Homework 2

Nick Sun

October 8, 2019

Question 1

Part a.

We can use the `aov()` function to solve this.

```r
summary(aov(V3 ~ as.factor(V1) + as.factor(V2), data = corn))
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)  
## as.factor(V1)  3   6360  2120.0   4.076 0.0125 *
## as.factor(V2)  2   1288   644.2   1.238 0.3002  
## Residuals     42  21846   520.2                 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The sums of squares associated with the blocks is 1288, the sums of squares associated with nitrogen is 6360, and the sums of squares associated with departures in additivity is 21846.

Part b.

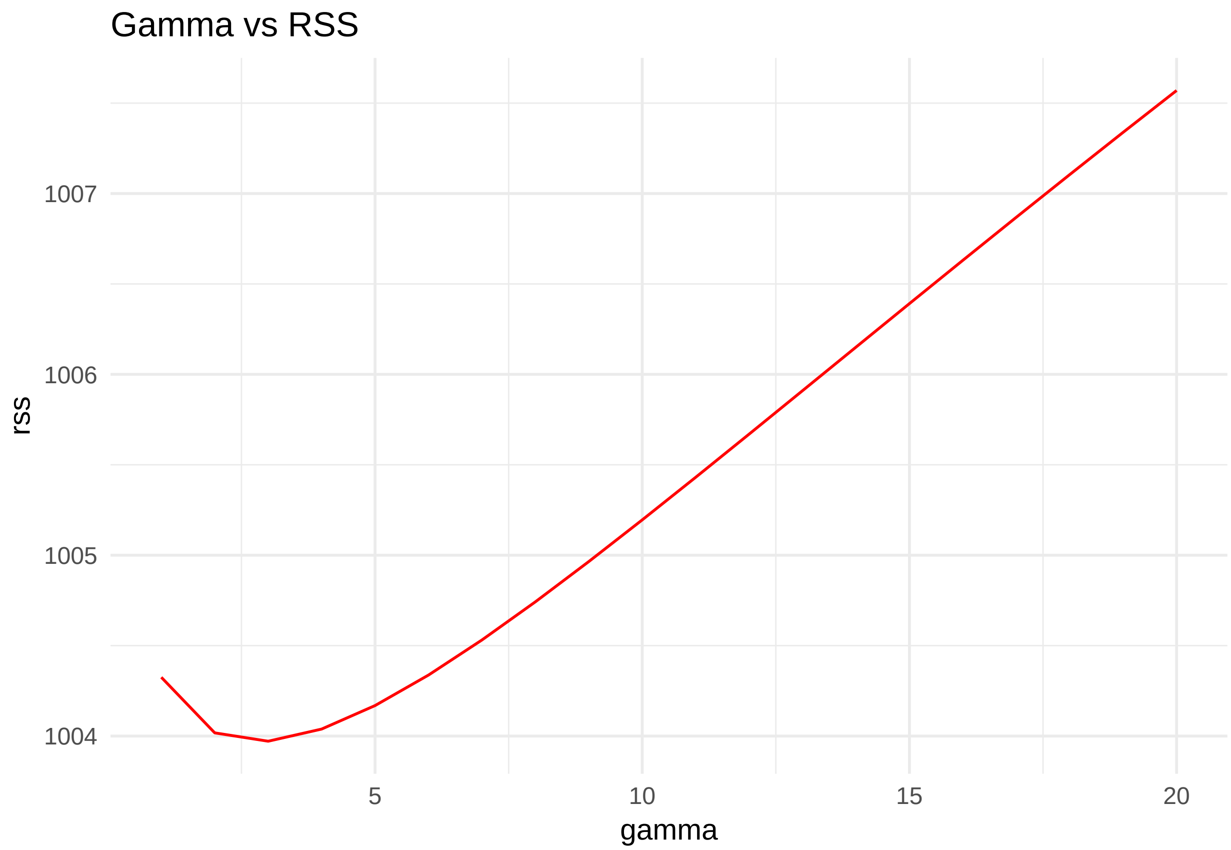
According to our test, no, blocking has not been effective. The p-value for the blocking factor V2 is not significant, indicating that there is no significant difference between the observed yields of the different blocks. We could spend the degrees of freedom associated with blocking on our residuals instead.

That being said, there may be experimentally advantageous reasons to block, for example, if the blocks are in different geographic locations and the researchers need to incorporate this into their model. That might be a case where practical significance outweighs statistical significance.

Part c.

