A3

Aly Abelwahed, Manish Suresh

08/03/2021

Question 1

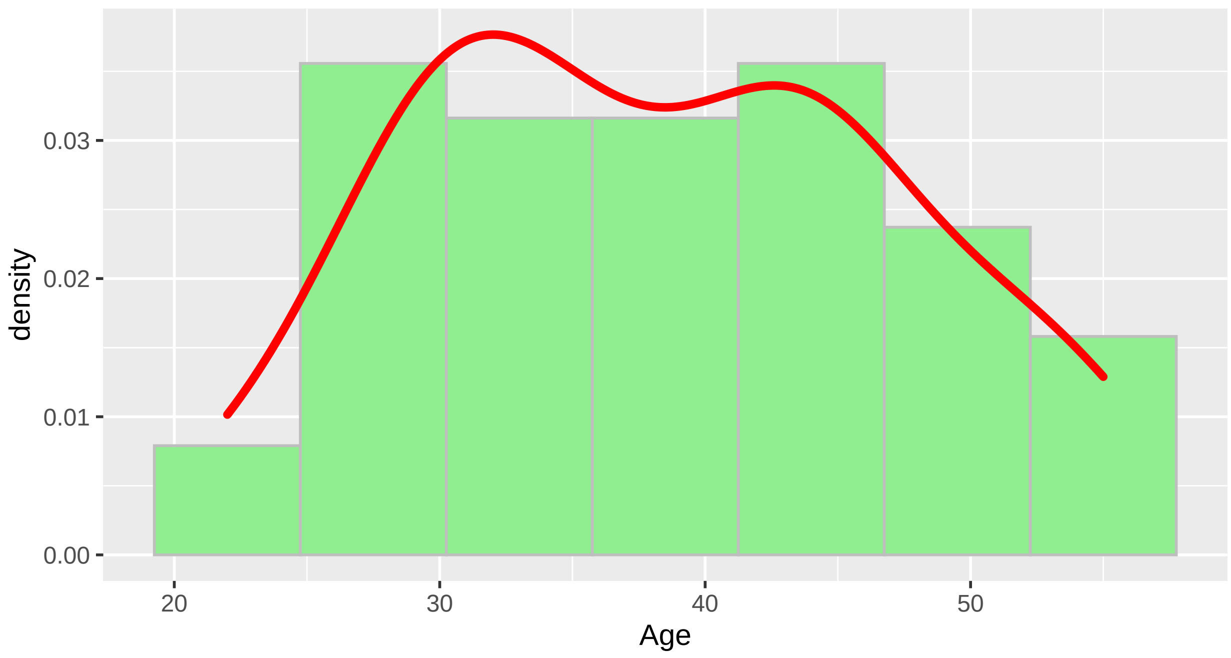
Part a

```
PatientSatisfactionData <- read_table("PatientSatisfaction.txt",
                                      col_names = c("Satisfaction", "Age",
                                                    "Illness_Severity", "Anxiety_Lvl"))
```

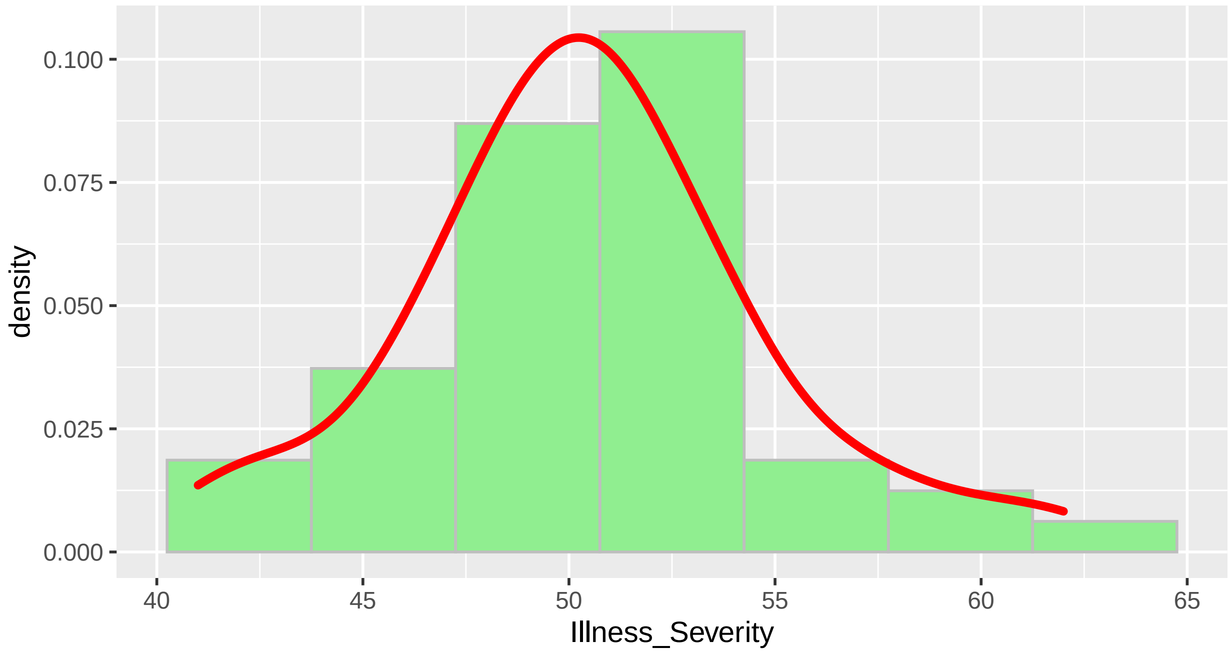
```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Satisfaction = col_double(),
##   Age = col_double(),
##   Illness_Severity = col_double(),
##   Anxiety_Lvl = col_double()
## )
```

Histogram of the predictor variables

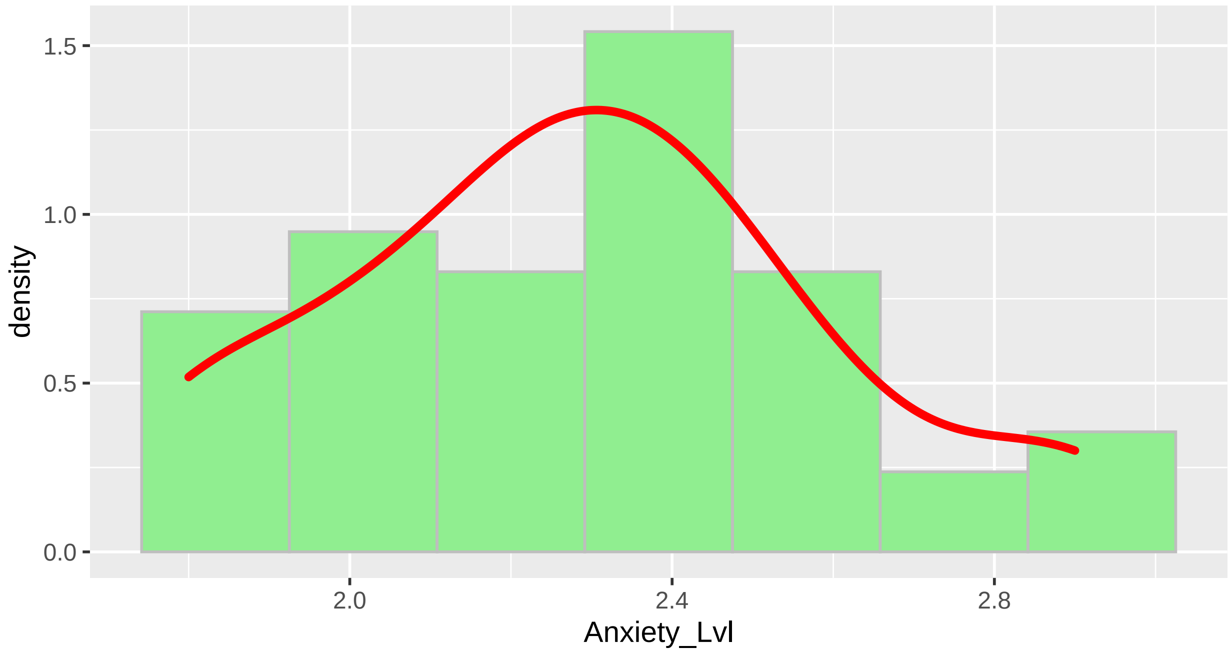
```
ggplot(PatientSatisfactionData, aes(x = Age, y = ..density..)) +
  geom_histogram(bins = 7, color = "grey", fill = "lightgreen") +
  geom_density(color = "Red", size = 1.5)
```



```r
ggplot(PatientSatisfactionData, aes(x = Illness_Severity, y = ..density..)) +
  geom_histogram(bins = 7, color = "grey", fill = "lightgreen") +
  geom_density(color = "Red", size = 1.5)
```



```r
ggplot(PatientSatisfactionData, aes(x = Anxiety_Lvl, y = ..density..)) +
  geom_histogram(bins = 7, color = "grey", fill = "lightgreen") +
  geom_density(color = "Red", size = 1.5)
```

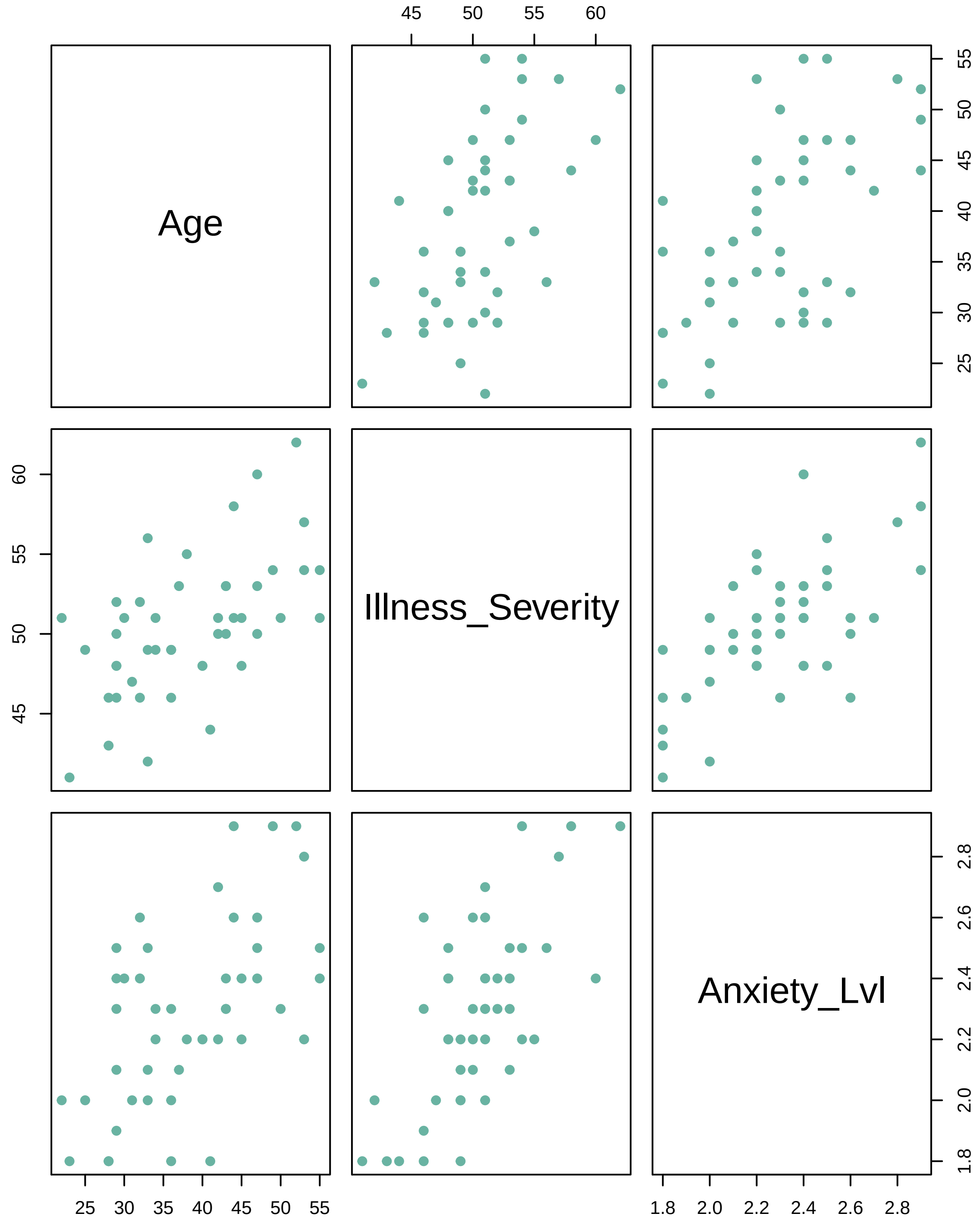


Noteworthy features revealed by these plots:

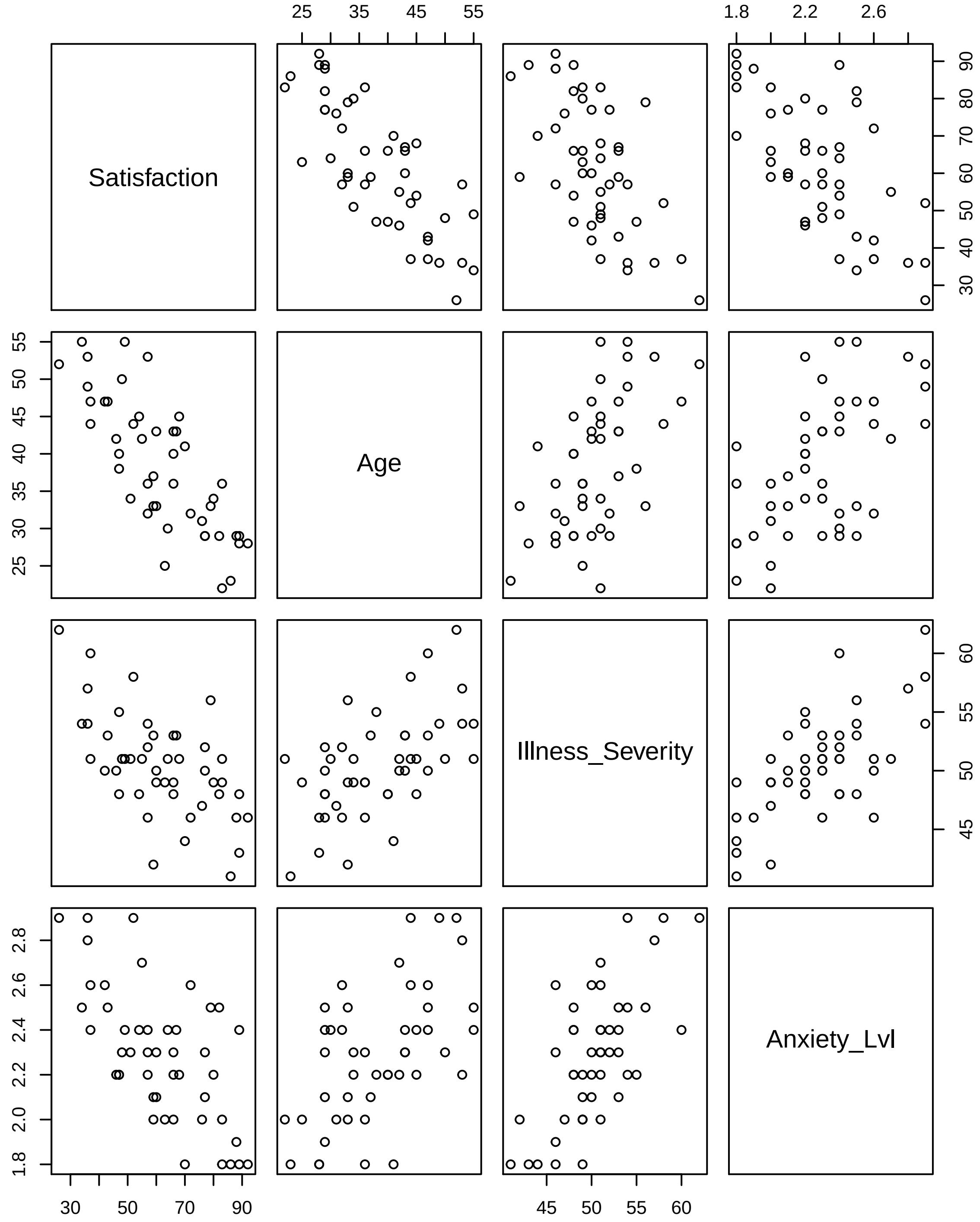
- Illness_severity is normally distributed and is slightly skewed to the right
- Age is normally distributed but is very largely spread out at the middle and it's also bimodal
- Anxiety_Lvl is almost entirely normal

Part b

```
pairs(with(PatientSatisfactionData, cbind(Age, Illness_Severity, Anxiety_Lvl)), pch=20 , cex=1.5 , col=
```



```
pairs(PatientSatisfactionData)
```



```
cor(with(PatientSatisfactionData, cbind(Age, Illness_Severity, Anxiety_Lvl)))
```

```
##                        Age Illness_Severity Anxiety_Lvl
## Age              1.0000000        0.5679505   0.5696775
## Illness_Severity 0.5679505        1.0000000   0.6705287
## Anxiety_Lvl      0.5696775        0.6705287   1.0000000
```

From the correlation matrix above, there is no definite cause for concern about multicollinearity as there are no variables that are highly correlated. One thing to be aware of is the correlation between Anxiety_Lvl and Illness_Severity as the correlation between them is 0.6705287 which is close to being considered as highly correlated but not quite there yet as it's not greater than 0.7. But other than that, there are no other causes of concern about multicollinearity.

Part c

```r
# Perform the Regression Fit
patient.regression <- lm(Satisfaction ~ Age + Illness_Severity + Anxiety_Lvl, PatientSatisfactionData)

# Display info about regression
patient.regression.coeffcients <- tidy(patient.regression)
patient.regression.coeffcients
```

```
## # A tibble: 4 x 5
##   term             estimate std.error statistic  p.value
##   <chr>               <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)       158.       18.1       8.74  5.26e-11
## 2 Age                -1.14      0.215    -5.31  3.81e- 6
## 3 Illness_Severity   -0.442     0.492    -0.898 3.74e- 1
## 4 Anxiety_Lvl       -13.5       7.10     -1.90  6.47e- 2
```

The estimated regression function is $Y = 158.491 - 1.142\,X_1 - 0.442\,X_2 - 13.47\,X_3$

In this case, $\hat{\beta}_2$ is interpreted as follows: For every unit for increase in satisfaction the illness_severity reduces by -0.442 provided that the other prediction variables are held constant

Part d

To test whether there is a regression relation, we perform the hypothesis testing.
The Null hypothesis is

$$H_0 : \beta_1 = \beta_2 = \beta_3 = 0$$

The alternative is

$$H_a : \text{At least one of } \beta_1, \beta_2, \beta_3 \text{ is not } 0$$

The test statistic is

$$F^* = \frac{MSR}{MSE}$$

The decision rule is

$$- \text{ Reject } H_0 \text{ if } F^* > F_{1-\alpha';\, p'-1, n-p'} - \text{ Do not Reject } H_0 \text{ if } F^* < F_{1-\alpha';\, p'-1, n-p'}$$

where $F_{1-\alpha';\, p'-1, n-p'}$ is the $1 - \alpha$ percentile of a $F(p' - 1,\, n - p')$ distribution

Basic Information

```r
# Store the Design Matrix and the response vector
X <- cbind(1,data.matrix(PatientSatisfactionData)[,2:4])
Y <- as.vector(data.matrix(PatientSatisfactionData)[,1])

n <- dim(PatientSatisfactionData)[1]
p.prime<- dim(PatientSatisfactionData)[2]
```

Simple Method: We can read the test statistic and P-Value from the Summary of the regression model

```
sum.patient.regression <- summary(patient.regression)
F.star <- sum.patient.regression$fstatistic[1]

p.value <- pf(F.star, p.prime - 1, n - p.prime, lower.tail = FALSE)

rejection <- abs(F.star) > qf(0.9, p.prime - 1, n - p.prime)

rejection
```

```
## value
##  TRUE
```

Calculating by hand

Plugging in the values

```
sum.Y.Squared <- t(Y) %*% Y
J <- matrix(rep(1,n), ncol = n, nrow = n)
H <- X %*% solve(t(X) %*% X) %*% t(X)
I <- diag(rep(1,n))

SST <- t(Y) %*% Y - 1/n * t(Y) %*% J %*% Y
SSR <- t(Y) %*% (H - 1/n * J) %*% Y
SSE <- t(Y) %*% (I - H) %*% Y

MSR <- SSR / 3
MSE <- SSE / (n - 4)
F.star.2 <- MSR[1]/MSE[1]
p.value.2 <- pf(F.star, p.prime - 1, n - p.prime, lower.tail = FALSE)

rejection <- F.star.2 > qf(0.9, p.prime - 1, n - p.prime)

rejection
```

```
## [1] TRUE
```

As you can see the test says we should reject the null hypothesis. In other words there is sufficient evidence to say that at least one of $\beta_1, \beta_2, \beta_3$ is not 0.

$$\text{The P-Value is } 1.5419726 \times 10^{-10}$$

Part e

To Calculate the coefficient of determination or R^2 its just $\frac{SSR}{SST}$.

```
R.Squared <- SSR[1]/SST[1]
R.Squared
```

```
## [1] 0.6821943
```

To Calculate the adjusted coefficient of determination or R^2_{adj} its just $1 - \left(\frac{n-1}{n-p'}\right)\frac{SSE}{SST}$.

```
Adj.R.Squared <- 1 - (n-1)/(n-p.prime) * SSE[1]/SST[1]
Adj.R.Squared
```

```
## [1] 0.6594939
```

R^2 explains the fraction of the variance in Y explained by the model. In other words the relationship between $X_1, \cdots, X_p$ and Y

R^2_{adj} accounts for the number of variables in the model.

