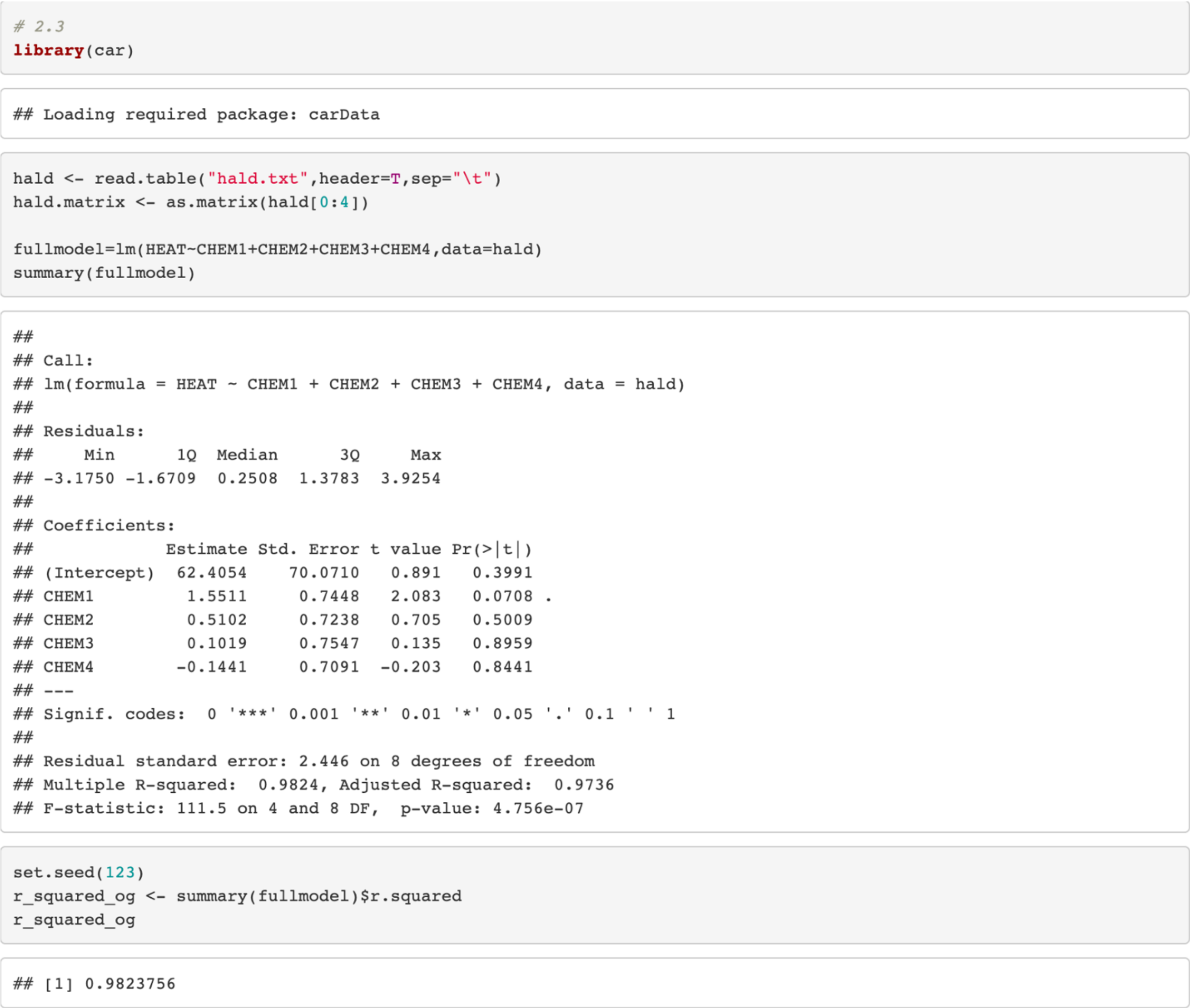
Nhut Cao - 906939

Computer Exercise 2

2.3

First read data from file. From the summary table, we see that R^2 is very high, and none of the explanatory variables is less than 5%.



```
# 2.3
library(car)
```

