Project 2

Andreas Hochrein *ID: 4855928* *hochr007@umn.edu*

March 11, 2016

Problem 1

1. Preparation

```r
# Install and import packages/libraries required

#install.packages(c("lme4", "car", "perm", "effects", "tseries", "FrF2", "RLRsim", "conf.design", "rsm"
install.packages("Stat5303libs_0.7-4.tgz",repos=NULL)
install.packages("cfcdae_0.8-3.tgz",repos=NULL)
install.packages("pbkrtest_0.4-6.tgz",repos=NULL)
require(car)
```

```
## Loading required package: car
```

```r
require(lme4)
```

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

```r
require(alr4)
```

```
## Loading required package: alr4
## Loading required package: effects
##
## Attaching package: 'effects'
##
## The following object is masked from 'package:car':
##
##     Prestige
```

```r
require(Stat5303libs)
```

```
## Loading required package: Stat5303libs
## Loading required package: mvtnorm
## Loading required package: perm
## Loading required package: tseries
## Loading required package: FrF2
## Loading required package: DoE.base
## Loading required package: grid
## Loading required package: conf.design
##
## Attaching package: 'conf.design'
```

```
##
## The following object is masked from 'package:lme4':
##
##     factorize
##
##
## 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 packages for Statistics 5303
```

```
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
```

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

```
##     operator      substrate    strength
##  Min.   :1.00   Min.   :1   Min.   :2.750
##  1st Qu.:1.75   1st Qu.:1   1st Qu.:6.905
```

```
##  Median :2.50   Median :2   Median :7.660
##  Mean   :2.50   Mean   :2   Mean   :7.409
##  3rd Qu.:3.25   3rd Qu.:3   3rd Qu.:8.255
##  Max.   :4.00   Max.   :3   Max.   :9.000
```

```
# Check the data types of the variables
lapply(joint, class)
```

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

```
joint_f <- with(joint, data.frame(operator=as.factor(operator), substrate=as.factor(substrate), strengt
lapply(joint_f, class)
```

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

After checking the data types of the variables in the imported dataset, I transformed the type of the variables operator and substrate from integer to factor.

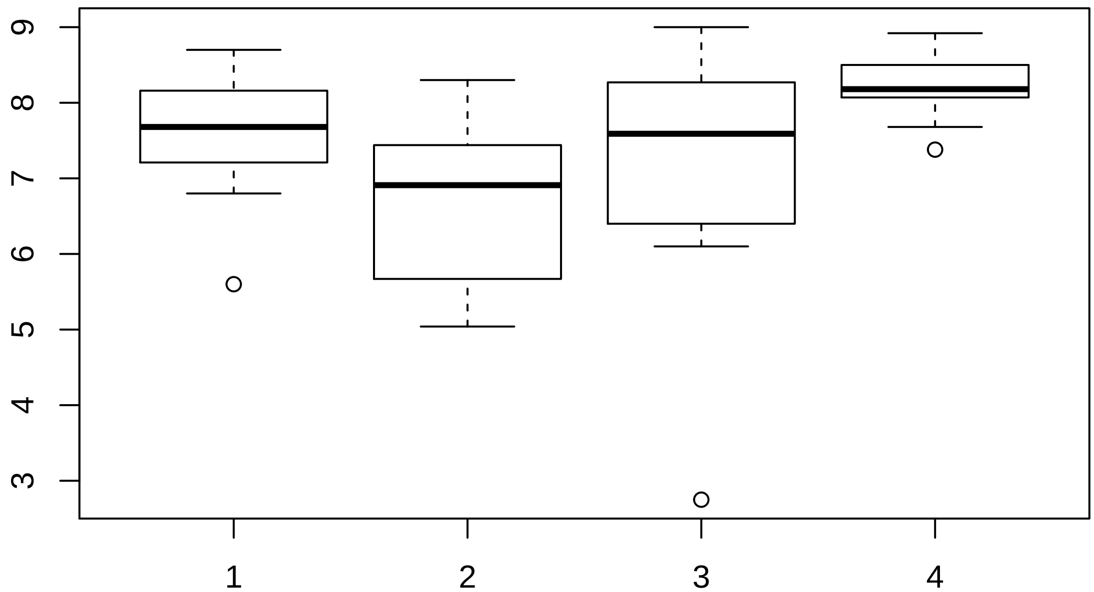
a) Analyze these data to determine if there is any evidence that the operators produce different mean shear strengths.

In other words, we have to build a model that contains variable strength as a response variable and operators as a regressor. We are interested in the relationship between these two.

2. Graphical Exploration

To get a better understanding of mean effects in the different operator groups and to see if there are potential outliers, we start the exploration of the data with a boxplot.

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



We can see that there are apparently one outlier each in group 1 and 4. The variance could to be greatest in groups 2 and 3. The differences in means are not that apparant on the plot and will require further analysis.

3. Model fitting and diagnostics

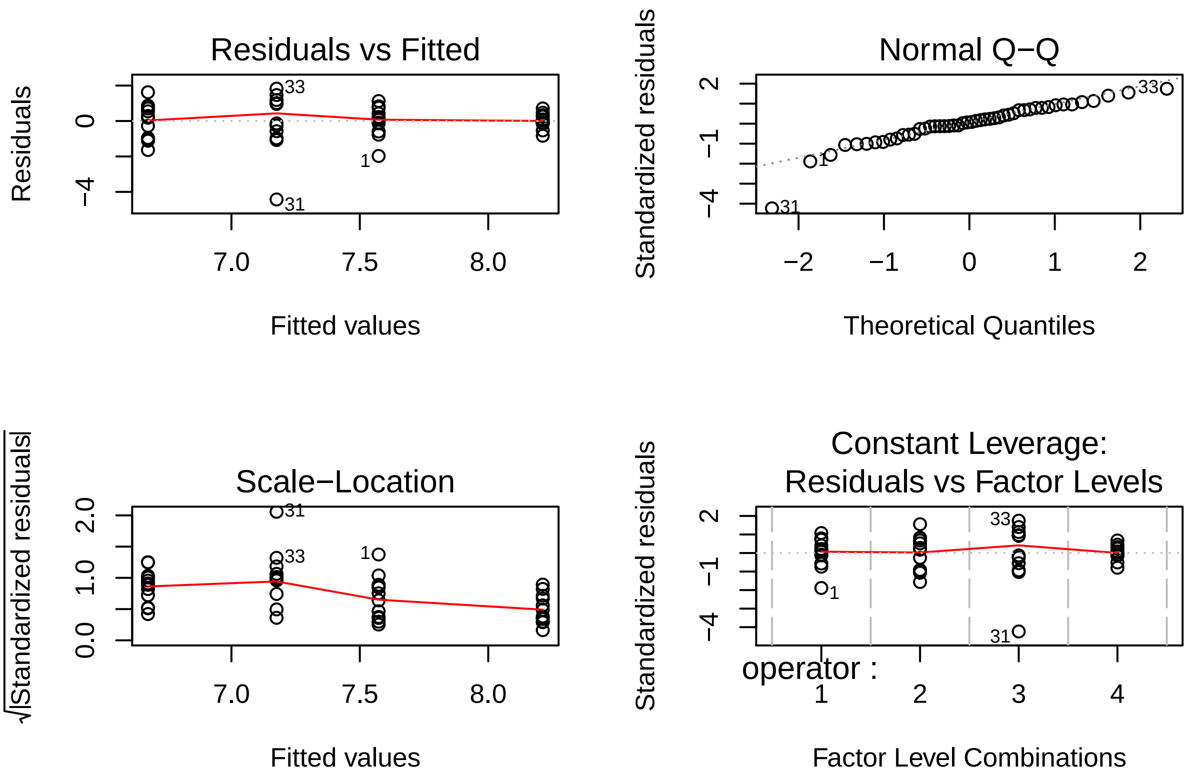
We will start the analysis by fitting the easiest and most straightforward model in this case:

strength ~ operator

```r
m1a <- lm(strength~operator, data=joint_f)
summary(m1a)
```

```
##
## Call:
## lm.default(formula = strength ~ operator, data = joint_f)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4258 -0.5433  0.0771  0.7173  1.8242
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.4092     0.1577  46.990   <2e-16 ***
## operator1     0.1642     0.2731   0.601   0.5508
## operator2    -0.7342     0.2731  -2.688   0.0101 *
## operator3    -0.2333     0.2731  -0.854   0.3975
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.092 on 44 degrees of freedom
## Multiple R-squared:  0.2244, Adjusted R-squared:  0.1715
## F-statistic: 4.243 on 3 and 44 DF,  p-value: 0.0102
```

```r
par(mfrow=c(2,2))
plot(m1a)
```



There are several problems with this first model. Only the difference between operator 4 and 2 appears to be significant, R-squared is only around .22 and the model fit thus low and the constant variance assumption seems to be violated. The residuals might also not be normally distributed.

4. Transformation

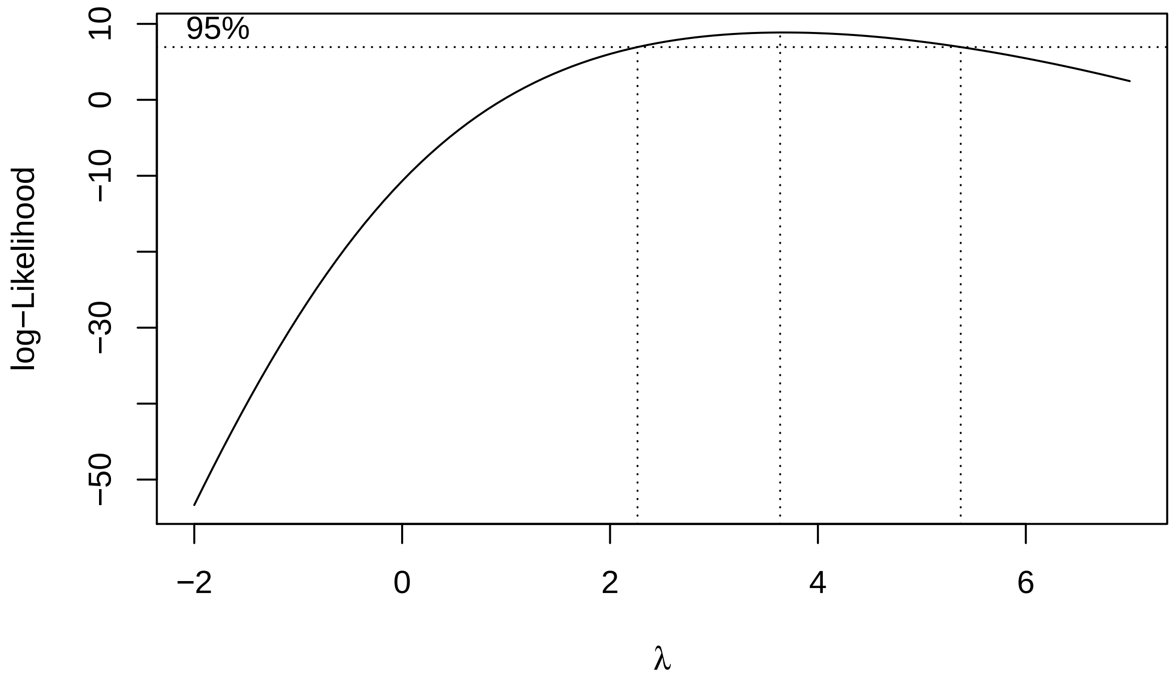
A next step is to analyze if there are any transformations that would be particulraly adequant for the given data. We cannot transform the regressor since it is a factor, i.e. not really a number. Thus, we explore if a transformation for the response makes sense.

```r
require(MASS)
```