Part f

```
# New Values
new.values <- data.frame(Age = 35, Illness_Severity = 45, Anxiety_Lvl = 2.2)
patient.prediction <- predict(patient.regression, new.values, interval = "prediction", level = 0.9)
patient.prediction
```

```
##        fit      lwr      upr
## 1 69.01029 51.50965 86.51092
```

With 90% confidence we can say that the new patient's satisfaction will be between 51.51 and 86.51

Question 2

Part a

$$H_0 : K'\beta - m = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} => \begin{bmatrix} 1 & 70 & 10 & 10 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} - \begin{bmatrix} 80 \\ 4 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Part b

The F* $= \frac{\frac{1800}{K}}{MSE} = \frac{\frac{1800}{K}}{\frac{SSE}{n-4}} = \frac{\frac{1800}{3}}{\frac{SSE}{n-4}} = \frac{600}{\frac{7800}{24-4}} = 1.5384615$

```
p.prime = 4
n = 24
F.star = 600/(7800/20)
p.value = pf(F.star, 3, n-p.prime, lower.tail = FALSE)
p.value
```

```
## [1] 0.2354379
```

Now, since p-value is 0.2354379 which is greater than $\alpha = 0.05$, we fail to reject the hypothesis H_0

Question 3

Part a

$$\mathbf{X'X} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ I_{11} & I_{21} & \cdots & I_{n1} \\ I_{12} & I_{22} & \cdots & I_{n2} \end{bmatrix} \begin{bmatrix} 1 & I_{11} & I_{12} \\ 1 & I_{21} & I_{22} \\ \vdots & \vdots & \vdots \\ 1 & I_{n1} & I_{n2} \end{bmatrix}$$

$$= \begin{bmatrix} n & \sum_{i=1}^n I_{i1} & \sum_{i=1}^n I_{i2} \\ \sum_{i=1}^n I_{i1} & \sum_{i=1}^n I_{i1}^2 & \sum_{i=1}^n I_{i1} I_{i2} \\ \sum_{i=1}^n I_{i2} & \sum_{i=1}^n I_{i1} I_{i2} & \sum_{i=1}^n I_{i2}^2 \end{bmatrix}$$

$$= \begin{bmatrix} n & n_A & n_B \\ n_A & n_A & 0 \\ n_B & 0 & n_B \end{bmatrix}$$

$$\mathbf{X'Y} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ I_{11} & I_{21} & \cdots & I_{n1} \\ I_{12} & I_{22} & \cdots & I_{n2} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^n Y_i \\ \sum_{i=1}^n I_{i1} Y_i \\ \sum_{i=1}^n I_{i2} Y_i \end{bmatrix}$$

$$= \begin{bmatrix} n\,\bar{y} \\ n_A\,\bar{y}_A \\ n_B\,\bar{y}_B \end{bmatrix}$$

Part b

Option 1 We need to find the inverse of $X'X$

So using linear row reduction to find the Inverse Matrix.

$$(X'X)^{-1} = \left[\begin{array}{ccc|ccc} n & n_A & n_B & 1 & 0 & 0 \\ n_A & n_A & 0 & 0 & 1 & 0 \\ n_B & 0 & n_B & 0 & 0 & 1 \end{array}\right]$$

$$= \left[\begin{array}{ccc|ccc} 1 & \frac{n_A}{n} & \frac{n_B}{n} & \frac{1}{n} & 0 & 0 \\ n_A & n_A & 0 & 0 & 1 & 0 \\ n_B & 0 & n_B & 0 & 0 & 1 \end{array}\right]$$

$$= \left[\begin{array}{ccc|ccc} 1 & \frac{n_A}{n} & \frac{n_B}{n} & \frac{1}{n} & 0 & 0 \\ 0 & \frac{n_A(n_B+n_C)}{n} & \frac{-\,n_B\,n_A}{n} & \frac{-n_A}{n} & 1 & 0 \\ 0 & \frac{-\,n_B\,n_A}{n} & \frac{n_B(n_A+n_C)}{n} & \frac{-n_B}{n} & 0 & 1 \end{array}\right]$$

$$= \left[\begin{array}{ccc|ccc} 1 & \frac{n_A}{n} & \frac{n_B}{n} & \frac{1}{n} & 0 & 0 \\ 0 & 1 & \frac{-\,n_B}{n_B+n_C} & \frac{-1}{n_B+n_C} & \frac{n}{n_A(n_B+n_C)} & 0 \\ 0 & \frac{-\,n_B\,n_A}{n} & \frac{n_B(n_A+n_C)}{n} & \frac{-n_B}{n} & 0 & 1 \end{array}\right]$$

$$= \left[\begin{array}{ccc|ccc} 1 & 0 & \frac{n_B}{n_B+n_C} & \frac{1}{n_B+n_C} & \frac{-1}{n_B+n_C} & 0 \\ 0 & 1 & \frac{-n_B}{n_B+n_C} & \frac{-1}{n_B+n_C} & \frac{n}{n_A(n_B+n_C)} & 0 \\ 0 & 0 & \frac{n_B n_C}{n_B+n_C} & \frac{-n_B}{n_B+n_C} & \frac{n_B}{n_B+n_C} & 1 \end{array}\right]$$

$$= \left[\begin{array}{ccc|ccc} 1 & 0 & \frac{n_B}{n_B+n_C} & \frac{1}{n_B+n_C} & \frac{-1}{n_B+n_C} & 0 \\ 0 & 1 & \frac{-n_B}{n_B+n_C} & \frac{-1}{n_B+n_C} & \frac{n}{n_A(n_B+n_C)} & 0 \\ 0 & 0 & 1 & \frac{-1}{n_C} & \frac{1}{n_C} & \frac{n_B+n_C}{n_B\,n_C} \end{array}\right]$$

$$= \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{1}{n_C} & \frac{-1}{n_C} & \frac{-1}{n_C} \\ 0 & 1 & 0 & \frac{-1}{n_C} & \frac{n\,n_C+n_B\,n_A}{n_C\,n_A(n_B+n_C)} & \frac{1}{n_C} \\ 0 & 0 & 1 & \frac{-1}{n_C} & \frac{1}{n_C} & \frac{n_B+n_C}{n_B\,n_C} \end{array}\right]$$

So

$$(X'X)^{-1} = \begin{bmatrix} \frac{1}{n_C} & \frac{-1}{n_C} & \frac{-1}{n_C} \\ \frac{-1}{n_C} & \frac{n\,n_C+n_B\,n_A}{n_A(n_B+n_C)} & \frac{1}{n_C} \\ \frac{-1}{n_C} & \frac{1}{n_C} & \frac{n_B+n_C}{n_B\,n_C} \end{bmatrix}$$

$$\hat{\beta} = (X'X)^{-1}X'Y = \begin{bmatrix} \frac{1}{n_C} & \frac{-1}{n_C} & \frac{-1}{n_C} \\ \frac{-1}{n_C} & \frac{n\,n_C+n_B\,n_A}{n_C\,n_A(n_B+n_C)} & \frac{1}{n_C} \\ \frac{-1}{n_C} & \frac{1}{n_C} & \frac{n_B+n_C}{n_B\,n_C} \end{bmatrix} \begin{bmatrix} n\,\bar{y} \\ n_A\,\bar{y}_A \\ n_B\,\bar{y}_B \end{bmatrix}$$

$$= \begin{bmatrix} \frac{n\,\bar{y}}{n_C} - \frac{n_A\,\bar{y}_A}{n_C} - \frac{n_B\,\bar{y}_B}{n_C} \\ \frac{-n\,\bar{y}}{n_C} + \frac{n\,n_C\,\bar{y}_A+n_B\,n_A\,\bar{y}_A}{n_C(n_B+n_C)} + \frac{n_B\,\bar{y}_B}{n_C} \\ \frac{-n\,\bar{y}}{n_C} + \frac{n_A\,\bar{y}_A}{n_C} + \frac{n_B\,\bar{y}_B + n_C\,\bar{y}_B}{n_C} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{n\,\bar{y} - n_A\,\bar{y}_A - n_B\,\bar{y}_B}{n_C} \\ \frac{-n\,\bar{y}(n_B+n_C)}{n_C} + \frac{n\,n_C\,\bar{y}_A+n_B\,n_A\,\bar{y}_A}{n_C(n_B+n_C)} + \frac{n_B\,\bar{y}_B(n_B+n_C)}{n_C} \\ \frac{-n\,\bar{y} + n_A\,\bar{y}_A + n_B\,\bar{y}_B + n_C\,\bar{y}_B}{n_C} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{n_A\,\bar{y}_A + n_B\,\bar{y}_B + n_C\,\bar{y}_C - n_A\,\bar{y}_A - n_B\,\bar{y}_B}{n_C} \\ \frac{(-n_A\,\bar{y}_A - n_B\,\bar{y}_B - n_C\,\bar{y}_C)\,(n_B+n_C) + n\,n_C\,\bar{y}_A+n_B\,n_A\,\bar{y}_A + n_B^2\,\bar{y}_B+n_B\,n_C\,\bar{y}_B}{n_C(n_B+n_C)} \\ \frac{-n_A\,\bar{y}_A - n_B\,\bar{y}_B - n_C\,\bar{y}_C + n_A\,\bar{y}_A + n_B\,\bar{y}_B + n_C\,\bar{y}_B}{n_C} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{n_C\,\bar{y}_C}{n_C} \\ \frac{-n_A\,n_B\,\bar{y}_A - n_B^2\,\bar{y}_B - n_C\,n_B\,\bar{y}_C - n_A\,n_C\,\bar{y}_A - n_B\,n_C\,\bar{y}_B - n_C^2\,\bar{y}_C + n\,n_C\,\bar{y}_A + n_B\,n_A\,\bar{y}_A + n_B^2\,\bar{y}_B + n_B\,n_C\,\bar{y}_B}{n_C(n_B+n_C)} \\ \frac{-\,n_C\,\bar{y}_C + n_C\,\bar{y}_B}{n_C} \end{bmatrix}$$

$$= \begin{bmatrix} \bar{y}_C \\ \frac{(-n_A\,n_B - n_A\,n_C + n\,n_C + n_B\,n_A)\bar{y}_A + (-n_B^2 - n_B\,n_C + n_B^2 + n_B\,n_C)\bar{y}_B + (-n_C\,n_B - n_C^2)\bar{y}_C}{n_C(n_B+n_C)} \\ \frac{n_C(\bar{y}_B - \bar{y}_C)}{n_C} \end{bmatrix}$$

$$= \begin{bmatrix} \bar{y}_C \\ \frac{(n\,n_C - n_A\,n_C)\bar{y}_A - (n_C^2 + n_C\,n_B)\bar{y}_C}{n_C(n_B+n_C)} \\ \bar{y}_B - \bar{y}_C \end{bmatrix}$$

$$= \begin{bmatrix} \bar{y}_C \\ \frac{n_C(n - n_A)\bar{y}_A - n_C(n_C + n_B)\bar{y}_C}{n_C(n_B+n_C)} \\ \bar{y}_B - \bar{y}_C \end{bmatrix}$$

$$= \begin{bmatrix} \bar{y}_C \\ \frac{n_C(n_B+n_C)\bar{y}_A - n_C(n_C+n_B)\bar{y}_C}{n_C(n_B+n_C)} \\ \bar{y}_B - \bar{y}_C \end{bmatrix}$$

$$= \begin{bmatrix} \bar{y}_C \\ \frac{n_C(n_B+n_C)(\bar{y}_A - \bar{y}_C)}{n_C(n_B+n_C)} \\ \bar{y}_B - \bar{y}_C \end{bmatrix}$$

$$= \begin{bmatrix} \bar{y}_C \\ \bar{y}_A - \bar{y}_C \\ \bar{y}_B - \bar{y}_C \end{bmatrix}$$

Option 2 Minimizing the Sum of squared errors $S(\beta_0, \beta_1, \beta_2) = \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 I_{1i} - \beta_2 I_{2i})^2$

$$\begin{aligned} \frac{\partial S(\beta_0, \beta_1, \beta_2)}{\beta_0} &= \sum_{i=1}^{n} 2(y_i - \beta_0 - \beta_1 I_{1i} - \beta_2 I_{2i})(-1) \\ 0 &= -2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 I_{1i} - \beta_2 I_{2i}) \\ 0 &= \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} \beta_0 - \sum_{i=1}^{n} \beta_1 I_{1i} - \sum_{i=1}^{n} \beta_2 I_{2i} \\ 0 &= n\bar{y} - n\beta_0 - n_A\beta_1 - n_B\beta_2 \\ \hat{\beta}_0 &= \bar{y} - \frac{n_A}{n}\beta_1 - \frac{n_B}{n}\beta_2 \end{aligned}$$

$$\begin{aligned} \frac{\partial S(\beta_0, \beta_1, \beta_2)}{\beta_1} &= \sum_{i=1}^{n} 2(y_i - \beta_0 - \beta_1 I_{1i} - \beta_2 I_{2i})(-1 I_{1i}) \\ 0 &= -2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 I_{1i} - \beta_2 I_{2i}) I_{1i} \\ 0 &= \sum_{i=1}^{n} y_i I_{1i} - \sum_{i=1}^{n} \beta_0 I_{1i} - \sum_{i=1}^{n} \beta_1 I_{1i} I_{1i} - \sum_{i=1}^{n} \beta_2 I_{2i} I_{1i} \\ 0 &= \bar{y}_A - n_A\beta_0 - n_A\beta_1 \\ \hat{\beta}_1 &= \bar{y}_A - \beta_0 \end{aligned}$$

$$\begin{aligned} \frac{\partial S(\beta_0, \beta_1, \beta_2)}{\beta_1} &= \sum_{i=1}^{n} 2(y_i - \beta_0 - \beta_1 I_{1i} - \beta_2 I_{2i})(-1 I_{2i}) \\ 0 &= -2\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 I_{1i} - \beta_2 I_{2i}) I_{2i} \end{aligned}$$

$$
\begin{aligned}
0 &= \sum_{i=1}^{n} y_i I_{2i} - \sum_{i=1}^{n} \beta_0 I_{2i} - \sum_{i=1}^{n} \beta_1 I_{2i} - \sum_{i=1}^{n} \beta_2 I_{2i} I_{2i} \\
0 &= \bar{y}_B - n_B \beta_0 - n_B \beta_2 \\
\hat{\beta}_2 &= \bar{y}_B - \beta_0
\end{aligned}
$$