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

```
hald <- read.table("hald.txt",header=T,sep="\t")
hald.matrix <- as.matrix(hald[0:4])

fullmodel=lm(HEAT~CHEM1+CHEM2+CHEM3+CHEM4,data=hald)
summary(fullmodel)
```

```
##
## Call:
## lm(formula = HEAT ~ CHEM1 + CHEM2 + CHEM3 + CHEM4, data = hald)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1750 -1.6709  0.2508  1.3783  3.9254
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  62.4054    70.0710   0.891   0.3991
## CHEM1         1.5511     0.7448   2.083   0.0708 .
## CHEM2         0.5102     0.7238   0.705   0.5009
## CHEM3         0.1019     0.7547   0.135   0.8959
## CHEM4        -0.1441     0.7091  -0.203   0.8441
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.446 on 8 degrees of freedom
## Multiple R-squared:  0.9824, Adjusted R-squared:  0.9736
## F-statistic: 111.5 on 4 and 8 DF,  p-value: 4.756e-07
```

```
set.seed(123)
r_squared_og <- summary(fullmodel)$r.squared
r_squared_og
```

```
## [1] 0.9823756
```

Now use permutation test to test the full model

```r
# Test full model
var_names <- colnames(hald[0:4])

fit_helper_d <- function(X, y, perm_var) {
  # Permute the values of perm_var
  X[,perm_var] <- sample(X[,perm_var])
  # LS estimate
  beta <- solve((t(X) %*% X)) %*% t(X) %*% y
  # Fitted values
  y_hat <- X %*% beta
  # R^2
  cor(y_hat, y)^2
}

perm_replicator_d <- function(n_perm, X, y, var_name) {
  # Generate n_perm permutation estimates of R^2 for var_name
  replicate(n_perm,  fit_helper_d(X, y, var_name))
}

# Sanity check
n <- nrow(hald)
Intercept <- rep(1, n)
X <- cbind(hald.matrix, Intercept)
y <- hald$HEAT

n_perm <- 2000
alpha <- 0.05

expl_var <- var_names[]
# Compute the permutation estimates for each expl. variable individually
r_squares <- sapply(expl_var, function(name) perm_replicator_d(n_perm, X, y, name))

p_values_perm <- apply(r_squares, 2, function(x) sum(x > r_squared_og)/length(x))
p_values_perm # Not exactly the same as in the model solutions since here things are done in a different order.
```

```
##  CHEM1  CHEM2  CHEM3  CHEM4
## 0.0725 0.4805 0.8900 0.8510
```

```r
p_values_perm < alpha
```

```
## CHEM1 CHEM2 CHEM3 CHEM4
## FALSE FALSE FALSE FALSE
```

This model is not appropriate as its explanatory variables are insignificant, and base on the p-value, CHEM3 and CHEM4 are significantly larger than 5%. So, first I remove CHEM4 and test the new model

```
# Test remove CHEM4
# permutation test for model without chem4
set.seed(123)
model_1 <- lm(HEAT~CHEM1+CHEM2+CHEM3,data=hald)
summary(model_1)
```

```
##
## Call:
## lm(formula = HEAT ~ CHEM1 + CHEM2 + CHEM3, data = hald)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2543 -1.4726  0.1755  1.5409  3.9711
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 48.19363    3.91330  12.315 6.17e-07 ***
## CHEM1        1.69589    0.20458   8.290 1.66e-05 ***
## CHEM2        0.65691    0.04423  14.851 1.23e-07 ***
## CHEM3        0.25002    0.18471   1.354    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.312 on 9 degrees of freedom
## Multiple R-squared:  0.9823, Adjusted R-squared:  0.9764
## F-statistic: 166.3 on 3 and 9 DF,  p-value: 3.367e-08
```

Here the R^2 value is slightly less than the previous value, and we can notice that CHEM1 and CHEM2 are significant here.

