each treatment combination) (we'll also calculate the mean and standard deviation (sd) for each treatment combination, but I just threw thos in for fun and to show what we can do). Look at the values in the column, Amplitude.length, and you should notice two things:

1) The values are not identical for all treatment combinations; therefore the data are *unbalanced*, and we need to be careful how we calculate p-values.
2) We have at least two data points in each treatment combination: this means that we can include an interaction between Genotype and RecordingCondition in our model.

Let's now consider our assumptions. The Question tells us that the data are independent, and randomly sampled. *(Note, however, that the original paper does not indicate whether subjects were allocated randomly to treatments, so we're being generous here to assume the assumption of random allocation is met! Happily, the original paper does provide info with respect to independence).* Therefore, so far, our data meet the assumptions of a GLM, and we can proceed to model our data.

Remembering that Amplitude is the dependent variable, we'll model the data as (we'll start with non-transformed data… for now, we'll not include the contrasts to account for unbalanced data):

```
amp.lm <- lm(Amplitude ~ Genotype + RecordingCondition +
               Genotype:RecordingCondition, data = amp)
```

Let's start by looking at the summary():

```
summary(amp.lm)
```

```
##
## Call:
## lm(formula = Amplitude ~ Genotype + RecordingCondition + Genotype:RecordingCondition,
##     data = amp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -128.35  -53.86  -17.55   30.94  324.43
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        42.462     17.659   2.405 0.017848 *
## GenotypeWT                         27.439     28.838   0.952 0.343413
## RecordingConditionb                62.686     23.489   2.669 0.008755 **
## RecordingConditionc                90.430     24.675   3.665 0.000381 ***
## GenotypeWT:RecordingConditionb    -22.102     35.976  -0.614 0.540238
## GenotypeWT:RecordingConditionc      8.451     41.861   0.202 0.840373
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 78.97 on 111 degrees of freedom
## Multiple R-squared:  0.1782, Adjusted R-squared:  0.1412
## F-statistic: 4.814 on 5 and 111 DF,  p-value: 0.0005066
```

Just for practice, let's use the coefficients to calculate the mean values of each treatment combination. Which treatment combination will the (Intercept) represent? By process of elimination (or thinking of the alphabet), we can infer that the (Intercept) represents the mean value of the Genotype R43Q in

RearingCondition ‘a’; this mean equals 42.462. Does that match the output from the summaryBy function? (Go and look!)

Let’s figure out a few more. What is the mean value of the Wt (Wild Type) in RecordingCondition ‘a’? This will equal the reference value (i.e., (Intercept)) for RearingCondition ‘a’ plus the difference between that treatment combination and the combination of Wt in RecordingCondition ‘a’; this latter value is given by ‘GenotypeWT’, which equals 27.439. Therefore, the mean of Wt in RearingCondition ‘a’ equals 42.462 + 27.439 = 69.901. Does that match the summaryBy output, above?

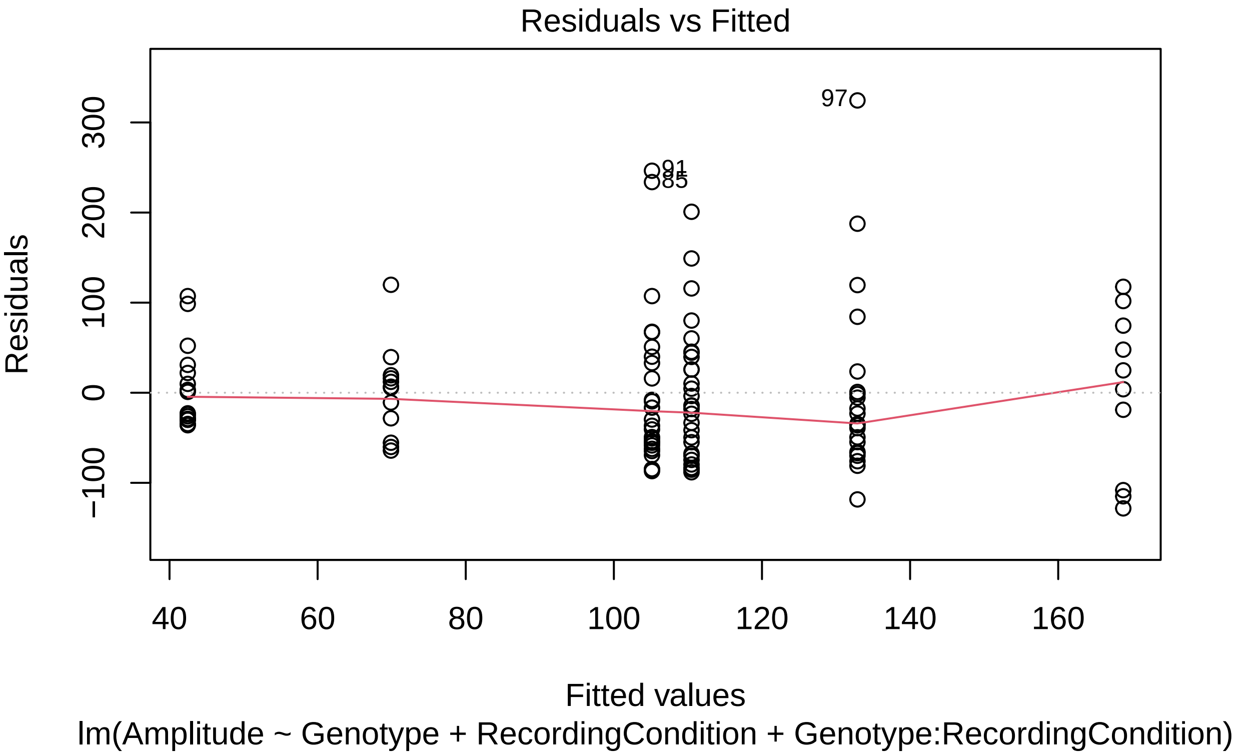
What about the mean of Genotype R43Q in RecordingCondition ‘b’? To find this mean, we add the (Intercept) (which gives the mean for Genotype R43Q in RecordingCondition ‘a’) to the coefficient for RecordingConditionb, which gives the difference between the mean for the (Intercept) (Genotype R43Q in RecordingCondition ‘a’) and the mean for Genotype R43Q in RecordingCondition ‘b’: 42.462 + 62.686 = 105.148. (Again, compare this to the output, above!)

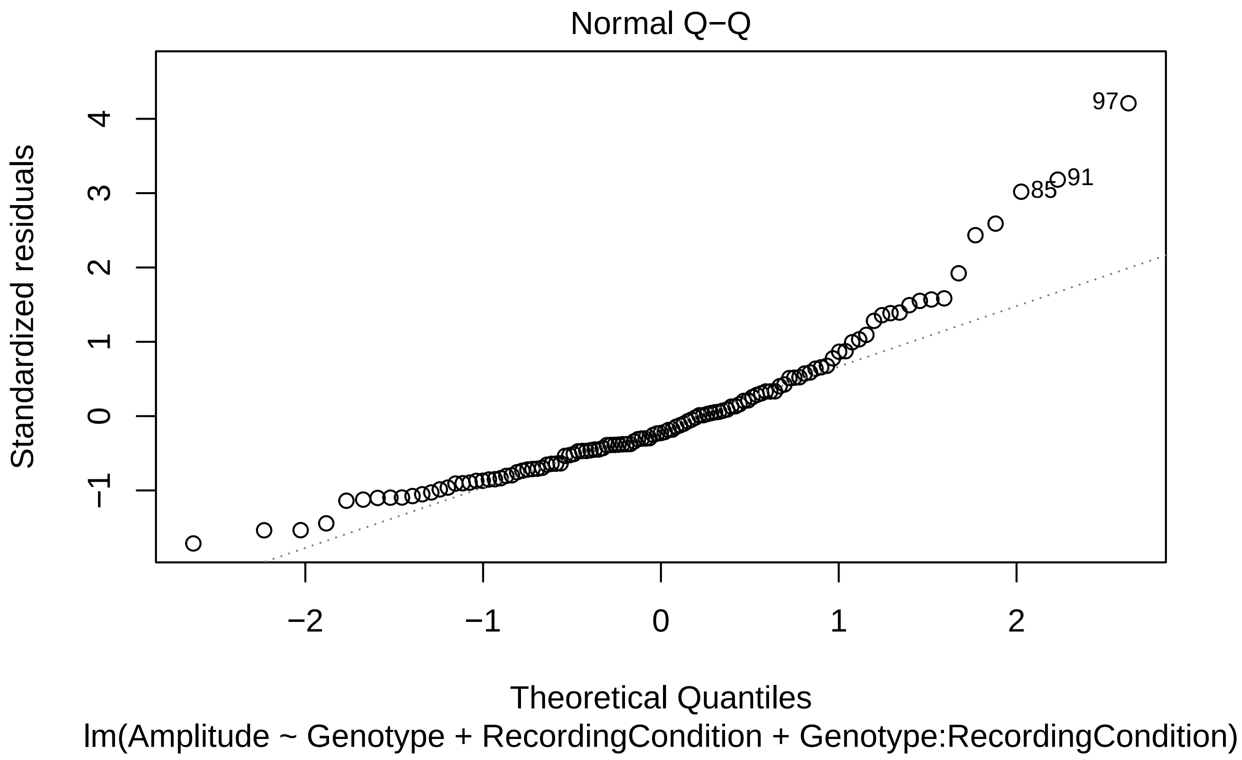
Finally, if we wanted the mean of the Genotype Wt in RearingCondition ‘c’, we would add: 42.462 + 27.439 + 90.430 + 8.451 = 168.782; this value matches the output from summaryBy(), above. Think about what we did here and convince yourself that this is correct.