Simplifying it,

$$
\begin{aligned}
\beta_0 &= \bar{y} - \frac{n_A}{n}\beta_1 - \frac{n_B}{n}\beta_2 \\
\beta_0 &= \frac{\sum_{i=1}^{n_A} y_A \, + \, \sum_{i=1}^{n_B} y_B \, + \, \sum_{i=1}^{n_C} y_C}{n} - \frac{n_A}{n}(\bar{y}_A - \beta_0) - \frac{n_B}{n}(\bar{y}_B - \beta_0) \\
\beta_0 &= \frac{n_A \, \bar{y}_A \, + \, n_B \, \bar{y}_B \, + \, n_C \, \bar{y}_C}{n} \, - \, \frac{n_A \, \bar{y}_A}{n} \, + \, \frac{n_A \, \beta_0}{n} \, - \, \frac{n_B \, \bar{y}_B}{n} \, + \, \frac{n_B \, \beta_0}{n} \\
n \, \beta_0 &= n_A \, \bar{y}_A \, + \, n_B \, \bar{y}_B \, + \, n_C \, \bar{y}_C \, - \, n_A \, \bar{y}_A \, + \, n_A \, \beta_0 \, - \, n_B \, \bar{y}_B \, + \, n_B \, \beta_0 \\
n \, \beta_0 &= n_C \, \bar{y}_C \, + \, n_A \, \beta_0 \, + \, n_B \, \beta_0 \\
n \, \beta_0 \, - \, n_A \, \beta_0 \, - \, n_B \, \beta_0 &= n_C \, \bar{y}_C \\
n_C \, \beta_0 &= n_C \, \bar{y}_C
\end{aligned}
$$

$$
\begin{aligned}
\hat{\beta}_0 &= \bar{y}_C \\
\hat{\beta}_1 &= \bar{y}_A - \bar{y}_C \\
\hat{\beta}_2 &= \bar{y}_B - \bar{y}_C
\end{aligned}
$$

as wanted

Part c

To calculate the SSE we need the residual $e_i = Y_i - \hat{Y}_i$
So

$$\hat{Y}_i = \begin{cases} \hat{\beta}_0 + \hat{\beta}_1 & \text{when } X_i = A \\ \hat{\beta}_0 + \hat{\beta}_2 & \text{when } X_i = B \\ \hat{\beta}_0 & \text{when } X_i = C \end{cases}$$

$$\hat{Y}_i = \begin{cases} \bar{y}_A & \text{when } X_i = A \\ \bar{y}_B & \text{when } X_i = B \\ \bar{y}_C & \text{when } X_i = C \end{cases}$$

Therefore the residual is

$$e_i = Y_i - \hat{Y}_i = \begin{cases} Y_i - \bar{y}_A & \text{when } X_i = A \\ Y_i - \bar{y}_B & \text{when } X_i = B \\ Y_i - \bar{y}_C & \text{when } X_i = C \end{cases}$$

Therefore SSE is

$$SSE = \sum_{i=1}^{n} e_i^2$$

$$SSE = \sum_{i=1}^{n} I_A(Y_i - \bar{y}_A)^2 + I_B(Y_i - \bar{y}_B)^2 + I_C(Y_i - \bar{y}_C)^2$$

$$SSE = \sum_{i=1}^{n_A}(Y_i - \bar{y}_A)^2 + \sum_{i=1}^{n_B}(Y_i - \bar{y}_B)^2 + \sum_{i=1}^{n_C}(Y_i - \bar{y}_C)^2$$

By using the unbiased estimator of the Variance we can say

$$SSE = (n_A - 1)\mathbb{V}_A + (n_B - 1)\mathbb{V}_B + (n_C - 1)\mathbb{V}_C$$

$$SSE = (n_A - 1)s_A^2 + (n_B - 1)s_B^2 + (n_C - 1)s_C^2$$

as wanted.

Question 4

Part a

```
egyptCotton <- read.table("egyptcttn.txt", quote="\"", comment.char="")
names(egyptCotton) <- c("Variety","Luminance","lnGrade")
egyptCotton$giza69 <- ifelse(egyptCotton$Variety == "Giza69", 1, 0)
egyptCotton$giza67 <- ifelse(egyptCotton$Variety == "Giza67", 1, 0)
egyptCotton$giza70 <- ifelse(egyptCotton$Variety == "Giza70", 1, 0)
egyptCotton$giza68 <- ifelse(egyptCotton$Variety == "Giza68", 1, 0)

### Direct coding of "new" dummy variables from "variety"
lnGradeGiza69 <- egyptCotton$lnGrade*egyptCotton$giza69
lnGradeGiza67 <- egyptCotton$lnGrade*egyptCotton$giza67
lnGradeGiza70 <- egyptCotton$lnGrade*egyptCotton$giza70
lnGradeGiza68 <- egyptCotton$lnGrade*egyptCotton$giza68

fit.FullModel <- lm(Luminance ~ lnGrade + giza69 + giza67 + giza70 + giza68 +
                      lnGradeGiza69 + lnGradeGiza67 + lnGradeGiza70 + lnGradeGiza68
                    , data = egyptCotton)
summary(fit.FullModel)
```

```
##
## Call:
## lm(formula = Luminance ~ lnGrade + giza69 + giza67 + giza70 +
##     giza68 + lnGradeGiza69 + lnGradeGiza67 + lnGradeGiza70 +
##     lnGradeGiza68, data = egyptCotton)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66004 -0.05597 -0.00598  0.10859  0.32705
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    78.8034     1.3976  56.386 7.47e-14 ***
## lnGrade         3.3137     0.4243   7.810 1.45e-05 ***
## giza69          7.2233     1.9765   3.655  0.00443 **
## giza67          2.0524     1.9765   1.038  0.32352
## giza70          5.1151     1.9765   2.588  0.02704 *
## giza68          5.0801     1.9765   2.570  0.02788 *
## lnGradeGiza69  -2.2741     0.6000  -3.790  0.00354 **
## lnGradeGiza67  -1.1507     0.6000  -1.918  0.08411 .
## lnGradeGiza70  -2.0709     0.6000  -3.452  0.00621 **
## lnGradeGiza68  -1.0797     0.6000  -1.800  0.10212
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.2907 on 10 degrees of freedom
## Multiple R-squared:  0.9794, Adjusted R-squared:  0.9609
## F-statistic: 52.82 on 9 and 10 DF,  p-value: 2.986e-07
```

Therefore, from the above the full model is:

$$78.8034042 + 3.3136504 lnGrade + 7.223323 giza69 + 2.0524441 giza67 + 5.1150993 giza70 + 5.0800919 giza68$$
$$-2.2741404 lnGradeGiza69 - 1.1506624 lnGradeGiza67 - 2.0709482 lnGradeGiza70 - 1.0797439 lnGradeGiza68$$

Part b

The model in part a follows the following format: $E(Luminance) = \beta_0 + \beta_1\text{lnGrade} + \beta_2\text{Giza69} + \beta_3\text{Giza67} + \beta_4\text{Giza70} + \beta_5\text{Giza68} + \beta_6\text{lnGradeGiza69} + \beta_7\text{lnGradeGiza67} + \beta_8\text{lnGradeGiza70} + \beta_9\text{lnGradeGiza68}$

And we want to test the following:

$H_0 : \beta_6 = \beta_7 = \beta_8 = \beta_9 = 0$

```r
#summary(fit.FullModel)
fullAnova = anova(fit.FullModel)
n = as.numeric(nrow(egyptCotton))
MSE.FullModel = as.numeric(fullAnova$`Mean Sq`[10])
SSE.FullModel = as.numeric(fullAnova$`Sum Sq`[10])
df.FullModel = n - as.numeric(length(fit.FullModel$coefficients))

#print("\n Reduced \n")
fit.reduced = lm(Luminance  ~ Variety + lnGrade, data = egyptCotton)
#summary(fit.reduced)
reducedAnova = anova(fit.reduced)
df.ReducedModel = n - as.numeric(length(fit.reduced$coefficients))
#df.ReducedModel
#reducedAnova
SSE.reduced = as.numeric(reducedAnova$`Sum Sq`[3])

# SSE.reduced = 0.216423
# SSE.FullModel = 0.211794
# df.ReducedModel = 432
# df.FullModel = 431
# MSE.FullModel = SSE.FullModel/df.FullModel

F.star = ((SSE.reduced-SSE.FullModel)/(df.ReducedModel-df.FullModel))/MSE.FullModel
F.star
```

```
## [1] 4.587634
```

```r
qf(0.95, df.ReducedModel-df.FullModel, df.FullModel)
```

```
## [1] 3.47805
```

```r
p.value = pf(F.star, df.ReducedModel-df.FullModel, df.FullModel, lower.tail= FALSE)
p.value
```

```
## [1] 0.02313061
```

Now, the p-value is 0.0231306 Which is less than 0.05. Hence, we reject the null hypothesis H_0

Part c

Let's assume that we fail to reject the null hypothesis in part (b) (i.e. $\beta_6 = \beta_7 = \beta_8 = \beta_9 = 0$).

Now, the new model is:

```
fit.newModel = lm(Luminance ~ lnGrade + giza69 + giza67 + giza70 + giza68, data=egyptCotton)
summary(fit.newModel)
```

```
##
## Call:
## lm(formula = Luminance ~ lnGrade + giza69 + giza67 + giza70 +
##     giza68, data = egyptCotton)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.86328 -0.18708  0.07463  0.23577  0.55422
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  83.1121     0.9085  91.487  < 2e-16 ***
## lnGrade       1.9986     0.2700   7.402 3.34e-06 ***
## giza69       -0.2275     0.2925  -0.778 0.449653
## giza67       -1.7175     0.2925  -5.872 4.06e-05 ***
## giza70       -1.6700     0.2925  -5.709 5.40e-05 ***
## giza68        1.5425     0.2925   5.273 0.000118 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4137 on 14 degrees of freedom
## Multiple R-squared:  0.9416, Adjusted R-squared:  0.9207
## F-statistic: 45.14 on 5 and 14 DF,  p-value: 3.818e-08
```

Now, the new model is: $83.1120959 + 1.9985515 lnGrade - 0.2275 giza69 - 1.7175 giza67 - 1.67 giza70 + 1.5425 giza68$

The model in part a follows the following format: $E(Luminance) = \beta_0 + \beta_1 \text{lnGrade} + \beta_2 \text{Giza69} + \beta_3 \text{Giza67} + \beta_4 \text{Giza70} + \beta_5 \text{Giza68}$

And we want to test the following:

$H_0 : \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$

```
newAnova = anova(fit.newModel)
MSE.newModel = as.numeric(newAnova$`Mean Sq`[6])
SSE.newModel = as.numeric(newAnova$`Sum Sq`[6])
df.newModel = as.numeric(newAnova$`Df`[6])

fit.newReduced = lm(Luminance  ~ lnGrade, data = egyptCotton)
newReducedAnova = anova(fit.newReduced)
SSE.newReduced = as.numeric(newReducedAnova$`Sum Sq`[2])

F.star = ((SSE.newReduced-SSE.newModel)/4)/MSE.newModel
```

```r
p.value = pf(F.star, 4, df.newModel, lower.tail= FALSE)
p.value
```

```
## [1] 1.066057e-07
```

Now, the p-value is 1.0660575×10^{-7} Which is less than 0.05. Hence, we reject the null hypothesis H_0

Part d

Since, in parts b & c of this question we rejected both the following hypothesis:

$H_0 : \beta_6 = \beta_7 = \beta_8 = \beta_9 = 0$
$H_0 : \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$

Therefore, the model I will be using the full model from part (a) of this question which is:

$$78.8034042 + 3.3136504 lnGrade + 7.223323 giza69 + 2.0524441 giza67 + 5.1150993 giza70 + 5.0800919 giza68$$
$$-2.2741404 lnGradeGiza69 - 1.1506624 lnGradeGiza67 - 2.0709482 lnGradeGiza70 - 1.0797439 lnGradeGiza68$$

Since in this question, the objective is not predicting nor estimating the mean response. Instead, we're simply looking for ways to describe the behavior of the response variable. Hence, according to slide 10 on lecture 18, the full model is the model I would choose for this data.