Now apply the permutation test for this model

```
var_names <- colnames(hald[0:3])

fit_helper_d <- function(X, y, perm_var) {
  # Permute the values of perm_var
  X[,perm_var] <- sample(X[,perm_var])
  # LS estimate
  beta <- solve((t(X) %*% X)) %*% t(X) %*% y
  # Fitted values
  y_hat <- X %*% beta
  # R^2
  cor(y_hat, y)^2
}

perm_replicator_d <- function(n_perm, X, y, var_name) {
  # Generate n_perm permutation estimates of R^2 for var_name
  replicate(n_perm,  fit_helper_d(X, y, var_name))
}

# Sanity check
n <- nrow(hald)
Intercept <- rep(1, n)
X <- cbind(hald.matrix[,-4], Intercept)
y <- hald$HEAT

n_perm <- 2000
alpha <- 0.05

expl_var <- var_names[]
# Compute the permutation estimates for each expl. variable individually
r_squares <- sapply(expl_var, function(name) perm_replicator_d(n_perm, X, y, name))

p_values_perm <- apply(r_squares, 2, function(x) sum(x > r_squared_og)/length(x))
p_values_perm # Not exactly the same as in the model solutions since here things are done in a different order.
```

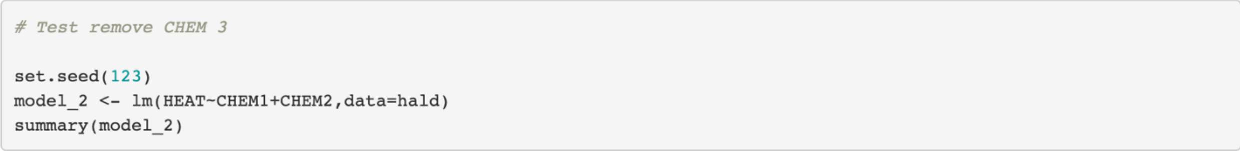
```
##  CHEM1  CHEM2  CHEM3
## 0.0000 0.0000 0.2065
```

```
p_values_perm < alpha
```

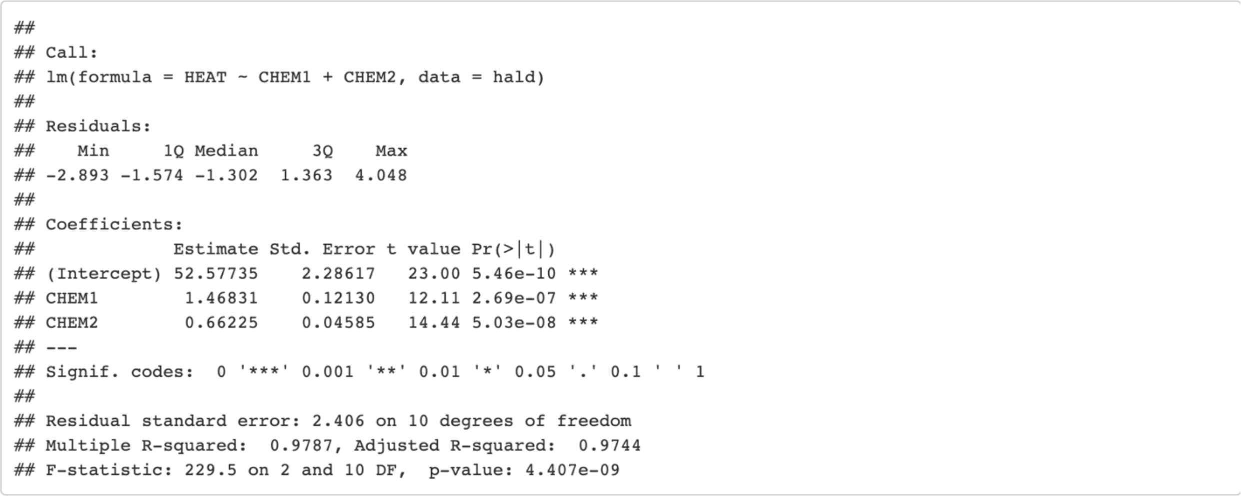
```
## CHEM1 CHEM2 CHEM3
##  TRUE  TRUE FALSE
```

This model is better than the full model, as there is only one variable CHEM3 is not significant. We continue to remove CHEM3 and test model with only CHEM1, CHEM2