```r
nit_adjusted <- function(gamma) {
  model <- lm(V3 ~ as.factor(V2) + I(log(corn$V1 + gamma)), data = corn)
  output <- 45*(summary(model)$sigma)
  return(output)
}

gamma <- 1:20
rss <- vector(mode = "numeric", length = length(gamma))
for (g in gamma) {
  rss[g] <- nit_adjusted(g)
}

gamma_rss <- data.frame(gamma, rss)
```

```r
ggplot(gamma_rss) +
  geom_line(mapping = aes(x = gamma,
                          y = rss),
            color = "red") +
  labs(
    title = "Gamma vs RSS"
  ) +
  theme_minimal()
```

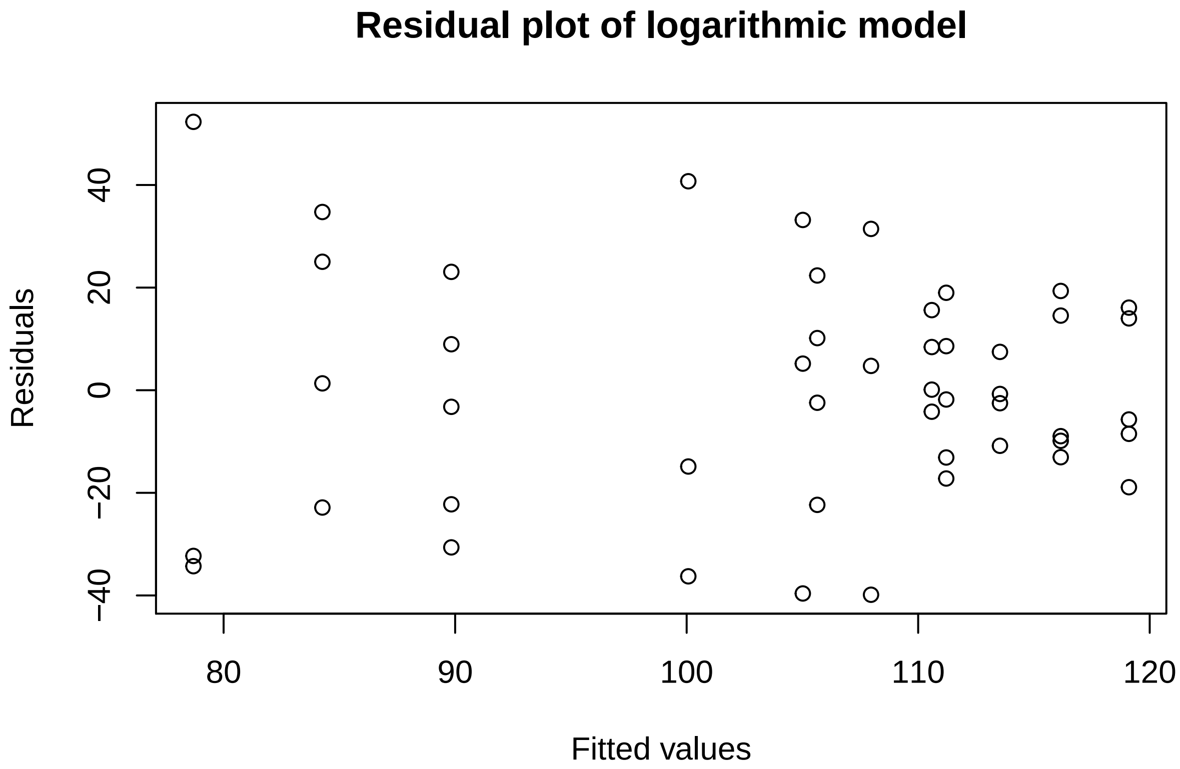


The value of γ with the smallest RSS appears to be 3. This will be our least squares estimate.

Computing a 95% Confidence interval for γ will have to involve either finding an MLE and using the asymptotic distribution of the MLE to create a Wald confidence interval **or** using some form of bootstrapping.

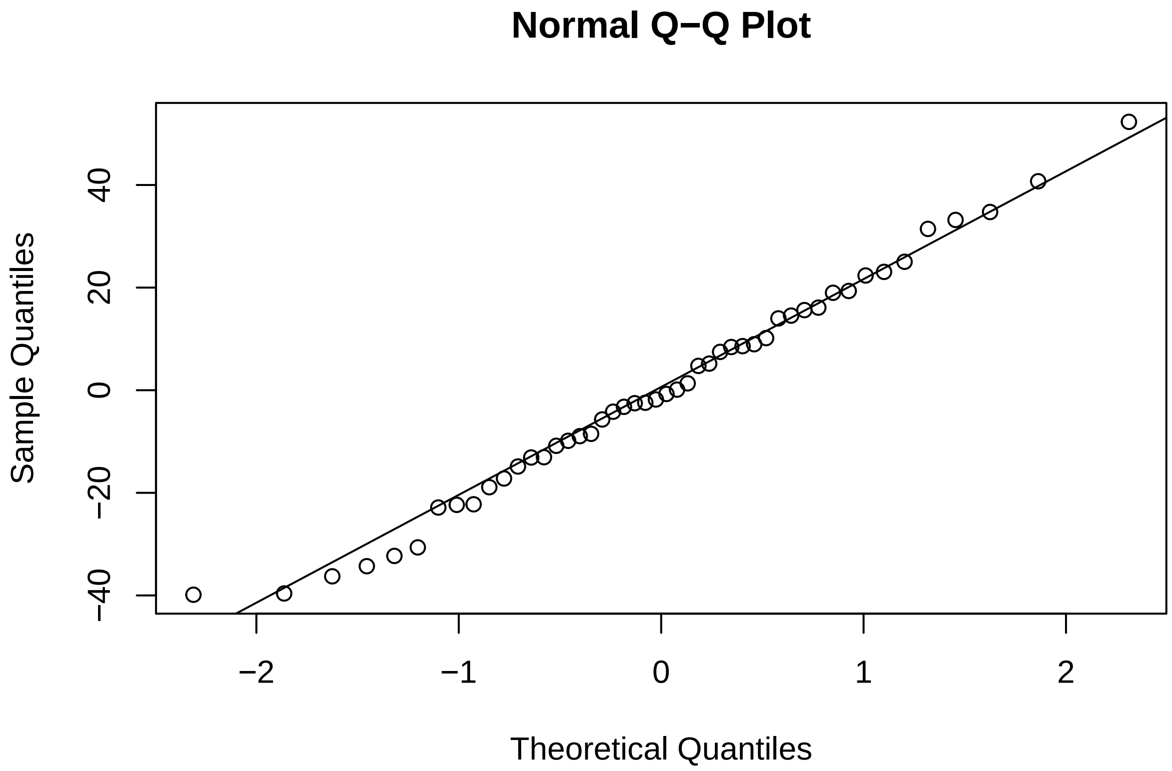
Part d.