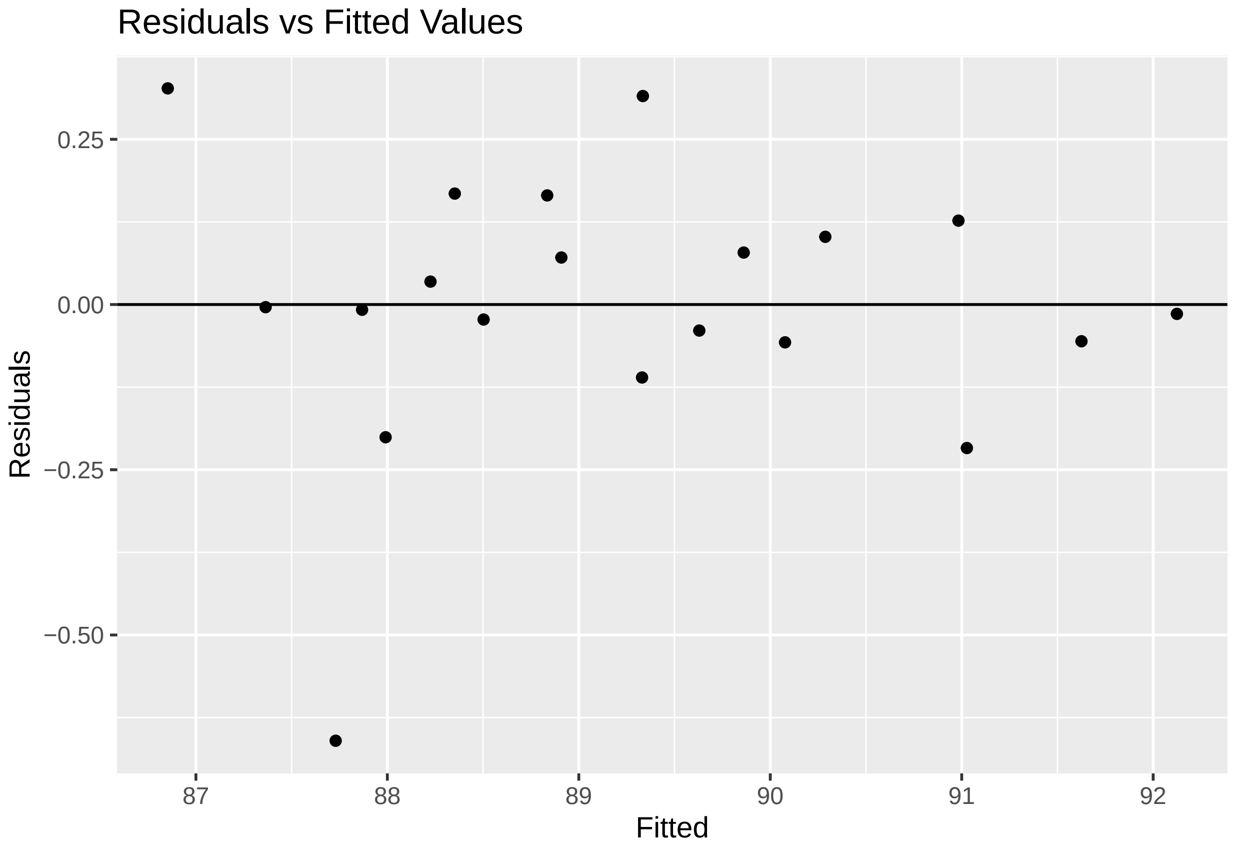
Part e

Assumptions:

- Errors are independent
- Error variance is contant (does not depend on the level of X)
- Errors are normally distriuted
- The relationship is linear

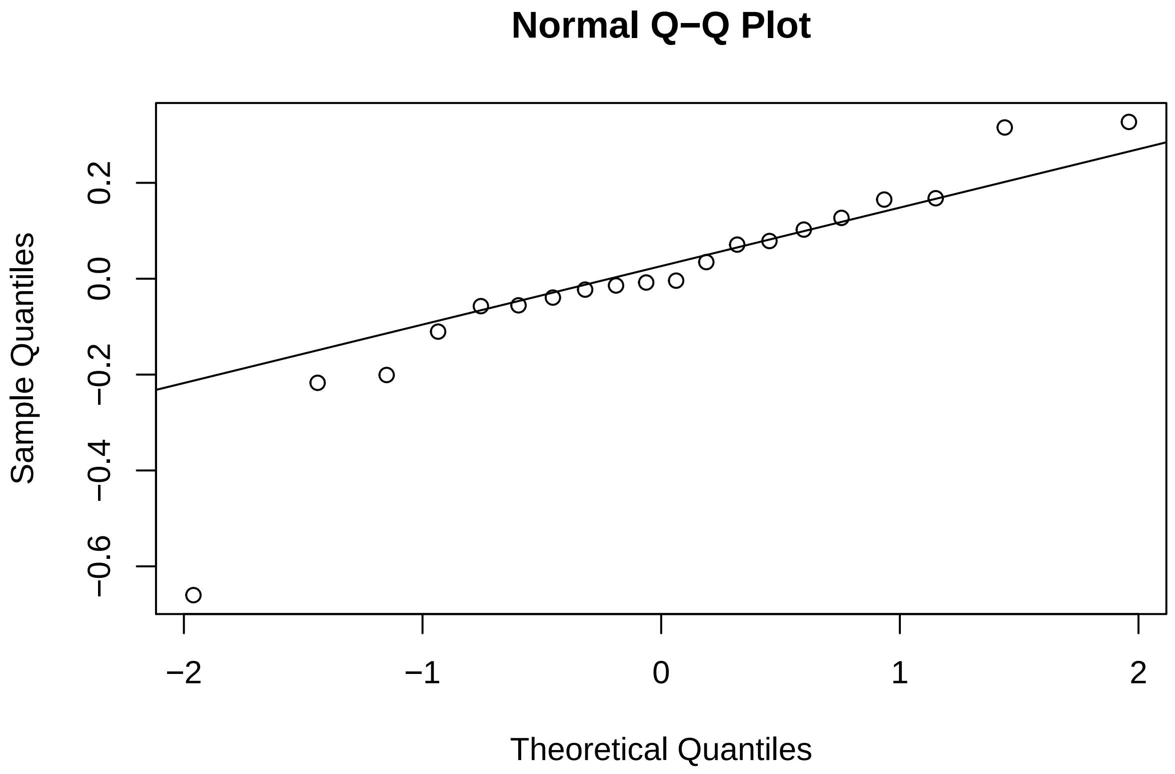
Residual Plot:

```r
#Store the residuals of the dataset
Residuals <- resid(fit.FullModel)

# store the fitted values of the regression
Fitted <- fitted.values(fit.FullModel)

# Combine them into data frame
graph.data <- cbind.data.frame(Residuals, Fitted)

# PLot the graph
ggplot(graph.data, aes(x=Fitted, y=Residuals)) + geom_point() +
ggtitle("Residuals vs Fitted Values") + geom_hline(yintercept = 0)
```



Based on the residual plot, we can't see any patterns, they are randomly scattered. Hence, the error variance seems to be constant, and the relationship is linear.

```
# Plot the graph to test the normality
qqnorm(Residuals)
qqline(Residuals)
```



```
# Perform the Shapiro test
shapiro.test(Residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Residuals
## W = 0.88183, p-value = 0.01908
```

Based on how spread the dots are on the Normal Q-Q plot and how far they are from forming a line y=x shape and the p-value of the Shapiro-Wilk's test which is 0.01908 which is much lower than 0.05, we cannot conclude that the residuals are coming from the normal population. Thus not all the regression assumptions are satisfied.

