Project 2

Bailey Perry

March 11, 2016

Problem 1 –

Introduction

Cardiac pacemakers contain electrical connections that are platinum pins soldered onto a substrate. The question of interest is whether different operators produce solder joints with the same strength. Twelve substrates are randomly assigned to four operators. Each operator solders four pins on each substrate, and then these solder joints are assessed by measuring the shear strength of the pins. Data from T. Kerkow.

Import Dataset and Summarize Data

```
pacemakers <- read.table("http://www.stat.umn.edu/~gary/book/fcdae.data/pr3.1", header = T)
pacemakers
```

```
##    operator substrate strength
## 1         1         1     5.60
## 2         1         1     6.80
## 3         1         1     8.32
## 4         1         1     8.70
## 5         1         2     7.64
## 6         1         2     7.44
## 7         1         2     7.48
## 8         1         2     7.80
## 9         1         3     7.72
## 10        1         3     8.40
## 11        1         3     6.98
## 12        1         3     8.00
## 13        2         1     5.04
## 14        2         1     7.38
## 15        2         1     5.56
## 16        2         1     6.96
## 17        2         2     8.30
## 18        2         2     6.86
## 19        2         2     5.62
## 20        2         2     7.22
## 21        2         3     5.72
## 22        2         3     6.40
## 23        2         3     7.54
## 24        2         3     7.50
## 25        3         1     8.36
## 26        3         1     7.04
## 27        3         1     6.92
## 28        3         1     8.18
## 29        3         2     6.20
## 30        3         2     6.10
## 31        3         2     2.75
## 32        3         2     8.14
## 33        3         3     9.00
## 34        3         3     8.64
## 35        3         3     6.60
## 36        3         3     8.18
## 37        4         1     8.30
## 38        4         1     8.54
## 39        4         1     7.68
## 40        4         1     8.92
## 41        4         2     8.46
## 42        4         2     7.38
## 43        4         2     8.08
## 44        4         2     8.12
## 45        4         3     8.68
## 46        4         3     8.24
## 47        4         3     8.09
## 48        4         3     8.06
```

Note that the operators are given numbers from 1-4 that identifies which operator performed the soldering process on the substrate. The substrate values identify which substrate the pin was place on, that is why the data has four values of 1,2, and 3 respectively in the substrate column. Each operator places 4 pins on 3 different substrates.

Part A:

Step 1: Data Pre-Processing

```
lapply(pacemakers, class)
```

```
## $operator
## [1] "integer"
##
## $substrate
## [1] "integer"
##
## $strength
## [1] "numeric"
```

Since operator is not an integer we want to evaluate, it is necessary to change that variable to be recognized as a factor in R.

```
pacemakers_adj <- with(pacemakers, data.frame(operator=as.factor(operator),
                                  substrate = substrate, strength = strength))
lapply(pacemakers_adj, class)
```

```
## $operator
## [1] "factor"
```

```
##
## $substrate
## [1] "integer"
##
## $strength
## [1] "numeric"
```

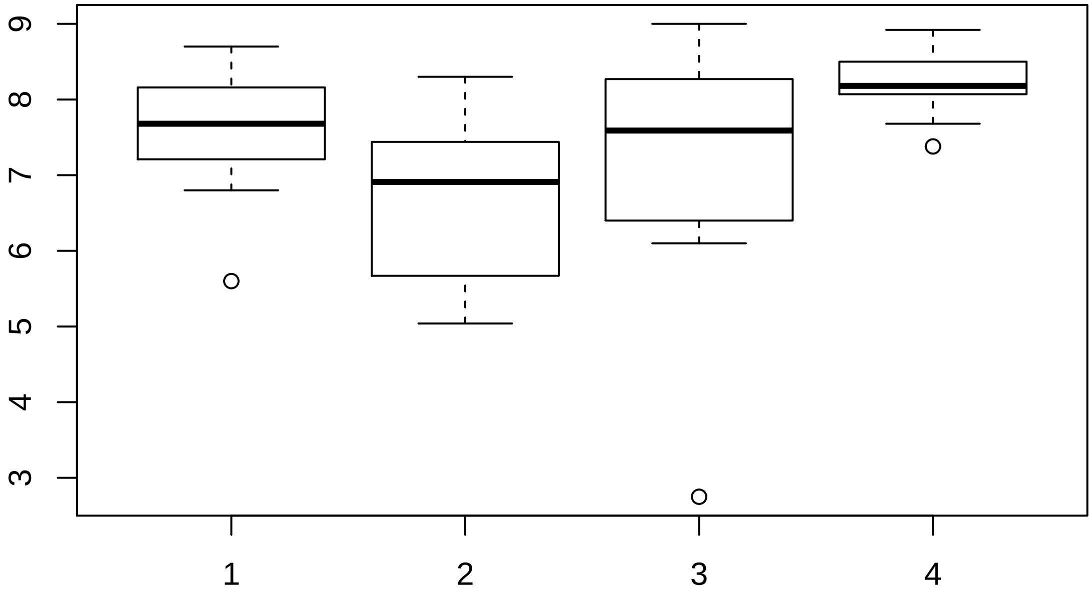
Step 2: Exploratory Data Analysis

```
summary(pacemakers_adj)
```

```
##  operator   substrate     strength
##  1:12     Min.   :1   Min.   :2.750
##  2:12     1st Qu.:1   1st Qu.:6.905
##  3:12     Median :2   Median :7.660
##  4:12     Mean   :2   Mean   :7.409
##           3rd Qu.:3   3rd Qu.:8.255
##           Max.   :3   Max.   :9.000
```

The summary for the categorical variable, operators, returned 12 for each operator because there were 4 pins associated with 3 substrates and each were done individually. This meant that each operator had 12 different pieces of data tied to their operator number.

```
boxplot(strength~operator, data=pacemakers_adj)
```



In the boxplots, we can see that the interquartile ranges (IQR) of operators 1,2, and 3 overlap, as well as operators 1,2, and 4 overlap. We note though that the IQR of operator 2 does not overlap with operator 4.