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

```r
par(mfrow=c(1,1))

# Fit boxcox plot with appropriate range
bc <- boxcox(m1a, lambda=seq(-2, 7, 1/10))
```




```r
# Find the max
with(bc, x[which.max(y)])
```

```
## [1] 3.636364
```

```r
# Based a our findings, let's fit new models and diagnose them:

summary(m2a<- lm((strength^3)~operator, data=joint_f))
```

```
##
## Call:
## lm.default(formula = (strength^3) ~ operator, data = joint_f)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -399.72 -105.72    4.42  107.96  308.49
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   435.58      21.16  20.585  < 2e-16 ***
## operator1      12.77      36.65   0.348  0.72922
## operator2    -119.97      36.65  -3.273  0.00207 **
## operator3     -15.07      36.65  -0.411  0.68300
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 146.6 on 44 degrees of freedom
## Multiple R-squared:  0.274,  Adjusted R-squared:  0.2244
## F-statistic: 5.534 on 3 and 44 DF,  p-value: 0.002596
```

```r
summary(m3a<- lm((strength^3.5)~operator, data=joint_f))
```

```
##
## Call:
## lm.default(formula = (strength^3.5) ~ operator, data = joint_f)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1146.05  -349.41     5.43   327.22  1006.46
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1217.97      66.12  18.420  < 2e-16 ***
## operator1      32.68     114.53   0.285  0.77672
## operator2    -381.11     114.53  -3.328  0.00178 **
## operator3     -37.43     114.53  -0.327  0.74538
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 458.1 on 44 degrees of freedom
## Multiple R-squared:  0.2782, Adjusted R-squared:  0.229
## F-statistic: 5.653 on 3 and 44 DF,  p-value: 0.002296
```

```r
summary(m4a<- lm((strength^4)~operator, data=joint_f))
```

```
##
## Call:
## lm.default(formula = (strength^4) ~ operator, data = joint_f)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3270.3 -1125.0    -8.4  1018.9  3233.6
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3416.08     204.23  16.727  < 2e-16 ***
## operator1      79.39     353.73   0.224  0.82346
## operator2   -1188.28     353.73  -3.359  0.00162 **
## operator3     -88.64     353.73  -0.251  0.80330
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1415 on 44 degrees of freedom
## Multiple R-squared:  0.2804, Adjusted R-squared:  0.2313
## F-statistic: 5.715 on 3 and 44 DF,  p-value: 0.002155
```

```r
summary(m5a<- lm((strength^4.5)~operator, data=joint_f))
```

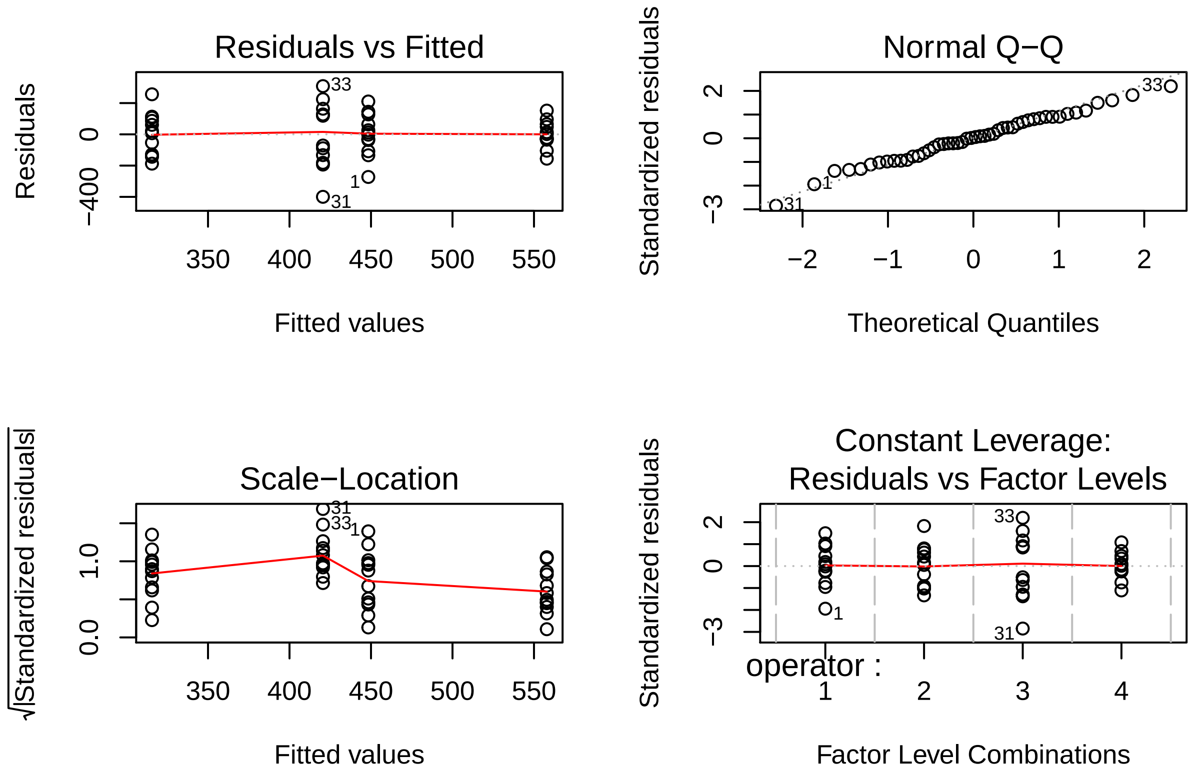
```
##
## Call:
## lm.default(formula = (strength^4.5) ~ operator, data = joint_f)
```

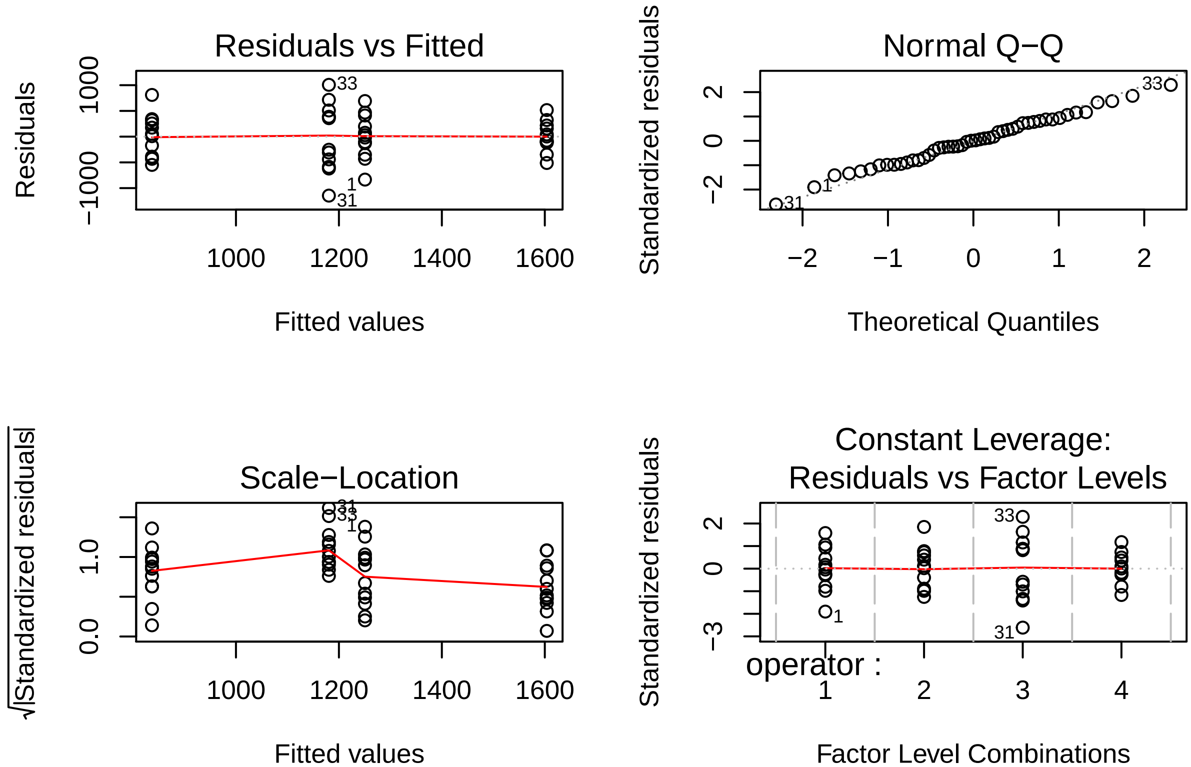
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9316.9 -3423.6   -99.1  2970.1 10271.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9607.9      625.5  15.360  < 2e-16 ***
## operator1      179.8     1083.4   0.166  0.86896
## operator2    -3654.9     1083.4  -3.373  0.00156 **
## operator3     -196.1     1083.4  -0.181  0.85718
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4334 on 44 degrees of freedom
## Multiple R-squared:  0.2809, Adjusted R-squared:  0.2319
## F-statistic: 5.731 on 3 and 44 DF,  p-value: 0.002121
```

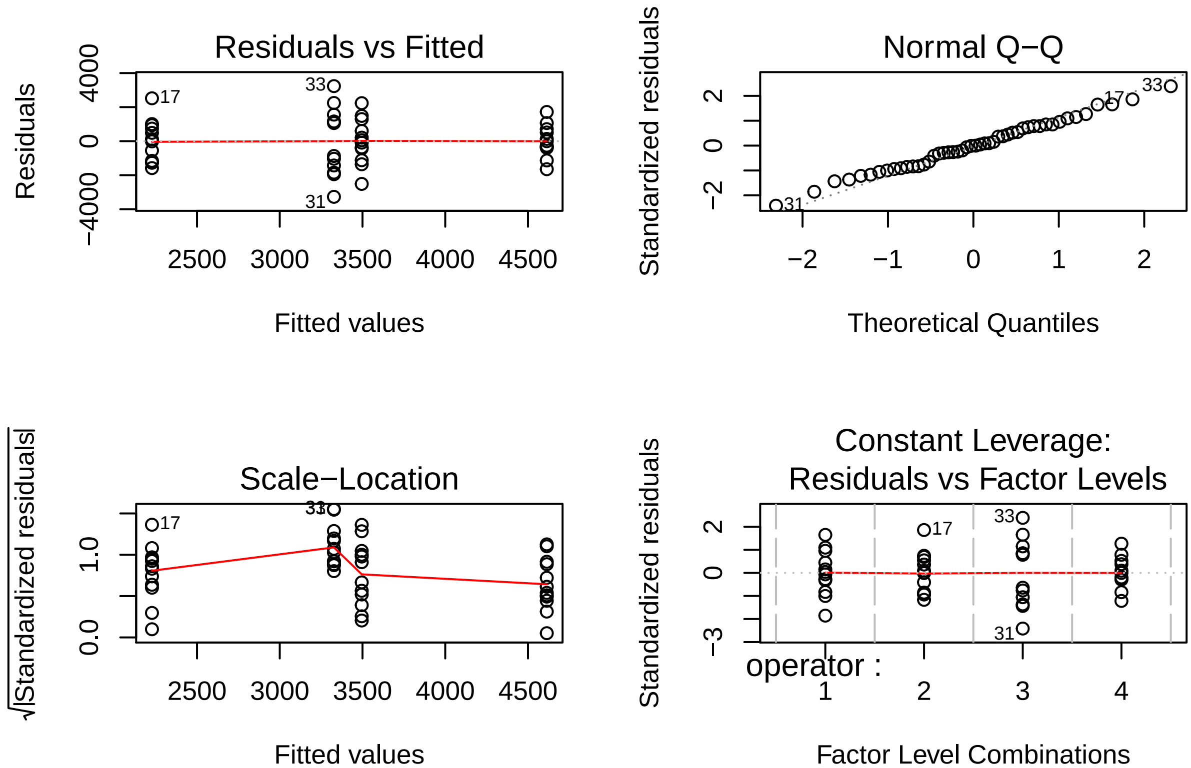
```
summary(m6a<- lm((strength^5)~operator, data=joint_f))
```

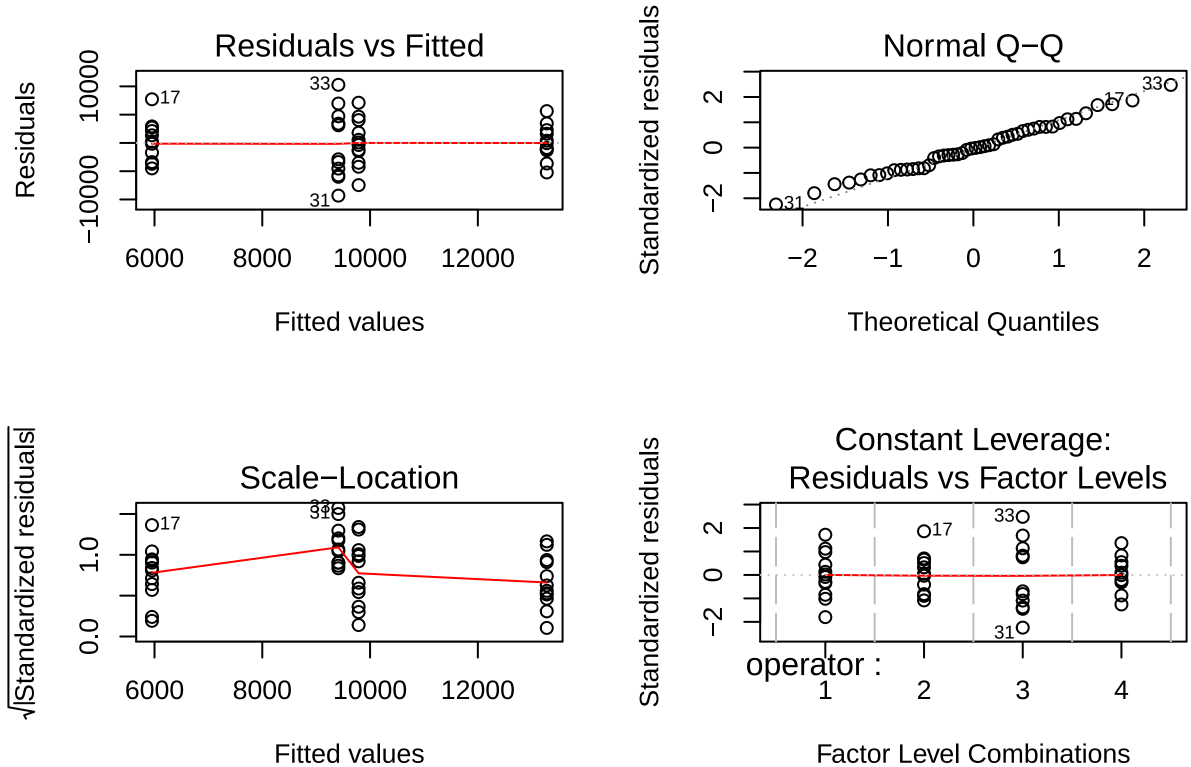
```
##
## Call:
## lm.default(formula = (strength^5) ~ operator, data = joint_f)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -26548 -10430   -513   8563  32344
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  27091.1     1904.2  14.227  < 2e-16 ***
## operator1      362.9     3298.2   0.110  0.91288
## operator2   -11127.8     3298.2  -3.374  0.00156 **
## operator3     -385.6     3298.2  -0.117  0.90745
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13190 on 44 degrees of freedom
## Multiple R-squared:  0.2802, Adjusted R-squared:  0.2311
## F-statistic:  5.71 on 3 and 44 DF,  p-value: 0.002167
```

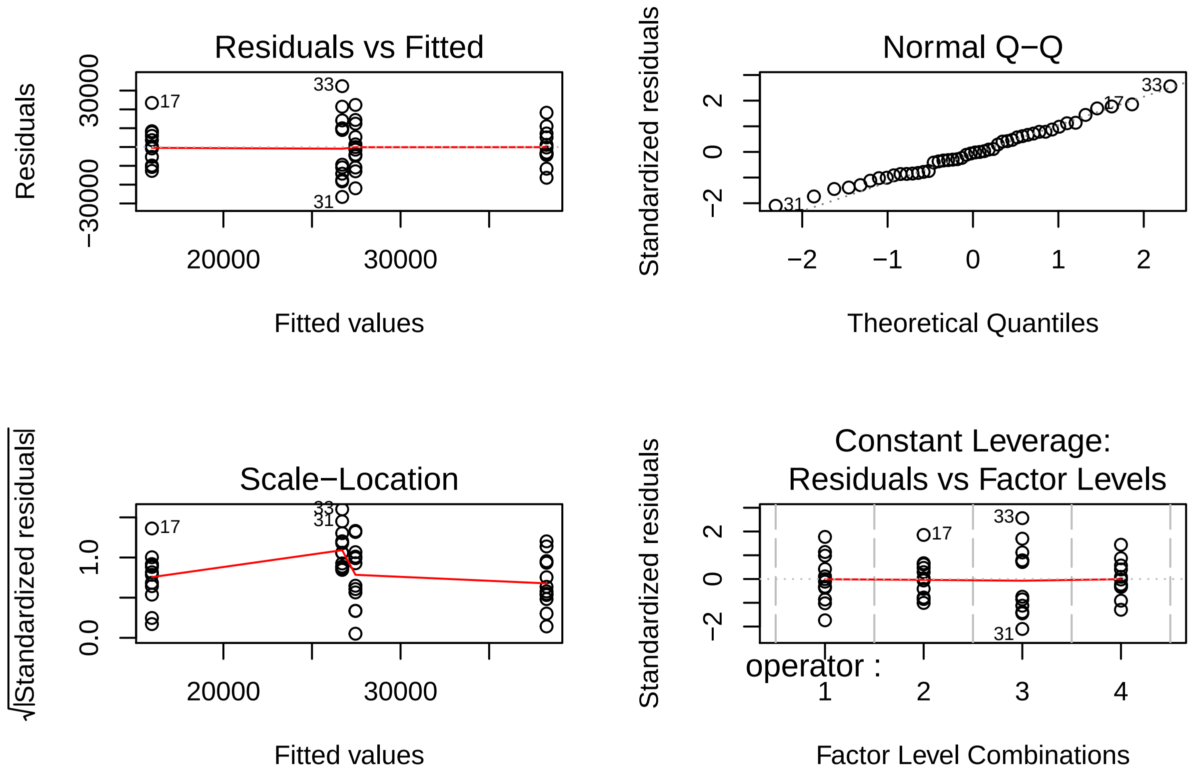
```
par(mfrow=c(2,2))
plot(m2a); plot(m3a); plot(m4a); plot(m5a); plot(m6a)
```


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
operator :
Factor Level Combinations


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
operator :
Factor Level Combinations


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
operator :
Factor Level Combinations


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
operator :
Factor Level Combinations




