# 205615894_stats101a_hw8

Takao

3/4/2022

## 1

```
inverter_data <- read.csv("inverter.csv", row.names = 1)
head(inverter_data)
```

```
##   x1 x2 x3 x4 x5     y
## 1  3  3  3  3  0 0.787
## 2  8 30  8  8  0 0.293
## 3  3  6  6  6  0 1.710
## 4  4  4  4 12  0 0.203
## 5  8  7  6  5  0 0.806
## 6 10 20  5  5  0 4.713
```

### (a)

```
mlr.model1 <- lm(y~x1+x2+x3+x4+x5, data = inverter_data)
summary(mlr.model1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5, data = inverter_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2915 -1.0794 -0.5519  1.2685  3.5009
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.85473    1.86922   1.527   0.1432
## x1          -0.29047    0.11742  -2.474   0.0230 *
## x2           0.20572    0.07506   2.741   0.0130 *
## x3           0.45444    0.18768   2.421   0.0256 *
## x4          -0.59419    0.21253  -2.796   0.0115 *
## x5           0.00464    0.01817   0.255   0.8012
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.196 on 19 degrees of freedom
## Multiple R-squared:  0.5584, Adjusted R-squared:  0.4422
## F-statistic: 4.805 on 5 and 19 DF,  p-value: 0.005239
```

This F-statistic is shown at the bottom of the output above. The p-value is calculated using a reference distribution F with 5 and 19 degrees of freedom. At a level of $\alpha = 0.01$, we *reject* the null hypothesis since the p-value is 0.00523865038 which is smaller than this value of $\alpha$. Therefore, we conclude that at least one of the predictors in the model has a significant explanatory power.

### (b)

```
summary(mlr.model1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5, data = inverter_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2915 -1.0794 -0.5519  1.2685  3.5009
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.85473    1.86922   1.527   0.1432
## x1          -0.29047    0.11742  -2.474   0.0230 *
## x2           0.20572    0.07506   2.741   0.0130 *
## x3           0.45444    0.18768   2.421   0.0256 *
## x4          -0.59419    0.21253  -2.796   0.0115 *
## x5           0.00464    0.01817   0.255   0.8012
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.196 on 19 degrees of freedom
## Multiple R-squared:  0.5584, Adjusted R-squared:  0.4422
## F-statistic: 4.805 on 5 and 19 DF,  p-value: 0.005239
```

1. The width of the NMOS device does have a significant association with the transient point in volts, when the predictors length of the NMOS device,width of the PMOS device, length of the PMOS device, and the temperature in celcius are all in the model.
2. The length of the NMOS device does have a significant association with the transient point in volts, when the predictors width of the NMOS device, width of the PMOS device, length of the PMOS device, and the temperature in celcius are all in the model.
3. The width of the PMOS device does have a significant association with the transient point in volts, when the predictors width of the NMOS device, length of the NMOS device, length of the PMOS device, and the temperature in celcius are all in the model.
4. The length of the PMOS device does have a significant association with the transient point in volts, when the predictors width of the NMOS device, length of the NMOS device, width of the PMOS device, and the temperature in celcius are all in the model.
5. The temperatrue in celcius does not have a significant association with the transient point in volts, when the predictors width of the NMOS device, length of the NMOS device, width of the PMOS device, and the length of the PMOS device are all in the model.

### (c)

```
mlr.model1mod <- lm(y~x1+x2+x3+x4, data = inverter_data)
summary(mlr.model1mod)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4, data = inverter_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2806 -1.1030 -0.6715  1.2499  3.5333
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.14825    1.43891   2.188  0.04072 *
## x1          -0.28999    0.11463  -2.530  0.01992 *
## x2           0.19919    0.06891   2.891  0.00904 **
## x3           0.45537    0.18321   2.486  0.02190 *
## x4          -0.60919    0.19942  -3.055  0.00625 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.144 on 20 degrees of freedom
## Multiple R-squared:  0.5569, Adjusted R-squared:  0.4683
## F-statistic: 6.284 on 4 and 20 DF,  p-value: 0.001917
```

This F-statistic is shown at the bottom of the output above. The p-value is calculated using a reference distribution F with 4 and 20 degrees of freedom. At a level of $\alpha = 0.01$, we *reject* the null hypothesis since the p-value is 0.00191658113 which is smaller than this value of $\alpha$. Therefore, we conclude that at least one of the predictors in the model has a significant explanatory power.

1. The width of the NMOS device does have a significant association with the transient point in volts, when the predictors length of the NMOS device,width of the PMOS device, and length of the PMOS device are all in the model.
2. The length of the NMOS device does have a significant association with the transient point in volts, when the predictors width of the NMOS device,width of the PMOS device, and length of the PMOS device are all in the model.
3. The width of the PMOS device device does have a significant association with the transient point in volts, when the predictors width of the NMOS device, length of the NMOS device, and length of the PMOS device are all in the model.
4. The length of the PMOS device device does have a significant association with the transient point in volts, when the predictors width of the NMOS device, length of the NMOS device, and width of the PMOS device are all in the model.