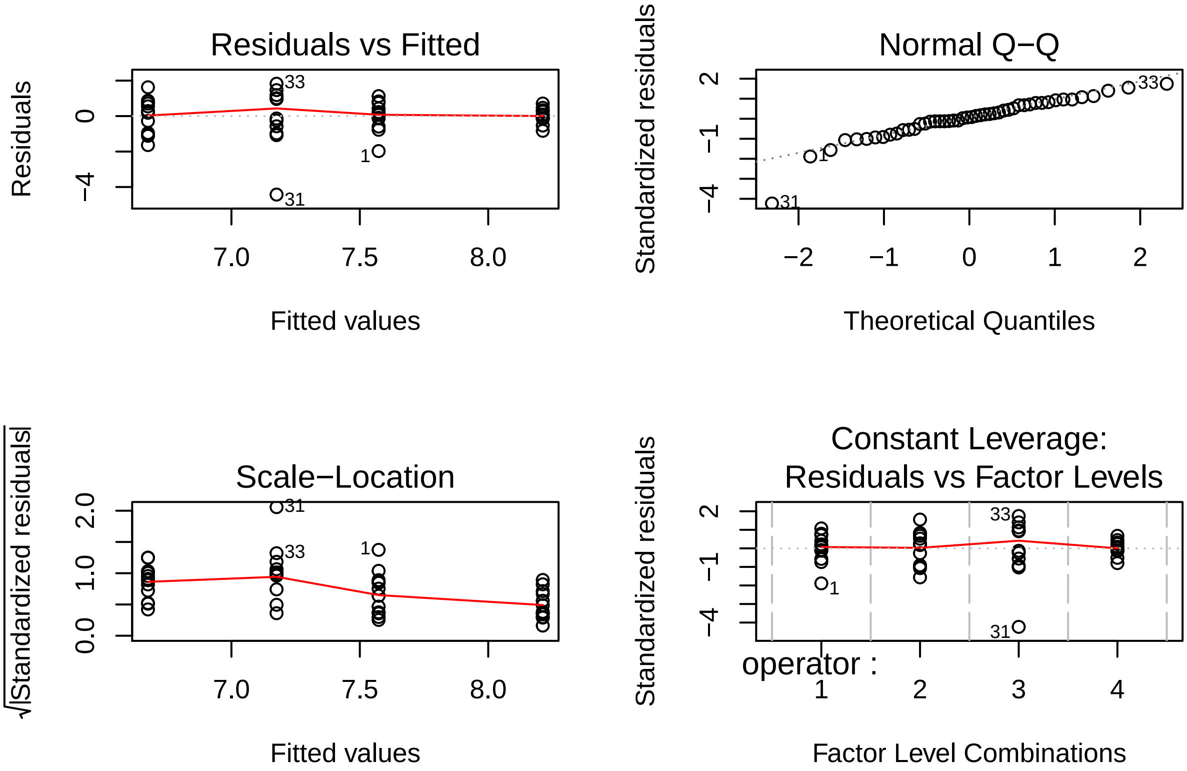
```r
lapply(with(pacemakers_adj, split(strength, operator)), mean)
```

```
## $`1`
## [1] 7.573333
##
## $`2`
## [1] 6.675
##
## $`3`
## [1] 7.175833
##
## $`4`
## [1] 8.2125
```

This output gives us the values of the sample mean responses for each of the different operators.

Step 3: Model Fitting and Diagnostics

```r
mod1 <- lm(strength~ operator, data = pacemakers_adj)
par(mfrow = c(2,2))
plot(mod1)
```



The assumptions do not seem to have any extreme violations, the data may contain an outlier, but for the most part they seem fit. The Residuals v Fitted plot is almost a straight horizontal line with most of the data evenly spread. The data from the second operator is something to be cautious of. The Normal QQ Plot also is relatively straight and fitting for the assumptions.

Step 4: Inference

Is there any evidence that the operators produce different mean shear strengths?

```
anova(mod1)
```

```
## Analysis of Variance Table
##
## Response: strength
##           Df Sum Sq Mean Sq F value Pr(>F)
## operator   3 15.189  5.0630  4.2427 0.0102 *
## Residuals 44 52.506  1.1933
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Part A Results: Based on the ANOVA output, we see that the p-value for operator is signficant. This p-value is less than 0.05, which makes it significant and tells us that the mean shear strengths for the operators are different. For this analysis, the null hypothesis would be that the mean shear strengths are all the same, and the alternative hypothesis would be that the null hypothesis is not true.

Part B:

Problem Set-Up: Workers 1 and 2 were experienced, whereas workers 3 and 4 were novices. Find a contrast to compare the experienced and novice workers and test the null hypothesis that experienced and novice works produce the same average shear strength.

To perform the linear contrast it is necessary to load two specific R packages.