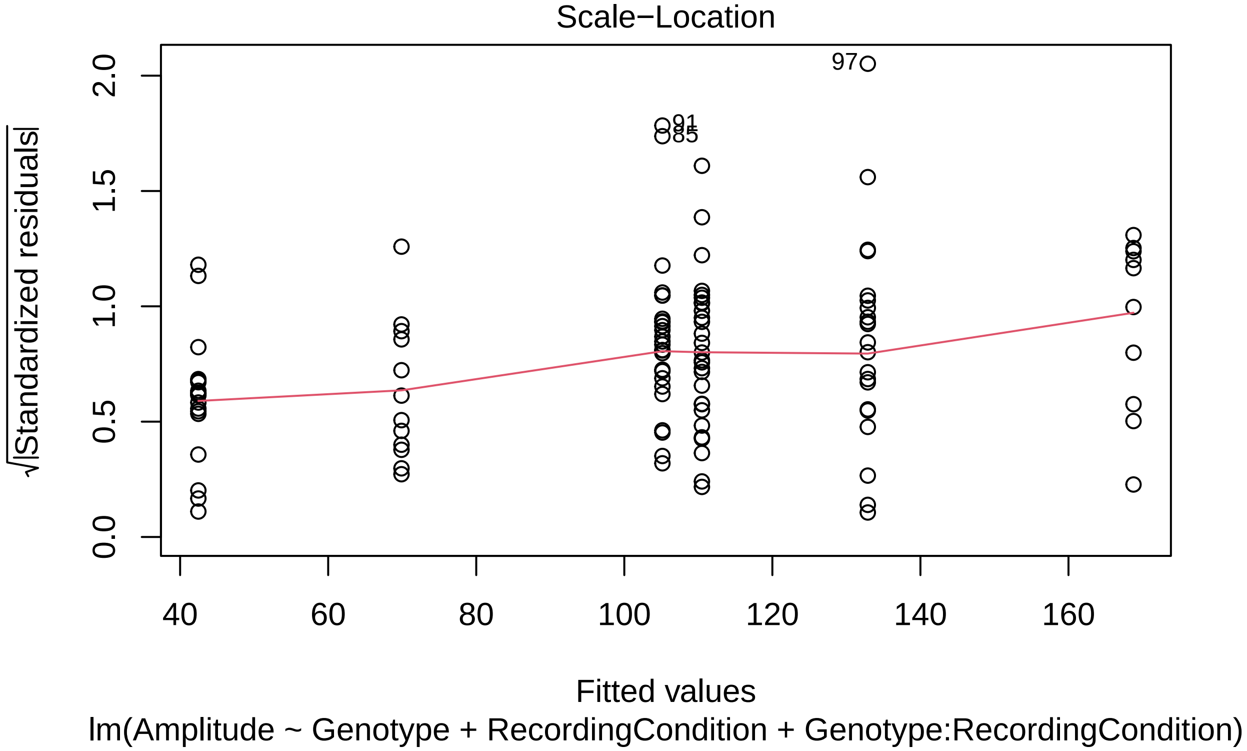
OK, that was enough of a distraction!

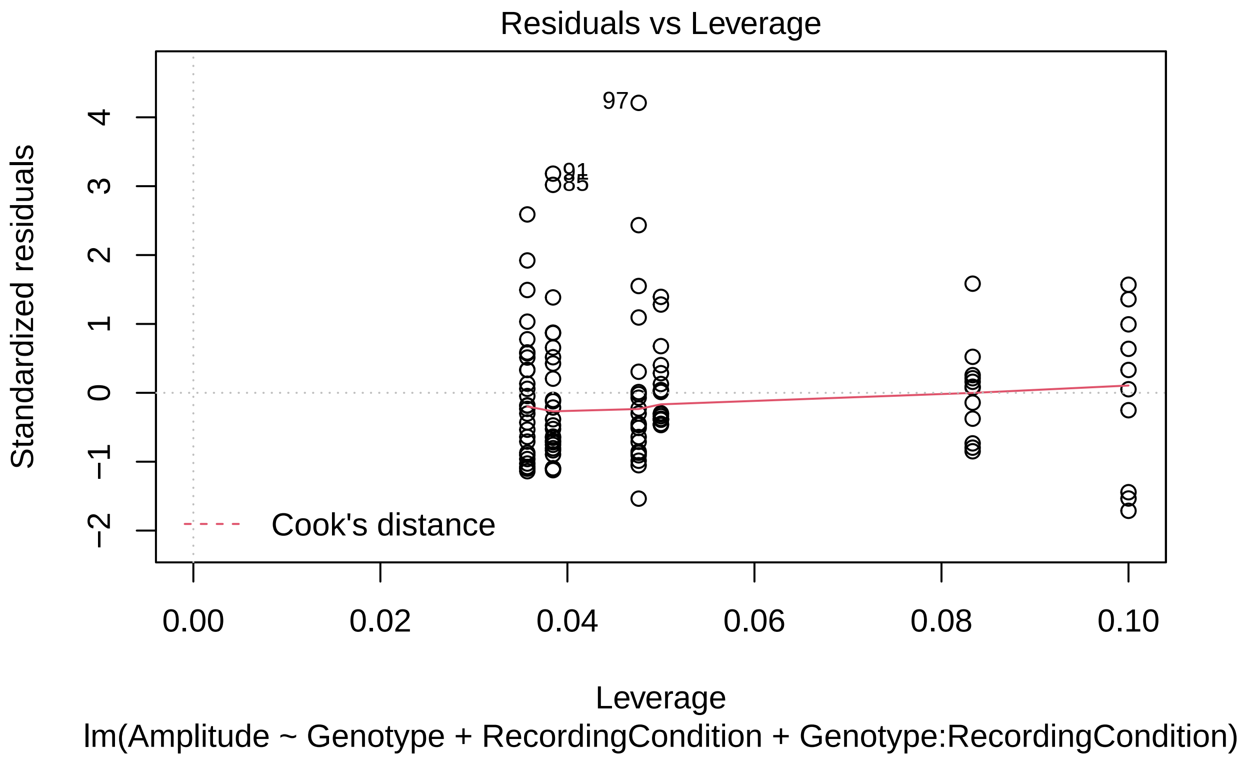
Now, let’s check our assumptions. Remember that these residuals are from a model that did *not* transform the data:

```
plot(amp.lm)
```




Normal Q−Q
Standardized residuals
4
3
2
1
0
−1
97
85
91
−2
−1
0
1
2
Theoretical Quantiles
lm(Amplitude ~ Genotype + RecordingCondition + Genotype:RecordingCondition)


Scale−Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
97
91
85
40
60
80
100
120
140
160
Fitted values
lm(Amplitude ~ Genotype + RecordingCondition + Genotype:RecordingCondition)