With this linear model, the R^2 value is less than these previous values, and both CHEM1 and CHEM2 show that they are significant.



```
# Test remove CHEM 3

set.seed(123)
model_2 <- lm(HEAT~CHEM1+CHEM2,data=hald)
summary(model_2)
```



```
##
## Call:
## lm(formula = HEAT ~ CHEM1 + CHEM2, data = hald)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.893 -1.574 -1.302  1.363  4.048
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 52.57735    2.28617   23.00 5.46e-10 ***
## CHEM1        1.46831    0.12130   12.11 2.69e-07 ***
## CHEM2        0.66225    0.04585   14.44 5.03e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.406 on 10 degrees of freedom
## Multiple R-squared:  0.9787, Adjusted R-squared:  0.9744
## F-statistic: 229.5 on 2 and 10 DF,  p-value: 4.407e-09
```

Now test the model using permutation test

```r
var_names <- colnames(hald[0:2])
fit_helper_d <- function(X, y, perm_var) {
  # Permute the values of perm_var
  X[,perm_var] <- sample(X[,perm_var])
  # LS estimate
  beta <- solve((t(X) %*% X)) %*% t(X) %*% y
  # Fitted values
  y_hat <- X %*% beta
  # R^2
  cor(y_hat, y)^2
}

perm_replicator_d <- function(n_perm, X, y, var_name) {
  # Generate n_perm permutation estimates of R^2 for var_name
  replicate(n_perm,  fit_helper_d(X, y, var_name))
}

# Sanity check
n <- nrow(hald)
Intercept <- rep(1, n)
X <- cbind(hald.matrix[,0:2], Intercept)
y <- hald$HEAT

n_perm <- 2000
alpha <- 0.05

expl_var <- var_names[]
# Compute the permutation estimates for each expl. variable individually
r_squares <- sapply(expl_var, function(name) perm_replicator_d(n_perm, X, y, name))

p_values_perm <- apply(r_squares, 2, function(x) sum(x > r_squared_og)/length(x))
p_values_perm # Not exactly the same as in the model solutions since here things are done in a different order.
```

```
## CHEM1 CHEM2
##     0     0
```

```r
p_values_perm < alpha
```

```
## CHEM1 CHEM2
##  TRUE  TRUE
```

Now the model is good, all the variables are significant with the level of significance 5%.

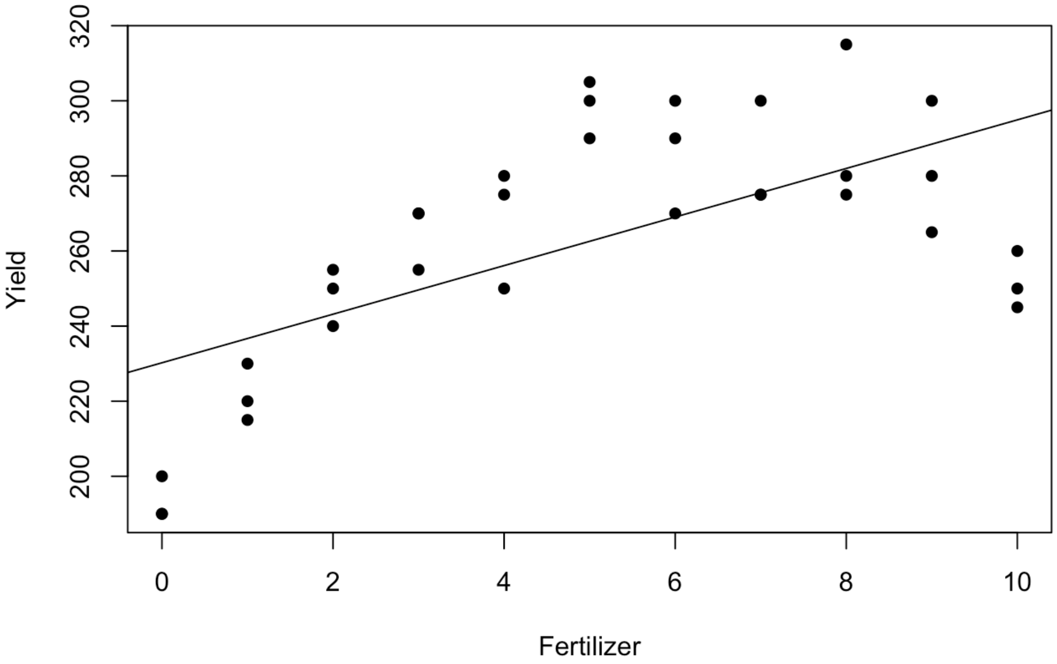
2.4

a)