```
require(Stat5303libs)
```

```
## Loading required package: Stat5303libs

## Loading required package: mvtnorm

## Loading required package: car

## Loading required package: perm

## Loading required package: effects

##
## Attaching package: 'effects'
## The following object is masked from 'package:car':
##
##     Prestige
```

```
## Loading required package: tseries
```

```
## Loading required package: FrF2
```

```
## Loading required package: DoE.base
```

```
## Loading required package: grid
```

```
## Loading required package: conf.design
```

```
##
## Attaching package: 'DoE.base'
```

```
## The following objects are masked from 'package:stats':
##
##     aov, lm
```

```
## The following object is masked from 'package:graphics':
##
##     plot.design
```

```
## The following object is masked from 'package:base':
##
##     lengths
```

```
## Loading required package: RLRsim
```

```
## Loading required package: rsm
```

```
## Loading required package: pbkrtest
```

```
## Loading required package: lme4
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'lme4'
```

```
## The following object is masked from 'package:conf.design':
##
##     factorize
```

```
## Loading packages for Statistics 5303
```

```r
require(cfcdae)
```

```
## Loading required package: cfcdae

## Warning: replacing previous import by 'lme4::VarCorr' when loading 'cfcdae'

## Warning: replacing previous import by 'lme4::ranef' when loading 'cfcdae'

## Warning: replacing previous import by 'lme4::fixef' when loading 'cfcdae'

## Warning: replacing previous import by 'lme4::lmList' when loading 'cfcdae'

##
## Attaching package: 'cfcdae'

## The following object is masked from 'package:stats':
##
##     power.anova.test
```

To know if the average mean shear strength for operators 1 and 2 is the same as average mean shear strength for operators 3 and 4, we perform a specific linear contrast. The null hypothesis for this would be: (mu1+mu2)/2 = (mu3+mu4)/2, and the alternative is that these averages are not equal.

To test this, we do the following:

```r
linear.contrast(mod1, operator, c(1/2, 1/2, -1/2, -1/2))
```

```
##   estimates        se   t-value    p-value  lower-ci   upper-ci
## 1     -0.57 0.3153476 -1.807529 0.07751935 -1.205541 0.06554129
```

Part B Results: Based on this output, the p-value is greater than 0.05 and the confidence interval contains zero, so we do not have enough evidence to reject the null hypothesis. This means we do not have enough evidence to conclude that the average mean shear strengths are different for the experienced versus the novice workers.

Part C:

Problem Set-Up: Test the null hypothesis that all pairs of workers produce solder joints with the same average strength against the alternative that some workers produce different average strengths.

This problem is similar to above, but this time we want to compare all pairs of operators. This is done as follows:

```r
linear.contrast(mod1, operator, allpairs = TRUE, confidence = 0.95)
```

```
##          estimates        se    t-value     p-value      lower-ci   upper-ci
## 1 - 2  0.8983333 0.4459688  2.0143411 0.050111802 -0.0004577793  1.7971244
## 1 - 3  0.3975000 0.4459688  0.8913179 0.377606988 -0.5012911127  1.2962911
## 1 - 4 -0.6391667 0.4459688 -1.4332093 0.158865845 -1.5379577793  0.2596244
## 2 - 3 -0.5008333 0.4459688 -1.1230232 0.267518140 -1.3996244460  0.3979578
## 2 - 4 -1.5375000 0.4459688 -3.4475504 0.001256974 -2.4362911127 -0.6387089
## 3 - 4 -1.0366667 0.4459688 -2.3245272 0.024775594 -1.9354577793 -0.1378756
```

Part C Results: From this test, we see that the pairs of operators 2 and 4, as well as operators 3 and 4 have different average mean shear strengths. Both of these pairs had p-values below 0.05, and had zero in the confidence interval. Since the test for 2-4 and for 3-4 both produced confidence intervals of purely negative values, it is safe to state that operator 4 produces stronger shears than operators 2 and 3. To be more conservative in our analysis, we also include a Tukey HSD pairwise comparison. Significant differences of this test are marked with a star.

```
pairwise(mod1, operator, confidence=0.95)
```

```
##
## Pairwise comparisons ( hsd ) of operator
##            estimate signif diff      lower      upper
##   1 - 2  0.8983333    1.190739 -0.2924060  2.0890726
##   1 - 3  0.3975000    1.190739 -0.7932393  1.5882393
##   1 - 4 -0.6391667    1.190739 -1.8299060  0.5515726
##   2 - 3 -0.5008333    1.190739 -1.6915726  0.6899060
## * 2 - 4 -1.5375000    1.190739 -2.7282393 -0.3467607
##   3 - 4 -1.0366667    1.190739 -2.2274060  0.1540726
```

Part C Results Updated: From this output, we determine that the only significant differences are found in the pair of operators 2 and 4. It still agrees with the test above in that operator 4 produces stronger shears than operator 2. It is noted that Tukey tests are often a more conservative analysis that tends to be better, and closer to reality.

Problem 2 –

Introduction

Pine oleoresin is obtained by tapping the trunks of pine trees. Tapping is done by cutting a hole in the bark and collecting the resin that oozes out. This experiment compares four shapes for the holes and the efficacy of acid treating the holes. Twenty-four pine trees are selected at random from a plantation, and the 24 trees are assigned at random to the eight combinations of hole shape (circular, diagonal slash, check, rectangular) and acid treatment (yes or no). The response is total grams of resin collected from the holes. Analyze these data to determine how the treatments affect resin yield.

Import Dataset and Summarize Data