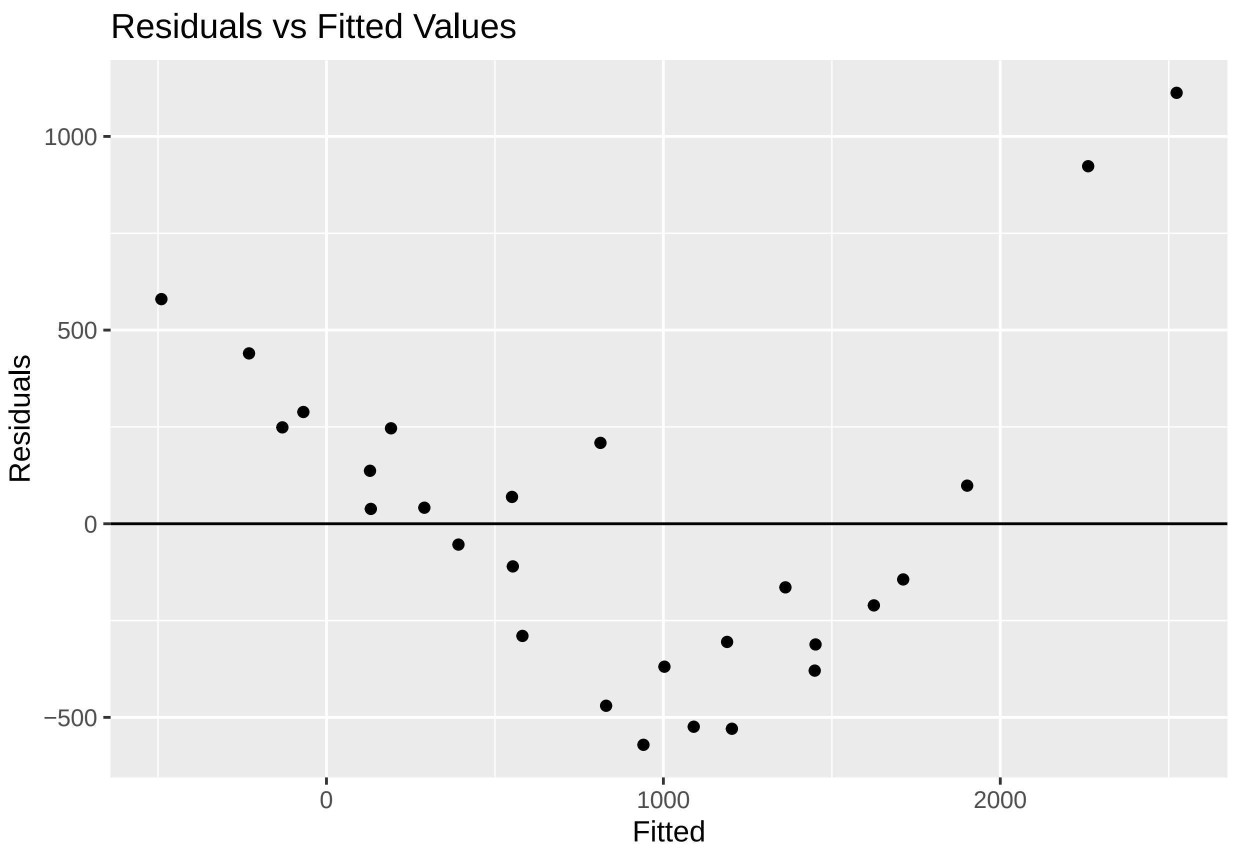
Question 5

Part a

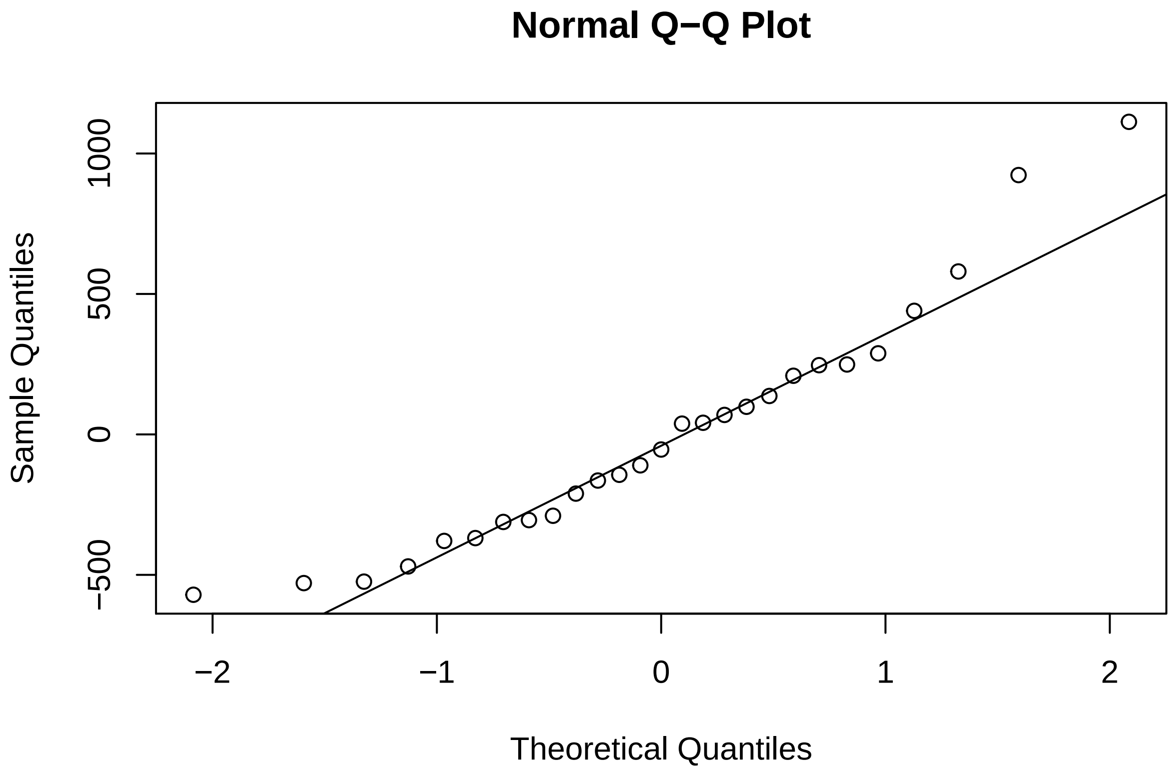
```
strengthWool <- read.table("StrengthWool.txt", quote="\"", comment.char="", header=TRUE)
fitaf <- lm(Cycles~factor(Len) + factor(Load) + factor(Amp), data = strengthWool)

summary(fitaf)
```

```
##
## Call:
## lm(formula = Cycles ~ factor(Len) + factor(Load) + factor(Amp),
##     data = strengthWool)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -570.81 -308.43  -53.81  227.57 1112.63
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1203.4      246.0   4.891 8.83e-05 ***
## factor(Len)300     421.4      227.8   1.850 0.079096 .
## factor(Len)350    1320.0      227.8   5.795 1.14e-05 ***
## factor(Load)45    -262.6      227.8  -1.153 0.262611
## factor(Load)50    -621.7      227.8  -2.729 0.012918 *
## factor(Amp)9      -811.6      227.8  -3.563 0.001948 **
## factor(Amp)10    -1071.7      227.8  -4.705 0.000136 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 483.2 on 20 degrees of freedom
## Multiple R-squared:  0.7692, Adjusted R-squared:  0.6999
## F-statistic: 11.11 on 6 and 20 DF,  p-value: 1.769e-05
```

```
rSquared = summary(fitaf)$r.squared

#Store the residuals of the dataset
Residuals <- resid(fitaf)

# store the fitted values of the regression
Fitted <- fitted.values(fitaf)

# Combine them into data frame
graph.data <- cbind.data.frame(Residuals, Fitted)

# PLot the graph
ggplot(graph.data, aes(x=Fitted, y=Residuals)) + geom_point() +
ggtitle("Residuals vs Fitted Values") + geom_hline(yintercept = 0)
```



```
# Plot the graph to test the normality
qqnorm(Residuals)
qqline(Residuals)
```



```
# Perform the Shapiro test
shapiro.test(Residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Residuals
## W = 0.9331, p-value = 0.08234
```

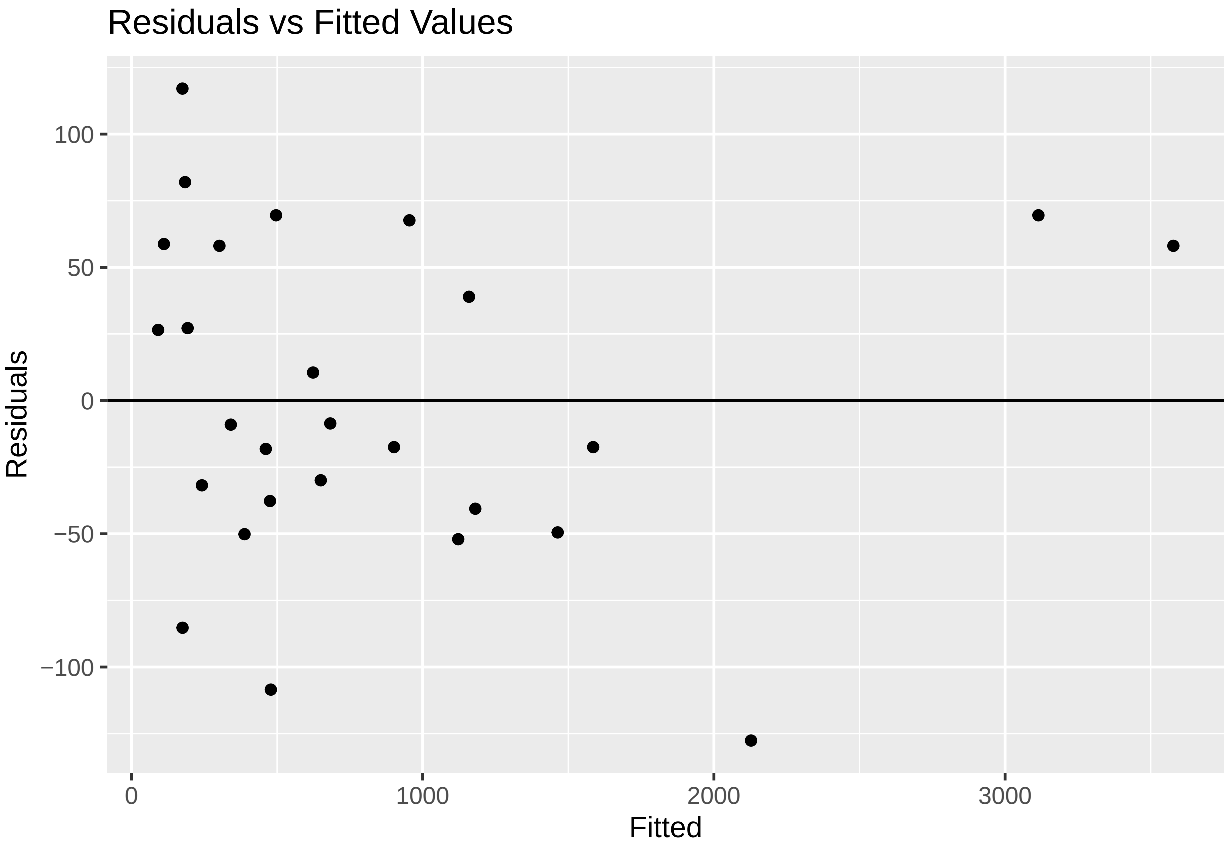
We can see from the summary of the model that $R^2 = 0.7691874$ which means that the predictors of this model explain roughly 77% of the variance of our response variable (Cycles). Also, looking at the residuals plot, it looks like there is the some quadratic relationship and that is slightly negative expanding. This means that we have a non-constant variance in our model. Also from the normal Q-Q plot, and the dots are not forming a great line y=x shape, hence our errors are somewhat normally distributed. This is also confirmed by the p-value of the shapiro-Wilk test which is approximatley 0.08 which is very closely above 0.05. As such, this model is not a very good fit for the data.