```r
gamma <- 3
model <- lm(V3 ~ V2 + I(log(corn$V1 + gamma)), data = corn)
plot(model$fitted.values, model$residuals,
     main = "Residual plot of logarithmic model",
     xlab = "Fitted values",
     ylab = "Residuals")
```



There appears to be some heteroskedasticity here.

```
qqnorm(model$residuals)
qqline(model$residuals)
```



The errors however do appear to be roughly normal. I would suggest that this tells us the logarithmic function isn't a close fit to the actual data, but it isn't awful.

An alternative approach to checking diagnostic plots is to use a logarithmic lack of fit test where we compare the fit of the logarithmic model to an oversaturated model where each $\hat{y}$ is just the group mean for that treatment combination (blocks x nitrogen).

```
saturated_model <- lm(V3 ~ factor(V1)*factor(V2),
                      data = corn)
summary(saturated_model)
```

```
##
## Call:
## lm(formula = V3 ~ factor(V1) * factor(V2), data = corn)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.200 -12.515  -0.887  14.255  57.067
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  73.9333    14.1143   5.238 7.24e-06 ***
## factor(V1)50                 22.6667    19.9607   1.136    0.264
## factor(V1)100                30.6667    19.9607   1.536    0.133
## factor(V1)150                32.8000    19.9607   1.643    0.109
## factor(V2)1                  19.8917    18.6715   1.065    0.294
## factor(V2)2                11.0867    17.8534   0.621    0.539
## factor(V1)50:factor(V2)1   -8.9167    26.4055  -0.338    0.738
## factor(V1)100:factor(V2)1  -8.9167    26.4055  -0.338    0.738
## factor(V1)150:factor(V2)1 -14.7500    26.4055  -0.559    0.580
## factor(V1)50:factor(V2)2    2.6133    25.2485   0.104    0.918
## factor(V1)100:factor(V2)2   0.8733    25.2485   0.035    0.973
## factor(V1)150:factor(V2)2   0.6800    25.2485   0.027    0.979
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.45 on 36 degrees of freedom
## Multiple R-squared:  0.2705, Adjusted R-squared:  0.04765
## F-statistic: 1.214 on 11 and 36 DF,  p-value: 0.3134
```

```
anova(saturated_model, model)
```

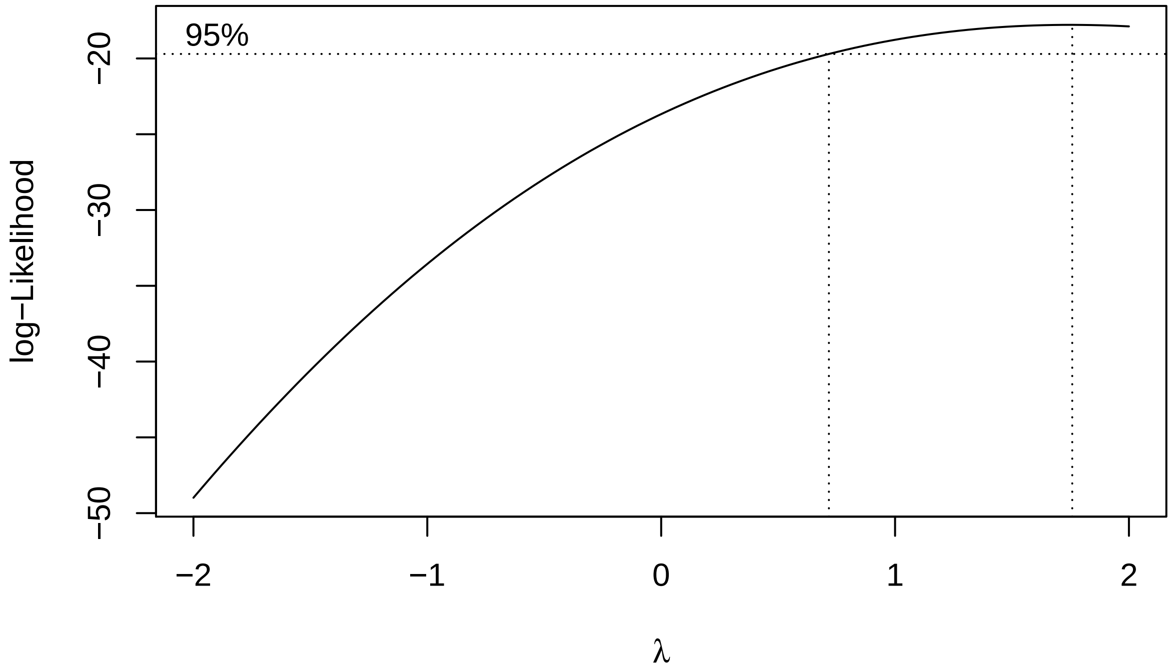
```
## Analysis of Variance Table
##
## Model 1: V3 ~ factor(V1) * factor(V2)
## Model 2: V3 ~ V2 + I(log(corn$V1 + gamma))
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     36 21515
## 2     45 22238 -9   -722.96 0.1344 0.9984
```

This high p-value indicates that we have no strong evidence of lack of fit!

Part e.

Using the `boxcox` function in R, we can identify the response transformation that gives us the greateset log-likelihood.