```
resin <- read.table("http://www.stat.umn.edu/~gary/book/fcdae.data/pr8.5", header = T)
resin
```

```
##   shape trt  y
## 1     1   1  9
## 2     2   1 43
## 3     3   1 60
## 4     4   1 77
## 5     1   1 13
## 6     2   1 48
## 7     3   1 65
## 8      4   1  70
## 9      1   1  12
## 10     2   1  57
## 11     3   1  70
## 12     4   1  91
## 13     1   2  15
## 14     2   2  66
## 15     3   2  75
## 16     4   2  97
## 17     1   2  13
## 18     2   2  58
## 19     3   2  78
## 20     4   2 108
## 21     1   2  20
## 22     2   2  73
## 23     3   2  90
## 24     4   2  99
```

The values in the sahpe column are 1-4 which correspond to the four different shapes that are cut into the trees. The treatment column has either 1 or 2 as the value, where 1 is no and 2 is yes, in regards to acid treatment.

Step 1: Data Pre-Processing

```
lapply(resin, class)
```

```
## $shape
## [1] "integer"
##
## $trt
## [1] "integer"
##
## $y
## [1] "integer"
```

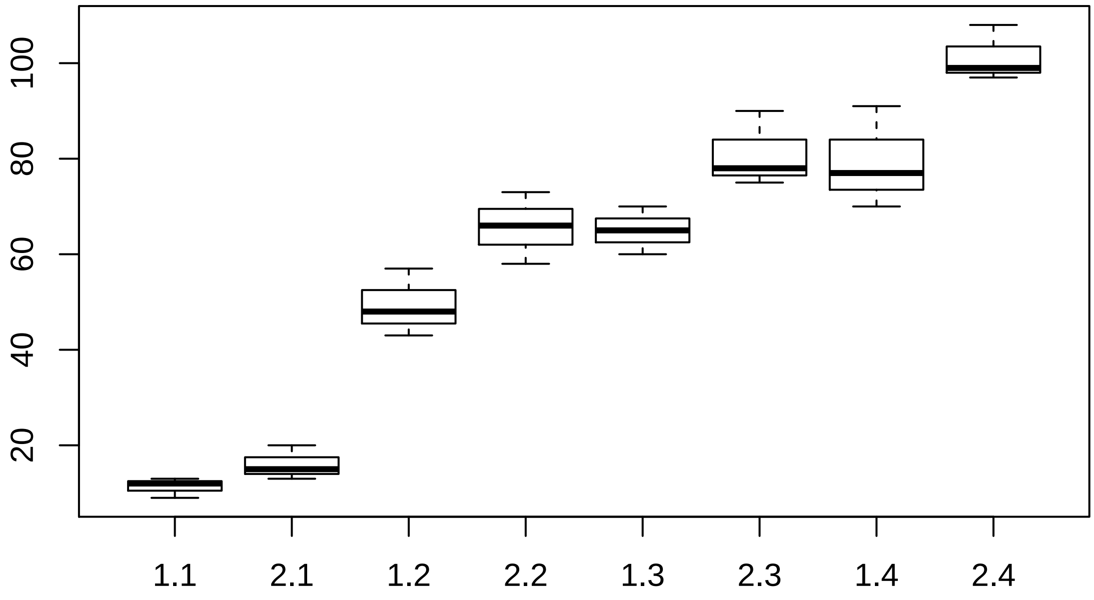
Since operator is not an integer we want to evaluate, it is necessary to change that variable to be recognized as a factor in R.

```
resin_adj <- with(resin, data.frame(shape=as.factor(shape),
                            trt = as.factor(trt), y = y))
lapply(resin_adj, class)
```

```
## $shape
## [1] "factor"
##
## $trt
## [1] "factor"
##
## $y
## [1] "integer"
```

Step 2: Exploratory Data Analysis

```
boxplot(y~trt + shape, data=resin_adj)
```



Looks from the sample that they might not be the same.

```
with(resin_adj, tapply(y, list(trt, shape), mean))
```

```
##          1        2  3         4
## 1 11.33333 49.33333 65  79.33333
## 2 16.00000 65.66667 81 101.33333
```

From these outputs, we see that the various combinations produce different sample mean responses.

Step 3: Fit the Model

- Additive Model