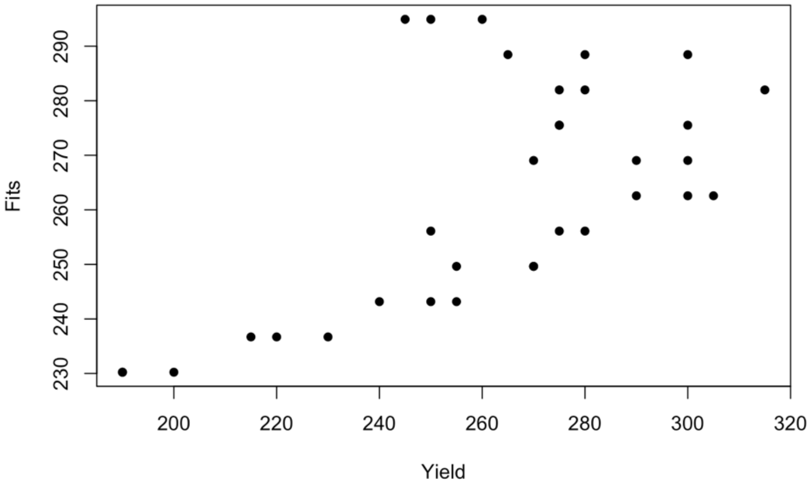
```r
# 2.4

crop_data <- read.table("crop.txt", header=T, sep='\t')

set.seed(123)

# a
crop_lm <- lm(Yield~Fertilizer, data=crop_data)
summary(crop_lm)
```

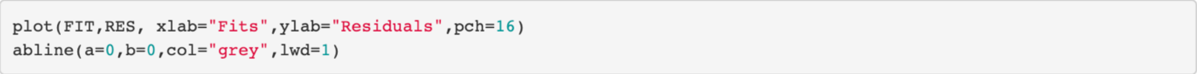
```
##
## Call:
## lm(formula = Yield ~ Fertilizer, data = crop_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.924 -16.697  -0.515  20.364  42.424
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  230.227      8.384  27.462  < 2e-16 ***
## Fertilizer     6.470      1.417   4.565 7.43e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.74 on 31 degrees of freedom
## Multiple R-squared:  0.402,  Adjusted R-squared:  0.3828
## F-statistic: 20.84 on 1 and 31 DF,  p-value: 7.432e-05
```

```r
FIT <- fitted(crop_lm)
RES <- resid(crop_lm)

plot(crop_data$Fertilizer,crop_data$Yield, ylab="Yield",xlab="Fertilizer",pch=16)
abline(crop_lm)
```

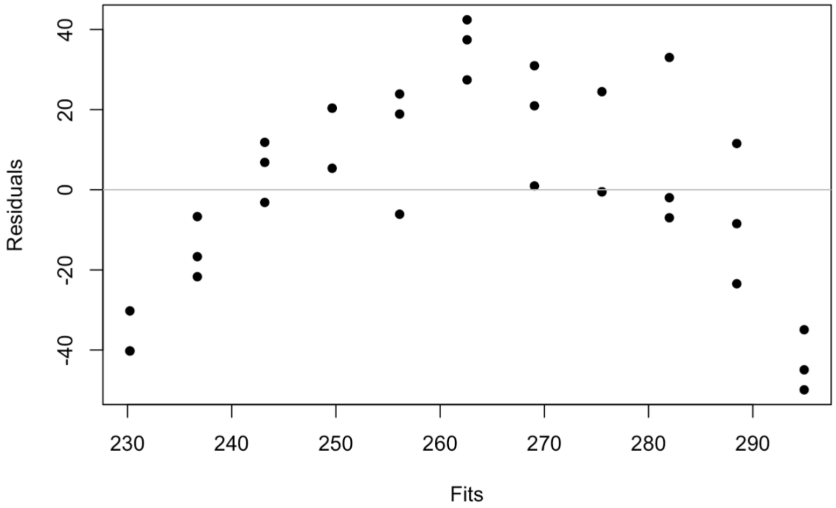


```
plot(crop_data$Yield,FIT, ylab="Fits",xlab="Yield",pch=16)
```





```
plot(FIT,RES, xlab="Fits",ylab="Residuals",pch=16)
abline(a=0,b=0,col="grey",lwd=1)
```