```
corncox <- boxcox(V3 ~ V2 + I(log(corn$V1 + gamma)),
       data = corn,
       plotit = TRUE)
```

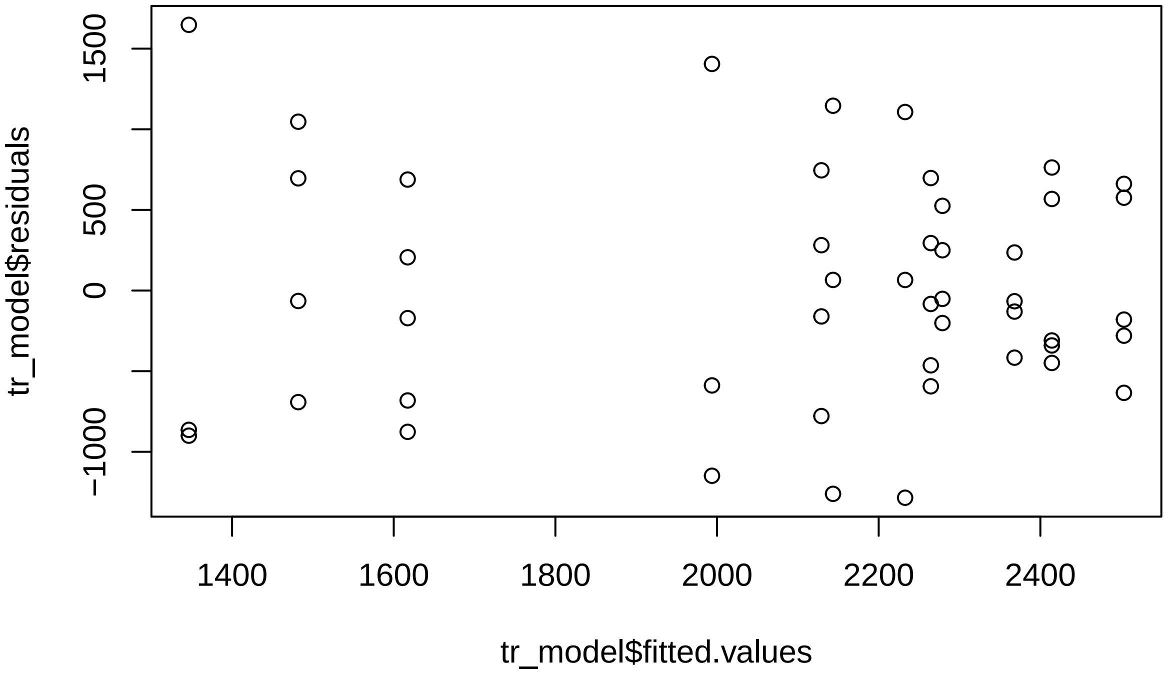