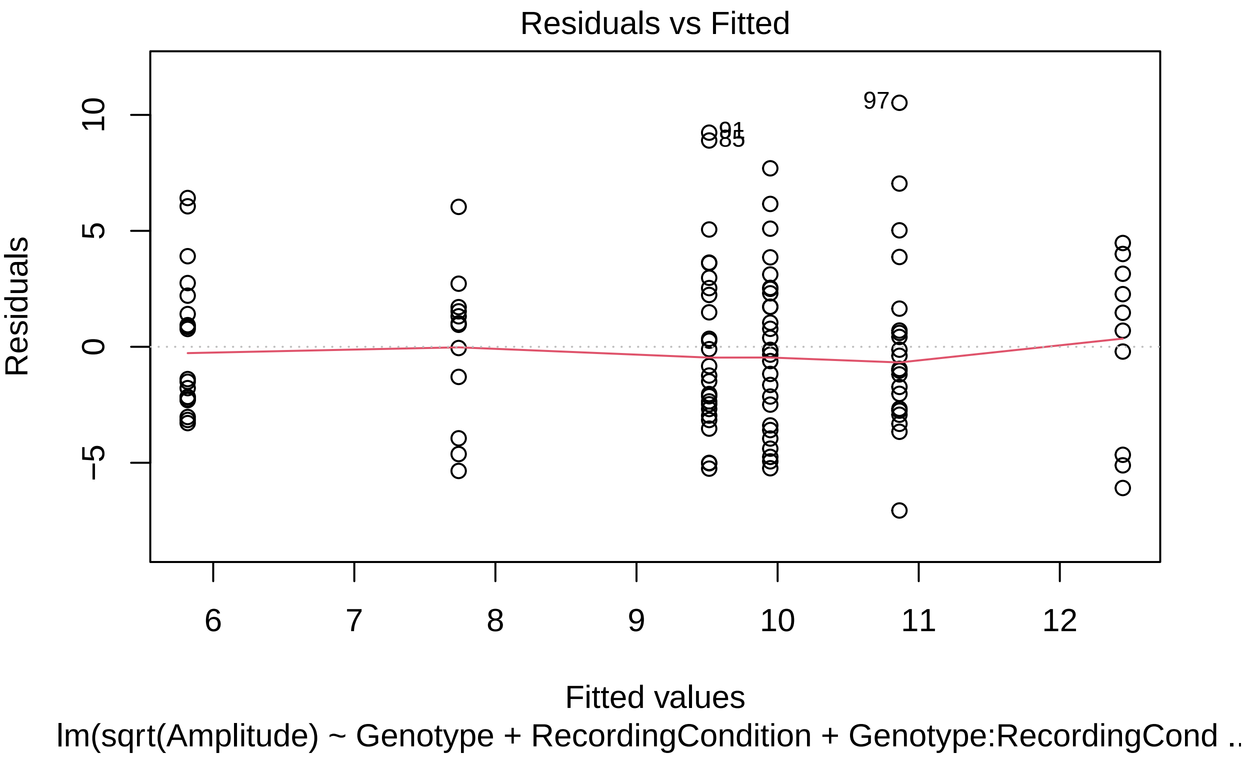
The first plot allows us to check the assumption of equal variance. As we anticipated, above, this plot suggests that we have a problem with equal variance: in general, we see that the vertical spread among the residuals increases as we move from left to right. The second plot (a qqplot) indicates that the data are not very normally distributed. The third plot also indicates the data violate the assumption of equal variance. This implies that we need to try to transform the data. Let's run the model again, but this time:

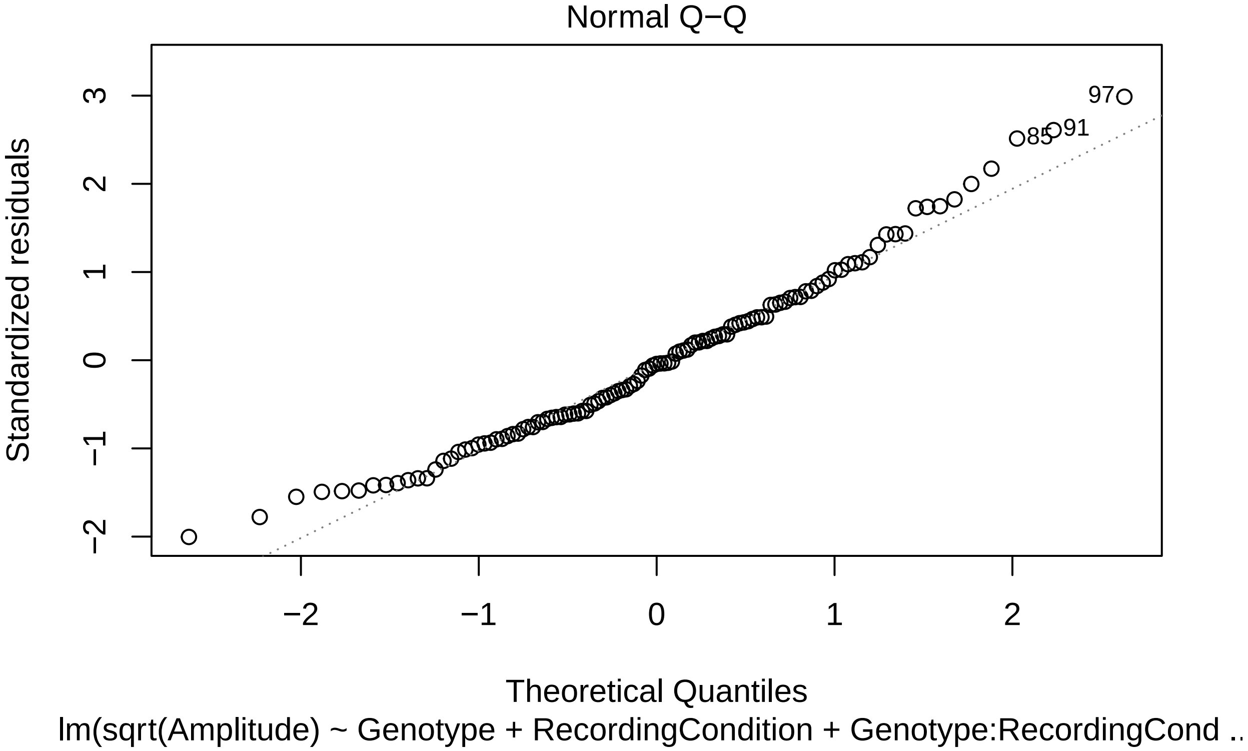
1) We'll sqrt-transform the dependent variable;
2) We'll add the 'contrasts' statement, to allows us to calculate p-values in a way that accounts for the data being unbalanced.

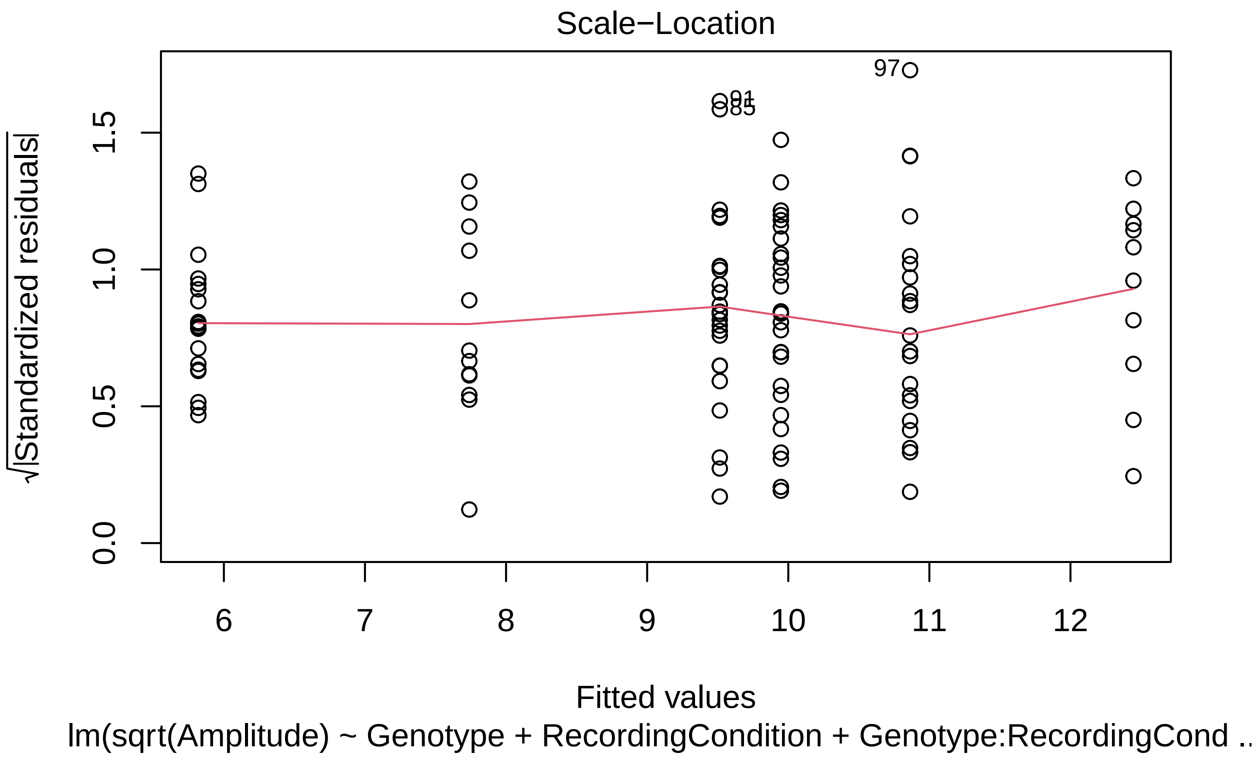
```
amp.sqrt.lm <- lm (sqrt(Amplitude) ~ Genotype + RecordingCondition +
                      Genotype:RecordingCondition, data = amp, contrasts =
                      list(Genotype = contr.sum, RecordingCondition = contr.sum))
```