```
m1 <- lm(y~ trt + shape, data = resin_adj)
```

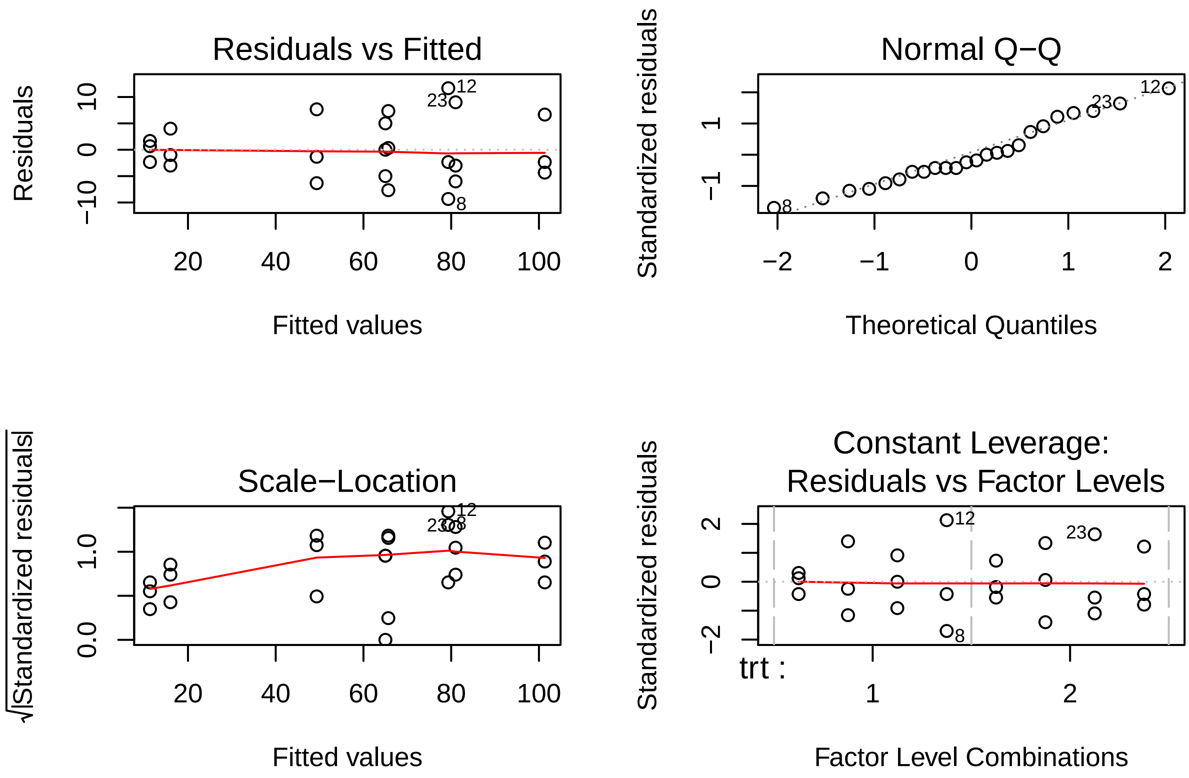
- Additive Model with Interaction

```
m2 <- lm(y~ trt*shape, data = resin_adj)
```

Which model do you go with?

To decide this, fit the model with interactions and run an anova. If the interaction term is not significant, just go with the additive model. The process is more complicated than just that, and it is known that it is necessary to evaluate the model diagnostics before preceding. But, that is the general overview.

```
par(mfrow = c(2,2))
plot(m2)
```



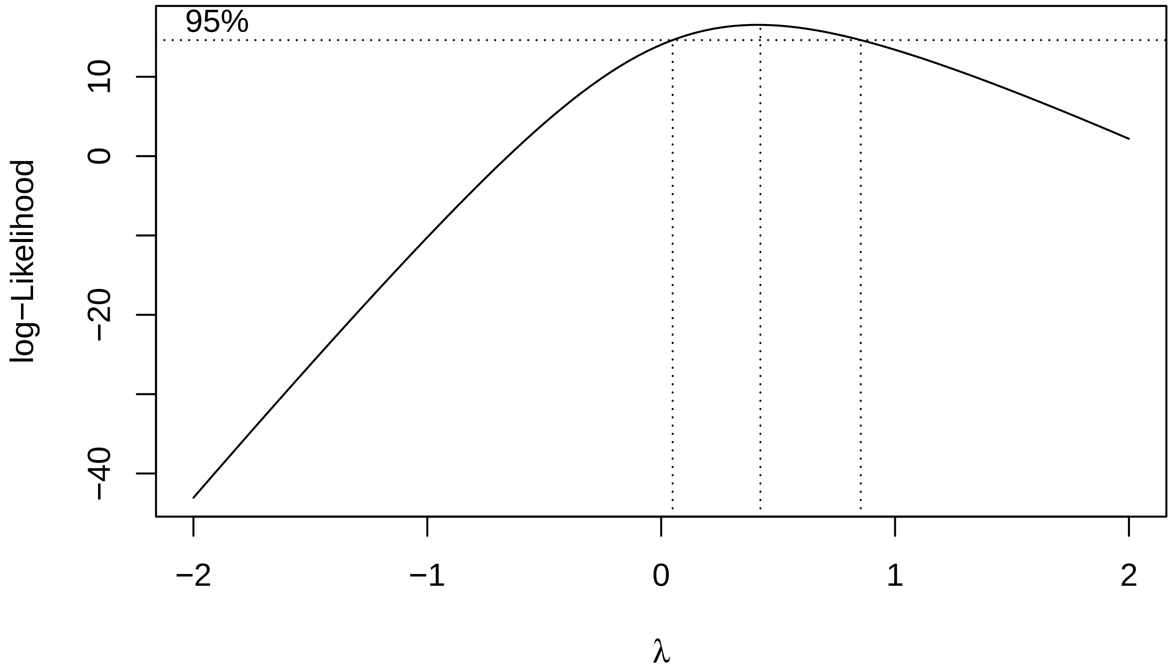
The plots give us the insight that the constant variance is most likely met, but there may be some issues with the assumption of normality. There may be outliers, and the line does not fit the QQ line as well as we would like.

To see if there is a better way to use this model, we attempt the boxcox command:

```
require(MASS)
```

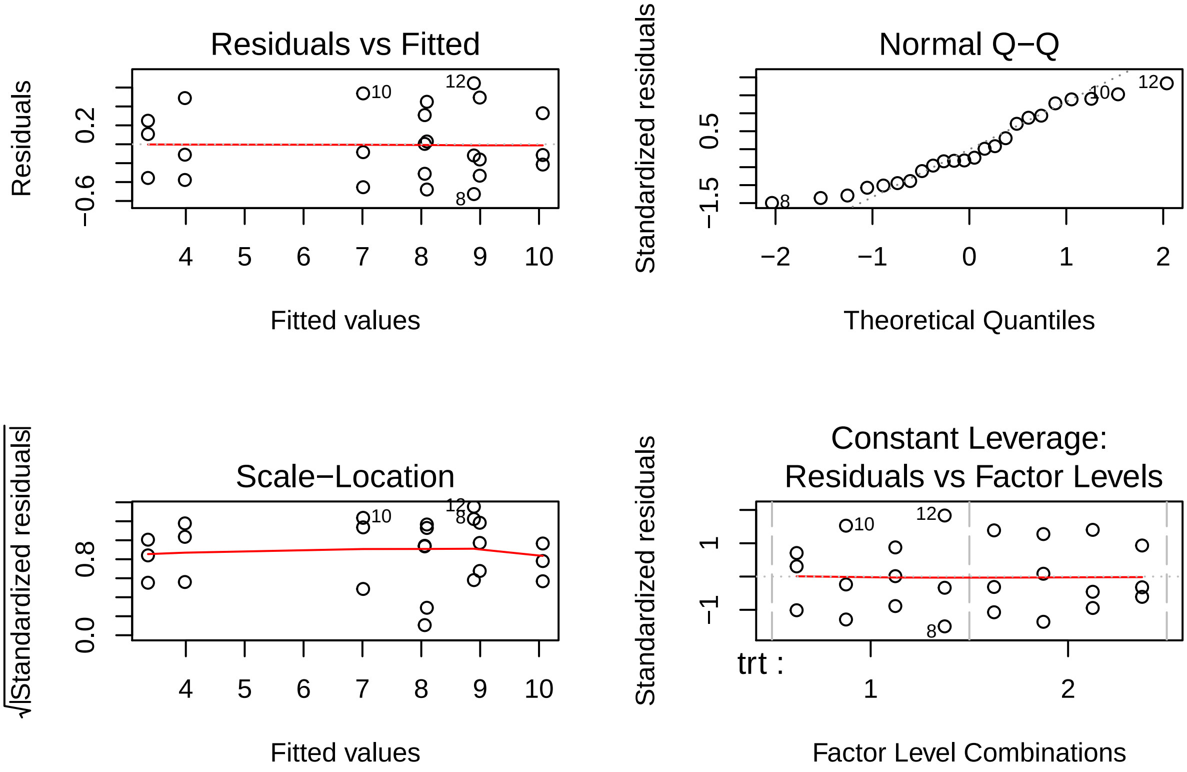
```
## Loading required package: MASS
```

```
boxcox(m2)
```



Based on the boxcox plot, we anticipate that the square-root transformation should stabilize the variability.

```
m3 <- lm(sqrt(y)~ trt * shape, data = resin_adj)
par(mfrow=c(2,2))
plot(m3)
```



Unfortunately, it did not make the plots any better, in fact for the Normal QQ Plot it created heavier tails in the plot and more divergence from the normal QQ line. This means we will go with the original model 2 since the assumptions were not grossly violated.

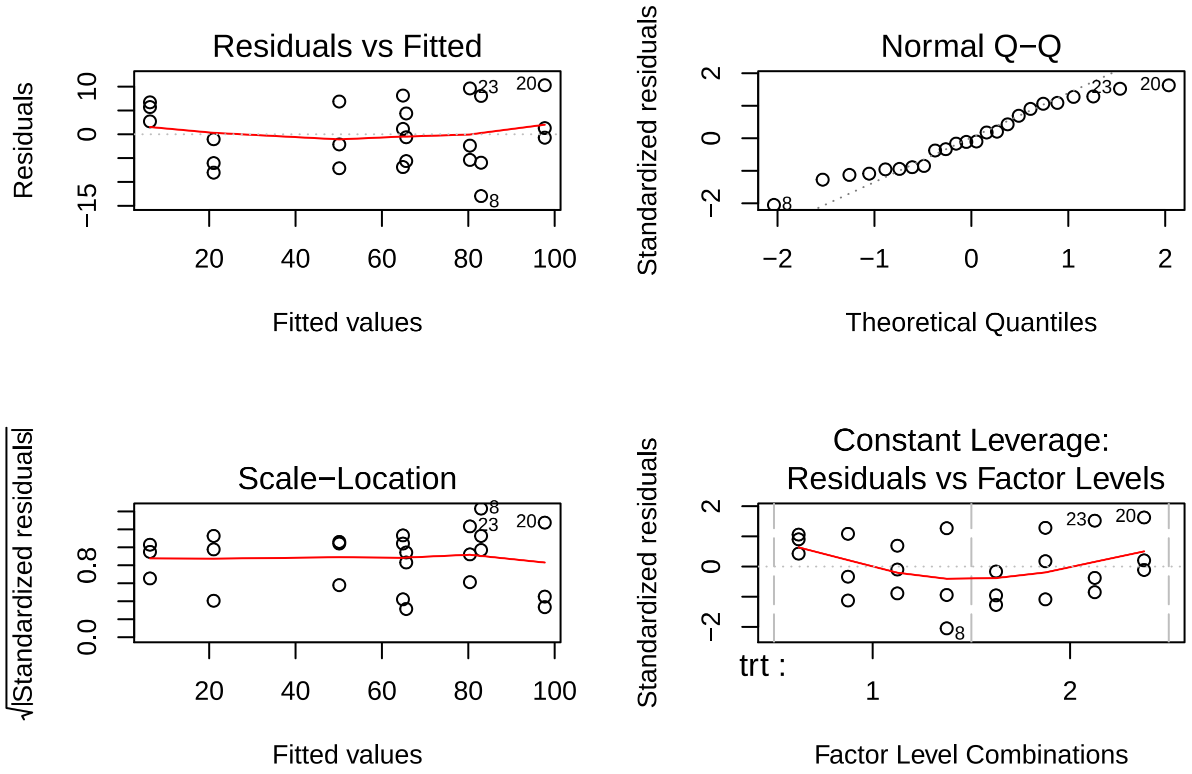
```
anova(m2)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df  Sum Sq Mean Sq  F value    Pr(>F)
## trt        1  1305.4  1305.4  28.9547 6.122e-05 ***
## shape      3 19407.5  6469.2 143.4932 8.934e-12 ***
## trt:shape  3   237.5    79.2   1.7557    0.1961
## Residuals 16   721.3    45.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA ouput states that treatment and shape are significant variables on their own, but are not significant as the interaction term.