```
lambda <- with(corncox, x[y == max(y)])
lambda
```

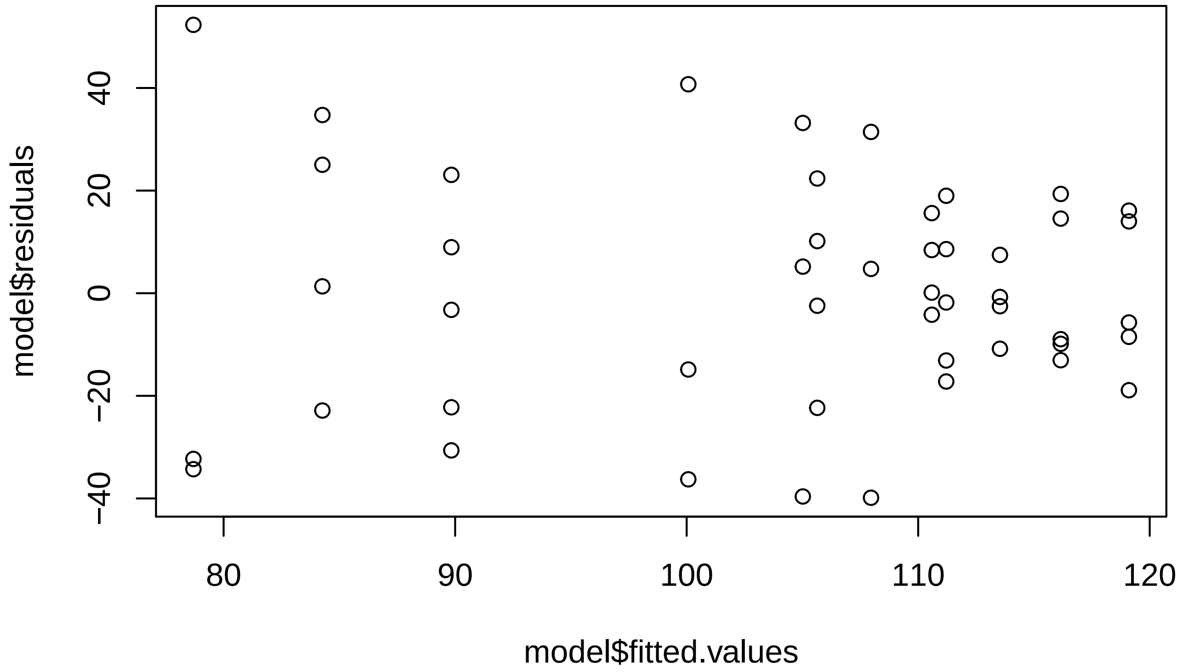
```
## [1] 1.757576
```

Applying this transformation to our response and refitting the model gives us the following probabilities.

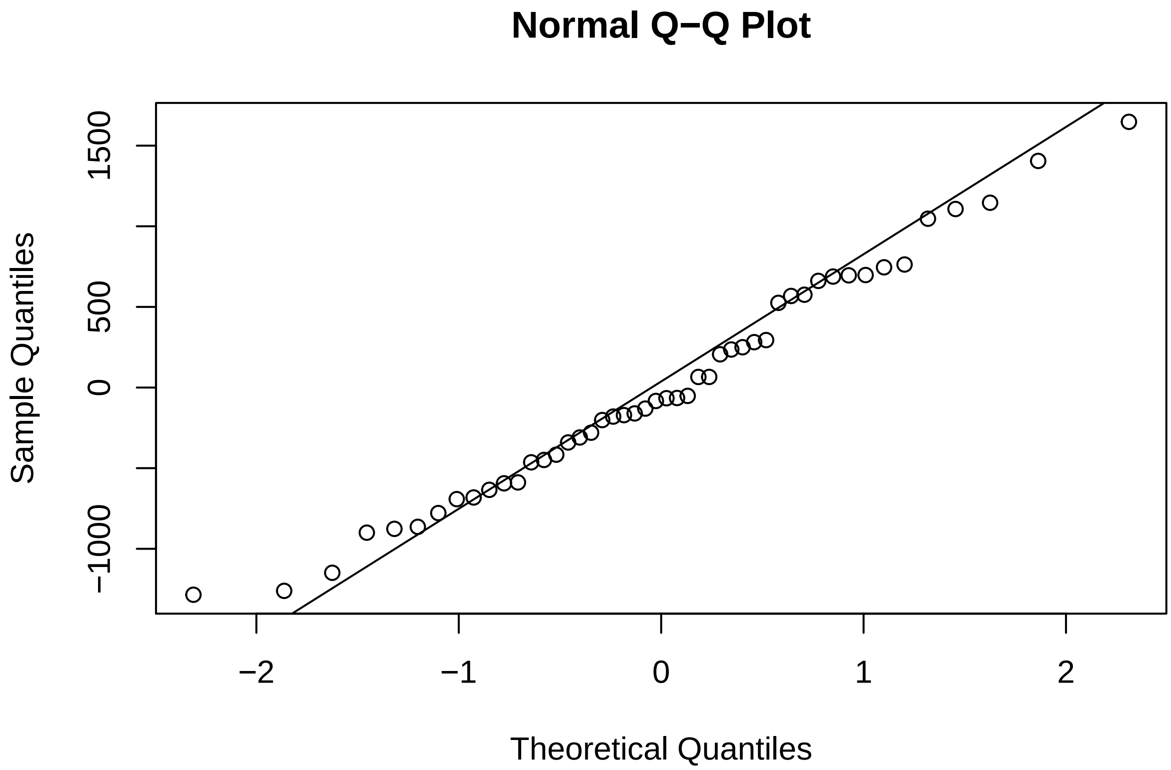
```
corn$tr_V3 <- (corn$V3^lambda - 1)/lambda