```r
# Compare R-squared
compR <-rbind(summary(m1a)$r.squared,summary(m2a)$r.squared,summary(m3a)$r.squared,summary(m4a)$r.squar
rownames(compR)<-c("m1","m2","m3","m4","m5","m6")
colnames(compR)<-c("R^2")
compR
```

```
##           R^2
## m1 0.2243711
## m2 0.2739501
## m3 0.2782045
## m4 0.2803926
## m5 0.2809480
## m6 0.2802093
```

In the covariant operator, level 2 is the only one significantly different from 4 in all models. The constant variance assumption seems to be satisfied in models 4 to 6 and normality of the residual seems to be more or less satisfied in all models, even though there might be derivations here. Since multiple R-squared is highest in model 5, I will choose this model for this problem.

5. Inference

After having shoosen the model, now it's time to explore if the means for shear strength in the different groups are all the same or if there are significant differences.

```r
anova(m5a)
```

```
## Analysis of Variance Table
```

```
##
## Response: (strength^4.5)
##           Df    Sum Sq   Mean Sq F value   Pr(>F)
## operator   3 322885699 107628566  5.7306 0.002121 **
## Residuals 44 826386478  18781511
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

6. Interpretation

The ANOVA output indicates that the means of shear strength are different for different operators since the p-value < .01. Thus, there is evidence that different operators produce different shear strengths and we reject the null hypothesis that all means are the same. The mean effects in the linear model indicate that operator 2 has a significant lower mean than operator 4 in this model.

---

b) 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.

This will be a parallel analysis process to a), but with the means for novice and experiences workers instead of for each operator.

```
novice <- as.factor(ifelse(joint_f$operator==3|joint_f$operator==4,1,0))
joint_bf <- cbind(joint_f,novice)
summary(joint_bf)
```

```
##  operator substrate    strength     novice
##  1:12     1:16      Min.   :2.750   0:24
##  2:12     2:16      1st Qu.:6.905   1:24
##  3:12     3:16      Median :7.660
##  4:12               Mean   :7.409
##                     3rd Qu.:8.255
##                     Max.   :9.000
```

```
lapply(joint_bf, class)
```

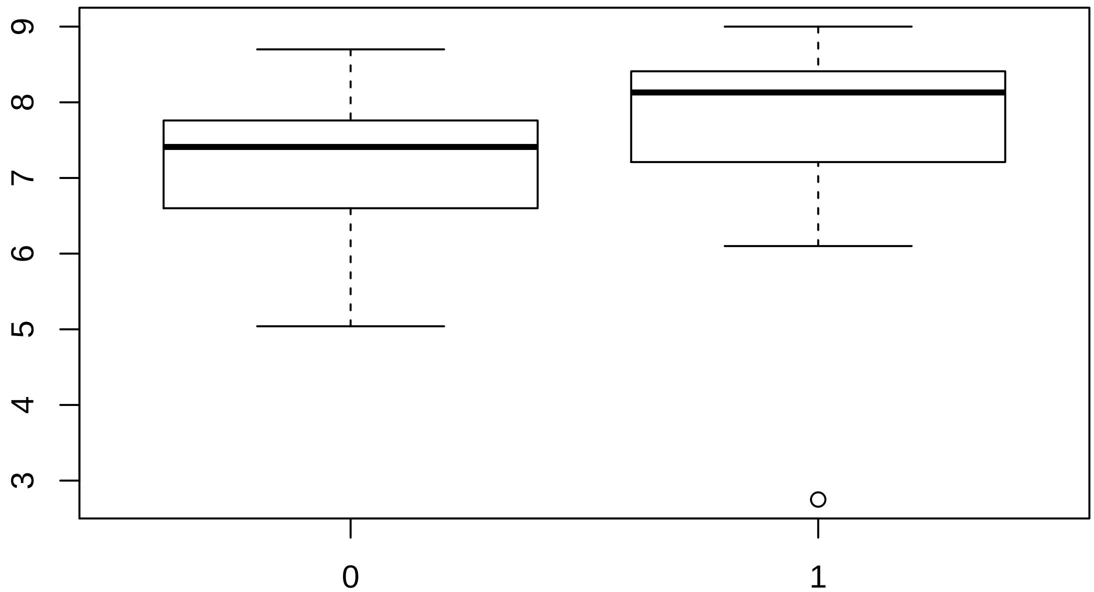
```
## $operator
## [1] "factor"
##
## $substrate
## [1] "factor"
##
## $strength
## [1] "numeric"
##
## $novice
## [1] "factor"
```

The added categorical variable novice has a value of 1 for the novice workers and 0 for experienced workers.

2. Graphical Exploration

To get a better understanding of mean effect in novice and experienced workers and to see if there are potential outliers, we start the exploration of the data with a boxplot.

```
par(mfrow=c(1,1))
boxplot(strength~novice, data=joint_bf)
```



We can see that there is one outlier for novice workers.The differences in means are not that apparent on the plot and will require further analysis.

3. Model fitting and diagnostics

We will start the analysis by fitting the easiest and most straightforward model in this case:

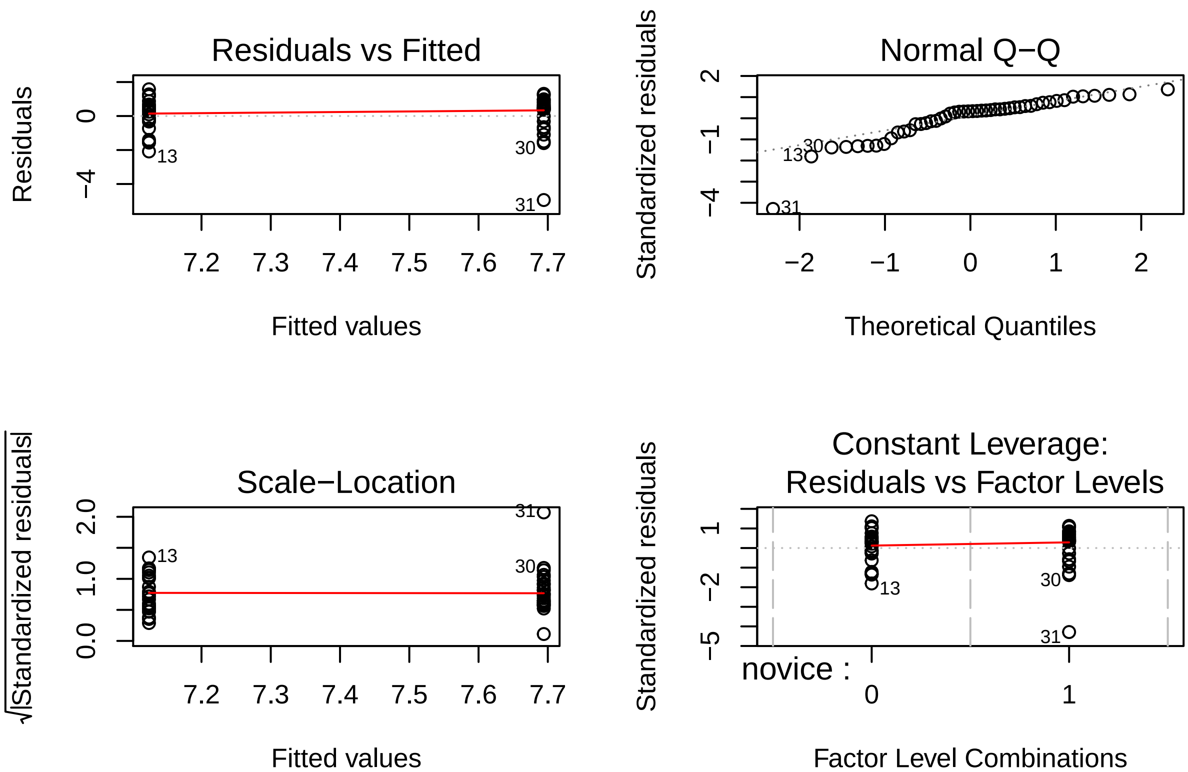
strength ~ novice

```
m1b <- lm(strength~novice, data=joint_bf)
summary(m1b)
```

```
##
## Call:
## lm.default(formula = strength ~ novice, data = joint_bf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9442 -0.4067  0.3808  0.6683  1.5758
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    7.409      0.170  43.588   <2e-16 ***
## novice1       -0.285      0.170  -1.677      0.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.178 on 46 degrees of freedom
## Multiple R-squared:  0.05759,    Adjusted R-squared:  0.03711
## F-statistic: 2.811 on 1 and 46 DF,  p-value: 0.1004
```

```r
par(mfrow=c(2,2))
plot(m1b)
```