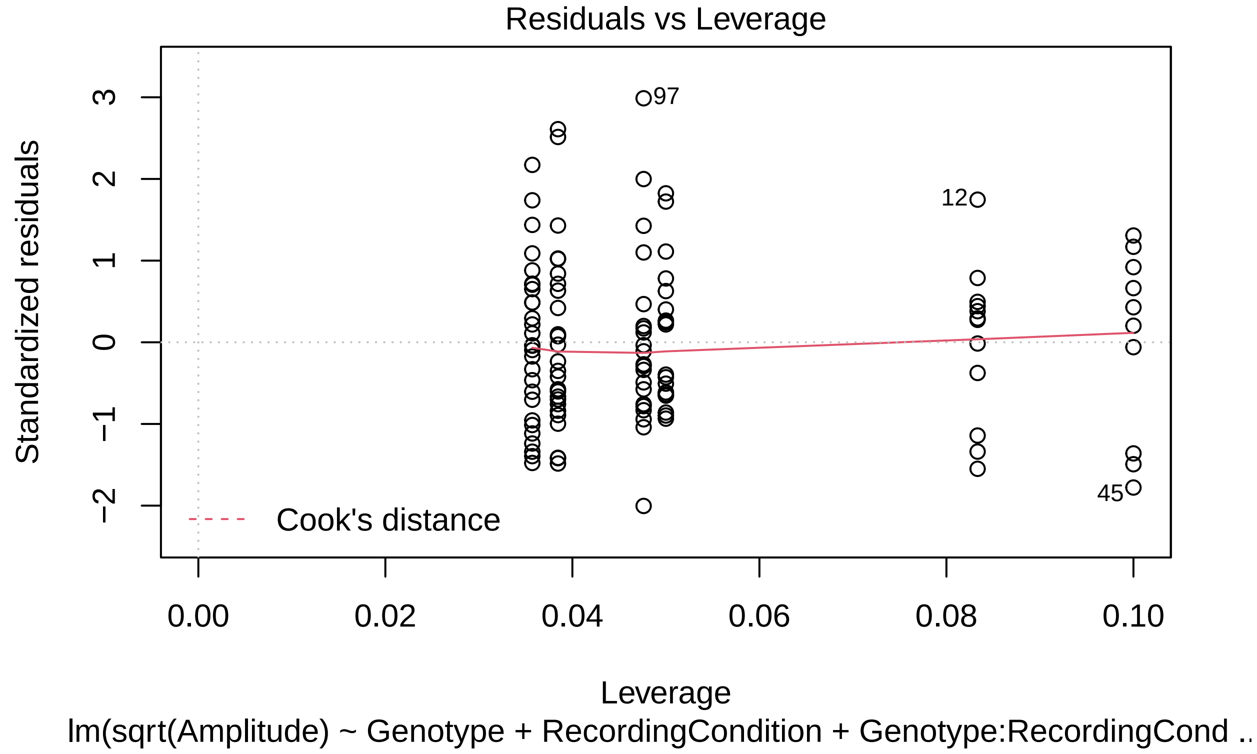
We'll plot the residuals again:

```
plot(amp.sqrt.lm)
```


Residuals vs Fitted
Residuals
Fitted values
lm(sqrt(Amplitude) ~ Genotype + RecordingCondition + Genotype:RecordingCond ..
97
91
85
10
5
0
−5
6
7
8
9
10
11
12


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(sqrt(Amplitude) ~ Genotype + RecordingCondition + Genotype:RecordingCond ..
97
85
91
3
2
1
0
−1
−2
−2
−1
0
1
2





These results indicate that the data now nicely meet the assumptions of equal variance and normally distributed residuals. (You should try log-transforming the data, and decide for yourself which transformation is better.)

Now we know that all the assumptions (random sampling, independence, equal variance and normality) are nicely met. We can look at our main results now. We'll use `Anova()` because we have unbalanced data:

```r
Anova(amp.sqrt.lm, type = 3)
```

```
## Anova Table (Type III tests)
##
## Response: sqrt(Amplitude)
##                              Sum Sq  Df  F value    Pr(>F)
## (Intercept)                  8935.5   1 686.0537 < 2.2e-16 ***
## Genotype                       43.6   1   3.3501   0.06988 .
## RecordingCondition            349.7   2  13.4263 5.996e-06 ***
## Genotype:RecordingCondition    12.6   2   0.4844   0.61735
## Residuals                    1445.7 111
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output suggests that we have little support for an interaction between `Genotype` and `RecordingCondition` (p = 0.61735, which is very large). However, we have strong support for the hypothesis that `Amplitude` differs among levels of `RecordingCondition` (p = 5.996e-06). We have ‘suggestive’ or ‘moderate’ evidence that `Genotype` affects `Amplitude` (p-value equals 0.06988).

Given that the plots, above, provided little suggestion for an interaction, and that the results from the model also suggest that there’s little evidence for an interaction, we can move forward by examining the effect sizes of each main effect, `Genotype` and `RecordingCondition`, separately; in each case we will average over the effects of the other Factor.

Let’s start with `Genotype`; we’ll work with data on the transformed scale:

```r
amp.sqrt.emmeans <- emmeans(amp.sqrt.lm, "Genotype")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```r
amp.sqrt.emmeans
```

```
##  Genotype emmean    SE  df lower.CL upper.CL
##  R43Q       8.73 0.444 111     7.85     9.61
##  WT        10.04 0.563 111     8.93    11.16
##
## Results are averaged over the levels of: RecordingCondition
## Results are given on the sqrt (not the response) scale.
## Confidence level used: 0.95
```

If we wished to report our results on the square-root scale, we could report the above means, SE’s and 95% CI’s.

Now let’s obtain effect sizes (again, on the transformed scale):

```r
amp.sqrt.pairs <- pairs(amp.sqrt.emmeans)
```

```
## Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
```

```r
amp.sqrt.pairs
```

```
##  contrast  estimate    SE  df t.ratio p.value
##  R43Q - WT    -1.31 0.717 111 -1.830  0.0699
##
## Results are averaged over the levels of: RecordingCondition
## Note: contrasts are still on the sqrt scale
```

Notice the output from `emmeans()`, indicating that *Note: Use ‘contrast(regrid(object), ...)’ to obtain contrasts of back-transformed estimates*. Therefore, it is possible to obtain back-transformed effect sizes for `sqrt` transformed data. However, this procedure is beyond this course. If you want to check it out for yourself, please see: https://cran.r-project.org/web/packages/emmeans/vignettes/transformations.html. (Back-transforming the effect sizes is easier with log-transformed data.)

Finally, we can obtain confidence intervals for the effect size:

```
confint(amp.sqrt.pairs)
```

```
##  contrast  estimate    SE  df lower.CL upper.CL
##  R43Q - WT    -1.31 0.717 111    -2.73    0.108
##
## Results are averaged over the levels of: RecordingCondition
## Note: contrasts are still on the sqrt scale
## Confidence level used: 0.95
```

We can, however, obtain the estimated marginal means for each level of `Genotype` for back-transformed data, which will make it easier to discuss our results:

```
emmeans(amp.sqrt.lm, "Genotype", type = "response")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
##  Genotype response    SE  df lower.CL upper.CL
##  R43Q         76.3  7.75 111     61.7     92.4
##  WT          100.9 11.31 111     79.7    124.6
##
## Results are averaged over the levels of: RecordingCondition
## Confidence level used: 0.95
## Intervals are back-transformed from the sqrt scale
```

These back-transformed (generalized) means indicate that the rate of FS cell activation in the R43Q genotype is about 3/4 that of the wild-type.

Now, let's perform a similar analysis for the effect if `RecordingCondition`:

```
amp.sqrt.emmeans.rc <- emmeans(amp.sqrt.lm, "RecordingCondition")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
amp.sqrt.emmeans.rc
```

```
##  RecordingCondition emmean    SE  df lower.CL upper.CL
##  a                    6.78 0.659 111     5.47     8.09
##  b                    9.73 0.491 111     8.76    10.71
##  c                   11.65 0.693 111    10.28    13.03
##
## Results are averaged over the levels of: Genotype
## Results are given on the sqrt (not the response) scale.
## Confidence level used: 0.95
```

Remember that these emmeans are on the transformed data scale.

Now, let's get the effect sizes:

```
amp.pairs.rc <- pairs(amp.sqrt.emmeans.rc)
```

```
## Note: Use 'contrast(regrid(object), ...)' to obtain contrasts of back-transformed estimates
```

```
amp.pairs.rc
```

```
##  contrast estimate    SE  df t.ratio p.value
##  a - b       -2.95 0.822 111 -3.591  0.0014
##  a - c       -4.88 0.956 111 -5.098  <.0001
##  b - c       -1.92 0.850 111 -2.263  0.0653
##
## Results are averaged over the levels of: Genotype
## Note: contrasts are still on the sqrt scale
## P value adjustment: tukey method for comparing a family of 3 estimates
```

e Note that the difference between conditions ‘a’ and ‘b’ or ‘c’ tend to be greater than between ‘b’ and ‘c’.

Let’s obtain confidence intervals for the effect sizes:

```
confint(amp.pairs.rc)
```

```
##  contrast estimate    SE  df lower.CL upper.CL
##  a - b       -2.95 0.822 111    -4.90  -0.9994
##  a - c       -4.88 0.956 111    -7.15  -2.6035
##  b - c       -1.92 0.850 111    -3.94   0.0953
##
## Results are averaged over the levels of: Genotype
## Note: contrasts are still on the sqrt scale
## Confidence level used: 0.95
## Conf-level adjustment: tukey method for comparing a family of 3 estimates
```

As above, let’s obtain the back-transformed (generalized) estimated marginal means for the three levels of `RecordingCondition`:

```
amp.response.emmeans.rc <- emmeans(amp.sqrt.lm, "RecordingCondition",
                                   type = "response")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
amp.response.emmeans.rc
```

```
##  RecordingCondition response    SE  df lower.CL upper.CL
##  a                      46.0  8.93 111     30.0     65.4
##  b                      94.7  9.57 111     76.7    114.6
##  c                     135.8 16.16 111    105.7    169.7
##
## Results are averaged over the levels of: Genotype
## Confidence level used: 0.95
## Intervals are back-transformed from the sqrt scale
```

Part (d)

With these results in hand, we can report them in something like the following…

A 2-Factor GLM revealed little evidence for an interaction between `Genotype` and `RecordingConditions` ($F_{2,111} = 0.484$, p = 0.617). Therefore, we focus our analysis on interpreting the main effects, `Genotype` and `RecordingCondition`.

The model revealed only weak evidence that `Genotype` affected `Amplitude` of FS cell activation ($F_{1,111} =$ 3.3501, p = 0.06988). The estimated marginal mean `Amplitude` for the `Genotype` R43Q (mean (SE) = 76.3 (7.75)) was approximately three-quarters that of the Wild-type (100.9 (11.31)); note that these means are *generalized* means. Therefore, the Wild-type tends to have greater FS cell activation, although evidence for a difference with R43Q is weak. On the square-root transformed scale, the estimated marginal means (SE) for `Genotype` R43Q and Wild-type equal 8.73 (0.444) and 10.04 (0.563), respectively. Therefore the difference between the means (on the transformed scale) equals 1.31 (0.717), with 95% CI’s that equal -2.73 to 0.108. *NOTE to students: I would like to say more about the range of the effect sizes, but, like many of you (I suspect), I find it difficult to wrap my head around differences of means of square-root transformed data. This* *is why I started with a general statement using the generalized means, to provide a sense of the size of the estimated difference between genotypes.*

In contrast, the model provided strong evidence for differences in `Amplitude` among levels of `RecordingCondition` ($F_{2,111} = 13.4263$, $p = 5.996 * 10^{-06}$). On the back-transformed (response scale), the generalized mean (SE) `Amplitude` of conditions 'a', 'b' and 'c' equal 46.0 (8.93), 94.7 (9.57) and 135.8 (16.16), respectively. Therefore, the generalized mean of `Amplitude` in `RecordingCondition` 'a' is approximately half that of `RecordingCondition` 'b', and a third that of `RecordingCondition` 'c', indicating that `RecordingCondition` strongly influences `Amplitude`. On the transformed scale, these means (SE's) equal 6.78 (0.659), 9.73 (0.491) and 11.65 (0.693), respectively. Still on the transformed scale, the difference between the mean of 'a' and 'b' equals 2.95 (0.822) (t-ratio = 3.591, df = 111, p = 0.0014), with 95% CI's of -4.90 to -0.9994. The difference between 'a' and 'c' is even greater (4.88 (0.956); t-ratio = 5.098, d.f. = 111, p <0.0001), with 95% CI's of -7.15 and -2.6035. Finally, evidence for a difference between the mean `Amplitude` of 'b' and 'c' is modest, and smaller in magnitude: 1.92 (0.850) (t-ratio = 2.263, d.f. = 111, p = 0.0653), where 95% CI's for this difference are -3.94 and 0.0953.

Question 3

Part (a)

Let's import the data:

```r
grow <- read.table("TwoGenesPilotData.csv", header = TRUE, sep = ",")
```

Let's look at the whole dataframe, as there are relatively few data:

```r
grow
```

```
##    pmDay4Growth Gene1 Gene2
## 1          20.1   Yes   Yes
## 2          22.5   Yes   Yes
## 3          21.5   Yes   Yes
## 4          23.9   Yes   Yes
## 5          23.9   Yes   Yes
## 6          26.4   Yes   Yes
## 7          22.8   Yes   Yes
## 8          24.6   Yes   Yes
## 9          24.1   Yes   Yes
## 10         23.3   Yes   Yes
## 11         21.4   Yes    No
## 12         21.8   Yes    No
## 13         20.2   Yes    No
## 14         22.7   Yes    No
## 15         23.5   Yes    No
## 16         27.1   Yes    No
## 17         24.1   Yes    No
## 18         21.2   Yes    No
## 19         22.7   Yes    No
## 20         21.2   Yes    No
## 21         22.9   Yes    No
## 22         22.6   Yes    No
## 23         22.3    No   Yes
## 24         22.9    No   Yes
## 25         24.7    No   Yes
## 26         25.7    No   Yes
## 27         21.4    No   Yes
## 28         22.7    No   Yes
## 29          21.6    No   Yes
## 30          23.0    No   Yes
## 31          22.2    No    No
## 32          22.0    No    No
## 33          23.7    No    No
## 34          21.7    No    No
## 35          26.0    No    No
## 36          21.5    No    No
## 37          21.7    No    No
## 38          28.4    No    No
## 39          20.3    No    No
## 40          22.1    No    No
## 41          24.4    No    No
## 42          23.1    No    No
```

We see three columns: `pmDay4Growth`, `Gene1` (coded as Yes or No) and `Gene2` (also coded as Yes or No). The hypothesis is that the presence or absense of the two genes would affect `pmDay4Growth`; therefore `pmDay4Growth` is the *dependent* variable.

Part (b)

Let's take a closer look at the experimental design. As above, we'll use the `summaryBy()` function to assess the extent of replication in the experiment:

```
summaryBy(pmDay4Growth ~ Gene1 + Gene2, data = grow, FUN = c(length, mean))
```