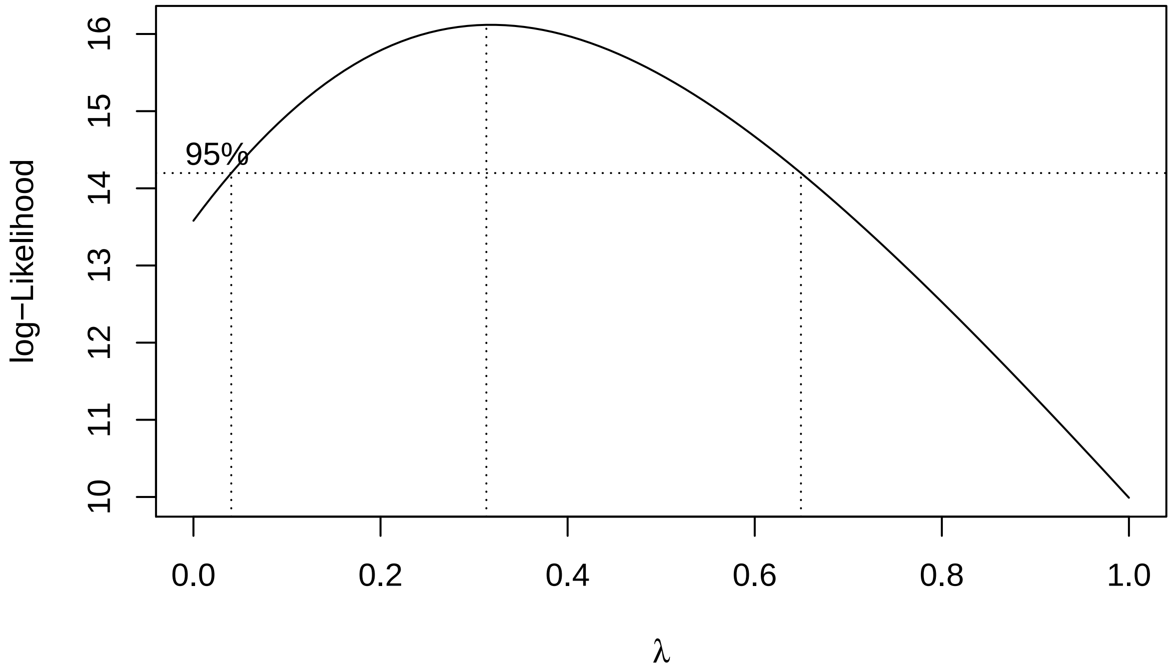
This means that we will go with the original additive model for this dataset:

```
par(mfrow = c(2,2))
plot(m1)
```



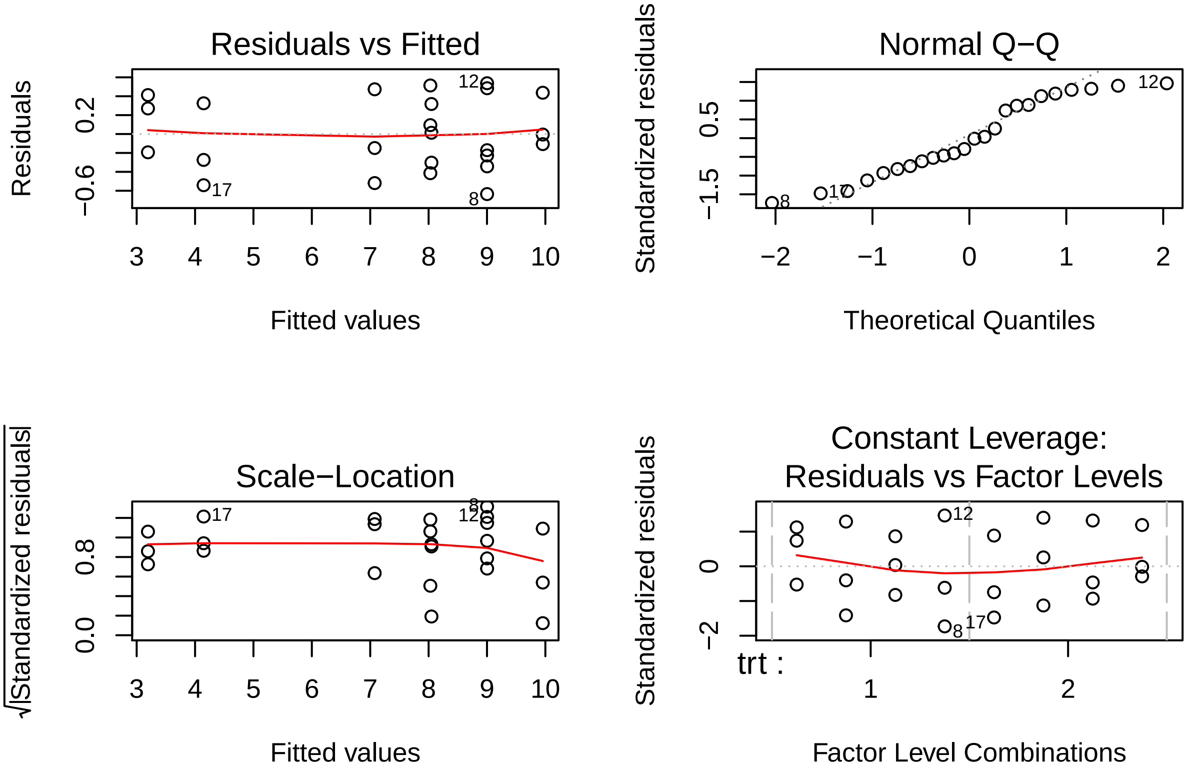
The Residuals v Fitted plot is bow shaped and the normal QQ plot does not fall on the line very well. Since the assumptions for the original dataset are not quite met, we again attempt to implement a boxcox model for this additive model.

```r
require(MASS)
boxcox(m1, lambda=seq(0, 1, 1/10))
```