Part b

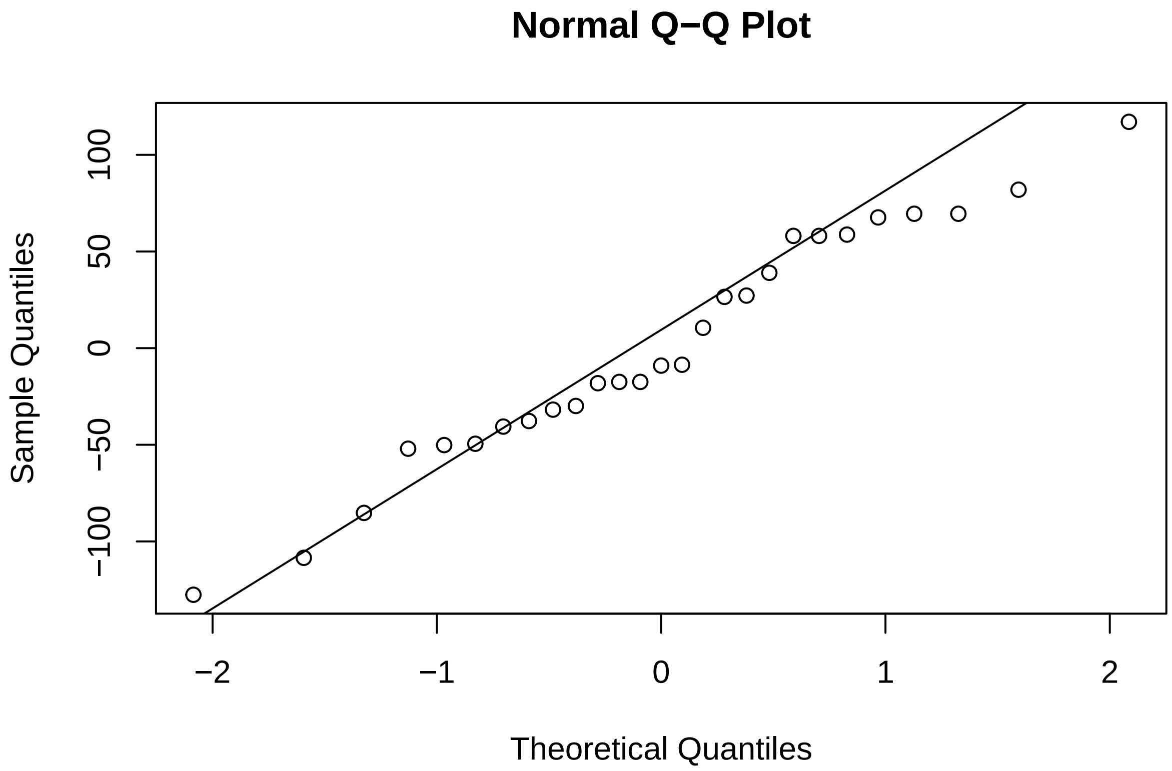
```
fitaf2 <- lm(Cycles~factor(Len) * factor(Load) + factor(Len) * factor(Amp)
             + factor(Load) * factor(Amp), data = strengthWool)
summary(fitaf2)
```

```
##
## Call:
## lm(formula = Cycles ~ factor(Len) * factor(Load) + factor(Len) *
##     factor(Amp) + factor(Load) * factor(Amp), data = strengthWool)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -127.593  -39.148   -9.037   58.074  117.074
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   6.826e+02  9.237e+01   7.390 7.69e-05 ***
## factor(Len)300                7.809e+02  1.161e+02   6.728 0.000148 ***
## factor(Len)350                2.895e+03  1.161e+02  24.946 7.13e-09 ***
## factor(Load)45               -2.041e+02  1.161e+02  -1.759 0.116697
## factor(Load)50               -5.077e+02  1.161e+02  -4.374 0.002368 **
## factor(Amp)9                 -2.944e+02  1.161e+02  -2.537 0.034879 *
## factor(Amp)10                -5.713e+02  1.161e+02  -4.923 0.001160 **
## factor(Len)300:factor(Load)45 -1.003e+02  1.271e+02  -0.789 0.452782
## factor(Len)350:factor(Load)45 -2.593e+02  1.271e+02  -2.040 0.075709 .
## factor(Len)300:factor(Load)50 -3.323e+02  1.271e+02  -2.614 0.030944 *
## factor(Len)350:factor(Load)50 -9.427e+02  1.271e+02  -7.414 7.52e-05 ***
## factor(Len)300:factor(Amp)9   -2.147e+02  1.271e+02  -1.688 0.129813
## factor(Len)350:factor(Amp)9   -1.698e+03  1.271e+02 -13.355 9.45e-07 ***
## factor(Len)300:factor(Amp)10  -4.310e+02  1.271e+02  -3.390 0.009502 **
## factor(Len)350:factor(Amp)10  -1.826e+03  1.271e+02 -14.362 5.40e-07 ***
## factor(Load)45:factor(Amp)9   -4.923e-14  1.271e+02   0.000 1.000000
## factor(Load)50:factor(Amp)9    3.613e+02  1.271e+02   2.842 0.021747 *
## factor(Load)45:factor(Amp)10   1.843e+02  1.271e+02   1.450 0.185155
## factor(Load)50:factor(Amp)10   5.717e+02  1.271e+02   4.496 0.002012 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 110.1 on 8 degrees of freedom
## Multiple R-squared:  0.9952, Adjusted R-squared:  0.9844
## F-statistic: 92.25 on 18 and 8 DF,  p-value: 2.537e-07
```

```r
rSquared = summary(fitaf2)$r.squared

#Store the residuals of the dataset
Residuals <- resid(fitaf2)

# store the fitted values of the regression
Fitted <- fitted.values(fitaf2)

# Combine them into data frame
graph.data <- cbind.data.frame(Residuals, Fitted)

# PLot the graph
ggplot(graph.data, aes(x=Fitted, y=Residuals)) + geom_point() +
ggtitle("Residuals vs Fitted Values") + geom_hline(yintercept = 0)
```



```
# Plot the graph to test the normality
qqnorm(Residuals)
qqline(Residuals)
```



```
# Perform the Shapiro test
shapiro.test(Residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Residuals
## W = 0.97117, p-value = 0.6331
```

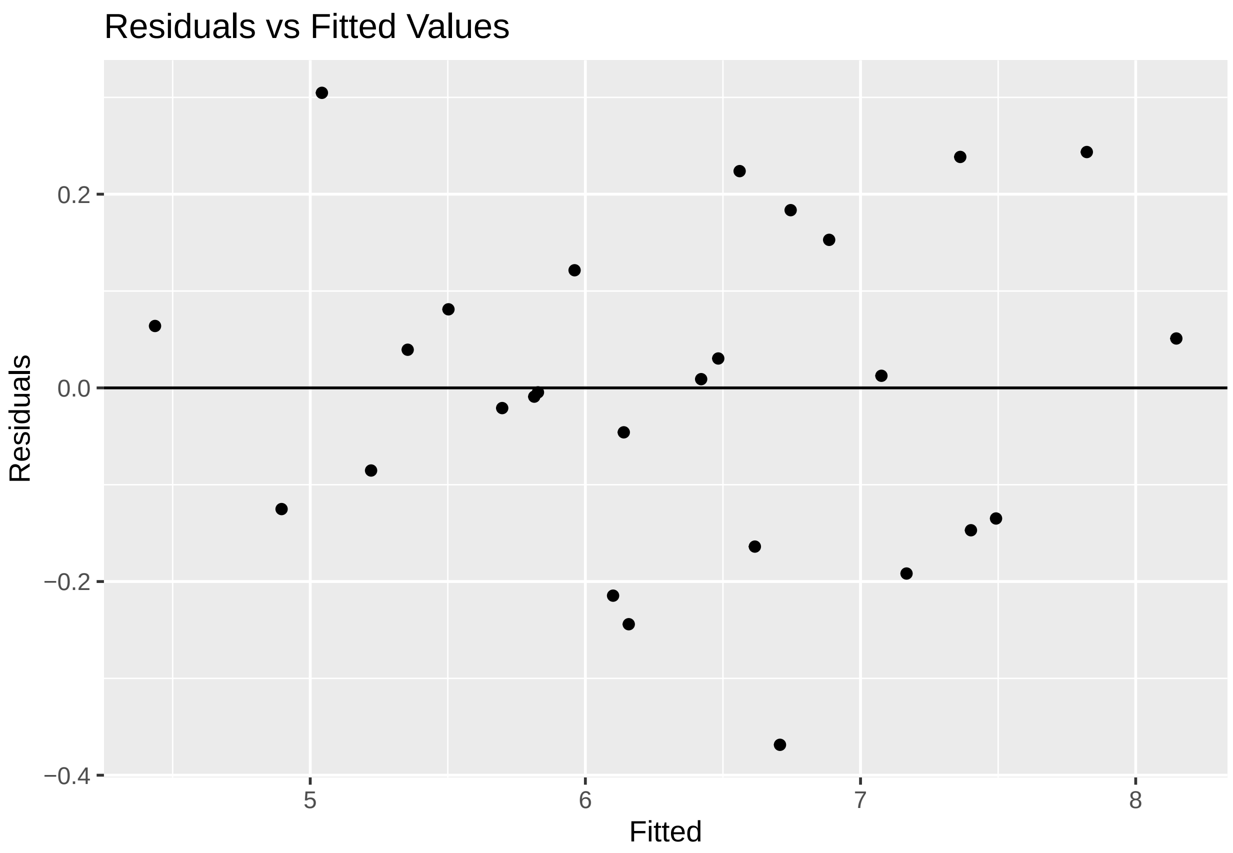
We can see from the summary of the model that $R^2 = 0.9952052$ which means that the predictors of this model explain almost 100% of the variance of our response variable (Cycles). Also, based on the residual plot, we cannot see any patterns, they are randomly scattered. Hence, the error variance seems to be constant, and the relationship is linear. In addition to that, Based on how closely the Normal Q-Q plot is forming a shape of a line y=x and the Shapiro-Wilk's test's p-value which is 0.6331 which is very largely greater than 0.05, we can conclude that the residuals are coming from the normal population. Thus all regression assumptions are satisfied and this model is a very good fit for the data and of course fits it much better than the model in part (a).

Part c

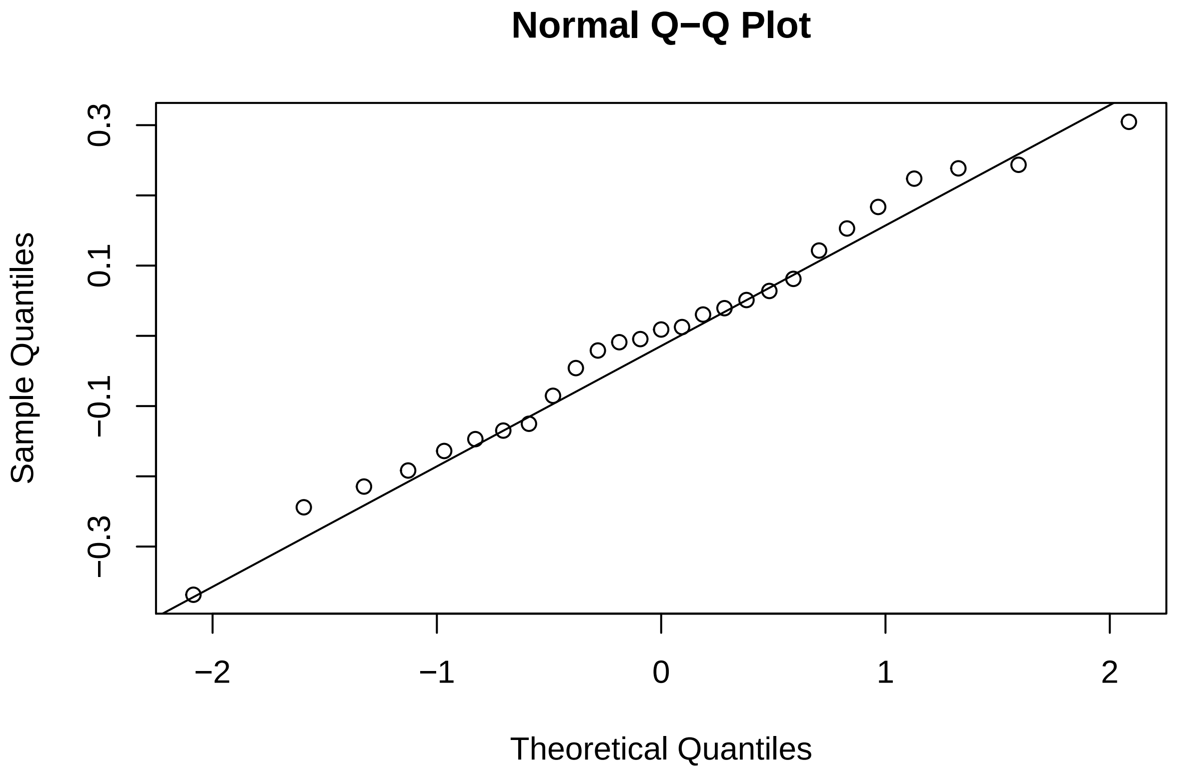
```
fitaf3 <- lm(log(Cycles)~factor(Len) + factor(Load) + factor(Amp), data = strengthWool)

summary(fitaf3)
```

```
##
## Call:
## lm(formula = log(Cycles) ~ factor(Len) + factor(Load) + factor(Amp),
##     data = strengthWool)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36860 -0.13002  0.00902  0.10129  0.30469
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     6.48287    0.09644  67.225  < 2e-16 ***
## factor(Len)300  0.91833    0.08928  10.286 1.97e-09 ***
## factor(Len)350  1.66477    0.08928  18.646 4.10e-14 ***
## factor(Load)45 -0.32529    0.08928  -3.643  0.00162 **
## factor(Load)50 -0.78524    0.08928  -8.795 2.62e-08 ***
## factor(Amp)9   -0.65521    0.08928  -7.339 4.31e-07 ***
## factor(Amp)10  -1.26173    0.08928 -14.132 7.19e-12 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.1894 on 20 degrees of freedom
## Multiple R-squared:  0.9691, Adjusted R-squared:  0.9598
## F-statistic: 104.5 on 6 and 20 DF,  p-value: 4.979e-14
```

```r
rSquared = summary(fitaf3)$r.squared

#Store the residuals of the dataset
Residuals <- resid(fitaf3)

# store the fitted values of the regression
Fitted <- fitted.values(fitaf3)

# Combine them into data frame
graph.data <- cbind.data.frame(Residuals, Fitted)

# PLot the graph
ggplot(graph.data, aes(x=Fitted, y=Residuals)) + geom_point() +
ggtitle("Residuals vs Fitted Values") + geom_hline(yintercept = 0)
```



```
# Plot the graph to test the normality
qqnorm(Residuals)
qqline(Residuals)
```



```
# Perform the Shapiro test
shapiro.test(Residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Residuals
## W = 0.98443, p-value = 0.9458
```