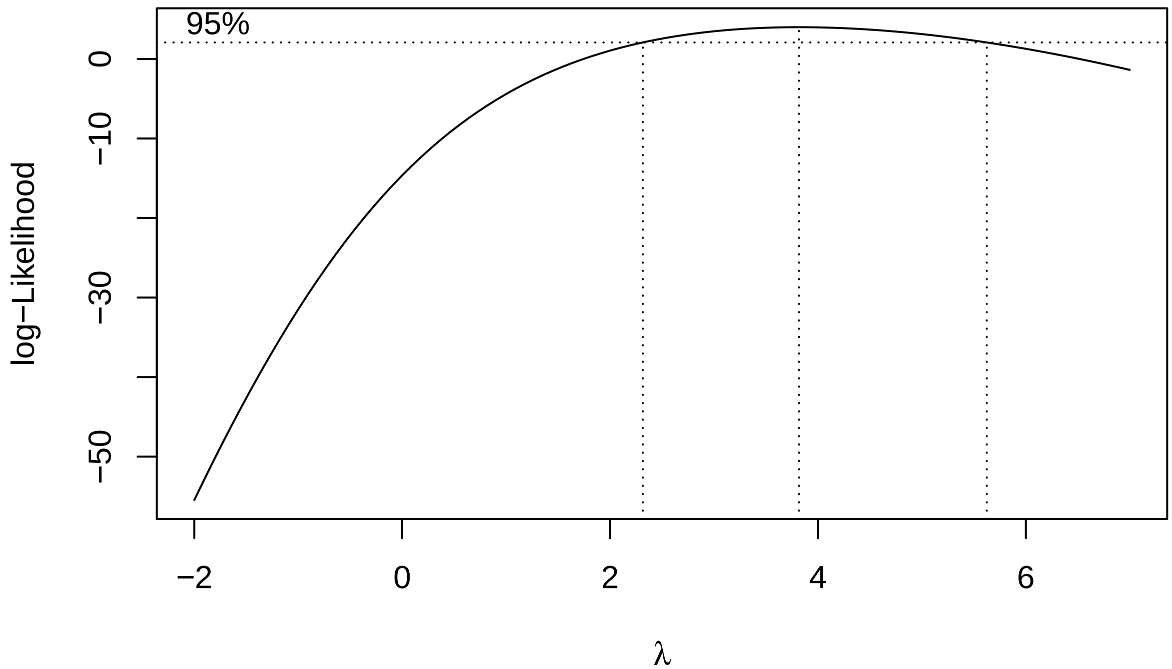
R-squared is only around .05 and the model fit thus low. The constant variance assumption is not completely satisfied. The residuals might also not be normally distributed. The difference in the mean effects of both levels in factor novice not seem to be significant.

4. Transformation

A next step is to analyze if there are any transformations that would be particularly adequant for the given data. We cannot transform the regressor since it is a factor, i.e. not really a number. Thus, we explore if a transformation for the response makes sense.

```r
require(MASS)
par(mfrow=c(1,1))

# Fit boxcox plot with appropriate range
bc <- boxcox(m1b, lambda=seq(-2, 7, 1/10))
```




```r
# Find the max
with(bc, x[which.max(y)])
```

```
## [1] 3.818182
```

```r
# Based a our findings, let's fit new models and diagnose them:

summary(m2b<- lm((strength^3)~novice, data=joint_bf))
```

```
##
## Call:
## lm.default(formula = (strength^3) ~ novice, data = joint_bf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -468.39  -95.39   39.11   93.20  276.52
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   435.58      22.97  18.967   <2e-16 ***
## novice1       -53.60      22.97  -2.334    0.024 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 159.1 on 46 degrees of freedom
## Multiple R-squared:  0.1059, Adjusted R-squared:  0.08645
## F-statistic: 5.448 on 1 and 46 DF,  p-value: 0.02402
```

```r
summary(m3b<- lm((strength^3.5)~novice, data=joint_bf))
```

```
##
```

```
## Call:
## lm.default(formula = (strength^3.5) ~ novice, data = joint_bf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1357.7  -320.3   109.5   285.5   898.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1217.97      71.65  16.998   <2e-16 ***
## novice1      -174.22      71.65  -2.431    0.019 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 496.4 on 46 degrees of freedom
## Multiple R-squared:  0.1139, Adjusted R-squared:  0.09461
## F-statistic: 5.911 on 1 and 46 DF,  p-value: 0.019
```

```r
summary(m4b<- lm((strength^4)~novice, data=joint_bf))
```

```
##
## Call:
## lm.default(formula = (strength^4) ~ novice, data = joint_bf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3913.3 -1049.1   297.1   858.4  2867.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3416.1      220.8  15.471   <2e-16 ***
## novice1       -554.4      220.8  -2.511   0.0156 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1530 on 46 degrees of freedom
## Multiple R-squared:  0.1205, Adjusted R-squared:  0.1014
## F-statistic: 6.305 on 1 and 46 DF,  p-value: 0.01562
```

```r
summary(m5b<- lm((strength^4.5)~novice, data=joint_bf))
```

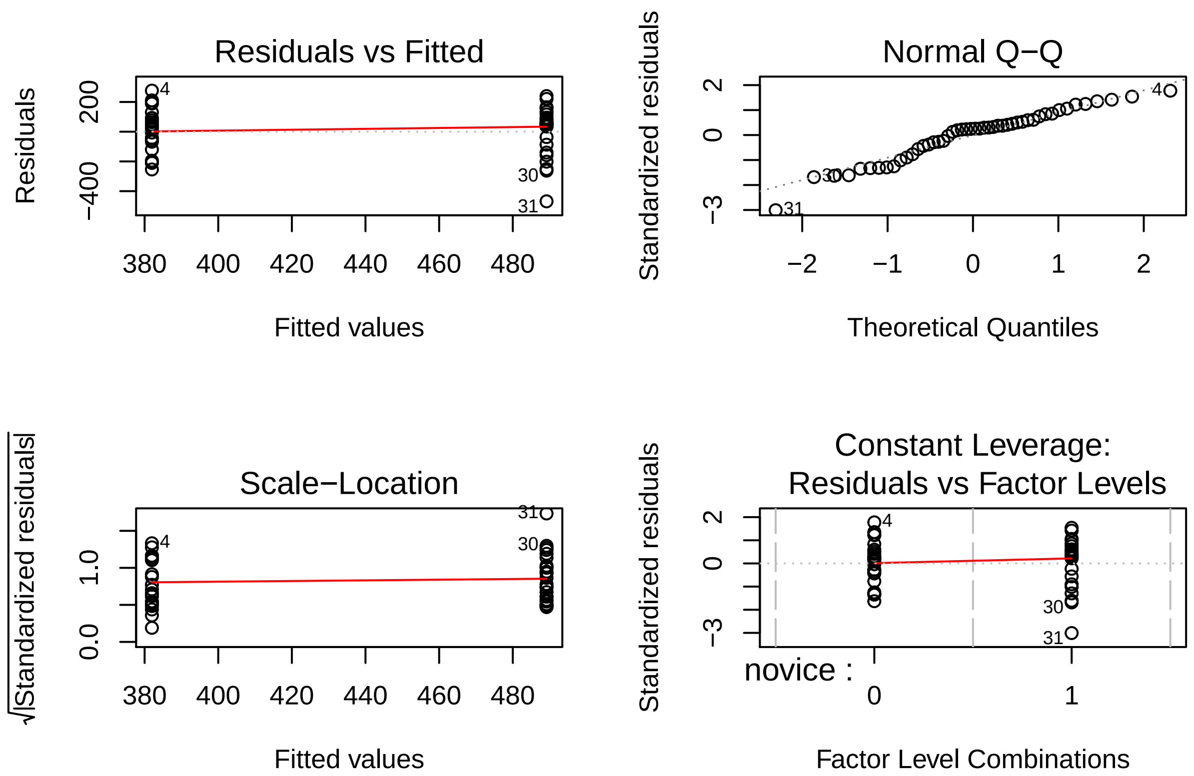
```
##
## Call:
## lm.default(formula = (strength^4.5) ~ novice, data = joint_bf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11250.6  -3371.7    782.6   2545.0   9027.8
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9607.9      674.4  14.246   <2e-16 ***
## novice1      -1737.6      674.4  -2.576   0.0133 *
```