Based on the boxcox plot, we anticipate that the square-root transformation should stabilize the variability.

```
m4 <- lm(sqrt(y)~ trt + shape, data = resin_adj)
par(mfrow=c(2,2))
plot(m4)
```



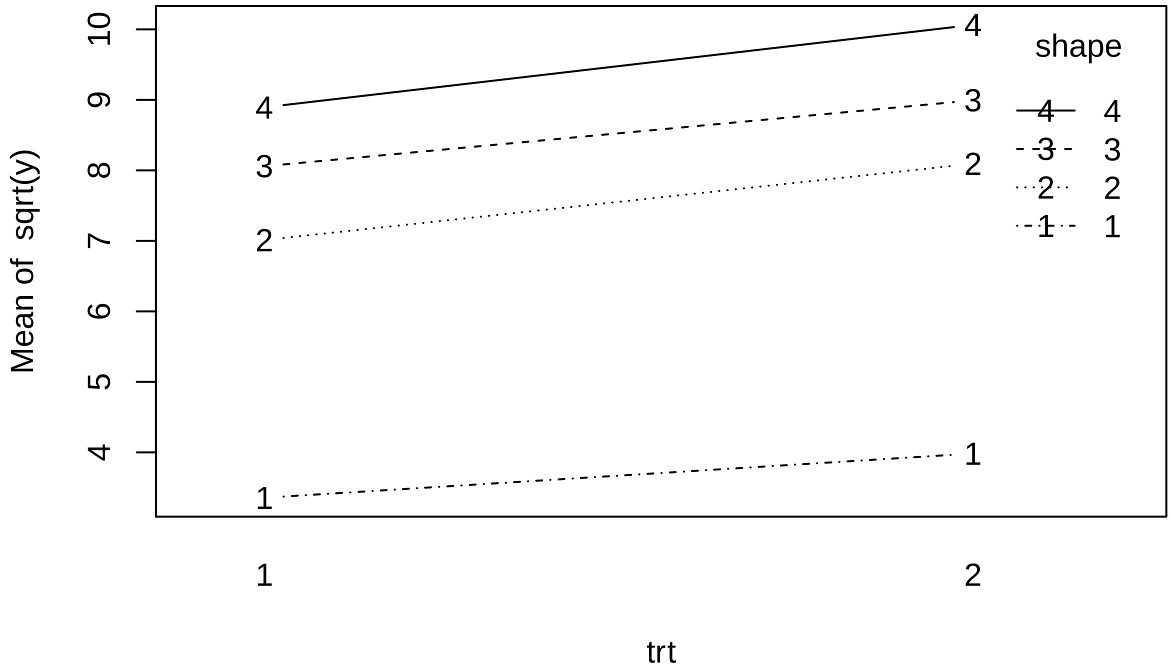
Overall, the boxcox transformation improved both the Residuals v Fitted plot and the Normal QQ Plot. So this is the model we will use for the ANOVA.

```r
anova(m4)
```

```
## Analysis of Variance Table
##
## Response: sqrt(y)
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## trt        1   5.456   5.456  32.051 1.855e-05 ***
## shape      3 116.935  38.978 228.964 4.342e-15 ***
## Residuals 19   3.235   0.170
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Even though interaction is not significant, we still fit the interaction plots because they give insight and information about the main effects.

```r
require(cfcdae)
with(resin_adj, interactplot(trt, shape, sqrt(y)))
```



Step 4: Conclusion

From the ANOVA output we see that both treatment and shape are significant factors of resin yield.

The interaction plots show us that the lines are all parallel. From this we can say, that treatment 2, using acid, resulted in higher resin yield, but there is the same difference in treatment 1 and treatment 2 between shapes. This means that resin yield depends on shape and treatment, but the extra yield from different treatments is the same for the four different shapes. This suggests no interaction between the plots which we saw above as well.