### (d)

```
anova(mlr.model1)
```

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

```
##
## Response: y
##           Df Sum Sq Mean Sq F value   Pr(>F)
## x1         1 13.045  13.045  2.7062 0.116401
## x2         1 47.708  47.708  9.8972 0.005320 **
## x3         1 11.871  11.871  2.4627 0.133082
## x4         1 42.879  42.879  8.8954 0.007652 **
## x5         1  0.314   0.314  0.0652 0.801250
## Residuals 19 91.587   4.820
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
anova(mlr.model1mod)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value   Pr(>F)
## x1         1 13.045  13.045  2.8389 0.107548
## x2         1 47.708  47.708 10.3825 0.004274 **
## x3         1 11.871  11.871  2.5835 0.123658
## x4         1 42.879  42.879  9.3316 0.006254 **
## Residuals 20 91.901   4.595
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

As we can see, the MSE decreased from the original model to the reduced model. This intuitively makes sense because the mean squared error decreases as the model error decreases (as the model has less error, the data points fall closer to the regression line, thus decreasing the MSE). In part c, we deleted x5 from the model which was proven to have no significant association with transient point in volts. Thus, by removing x5, an “insignificant” regressor, the MSE decreased, which indicates that the model error decreased.

### (e)

```
rstudent(mlr.model1mod)
```

```
##           1           2           3           4           5           6
## -0.80198823 -4.99898332 -0.39957789  2.22882813 -0.52268047  0.62841782
##           7           8           9          10          11          12
## -0.45288375  2.21003091  1.37195958 -0.42874767  0.75433789 -0.32058755
##          13          14          15          16          17          18
## -0.36965812  1.91673223  0.08151476 -0.69907607 -0.79464770  0.27842284
##          19          20          21          22          23          24
##  0.59987038  0.59608894 -0.12246789  0.71898132 -0.73233609 -0.82617019
##          25
## -0.45518236
```

```
rstudent(mlr.model1)
```

```
##           1           2           3           4           5           6
## -0.74636279 -4.91248064 -0.32675792  2.35943253 -0.45105537  0.67428946
##           7           8           9          10          11          12
## -0.40190965  2.13768006  1.39994211 -0.39004841  0.73613919 -0.26336583
##          13          14          15          16          17          18
## -0.36941593  1.85541149  0.02927331 -0.67101105 -0.77164762  0.23863358
##          19          20          21          22          23          24
##  0.57796276  0.53281741 -0.17930364  0.65676465 -0.81439954 -0.93072036
##          25
## -0.57815006
```

Looking at the output given above, it seems like observation 2 seems unusually large. Additionally, observation 4 and 8 can possibly be unusually large as well.

### (f)

```
inverter_data_mod <- inverter_data[-2, ]
inverter_data_mod
```

```
##    x1 x2 x3 x4 x5     y
## 1   3  3  3  3  0 0.787
## 3   3  6  6  6  0 1.710
## 4   4  4  4 12  0 0.203
## 5   8  7  6  5  0 0.806
## 6  10 20  5  5  0 4.713
## 7   8  6  3  3 25 0.607
## 8   6 24  4  4 25 9.107
## 9   4 10 12  4 25 9.210
## 10 16 12  8  4 25 1.365
## 11  3 10  8  8 25 4.554
## 12  8  3  3  3 25 0.293
## 13  3  6  3  3 50 2.252
## 14  3  8  8  3 50 9.167
## 15  4  8  4  8 50 0.694
## 16  5  2  2  2 50 0.379
## 17  2  2  2  3 50 0.485
## 18 10 15  3  3 50 3.345
## 19 15  6  2  3 50 0.208
## 20 15  6  2  3 75 0.201
## 21 10  4  3  3 75 0.329
## 22  3  8  2  2 75 4.966
## 23  6  6  6  4 75 1.362
## 24  2  3  8  6 75 1.515
## 25  3  3  8  8 75 0.751
```