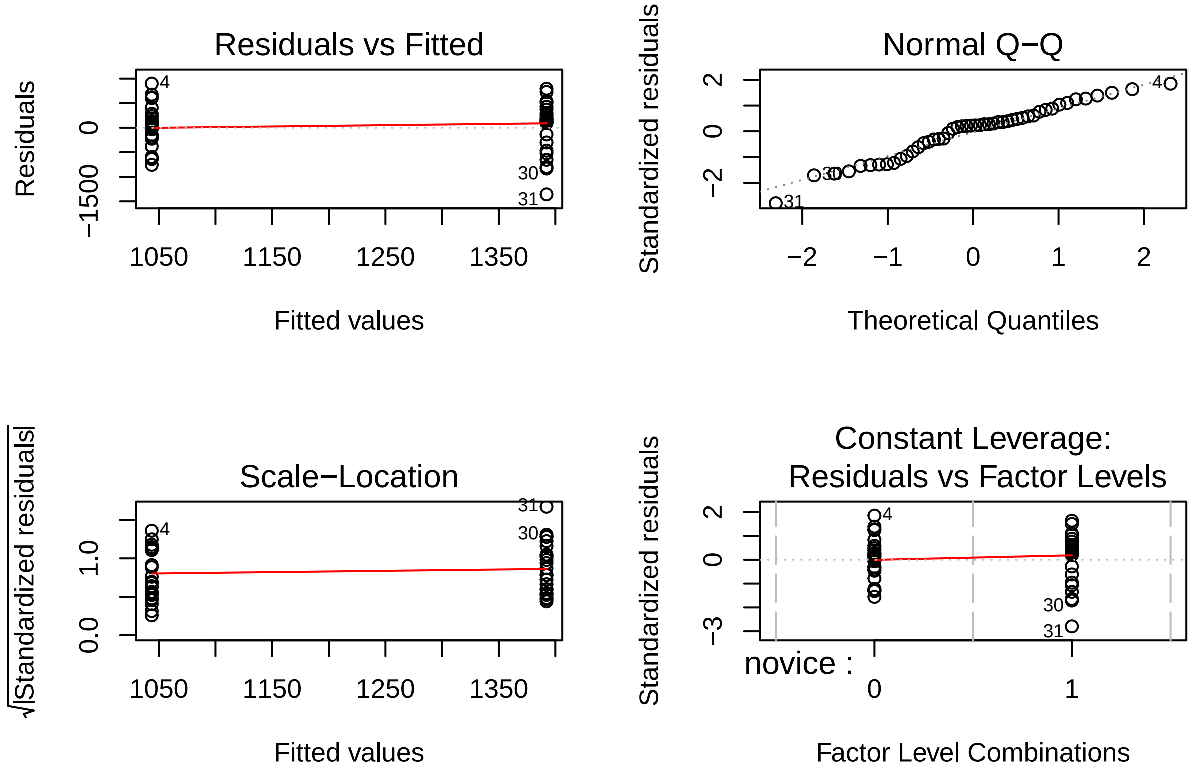
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4673 on 46 degrees of freedom
## Multiple R-squared:  0.1261, Adjusted R-squared:  0.1071
## F-statistic: 6.637 on 1 and 46 DF,  p-value: 0.01326
```

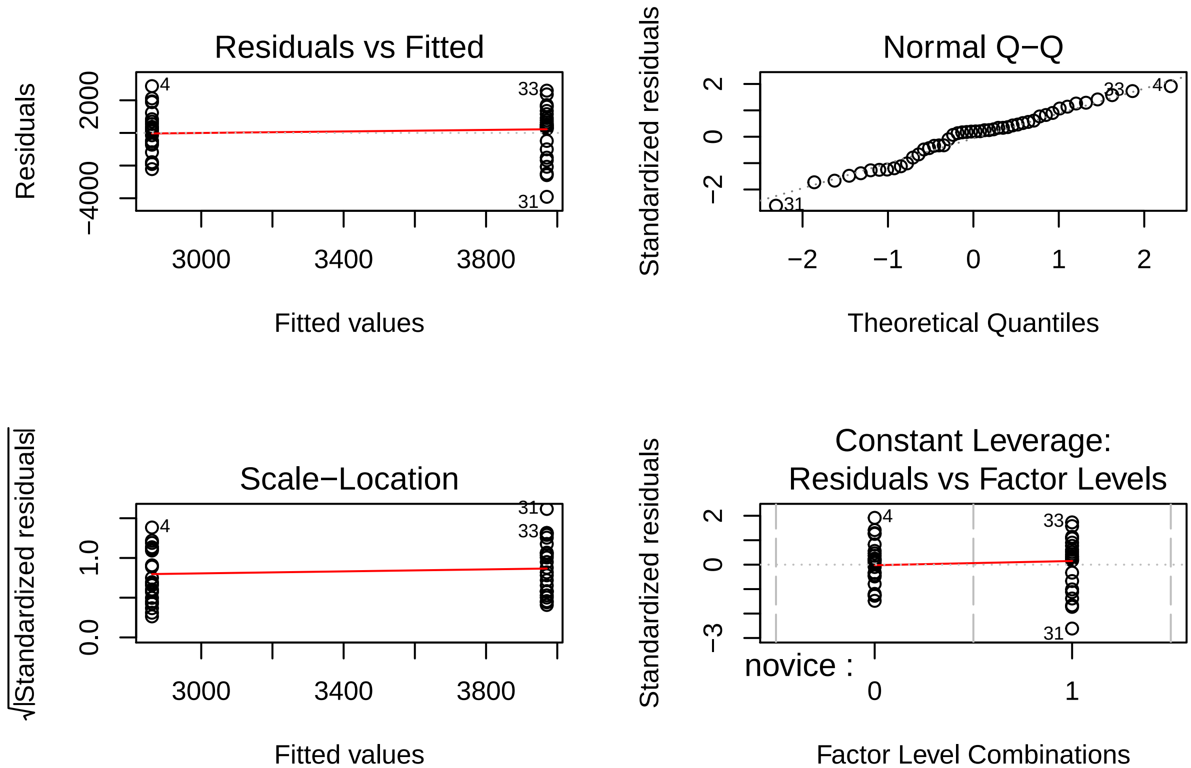
```
summary(m6b<- lm((strength^5)~novice, data=joint_bf))
```

```
##
## Call:
## lm.default(formula = (strength^5) ~ novice, data = joint_bf)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -32316 -10679   1994   7463  28133
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    27091       2047   13.24   <2e-16 ***
## novice1        -5382       2047   -2.63   0.0116 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14180 on 46 degrees of freedom
## Multiple R-squared:  0.1307, Adjusted R-squared:  0.1118
## F-statistic: 6.916 on 1 and 46 DF,  p-value: 0.01158
```

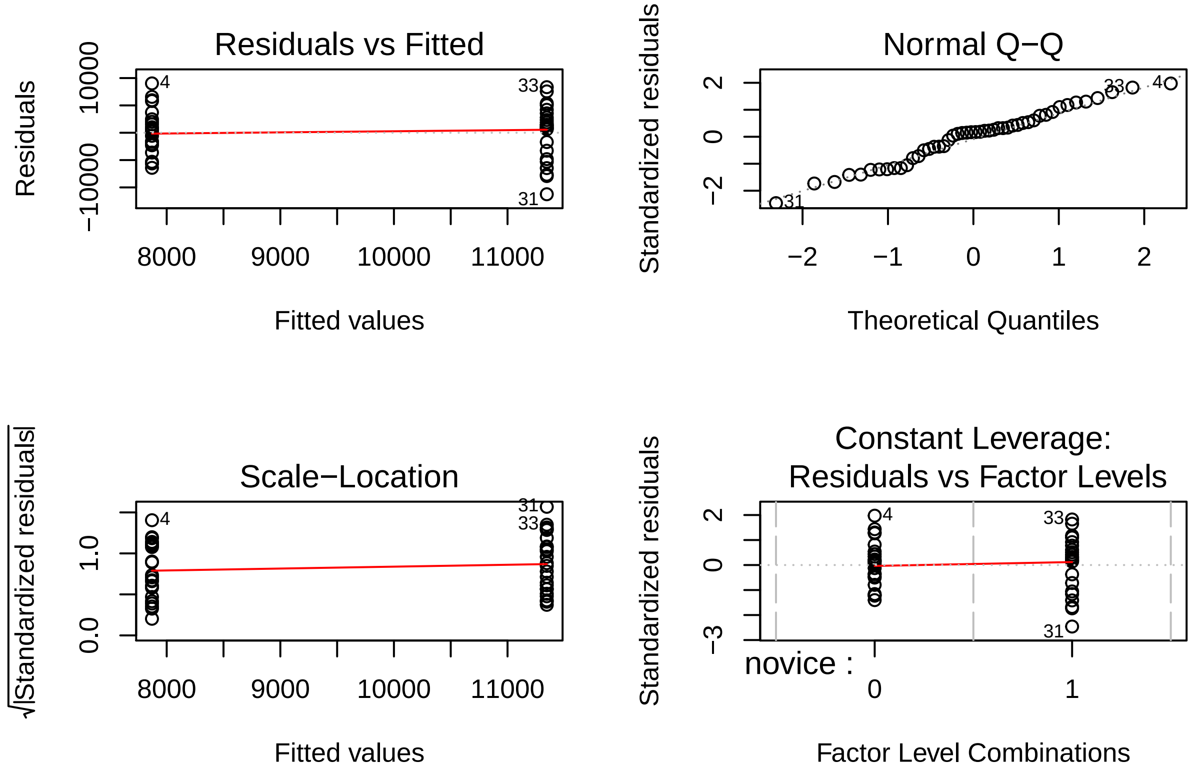
```
par(mfrow=c(2,2))
plot(m2b); plot(m3b); plot(m4b); plot(m5b); plot(m6b)
```


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
novice :
Factor Level Combinations

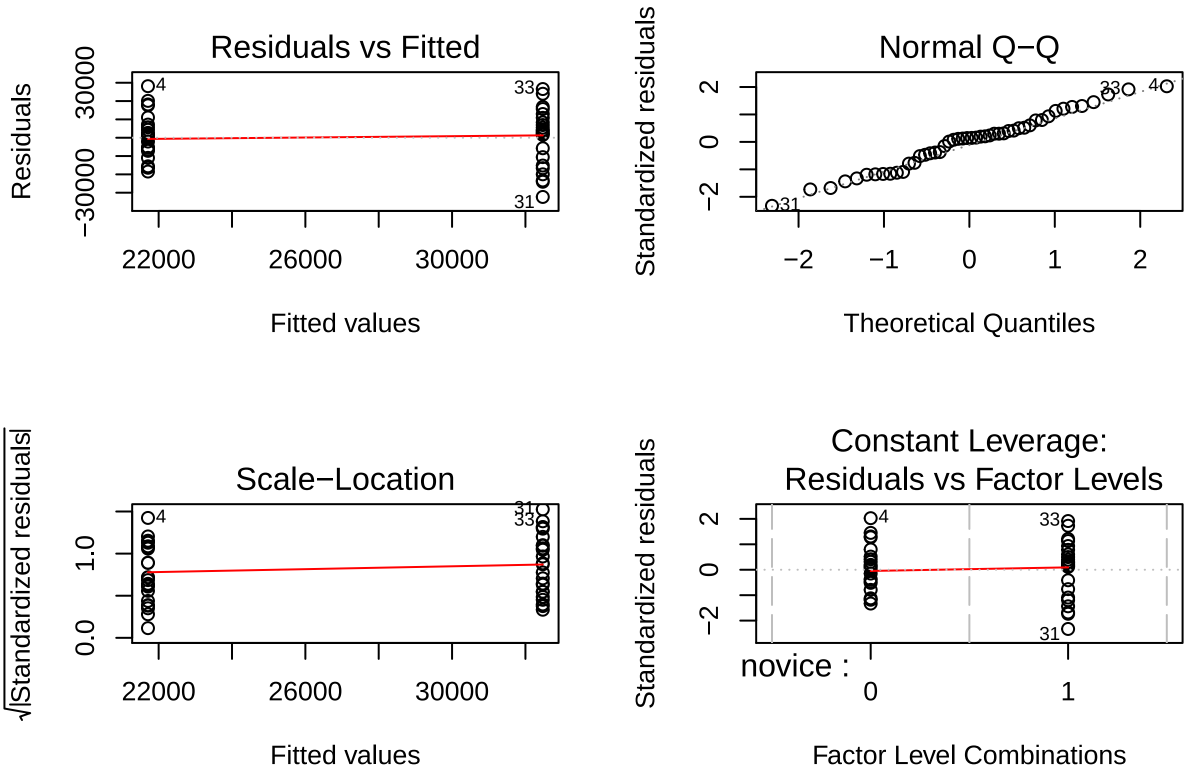

Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
novice :
Factor Level Combinations



Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
novice :
Factor Level Combinations



Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
novice :
Factor Level Combinations




```r
# Compare R-squared
compR <-rbind(summary(m1b)$r.squared,summary(m2b)$r.squared,summary(m3b)$r.squared,summary(m4b)$r.square
rownames(compR)<-c("m1","m2","m3","m4","m5","m6")
colnames(compR)<-c("R^2")
compR
```

```
##            R^2
## m1 0.05759331
## m2 0.10588881
## m3 0.11387400
## m4 0.12054290
## m5 0.12609456
## m6 0.13069974
```

The difference in the levels of covariant factor novice is significant in models 2, 3, 4, 5 and 6. The constant variance assumption seems to be violated in all models and the normality assumption might not be fullfilled in this model. Since multiple R-squared is highest in model 6, I will choose this model for this problem.

5. Inference

After having shoosen the model, now it's time to explore if the means for shear strength in the two levels novice are all the same or if there are significant differences.

```r
anova(m6b)
```

```
## Analysis of Variance Table
##
```

```
## Response: (strength^5)
##           Df     Sum Sq    Mean Sq F value  Pr(>F)
## novice     1 1390594192 1390594192  6.9161 0.01158 *
## Residuals 46 9249015360  201065551
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
linear.contrast(m6b, novice, allpairs=T, confidence=.95)
```

```
##       estimates       se   t-value    p-value  lower-ci  upper-ci
## 1 - 2 -10764.89 4093.344 -2.629852 0.01157889 -19004.36 -2525.416
```

6. Interpretation

The ANOVA output and pairwise comparison indicates that the means of shear strength are different for novices and experienced workers since the p-value $< .05$. Thus, there is evidence that operators with different experience levels produce different shear strengths. The mean effects in the linear model indicate that novice workers produce a lower mean strength than the experienced workers.

---

c) 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.

For this question we will focus on the same variables as in part a). Thus, I will use the same model that I used in part a), model m5a: Strength^4.5 ~ Operator. I have already done modeling, testing of assumptions, and anova in part a) for this model. For this question I will analyze between which operators, if any, there are differences in the mean shear strengths measured.

H0: means for all workers are equal H1: At least one of the means for the between two workers are different

Test:

```r
linear.contrast(m5a, operator, allpairs=T, confidence=.95)
```

```
##        estimates       se   t-value      p-value    lower-ci    upper-ci
## 1 - 2  3834.6854 1769.252  2.167405 0.0356575933    268.9928  7400.37807
## 1 - 3   375.9201 1769.252  0.212474 0.8327185091  -3189.7726  3941.61273
## 1 - 4 -3491.4618 1769.252 -1.973411 0.0547497943  -7057.1544    74.23087
## 2 - 3 -3458.7653 1769.252 -1.954931 0.0569624497  -7024.4580   106.92729
## 2 - 4 -7326.1472 1769.252 -4.140817 0.0001541798 -10891.8398 -3760.45456
## 3 - 4 -3867.3819 1769.252 -2.185886 0.0341889759  -7433.0745  -301.68921
```

As the linear.contrast test indicates, there are significant differences in the shear strengths produced by some workers if we compare them in pairs on none for others. At a confidence level of .95, we find significant differences between workers 1 and 2, 2 and 4, and 3 and 4. Thus, we reject the Null hypothesis that all the mean strength produced by all workers are the same.

Problem 2

The data in this problem was collected through a random experiment. Shape and treatment are the covariants and factors. Shape has four different levels (1=circular, 2=diagonal, 3=check, 4=rectangular) and treatment two (2=acid treatment and 1=control) The total grams of resin collected is the numeric response variable.

1. Preparation

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