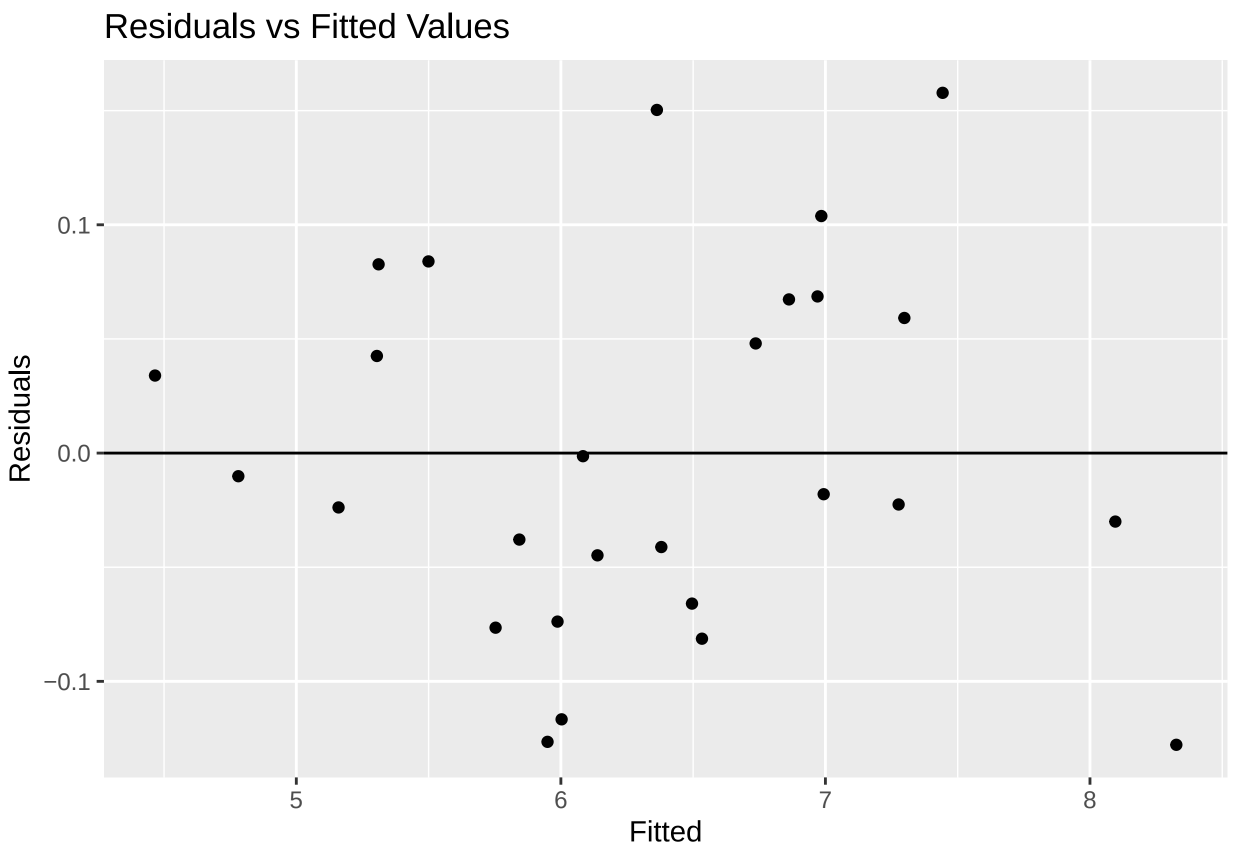
Since R treats log in base as the natural logarithm. So the log transformation in the model above is actually a Ln transformation. As we can see from the summary of the model that $R^2 = 0.9690801$ which means that the predictors of this model explain almost 97% of the variance of our response variable (Cycles). Also, based on the residual plot, we cannot see any patterns, they are randomly scattered. Hence, the error variance seems to be constant, and the relationship is linear. In addition to that, Based on how very closely the Normal Q-Q plot is forming a shape of a line y=x and the Shapiro-Wilk's test's p-value which is 0.9458 which is very largely greater than 0.05, we can conclude that the residuals are coming from the normal population. Thus all regression assumptions are satisfied and this model is a very good fit for the data.

Part d

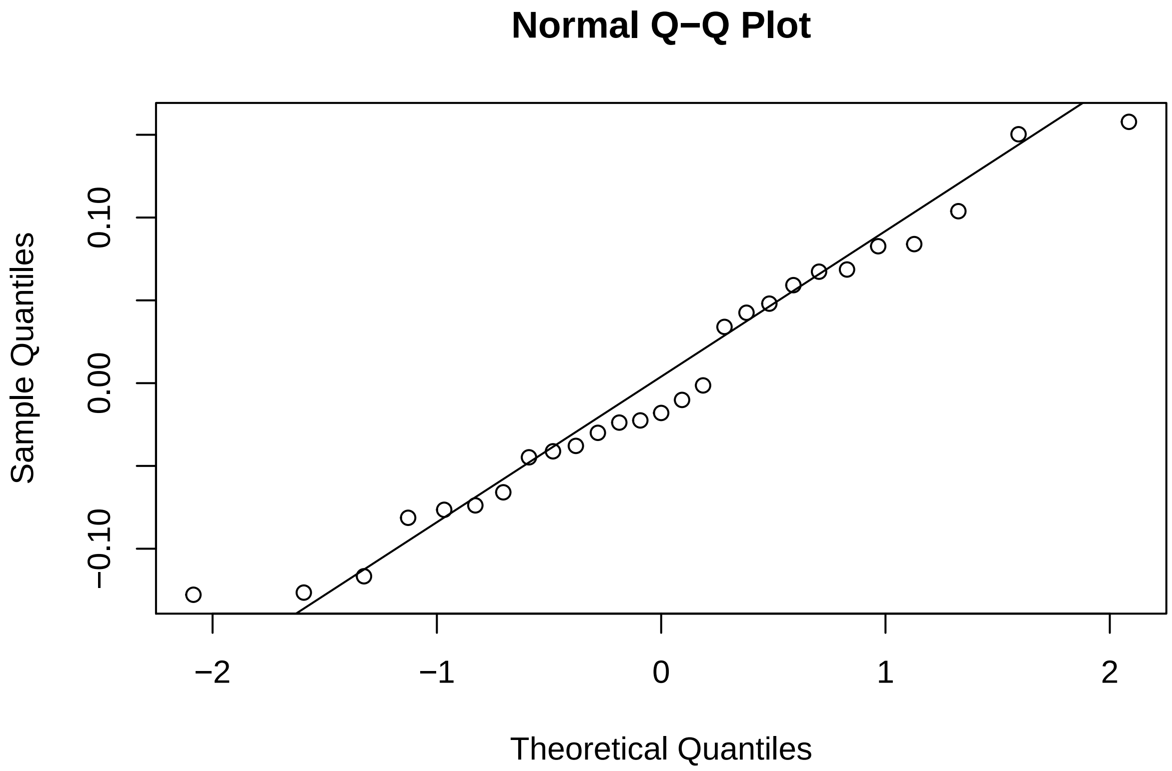
```
fitaf4 <- lm(log(Cycles)~factor(Len) * factor(Load) + factor(Len) * factor(Amp) +
               factor(Load) * factor(Amp), data = strengthWool)
summary(fitaf2)
```

```
##
## Call:
## lm(formula = Cycles ~ factor(Len) * factor(Load) + factor(Len) *
##     factor(Amp) + factor(Load) * factor(Amp), data = strengthWool)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -127.593  -39.148   -9.037   58.074  117.074
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   6.826e+02  9.237e+01   7.390 7.69e-05 ***
## factor(Len)300                7.809e+02  1.161e+02   6.728 0.000148 ***
## factor(Len)350                2.895e+03  1.161e+02  24.946 7.13e-09 ***
## factor(Load)45               -2.041e+02  1.161e+02  -1.759 0.116697
## factor(Load)50               -5.077e+02  1.161e+02  -4.374 0.002368 **
## factor(Amp)9                 -2.944e+02  1.161e+02  -2.537 0.034879 *
## factor(Amp)10                -5.713e+02  1.161e+02  -4.923 0.001160 **
## factor(Len)300:factor(Load)45 -1.003e+02  1.271e+02  -0.789 0.452782
## factor(Len)350:factor(Load)45 -2.593e+02  1.271e+02  -2.040 0.075709 .
## factor(Len)300:factor(Load)50 -3.323e+02  1.271e+02  -2.614 0.030944 *
## factor(Len)350:factor(Load)50 -9.427e+02  1.271e+02  -7.414 7.52e-05 ***
## factor(Len)300:factor(Amp)9   -2.147e+02  1.271e+02  -1.688 0.129813
## factor(Len)350:factor(Amp)9   -1.698e+03  1.271e+02 -13.355 9.45e-07 ***
## factor(Len)300:factor(Amp)10  -4.310e+02  1.271e+02  -3.390 0.009502 **
## factor(Len)350:factor(Amp)10  -1.826e+03  1.271e+02 -14.362 5.40e-07 ***
## factor(Load)45:factor(Amp)9   -4.923e-14  1.271e+02   0.000 1.000000
## factor(Load)50:factor(Amp)9    3.613e+02  1.271e+02   2.842 0.021747 *
## factor(Load)45:factor(Amp)10   1.843e+02  1.271e+02   1.450 0.185155
## factor(Load)50:factor(Amp)10   5.717e+02  1.271e+02   4.496 0.002012 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 110.1 on 8 degrees of freedom
## Multiple R-squared:  0.9952, Adjusted R-squared:  0.9844
## F-statistic: 92.25 on 18 and 8 DF,  p-value: 2.537e-07
```

```r
rSquared = summary(fitaf4)$r.squared

#Store the residuals of the dataset
Residuals <- resid(fitaf4)

# store the fitted values of the regression
Fitted <- fitted.values(fitaf4)

# Combine them into data frame
graph.data <- cbind.data.frame(Residuals, Fitted)

# PLot the graph
ggplot(graph.data, aes(x=Fitted, y=Residuals)) + geom_point() +
ggtitle("Residuals vs Fitted Values") + geom_hline(yintercept = 0)
```



```
# Plot the graph to test the normality
qqnorm(Residuals)
qqline(Residuals)
```



```
# Perform the Shapiro test
shapiro.test(Residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Residuals
## W = 0.96517, p-value = 0.4806
```

As we can see from the summary of the transformed model that $R^2 = 0.9928494$ which means that the predictors of this model explain almost 99% of the variance of our response variable (Cycles). Also, based on the residual plot, we cannot see any patterns, they are randomly scattered. Hence, the error variance seems to be constant, and the relationship is linear (same as the model in part (c)). In addition to that, Based on how very closely the Normal Q-Q plot is forming a shape of a line y=x and the Shapiro-Wilk's test's p-value which is 0.4806 which is very largely greater than 0.05, we can conclude that the residuals are coming from the normal population. Hence, there is no clear difference between this model and model in part (c), in fact, they are closely similar. Hence, in the transformed scale, the model with the interactions is no better than the model with main effects only.