```
mlr.model1modmod <- lm(y ~ x1 + x2 + x3 + x4, data = inverter_data_mod)
summary(mlr.model1modmod)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4, data = inverter_data_mod)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -1.4894 -0.9324 -0.6098  0.7224  3.3659 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  1.52430    1.02317   1.490 0.152692    
## x1          -0.30606    0.07736  -3.956 0.000847 ***
## x2           0.37439    0.05820   6.433 3.63e-06 ***
## x3           0.44957    0.12354   3.639 0.001746 ** 
## x4          -0.46557    0.13750  -3.386 0.003102 ** 
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 1.445 on 19 degrees of freedom
## Multiple R-squared:  0.8044, Adjusted R-squared:  0.7632 
## F-statistic: 19.53 on 4 and 19 DF,  p-value: 1.604e-06
```

The Multiple R squared for this model is 0.804369567 which is considerably higher than the multiple R squared for the original model which was 0.558413452, and 0.556898857 for the modified model in part c. This is because the second observation was a bad leverage point and thus, when utilizing this data point in the model to construct linear models, the model was highly influenced by this one point and thus caused the multiple R squared to go down. The newly modified model with the removed observation has a significantly better fit using the linear model compared to the other two models.

### (g)

```
summary(mlr.model1modmod)
```

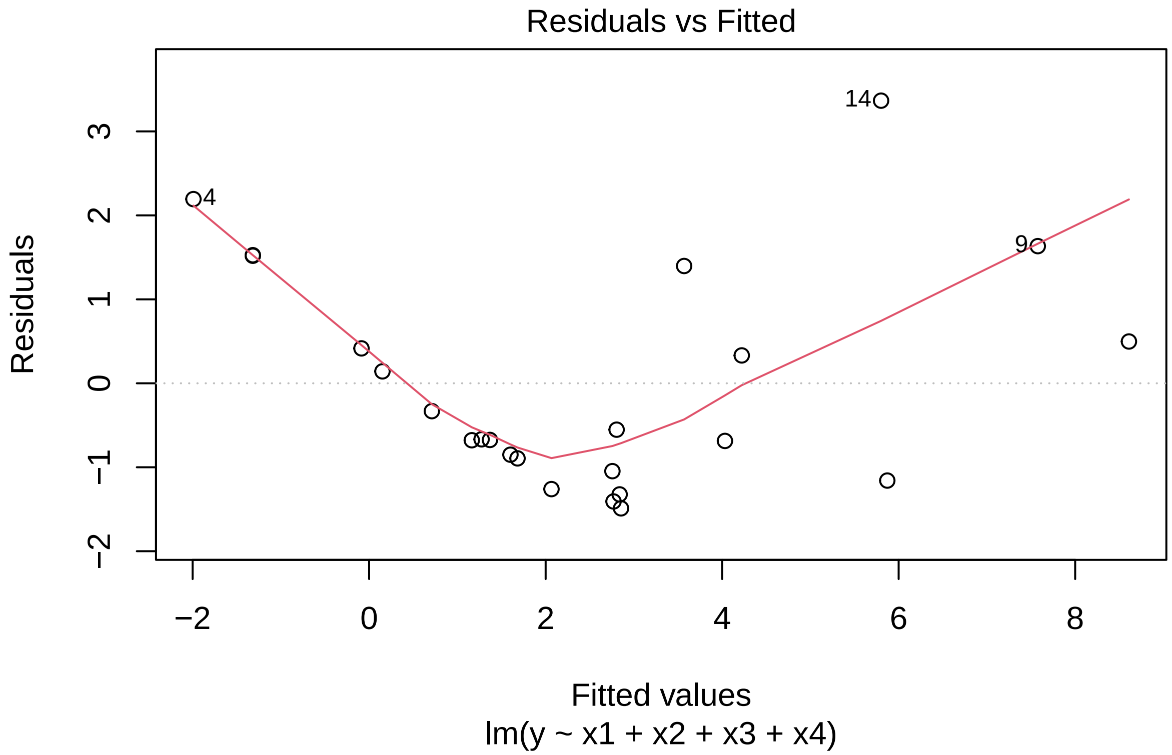
```
## 
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4, data = inverter_data_mod)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -1.4894 -0.9324 -0.6098  0.7224  3.3659 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  1.52430    1.02317   1.490 0.152692    
## x1          -0.30606    0.07736  -3.956 0.000847 ***
## x2           0.37439    0.05820   6.433 3.63e-06 ***
## x3           0.44957    0.12354   3.639 0.001746 ** 
## x4          -0.46557    0.13750  -3.386 0.003102 ** 
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 1.445 on 19 degrees of freedom
## Multiple R-squared:  0.8044, Adjusted R-squared:  0.7632 
## F-statistic: 19.53 on 4 and 19 DF,  p-value: 1.604e-06
```

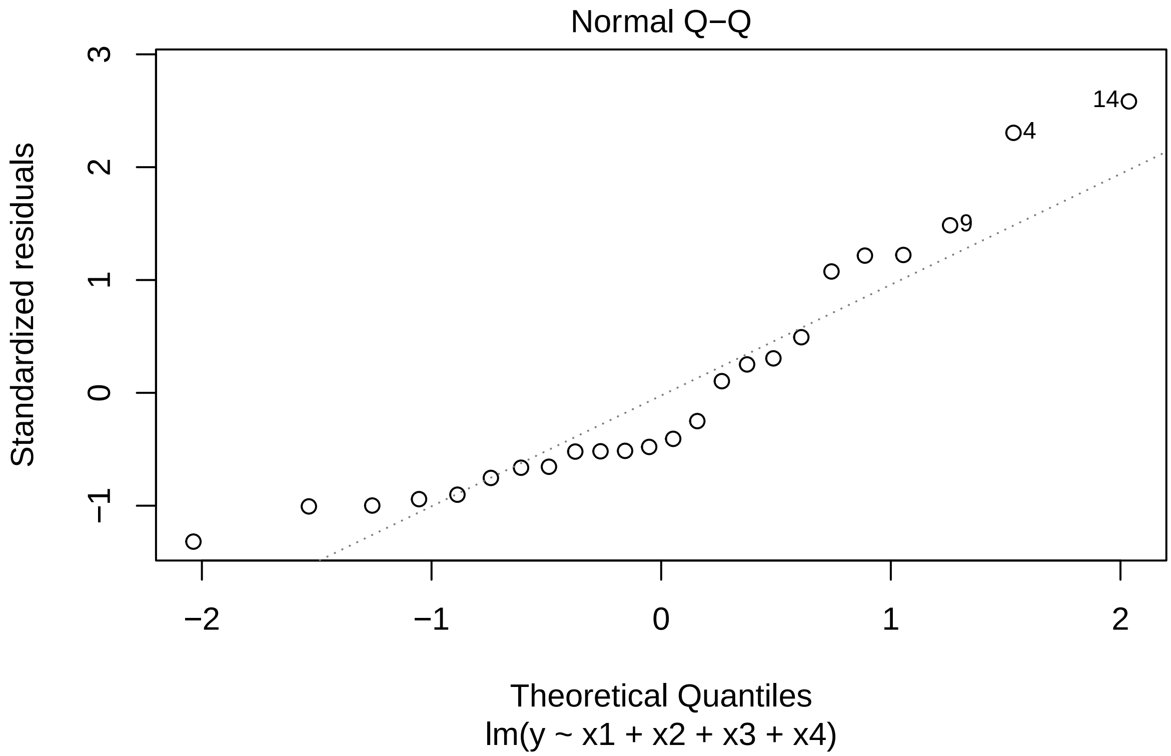
This F-statistic is shown at the bottom of the output above. The p-value is calculated using a reference distribution F with 4 and 19 degrees of freedom. At a level of $\alpha = 0.01$, we *reject* the null