```
##      shape           trt            y         
##  Min.   :1.00   Min.   :1.0   Min.   :  9.00  
##  1st Qu.:1.75   1st Qu.:1.0   1st Qu.: 37.25  
##  Median :2.50   Median :1.5   Median : 65.50  
##  Mean   :2.50   Mean   :1.5   Mean   : 58.62  
##  3rd Qu.:3.25   3rd Qu.:2.0   3rd Qu.: 77.25  
##  Max.   :4.00   Max.   :2.0   Max.   :108.00  
```

```
# Check the data types of the variables and adjust if necessary
lapply(pine, class)
```

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

```
pine_f <- with(pine, data.frame(shape=as.factor(shape), trt=as.factor(trt), gramsResin=y))
lapply(pine_f, class)
```

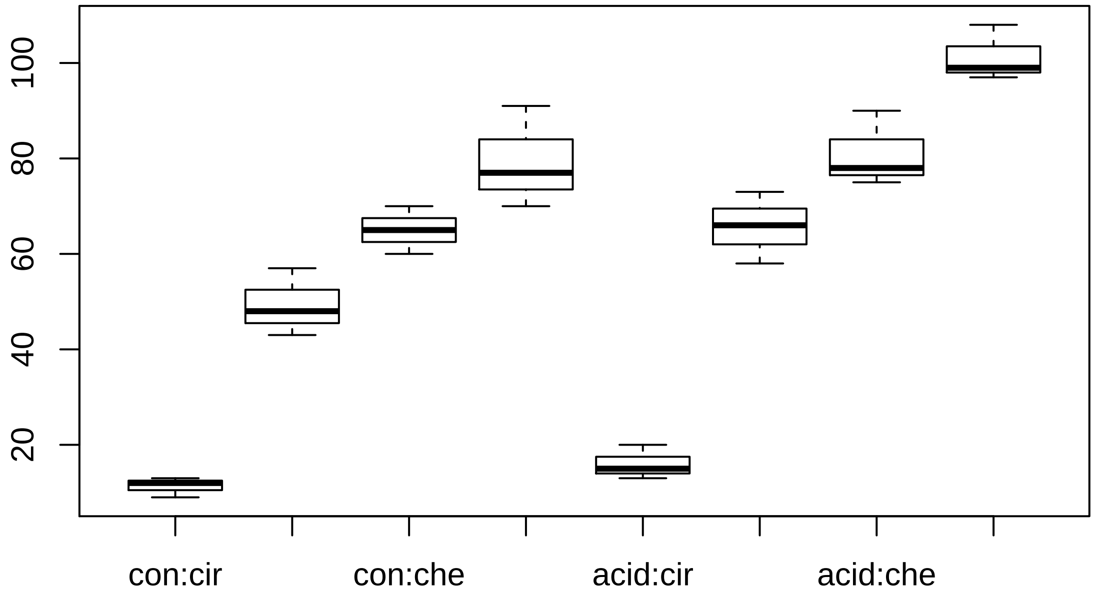
```
## $shape
## [1] "factor"
## 
## $trt
## [1] "factor"
## 
## $gramsResin
## [1] "integer"
```

Since both of the covariants had the data type integer after the import we changed them to factors.

2. Graphical Exploration

To get a better understanding of how the different combinations of covariant levels relate to the grams of resin produced and to see if there are potential outliers, we start the exploration of the data with a boxplot.

```
par(mfrow=c(1,1))
boxplot(gramsResin ~ shape + trt, data=pine_f, names=c("con:cir", "con:dia","con:che","con:rec","acid:c
```




```
# Summarize means in table
with(pine_f, tapply(gramsResin, list(shape,trt), mean))
```

```
##          1         2
## 1 11.33333  16.00000
## 2 49.33333  65.66667
## 3 65.00000  81.00000
## 4 79.33333 101.33333
```

In the boxplot, it appears that there are differnt outcomes for the different shapes as well as for the different treatments. To find out if these differences are significant and if there are interactions will require further analysis.

3. Model fitting and diagnostics

We will start the analysis by fitting the additive model with interactions:

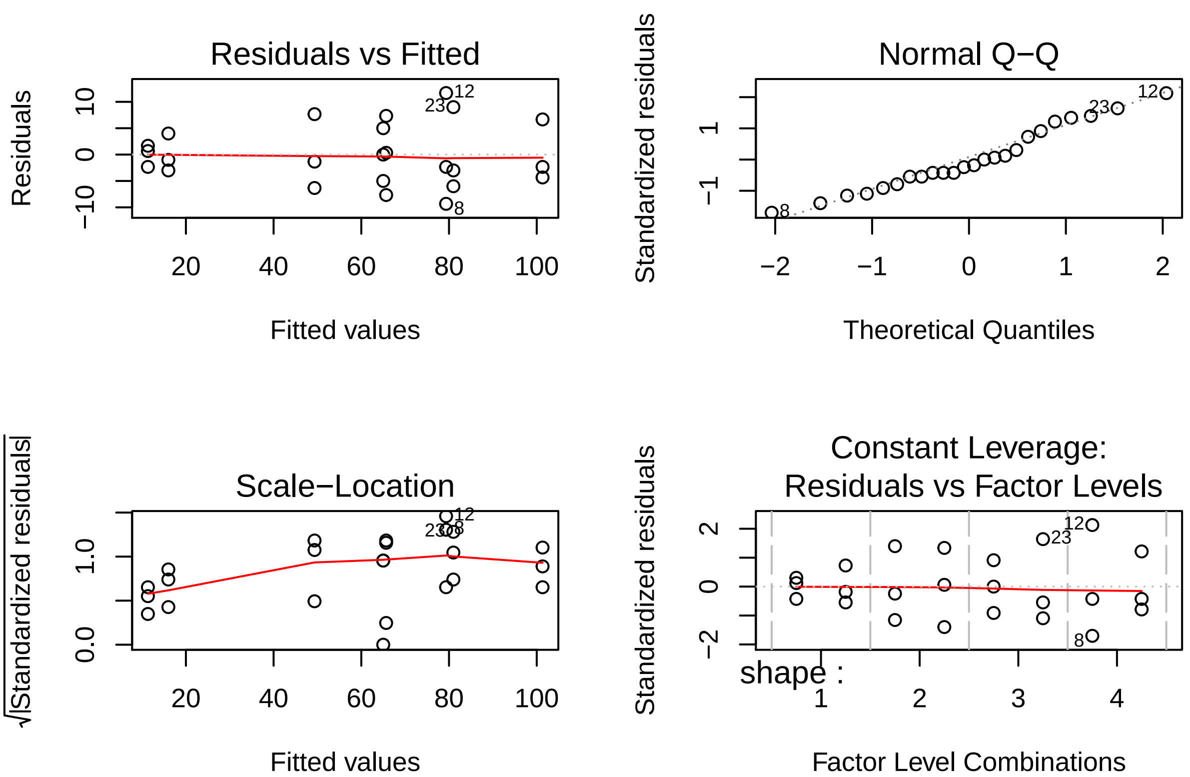
gramsResin ~ shape*trt

```
m1 <- lm(gramsResin~shape*trt, data=pine_f)
summary(m1)
```

```
##
## Call:
## lm.default(formula = gramsResin ~ shape * trt, data = pine_f)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.333 -3.333 -1.167  4.250 11.667
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  58.6250     1.3706  42.774  < 2e-16 ***
## shape1      -44.9583     2.3739 -18.939 2.21e-12 ***
## shape2       -1.1250     2.3739  -0.474   0.6420
## shape3       14.3750     2.3739   6.055 1.67e-05 ***
## trt1         -7.3750     1.3706  -5.381 6.12e-05 ***
## shape1:trt1   5.0417     2.3739   2.124   0.0496 *
## shape2:trt1  -0.7917     2.3739  -0.333   0.7431
## shape3:trt1  -0.6250     2.3739  -0.263   0.7957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.714 on 16 degrees of freedom
## Multiple R-squared:  0.9667, Adjusted R-squared:  0.9522
## F-statistic: 66.39 on 7 and 16 DF,  p-value: 1.241e-10
```

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



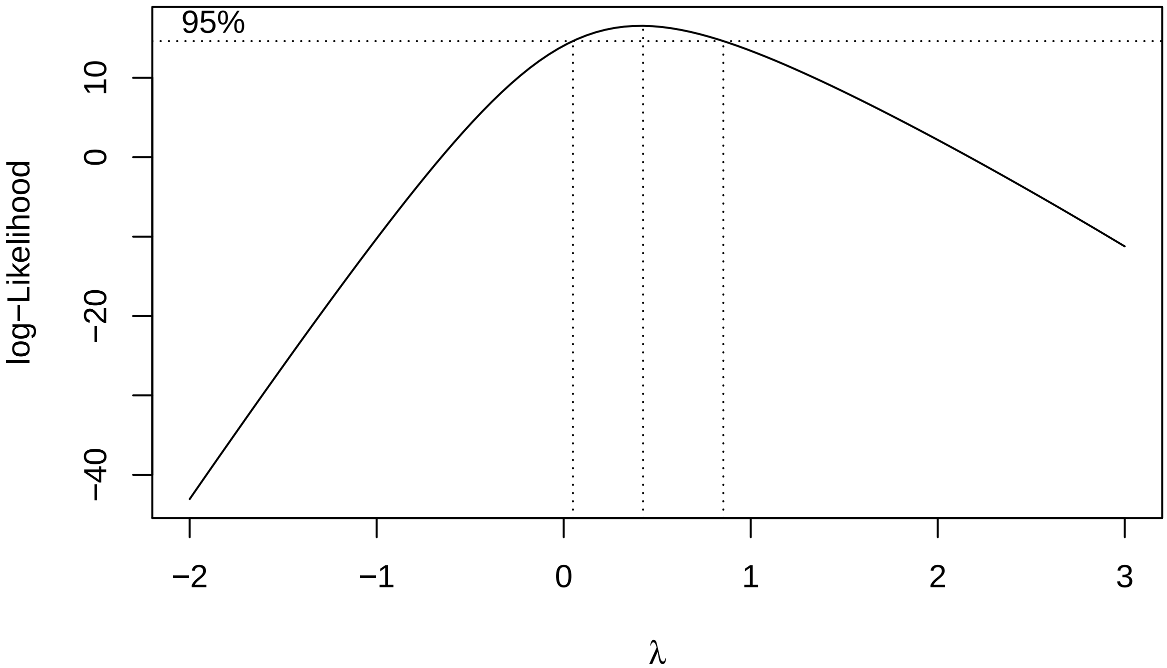
R-squared is high in this model around .96. The constant variance assumption is might not be completely satisfied. The residuals might not be normally distributed. The interaction between circular shape and control appears to be significant at a .95 confidence level.

4. Transformation

A next step is to analyze if there are any transformations that would be particularly adequant for the given data. We cannot transform the regressors since it is a factor, i.e. not really a number. Thus, we explore if a transformation for the response makes sense.