```
##   Gene1 Gene2 pmDay4Growth.length pmDay4Growth.mean
## 1    No    No                  12          23.09167
## 2    No   Yes                   8          23.03750
## 3   Yes    No                  12          22.61667
## 4   Yes   Yes                  10          23.31000
```

The output reveals:

1) the experiment is replication at all treatment combination (minimum sample size for each treatment combination is 8); therefore, we have sufficient replication to model an interaction between the two genes.

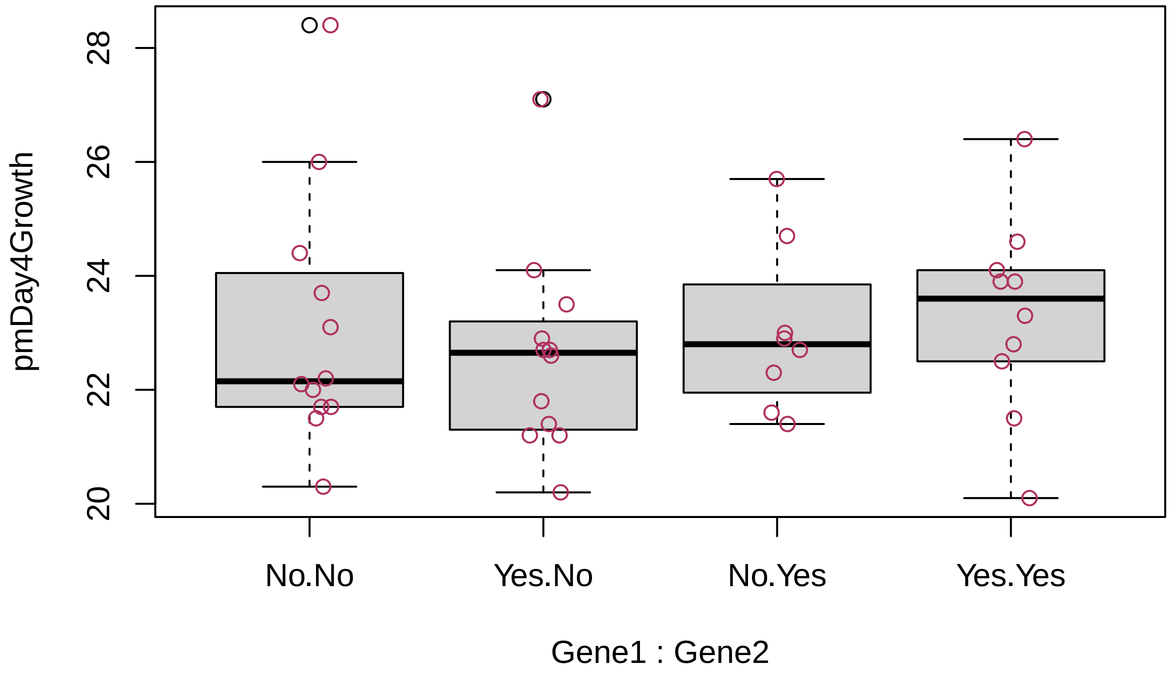
2) the data are *unbalanced*; we will need to account for this when calculating p-values.

We can also see from the dataframe (`grow`), above, that the experiment includes all possible combinations of treatment levels; this is consistent wil modeling the data as a 2-Factor GLM. Lacking any information to the contrary, we will assume that the data are independent (also allowing us to model the data as a 2-Factor GLM) and randomly sampled. Given these observations, we can model the data as a 2-Factor GLM that includes an interaction between `Gene1` and `Gene2`.

Part (c)

Let's plot the data. Given that `pmDay4Growth` is the dependent variable, we can plot the data as:

```
boxplot(pmDay4Growth ~ Gene1*Gene2, data = grow)
stripchart(pmDay4Growth ~ Gene1*Gene2, data = grow, add = TRUE, vertical = TRUE,
           method = "jitter", pch = 21, col = "maroon")
```



What do we learn from this plot?

1) Variance is likely equal among treatment combinations (notice similar breadth of the boxplots)
2) The residuals are likely normally distributed (notice that the boxplots are fairly symmetrical).
3) There seems to be little evidence for an effect of either `Gene1` or `Gene2`, or an interaction between them, on `pmDay4Growth`.
4) We expect the value of the `(Intercept)` (which will be combination 'No.No') will be around 23, and other values will be similar.

We'll not with making an interaction plot this time, because the lack of any treatment effects seems pretty clear from the boxplots.

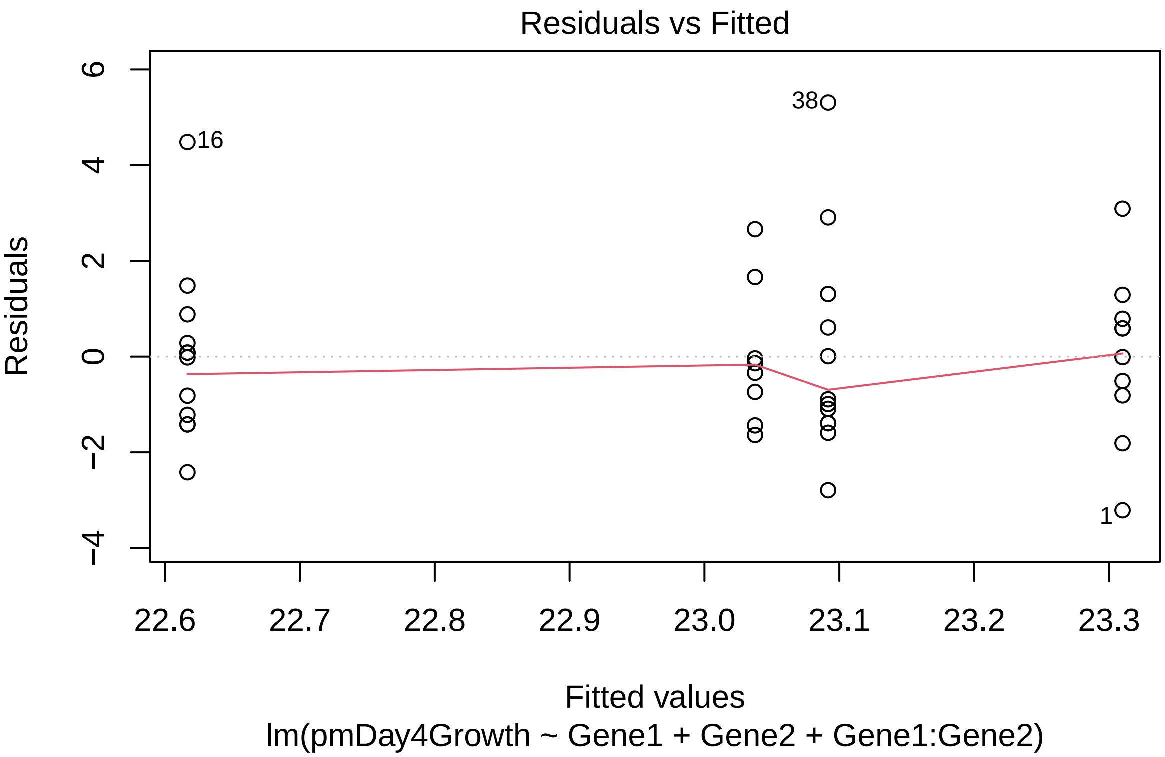
Part (d)

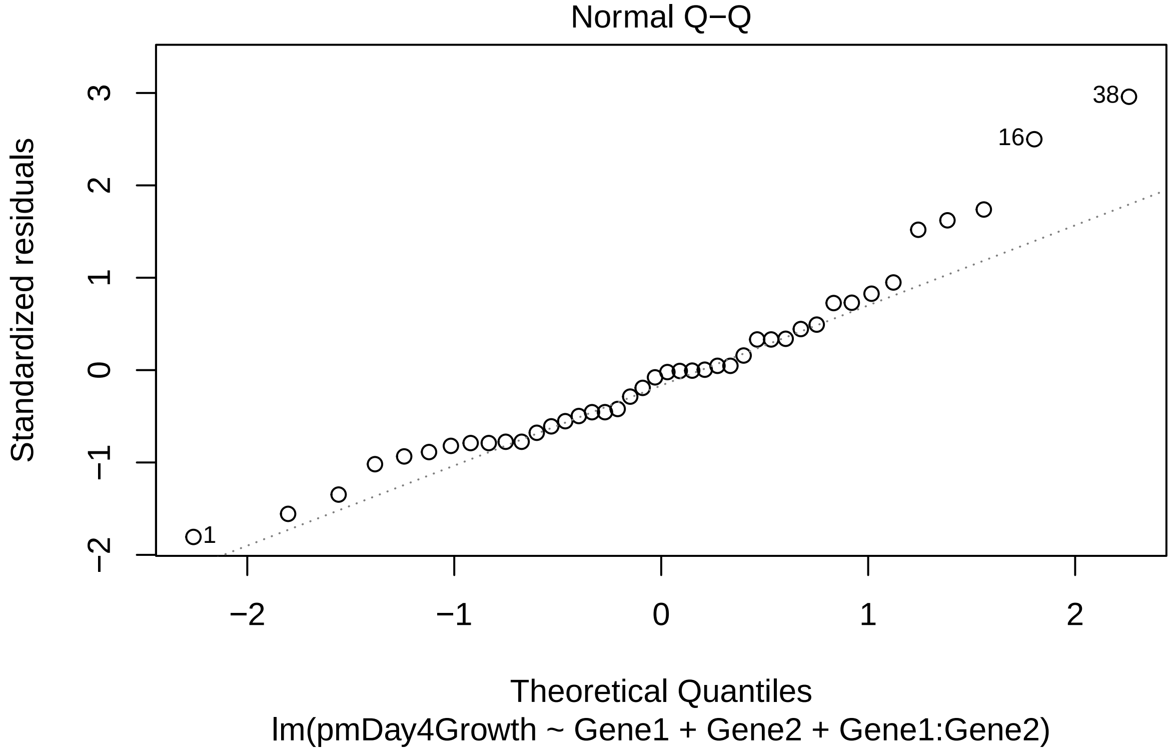
Let's model the data:

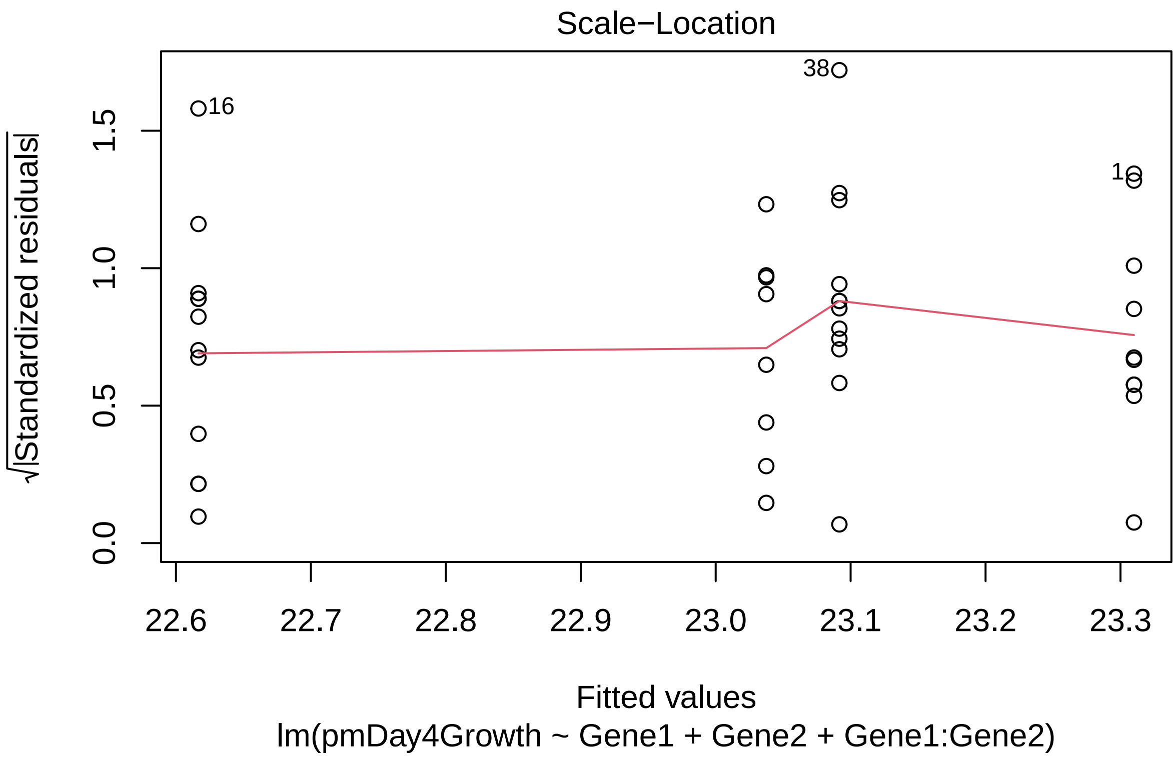
```
grow.lm <- lm(pmDay4Growth ~ Gene1 + Gene2 + Gene1:Gene2, data = grow,
              contrasts = list(Gene1 = contr.sum, Gene2 = contr.sum))
```

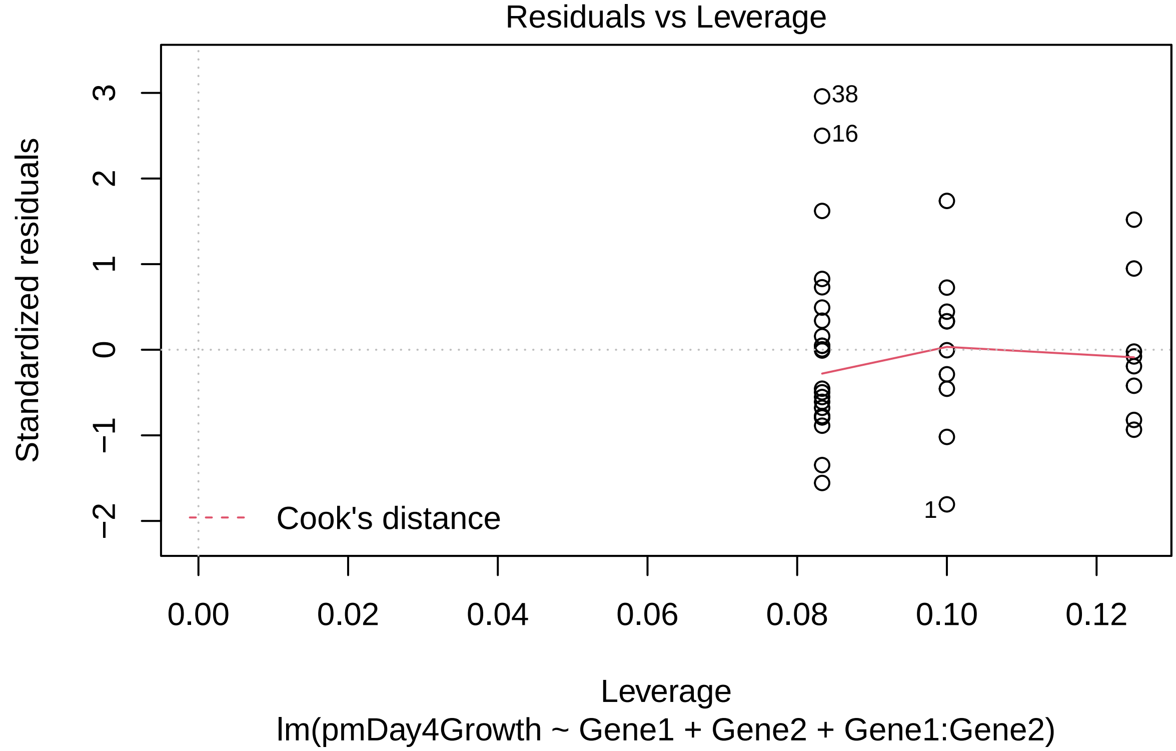
(Does it strike you that we seem to analyze a lot of datasets involving growth? That's biology for you!) Let's plot the residuals:

```
plot(grow.lm)
```


Residuals vs Fitted
Residuals
6
4
2
0
−2
−4
16
38
1
22.6
22.7
22.8
22.9
23.0
23.1
23.2
23.3
Fitted values
lm(pmDay4Growth ~ Gene1 + Gene2 + Gene1:Gene2)


Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
38
16
1
−2
−1
0
1
2
Theoretical Quantiles
lm(pmDay4Growth ~ Gene1 + Gene2 + Gene1:Gene2)





The first plot indicates (as we guessed from the boxplot) that the data nicely meet the assumption of equal variance. The second plot suggests that the data are relatively normally distributed. The third plot shows a somewhat ragged red line, but overall this looks OK to me; my impression is that this third plot (like the first one) suggests the data meet the assumption of equal variance. As we'd said above, we'll assume that the data are randomly sampled and independent. Therefore, we're satisfied that the data meet the assumptions of the model.

Now that we're satisfied that the assumptions are met, let's look at the results in terms of p-values and effect sizes. Remember that we need to account for unbalanced data:

```
Anova(grow.lm, type = 3)
```

```
## Anova Table (Type III tests)
##
## Response: pmDay4Growth
##              Sum Sq Df   F value Pr(>F)
## (Intercept) 21636.5  1 6164.5275 <2e-16 ***
## Gene1           0.1  1    0.0298 0.8638
## Gene2           1.0  1    0.2972 0.5888
## Gene1:Gene2     1.4  1    0.4065 0.5276
## Residuals     133.4 38
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we guessed from the boxplot, the p-values suggest that we have very little evidence that either `Gene1` or `Gene2` (via main effects or through an interaction) affect `pmDay4Growth`.