hypothesis since the p-value is 1.60411554e-06 which is smaller than this value of $\alpha$. Therefore, we conclude that at least one of the predictors in the model has a significant explanatory power.

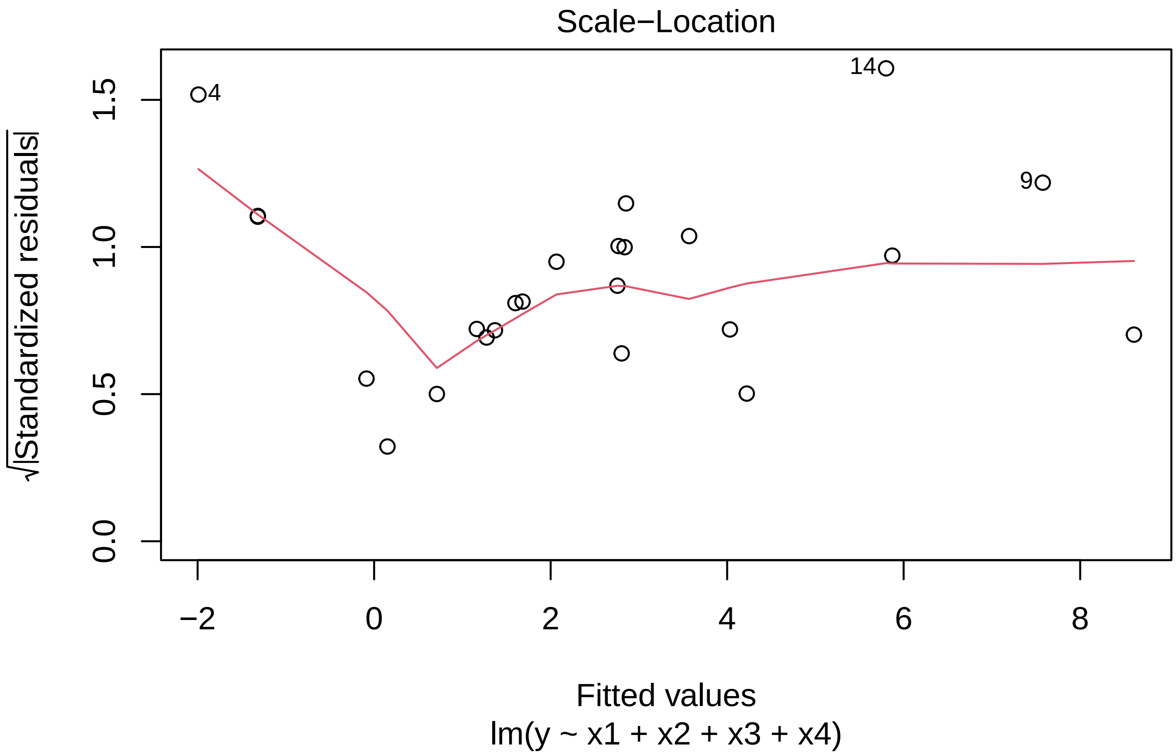
1. The width of the NMOS device does have a significant association with the transient point in volts, when the predictors length of the NMOS device,width of the PMOS device, and length of the PMOS device are all in the model.
2. The length of the NMOS device does have a significant association with the transient point in volts, when the predictors width of the NMOS device,width of the PMOS device, and length of the PMOS device are all in the model.
3. The width of the PMOS device device does have a significant association with the transient point in volts, when the predictors width of the NMOS device, length of the NMOS device, and length of the PMOS device are all in the model.
4. The length of the PMOS device device does have a significant association with the transient point in volts, when the predictors width of the NMOS device, length of the NMOS device, and width of the PMOS device are all in the model.

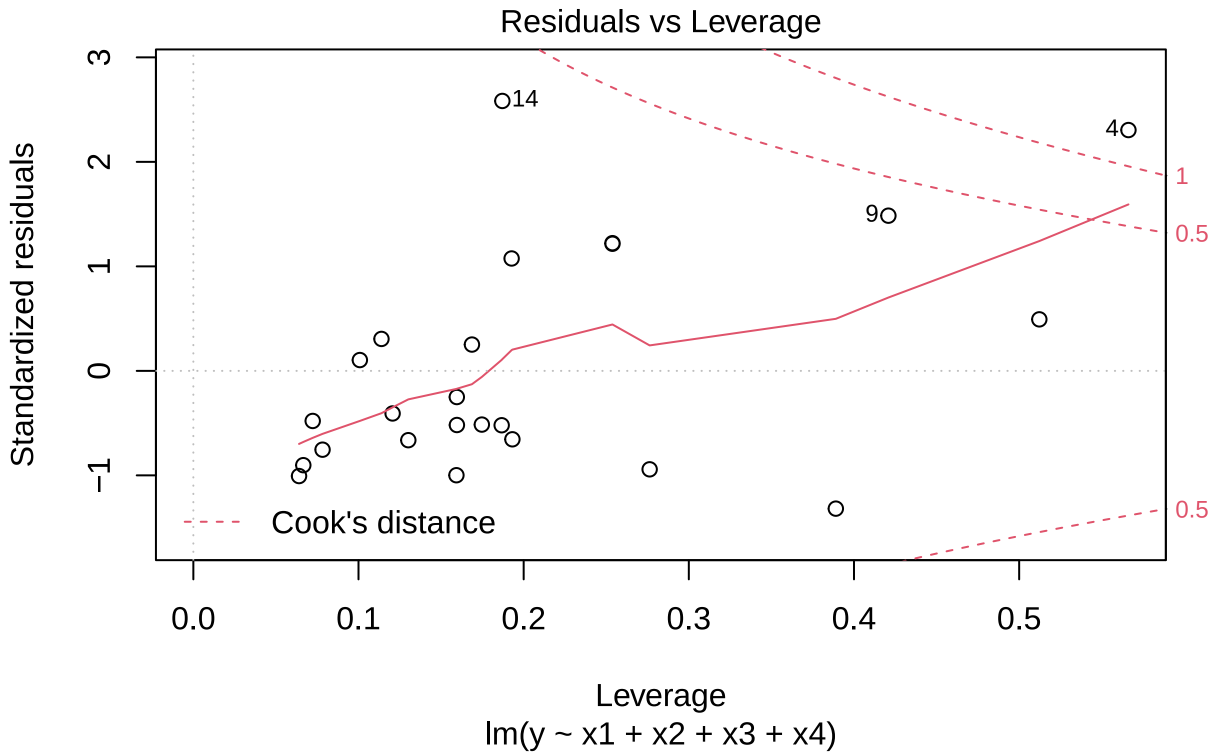
**(h)**