```r
par(mfrow=c(1,1))

# Fit boxcox plot with appropriate range
bc <- boxcox(m1, lambda=seq(-2, 3, 1/10))
```




```r
# Find the max
with(bc, x[which.max(y)])
```

```
## [1] 0.4242424
```

```r
# Based a our findings, let's fit new models and diagnose them:

summary(m2<- lm(log(gramsResin)~shape*trt, data=pine_f))
```

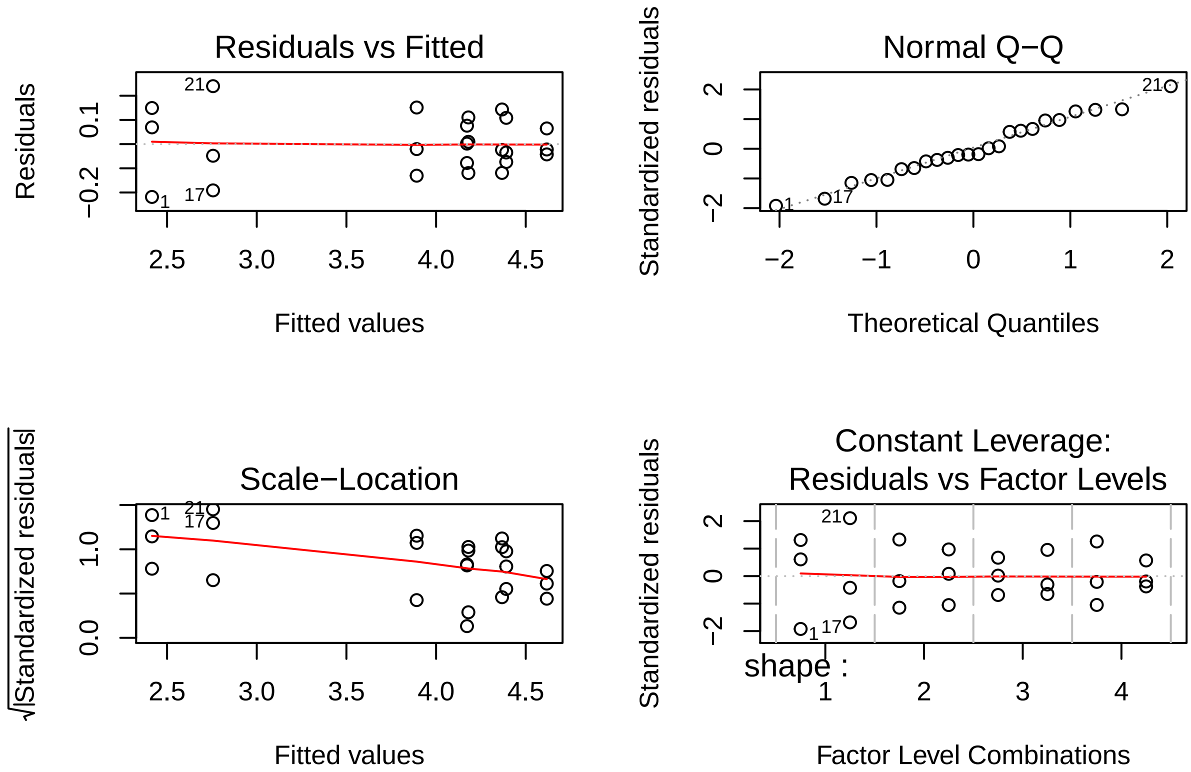
```
##
## Call:
## lm.default(formula = log(gramsResin) ~ shape * trt, data = pine_f)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21847 -0.07490 -0.02141  0.08418  0.23949
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.849091   0.028418 135.447  < 2e-16 ***
## shape1      -1.263122   0.049221 -25.662 1.99e-14 ***
## shape2       0.186911   0.049221   3.797 0.001581 **
## shape3       0.432782   0.049221   8.793 1.59e-07 ***
## trt1        -0.137181   0.028418  -4.827 0.000186 ***
```

```
## shape1:trt1 -0.033094   0.049221  -0.672 0.510943
## shape2:trt1 -0.007003   0.049221  -0.142 0.888631
## shape3:trt1  0.027717   0.049221   0.563 0.581156
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1392 on 16 degrees of freedom
## Multiple R-squared:  0.9781, Adjusted R-squared:  0.9685
## F-statistic: 102.1 on 7 and 16 DF,  p-value: 4.462e-12
```

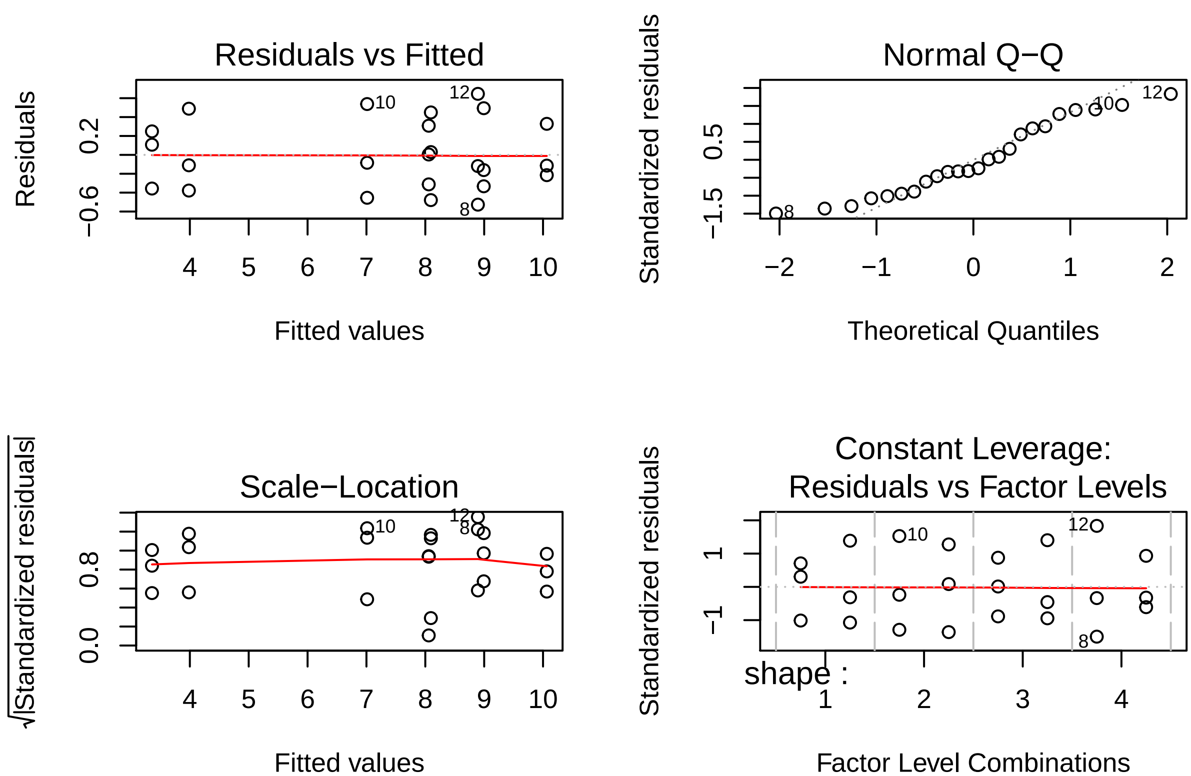
```
summary(m3<- lm((gramsResin^.5)~shape*trt, data=pine_f))
```

```
##
## Call:
## lm.default(formula = (gramsResin^0.5) ~ shape * trt, data = pine_f)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5271 -0.3174 -0.0971  0.3134  0.6457
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.306887   0.088074  82.963  < 2e-16 ***
## shape1      -3.636833   0.152548 -23.841 6.28e-14 ***
## shape2       0.246329   0.152548   1.615    0.126
## shape3       1.218725   0.152548   7.989 5.64e-07 ***
## trt1        -0.476811   0.088074  -5.414 5.74e-05 ***
## shape1:trt1  0.163308   0.152548   1.071    0.300
## shape2:trt1 -0.064579   0.152548  -0.423    0.678
## shape3:trt1  0.009474   0.152548   0.062    0.951
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4315 on 16 degrees of freedom
## Multiple R-squared:  0.9763, Adjusted R-squared:  0.9659
## F-statistic: 94.11 on 7 and 16 DF,  p-value: 8.412e-12
```

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


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
shape :
Factor Level Combinations

```r
plot(m3)
```


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
shape :
Factor Level Combinations

```
# Compare R-squared
compR <-rbind(summary(m1)$r.squared,summary(m2)$r.squared,summary(m3)$r.squared)
rownames(compR)<-c("m1","m2","m3")
colnames(compR)<-c("R^2")
compR
```

```
##          R^2
## m1 0.9667153
## m2 0.9781069
## m3 0.9762892
```

While both of the transformed models have significant differences in the mean effects, neither seems to have significant interactions. From the diagnostics, we can see that there is no significant improvment in satisfying the normality assumption. The constant variance assumption is approx. satisfied in all three models. Multiple R-squared is approx. the same in all three models.

Thus, it is reasonable to go with the first, untransformed model m1 since it is easiest to interprete and understand.

5. ANOVA

Now, we will test if the interaction in this model actually is significant.

```
anova(m1)
```

```
## Analysis of Variance Table
##
## Response: gramsResin
##           Df  Sum Sq Mean Sq  F value    Pr(>F)
## shape      3 19407.5  6469.2 143.4932 8.934e-12 ***
## trt        1  1305.4  1305.4  28.9547 6.122e-05 ***
## shape:trt  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
```

Anova test indicates that the interaction between the two covariants is not significant. Thus, we will choose an additive model without interaction, which will further simplify the model.

6. Refitting the model and diagnostics

The model we fit is:

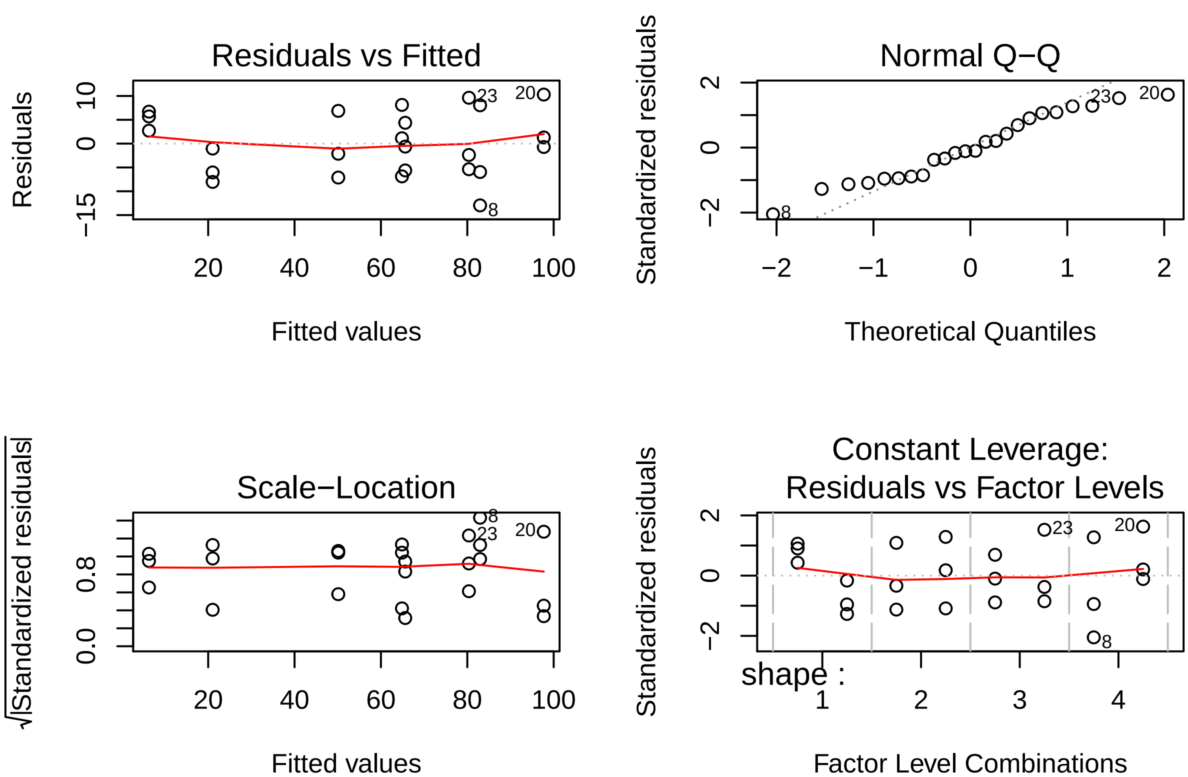
gramsResin ~ shape + trt

```
m0 <- lm(gramsResin~shape+trt, data=pine_f)
summary(m0)
```

```
##
## Call:
## lm.default(formula = gramsResin ~ shape + trt, data = pine_f)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.9583  -5.7083  -0.6667   5.9583  10.2917
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   58.625      1.450  40.430  < 2e-16 ***
## shape1       -44.958      2.512 -17.901 2.37e-13 ***
## shape2        -1.125      2.512  -0.448    0.659
## shape3        14.375      2.512   5.724 1.62e-05 ***
## trt1          -7.375      1.450  -5.086 6.56e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.104 on 19 degrees of freedom
## Multiple R-squared:  0.9558, Adjusted R-squared:  0.9464
## F-statistic: 102.6 on 4 and 19 DF,  p-value: 1.377e-12
```