Base on these three plots, we can see that the scatter plots illustrate the goodness of the model:

- ➢ The closer the points are to the line, the better the model is
- ➢ Outliers are easily seen

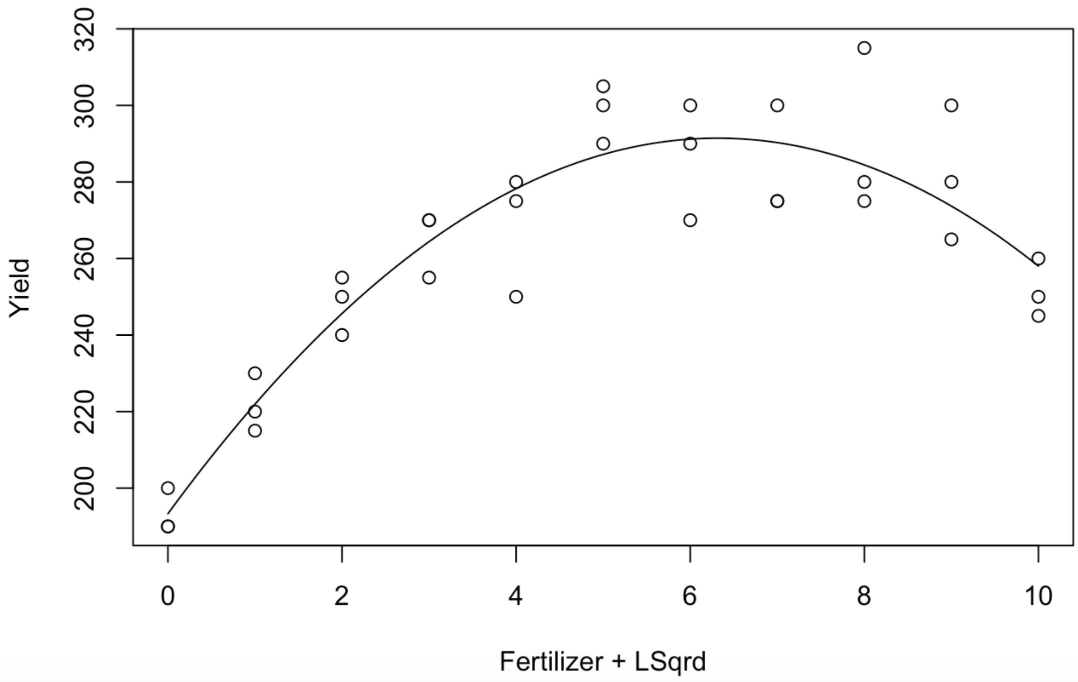
However, in this model, we can see that the points are not close to the line, and there are too many outliers. Hence this model is not sufficient.

b)

```
# b
set.seed(123)
crop_lm2 <- lm(Yield~Fertilizer+LSqrd, data=crop_data)
summary(crop_lm2)
```

```
##
## Call:
## lm(formula = Yield ~ Fertilizer + LSqrd, data = crop_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.256  -8.007  -1.196   6.690  30.554
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 193.3100     5.6511  34.207  < 2e-16 ***
## Fertilizer   31.0812     2.6292  11.821 8.11e-13 ***
## LSqrd        -2.4611     0.2532  -9.719 8.85e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.85 on 30 degrees of freedom
## Multiple R-squared:  0.8559, Adjusted R-squared:  0.8463
## F-statistic: 89.07 on 2 and 30 DF,  p-value: 2.407e-13
```

```
x <- crop_data$Fertilizer # x^2 = LSqrd
beta_1 <- 31.0812
beta_2 <- -2.4611
const_term <- 193.3100

plot(crop_data$Fertilizer, crop_data$Yield, xlab="Fertilizer + LSqrd",ylab="Yield")
curve((const_term + beta_1*x + beta_2*x^2), from=0, to=10, ylab='Yield', xlab="Fertilizer + LSqrd", add=TRUE)
```



The plot above illustrate the goodness of the model:

- The closer the points are to the line, the better the model is
- Outliers are easily seen

This plot shows that this model is good, as the points are close to the fit line, and the outliers are also visible. Hence, this model is sufficient

c)

Base on the results obtained from a and b, the model in part b is more suitable.