```
plot(mlr.model1modmod)
```



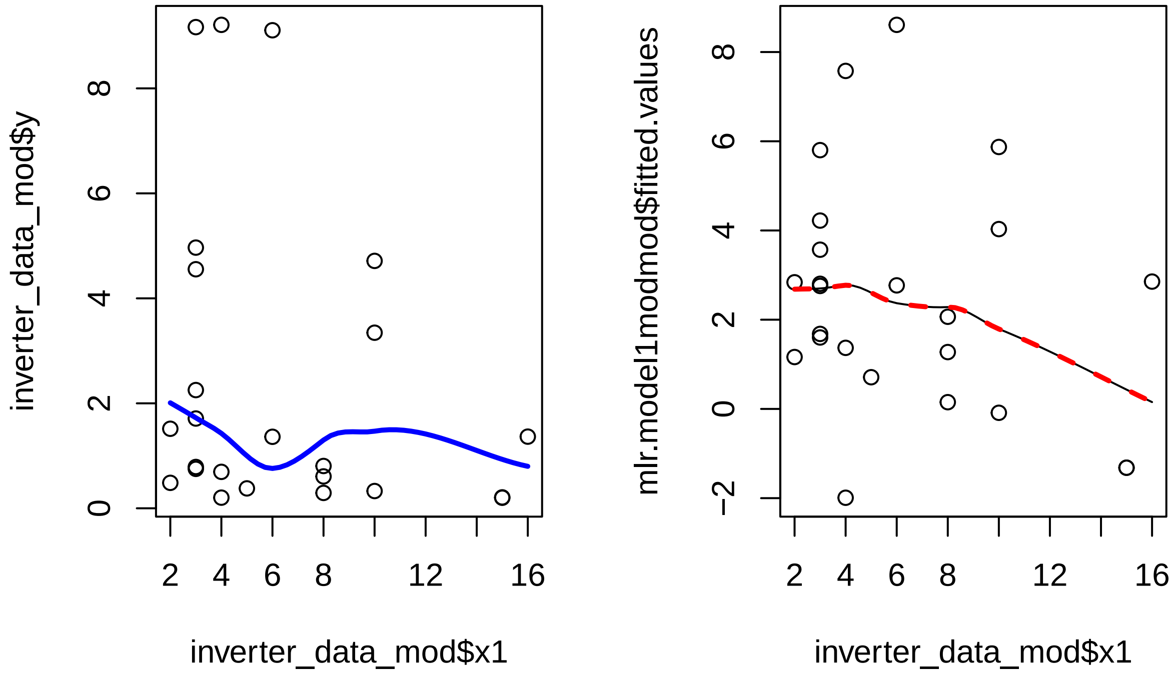

Normal Q−Q
Standardized residuals
3
2
1
0
−1
14
4
9
−2
−1
0
1
2
Theoretical Quantiles
lm(y ~ x1 + x2 + x3 + x4)


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
−2
0
2
4
6
8
4
14
9
Fitted values
lm(y ~ x1 + x2 + x3 + x4)



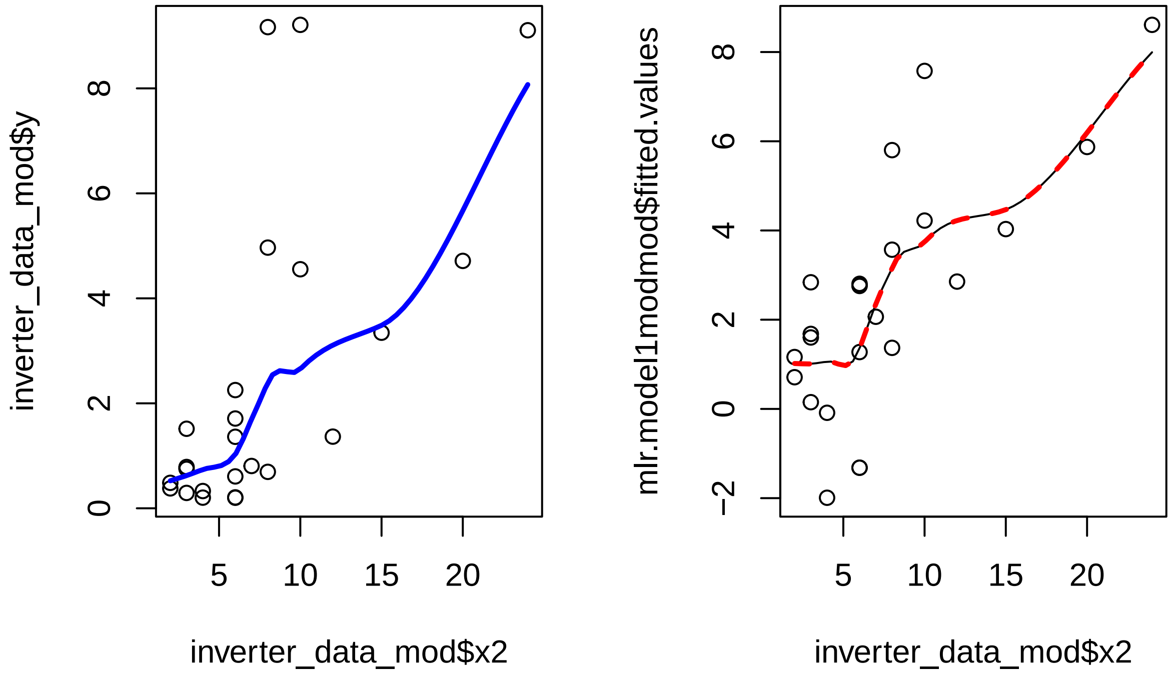
2*4/25 A point is considered a high leverage point if their levarage score is greater than 0.32. There seems to be four high levarage points in this data set. Now we look at the scale-location plot to evaluate the assumption of constant variance of the errors. There seems to be a clear trend/pattern looking at the red line in this plot. Not considering the high levarage points, we look at the residuals vs fitted plot, and this plot shows the same conclusion where there is a clear trend in the red line. Thus, we can say that the assumption of constant variance is not met.