Regardless, it can still be useful to calculate effect sizes. As we have no evidence for any interaction, we'll obtain effect sizes for each gene, while averaging over the effects of the other gene (i.e., we'll calculate the average effect of each gene). We'll start with `Gene1`, averaging over `Gene2`:

```
grow.emmeans.1 <- emmeans(grow.lm, "Gene1")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
grow.emmeans.1
```

```
##  Gene1 emmean    SE df lower.CL upper.CL
##  No      23.1 0.428 38     22.2     23.9
##  Yes     23.0 0.401 38     22.2     23.8
##
## Results are averaged over the levels of: Gene2
## Confidence level used: 0.95
```

These results reveal that mean growth is nearly identical between the two genotypes of `Gene1`. This is much more informative than simply reporting a p-value: a p-value might be large due to a lack of effect, or due to a small sample size, or both. These means clarify matters.

Let's obtain the effect size and SE for `Gene1`

```
grow.pairs.1 <- pairs(grow.emmeans.1)
grow.pairs.1
```

```
##  contrast estimate    SE df t.ratio p.value
##  No - Yes    0.101 0.586 38 0.173   0.8638
##
## Results are averaged over the levels of: Gene2
```

And, let's obtain a confidence interval for this effect size:

```
confint(grow.pairs.1)
```

```
##  contrast estimate    SE df lower.CL upper.CL
##  No - Yes    0.101 0.586 38    -1.09     1.29
##
## Results are averaged over the levels of: Gene2
## Confidence level used: 0.95
```

This confidence interval (-1.09 to 1.29) provides more insight, yet. Now learned above (from the `emmeans()` and `pairs()` functions) that the mean between-genotypes difference in growth for `Gene1` is small (0.101); the confidence intervals further suggest that the **range** of likely effects of `Gene1` is relatively small. It is reasonable to conclude that the absence of `Gene1` causes a change in growth that ranges from about a 4.7% decrease (1.09 / 23.0) to an increase of 5.6% (1.29 / 23.0) in growth.

Let's repeat this exercise for `Gene2`:

```
grow.emmeans.2 <- emmeans(grow.lm, "Gene2")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
grow.emmeans.2
```

```
##  Gene2 emmean    SE df lower.CL upper.CL
##  No      22.9 0.382 38     22.1     23.6
##  Yes     23.2 0.444 38     22.3     24.1
##
## Results are averaged over the levels of: Gene1
## Confidence level used: 0.95
```

The results for `Gene2` are similar to thise for `Gene1`: we see that the difference between genotypes is only about 0.3 units of growth (compared to a mean of of about 23). Now let's estimate this effect size and its SE:

```
grow.pairs.2 <- pairs(grow.emmeans.2)
grow.pairs.2
```

```
##  contrast estimate    SE df t.ratio p.value
##  No - Yes    -0.32 0.586 38 -0.545  0.5888
##
## Results are averaged over the levels of: Gene1
```

As with `Gene1`, we can report these results (effect size and its SE).

Now let's obtain 95% CI's for this effect size:

```
confint(grow.pairs.2)
```

```
##  contrast estimate    SE df lower.CL upper.CL
##  No - Yes    -0.32 0.586 38    -1.51    0.867
##
## Results are averaged over the levels of: Gene1
## Confidence level used: 0.95
```

The range of possible effect sizes for `Gene2` is comparable to that observed for `Gene1`. The 95% CI's suggest that the loss of `Gene2` results in a change in growth ranging from a loss of about 6.5% (1.51 / 23.2) to an increase of about 3.7% (0.867 / 23.2).

How would we report these findings? First, we'd present a nice plot; we could use a plot like the boxplot, above, but remember to:

1) label the axes correctly;
2) select a suitable range for the y-axis, starting at zero
3) add an appropriate figure caption.

I leave that to you!

Next, we might present our findings as follows:

We used a 2-Factor GLM (function `lm()` in **R**) to quantify the effects of `Gene1` and `Gene2`, and their interaction on `pmDay4Growth`; the analysis accounted for unbalanced data. This analysis revealed little evidence for an interaction between the two genes on ’pmDay4Growth(Gene1:Gene2‘ interaction; $F_{1,38} = 0.4065$, p = 0.5276). Therefore our analysis focused on quantifying effect of each gene on growth.

Analyses suggest little evidence that `Gene1` affects growth ($F_{1,38} = 0.0298$, p = 0.8638). Specifically, the mean (SE) growth of individuals with and without `Gene1` equaled 23.0 (0.401) and 23.1 (0.428), respectively. Hence, the mean effect size (and SE) equaled 0.101 (0.586), with 95% CI's that range from -1.09 to 1.29. This 95% CI suggesting that losing `Gene1` results in a change of growth ranging from a 4.7% decrease to a 5.6% increase in growth.

Similarly, we identified little evidence that `Gene2` affects growth ($F_{1,38} = 0.2972$ p = 0.5888). The mean (SE) growth of individuals with and without `Gene1` equaled 23.2 (0.444) and 22.9 (0.382), respectively. This results in a mean (SE) effect size of 0.32 (0.586), with 95% CI's of -1.51 and 0.867. These 95% CI's indicate that losing `Gene2` results in a change in growth ranging from a loss of about 6.5% to an increase of about 3.7%.

Question 4

Part (a)

Let's import the data:

```r
glucose <- read.table("BloodGlucoseLabData.csv", header = TRUE, sep = ',')
```

Let's get a summary of the data:

```r
summary(glucose)
```

```
##   treatment              time       subject            glucose      
##  Length:190         Min.   :  0   Length:190         Min.   : 4.100  
##  Class :character   1st Qu.: 30   Class :character   1st Qu.: 5.600  
##  Mode  :character   Median : 60   Mode  :character   Median : 6.300  
##                     Mean   : 60                      Mean   : 6.652  
##                     3rd Qu.: 90                      3rd Qu.: 7.400  
##                     Max.   :120                      Max.   :13.400  
```

And, let's look at the top of the dataframe:

```r
head(glucose)
```

```
##   treatment time subject glucose
## 1       HGL    0      H1     7.0
## 2       HGL    0      H2     5.7
## 3       HGL    0      H3     6.2
## 4       HGL    0      H4     6.3
## 5       HGL    0      H5     5.8
## 6       HGL    0      H6     5.6
```

We see 4 columns: `treatment`, `time`, `subject`, and `glucose`. The first 3 columns contain data regarding Factors, and the last is a continuous variable that measures glucose levels in blood (`glucose`). The hypothesis is that `glucose` will change over `time` and depend on `treatment`; therefore `glucose` is the dependent variable.

Part (b)

This is where our answer to this question ends. The goal of this experiment was to determine whether the trajectory of glucose levels differed among `treatment` levels over `time`; this would suggest at least a 2-Factor analysis. However, recall that the question informed us that individual subjects were measured repeatedly over time. This means that the data are not independent over time, and we cannot proceed with a simple 2-factor GLM to address our goal. We need more advanced techniques to analyze these data.

It is important to critically assess every experiment to ensure that its characteristics match any analysis we wish to perform. If we'd tried to run a 2-Factor GLM with the glucose data we would have obtained untrustworthy results.