tr_model <- lm(tr_V3 ~ V2 + I(log(corn$V1 + gamma)),
               data = corn)

plot(tr_model$fitted.values,
     tr_model$residuals)
```



```
plot(model$fitted.values,
     model$residuals)
```



```
qqnorm(tr_model$residuals)
qqline(tr_model$residuals)
```



The transformation actually made everything worse!

Part f.

Blocking was not effective, so in later experiments it might be better to not include it as an effect and have additional error degrees of freedom. The log transformation performed the best of the models we tried, noting that even then there is evidence of nonconstant variance. However, unless we are trying to make predictions, this heteroskedasticity should be fine.

Question 2

```
temp <- c(rep(0, 4),
          rep(10, 4),
          rep(20, 4))

weeks <- c(rep(c(2,4,6,8), 3))

beans <- data.frame(temp,
           weeks,
           con = c(45, 47, 46, 46,
                   45, 43, 41, 37,
                   34, 28, 21, 16))
beans
```

```
##    temp weeks con
## 1     0     2  45
## 2     0     4  47
## 3     0     6  46
## 4     0     8  46
## 5    10     2  45
## 6    10     4  43
## 7    10     6  41
## 8    10     8  37
## 9    20     2  34
## 10   20     4  28
## 11   20     6  21
## 12   20     8  16
```

The regression model here is

$$y_{t,T} = e^{-\alpha-\beta_T t+\epsilon_{t,T}}, \epsilon_{t,T} \sim$$

where $y_{t,T}$ is the concentration of ascorbic acid at time t with temperature T, β_T is the interaction effect of temperature T and storage time, and e^{α} is the intial concentration.

The score equations for this model are provided on a separate sheet of paper.

```r
log_model <- glm(con ~ factor(temp):weeks,
                 data = beans,
                 family = gaussian(link="log"))
summary(log_model)
```

```
## 
## Call:
## glm(formula = con ~ factor(temp):weeks, family = gaussian(link = "log"), 
##     data = beans)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -1.49820  -0.38546   0.08059   0.80866   0.86028  
## 
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)          3.8354389  0.0186464 205.694 3.49e-16 ***
## factor(temp)0:weeks  -0.0007716  0.0037540  -0.206 0.842286    
## factor(temp)10:weeks -0.0236121  0.0040351  -5.852 0.000382 ***
## factor(temp)20:weeks -0.1329785  0.0061597 -21.588 2.23e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for gaussian family taken to be 1.124727)
## 
##     Null deviance: 1226.9167  on 11  degrees of freedom
## Residual deviance:    8.9978  on  8  degrees of freedom
## AIC: 40.599
## 
## Number of Fisher Scoring iterations: 3
```

Part b.

We will use these two facts to construct a confidence interval:

$$\begin{aligned}\frac{1}{2}e^{\alpha} &= e^{\alpha-\beta_i h}\\ \frac{1}{2} &= e^{-\beta_T t}\\ t &= \frac{ln(2)}{\beta_T}\end{aligned}$$

and that $\frac{\hat{beta_i}-\beta_i}{SE(\beta_i)} \sim t_8$.

We can construct a $(1-\alpha)100\%$ confidence interval for each β_T, $T \in (0, 10, 20)$:

$\left(\hat{\beta_T} - 2.306SE(\hat{\beta_T}), \hat{\beta_T} + 2.306SE(\hat{\beta_T})\right)$

Noting that for each T, t is just a function of β_T, we can get $\hat{t} = \frac{ln(2)}{\hat{\beta_T}}$ and a $(1-\alpha)100\%$ CI as:

$$\left(\frac{ln(2)}{\hat{\beta_T} + 2.306SE(\hat{\beta_T})}, \frac{ln(2)}{\hat{\beta_T} - 2.306SE(\hat{\beta_T})}\right)$$

Plugging in the values from the model summary, we get that $\hat{t_0} = 898.324$, $\hat{t_{10}} = 29.356$, $\hat{t_{20}} = 5.212$