```
par(mfrow=c(1,2))
plot(inverter_data_mod$x1, inverter_data_mod$y)
lines(loess.smooth(inverter_data_mod$x1, inverter_data_mod$y), lwd = 2.5, col = "blue")
scatter.smooth(inverter_data_mod$x1, mlr.model1modmod$fitted.values)
lines(loess.smooth(inverter_data_mod$x1, mlr.model1modmod$fitted.values), lwd = 2.5, col = "red", lty =
```



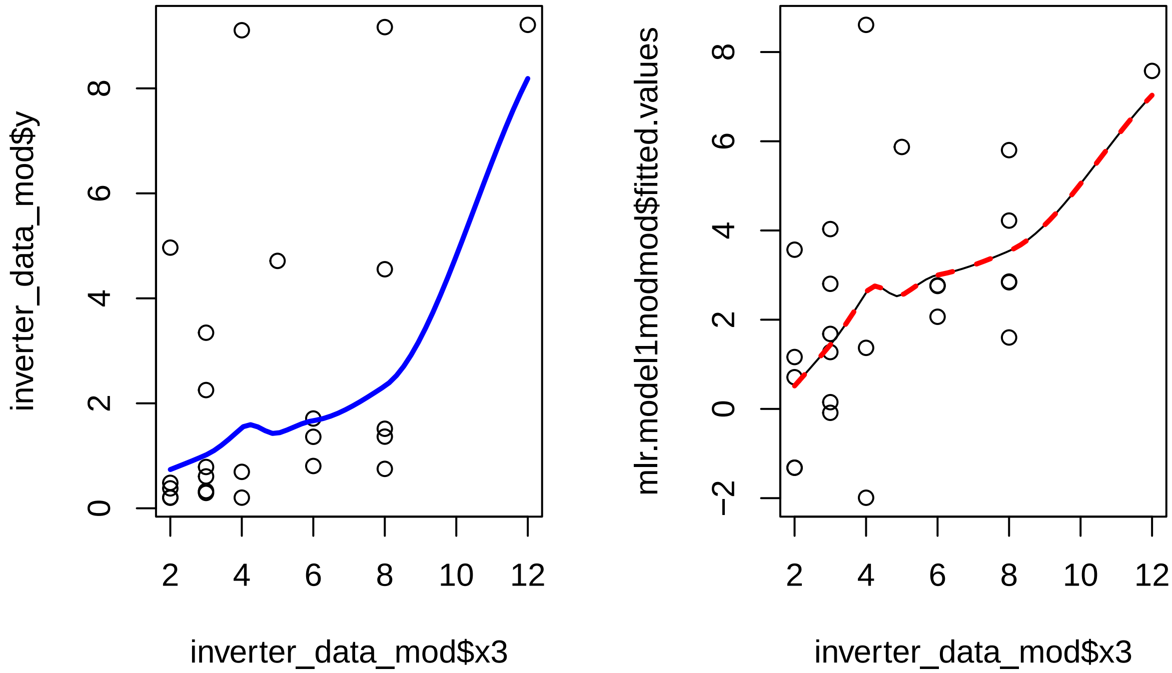
The two plots agree with each other, thus, the model is correctly capturing the relationship between y and x1.