```r
par(mfrow=c(2,2))
plot(m0)
```



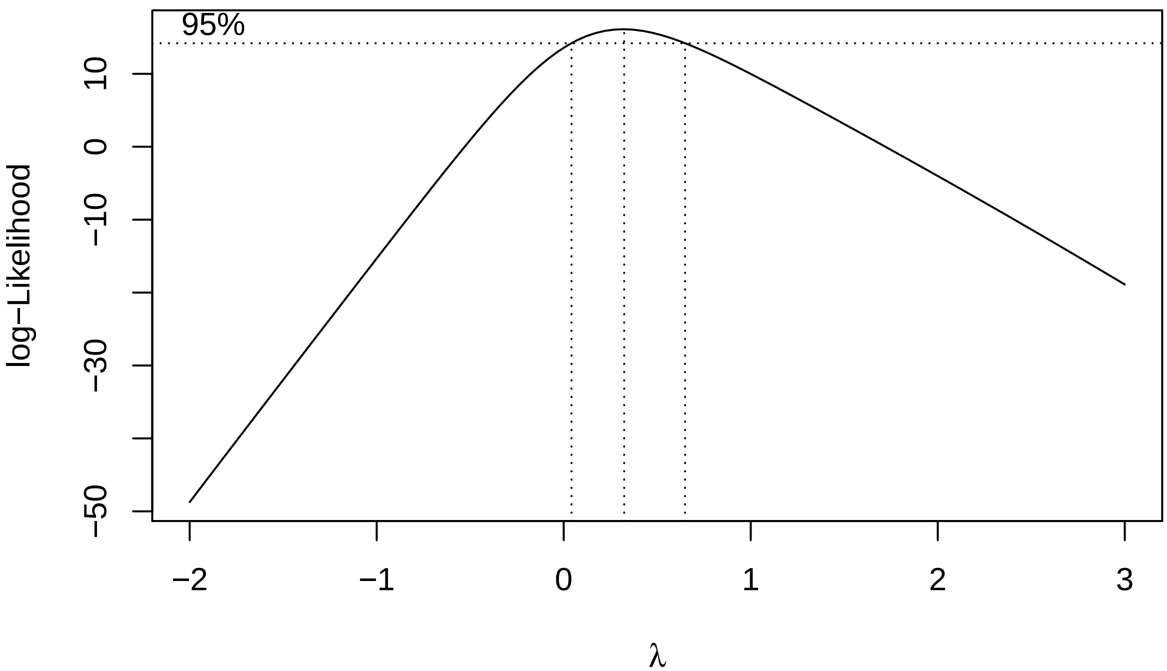
R-squared did not get much lower and is still around .95. The constant variance assumption is probably not completely satisfied. The residuals might also not be normally distributed. The mean effects for the treatment and shapes appear to be significantly diffent.

7. Transformations

A next step is to analyze if there are any transformations that would be particularly adequant for the given data and help us to better satisfy the assumptions. We cannot transform the regressors since it is a factor, i.e. not really a number. Thus, we explore if a transformation for the response makes sense.

```r
par(mfrow=c(1,1))

# Fit boxcox plot with appropriate range
bc <- boxcox(m0, lambda=seq(-2, 3, 1/10))
```




```r
# Find the max
with(bc, x[which.max(y)])
```

```
## [1] 0.3232323
```

```r
# Based a our findings, let's fit new models and diagnose them:

summary(m4<- lm(log(gramsResin)~shape+trt, data=pine_f))
```

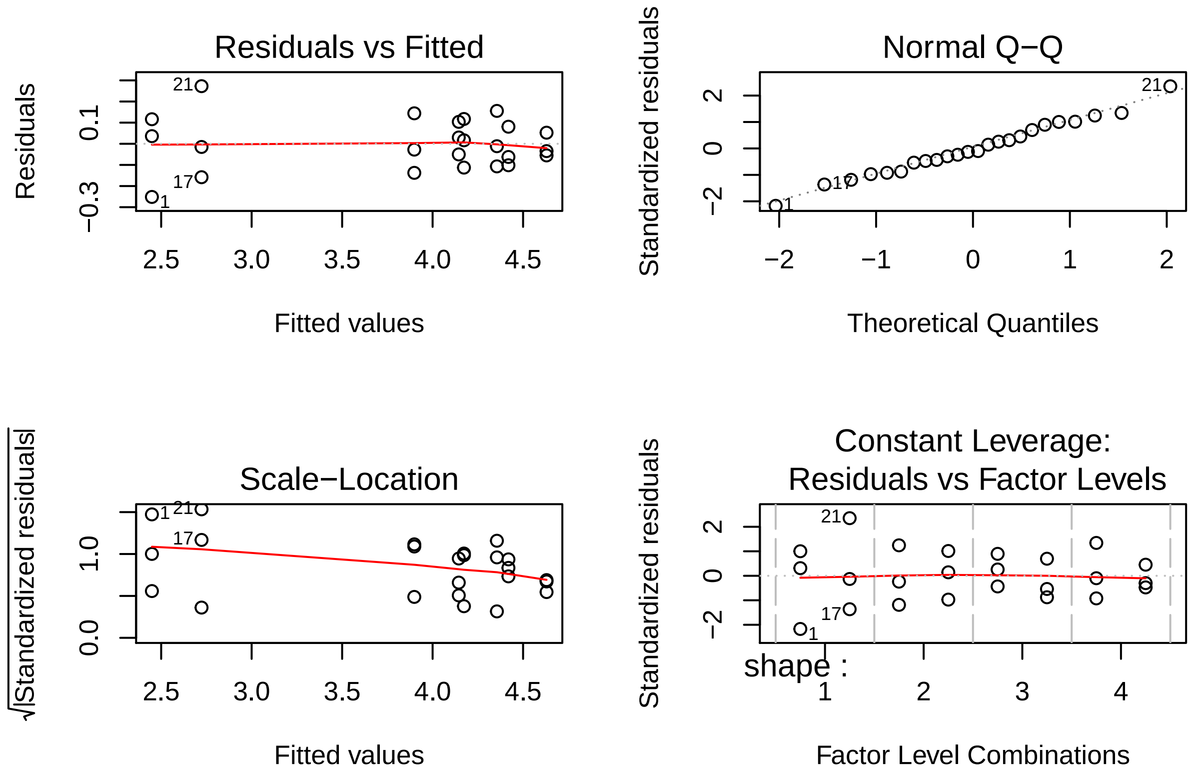
```
##
## Call:
## lm.default(formula = log(gramsResin) ~ shape + trt, data = pine_f)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25156 -0.07215 -0.01332  0.08652  0.27258
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.84909    0.02659 144.736  < 2e-16 ***
```

```
## shape1      -1.26312    0.04606 -27.422  < 2e-16 ***
## shape2       0.18691    0.04606   4.058 0.000671 ***
## shape3       0.43278    0.04606   9.396 1.42e-08 ***
## trt1        -0.13718    0.02659  -5.158 5.59e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1303 on 19 degrees of freedom
## Multiple R-squared:  0.9772, Adjusted R-squared:  0.9724
## F-statistic: 203.9 on 4 and 19 DF,  p-value: 2.551e-15
```

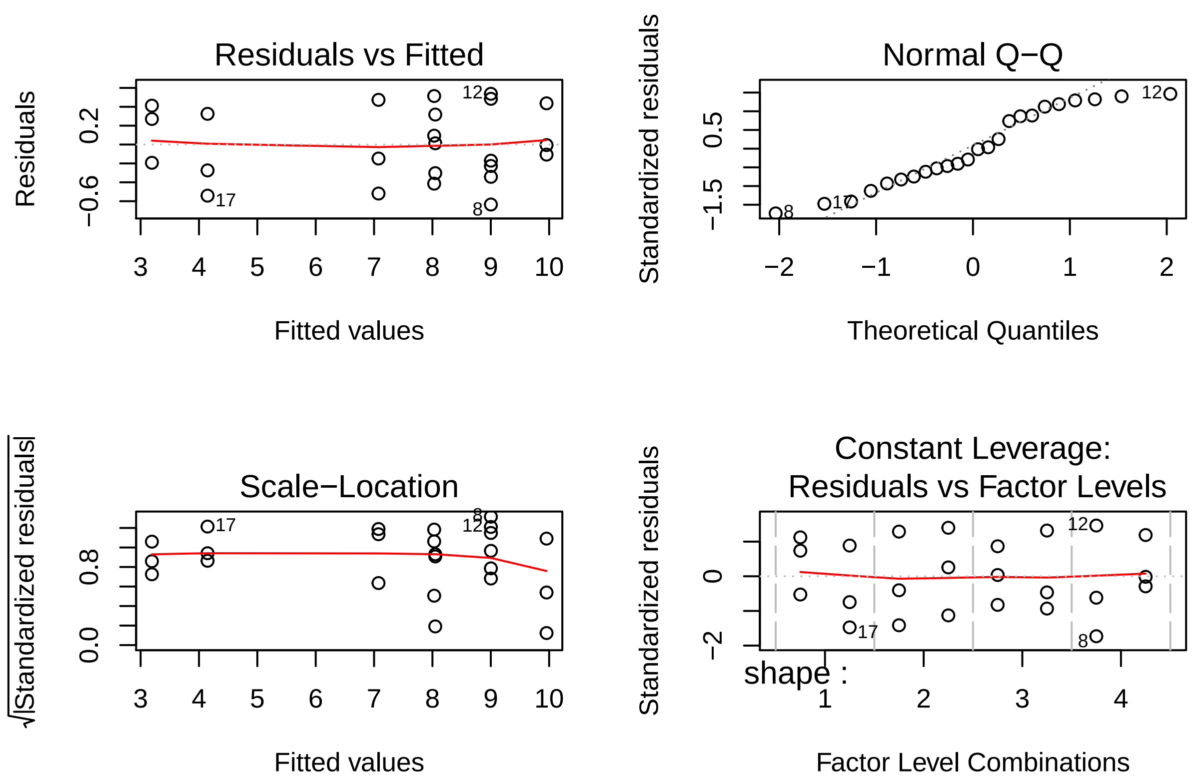
```
summary(m5<- lm((gramsResin^.5)~shape+trt, data=pine_f))
```

```
##
## Call:
## lm.default(formula = (gramsResin^0.5) ~ shape + trt, data = pine_f)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.63525 -0.28112 -0.05611  0.34703  0.53754
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.30689    0.08422  86.758  < 2e-16 ***
## shape1      -3.63683    0.14588 -24.931 5.61e-16 ***
## shape2       0.24633    0.14588   1.689    0.108
## shape3       1.21873    0.14588   8.355 8.75e-08 ***
## trt1        -0.47681    0.08422  -5.661 1.86e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4126 on 19 degrees of freedom
## Multiple R-squared:  0.9743, Adjusted R-squared:  0.9688
## F-statistic: 179.7 on 4 and 19 DF,  p-value: 8.183e-15
```

```
par(mfrow=c(2,2))
plot(m4)
```


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
shape :
Factor Level Combinations

```r
plot(m5)
```


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
shape :
Factor Level Combinations

```r
# Compare R-squared
compR <-rbind(summary(m0)$r.squared,summary(m4)$r.squared,summary(m5)$r.squared)
rownames(compR)<-c("m0","m4","m5")
colnames(compR)<-c("R^2")
compR
```

```
##          R^2
## m0 0.9557582
## m4 0.9772318
## m5 0.9742528
```

The model with the log-transformed response variable, m4, appears to satisfy the constant variance and normality assumption well. Moreover, all mean effects appear to be significnat different from each other in this model and it has the best model fit of all the additive model without interactions that I explored. Thus, it is reasonable to go with the m4.

8. Inference

After having choosen the model, now it's time to explore if the mean effects for shape and particularly treatments are actually differnt and how they differ based on a pairwise comparison.

```r
anova(m4)
```

```
## Analysis of Variance Table
##
## Response: log(gramsResin)
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## shape      3 13.3903  4.4634 262.963 1.207e-15 ***
## trt        1  0.4516  0.4516  26.609 5.589e-05 ***
## Residuals 19  0.3225  0.0170
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
linear.contrast(m4, trt, allpairs=T, confidence=.95)
```

```
##        estimates         se   t-value      p-value   lower-ci   upper-ci
## 1 - 2 -0.2743615 0.05318773 -5.15836 5.588531e-05 -0.3856847 -0.1630383
```

9. Interpretation

The ANOVA output indicates there is a difference of resign yield for the different shapes and the treatment. Since we are mainly interested on the impact of the acid treatment in this problem, I did a comparison of the mean difference between the control and treatment group over all shapes. From this comparison, we can conclude that the treatment increases the resin yield significantly and reject the null hypothesis that it doesn't. In numbers, it appears that on average the yield without treatment is 27% lower than with the treatment.