```
par(mfrow=c(1,2))
plot(inverter_data_mod$x2, inverter_data_mod$y)
lines(loess.smooth(inverter_data_mod$x2, inverter_data_mod$y), lwd = 2.5, col = "blue")
scatter.smooth(inverter_data_mod$x2, mlr.model1modmod$fitted.values)
lines(loess.smooth(inverter_data_mod$x2, mlr.model1modmod$fitted.values), lwd = 2.5, col = "red", lty =
```



The two plots agree with each other, thus, the model is correctly capturing the relationship between y and x2.

```
par(mfrow=c(1,2))
plot(inverter_data_mod$x3, inverter_data_mod$y)
lines(loess.smooth(inverter_data_mod$x3, inverter_data_mod$y), lwd = 2.5, col = "blue")
scatter.smooth(inverter_data_mod$x3, mlr.model1modmod$fitted.values)
lines(loess.smooth(inverter_data_mod$x3, mlr.model1modmod$fitted.values), lwd = 2.5, col = "red", lty =
```



The two plots agree with each other, thus, the model is correctly capturing the relationship between y and x3.

```
par(mfrow=c(1,2))
plot(inverter_data_mod$x4, inverter_data_mod$y)
lines(loess.smooth(inverter_data_mod$x4, inverter_data_mod$y), lwd = 2.5, col = "blue")
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## pseudoinverse used at 3
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## neighborhood radius 1
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## There are other near singularities as well. 1
```

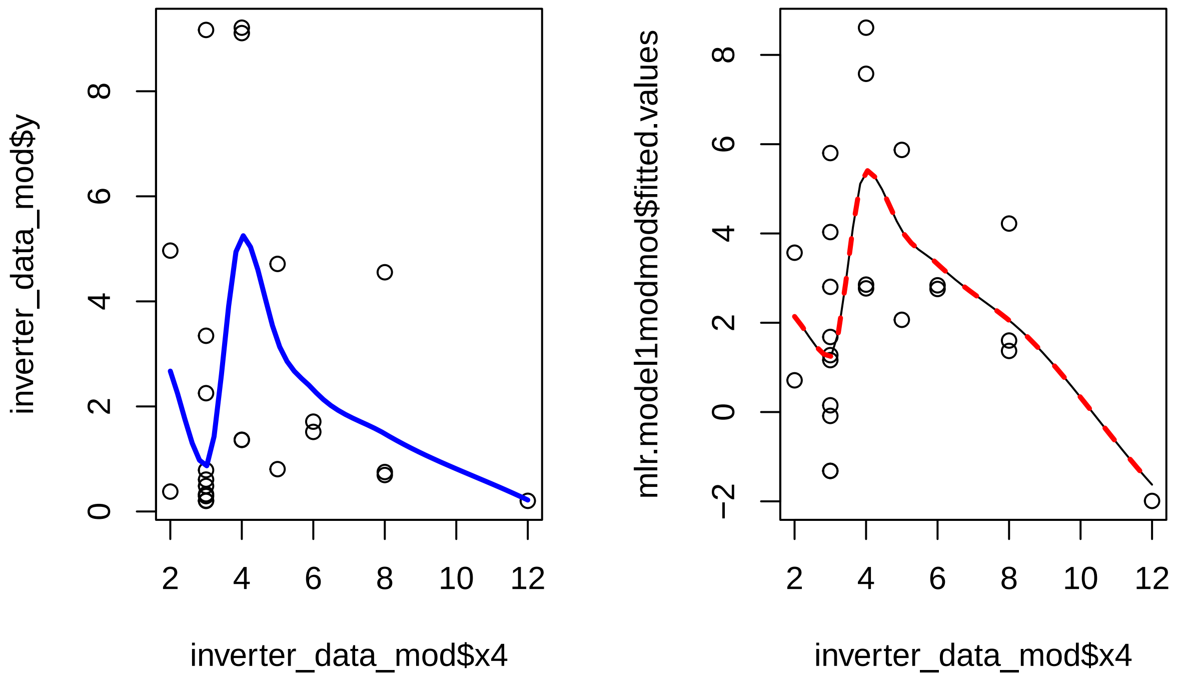
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## pseudoinverse used at 3
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## neighborhood radius 1
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## pseudoinverse used at 3

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## neighborhood radius 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FALSE, :
## There are other near singularities as well. 1
```



The two plots agree with each other, thus, the model is correctly capturing the relationship between y and x4.

## 2

1. I really was able to analytically determine the relationship between two variables. For example, when I took AP statistics in high school, I learned about one sample t tests and one sample proportion tests, however, I was not able to get a complete grasp of the concepts as I basically was mimicing what I have dow during lectures. Through this course, I was really about to understand the real-world applications and how different testing methods actually works and takes place.

2. I learned about regression diagnostics. Although there could be significance in the made conclusinos initially, when an individual closely looks at and analyzes the asssumptions made, these conclusions can

turn out to be invalid. For example, unless closely analyzing the assumption of constant variacnes using plots of residuals, the complete model can turn useless and additionally, there are mutiple methods to complete these analytical procedures.

3. I was able to learn about applicable R codes and techniques. Until I took this course, I was only educated with the basic R codes and simple functions. Through this course, I was able to really dig into certain functions such as lm() and plot() and anova(). I was able to touch the surface of what to do as a data analysist or data scientist performing tests for significant of regressions and correlations.

Thank you so much for an amazing quarter!