# Assignment8

## #question1

library(tidyverse)

library(GGally)

diamonds <- read\_csv("diamonds.csv")

set.seed(17053777)

my\_diamonds <- diamonds %>%

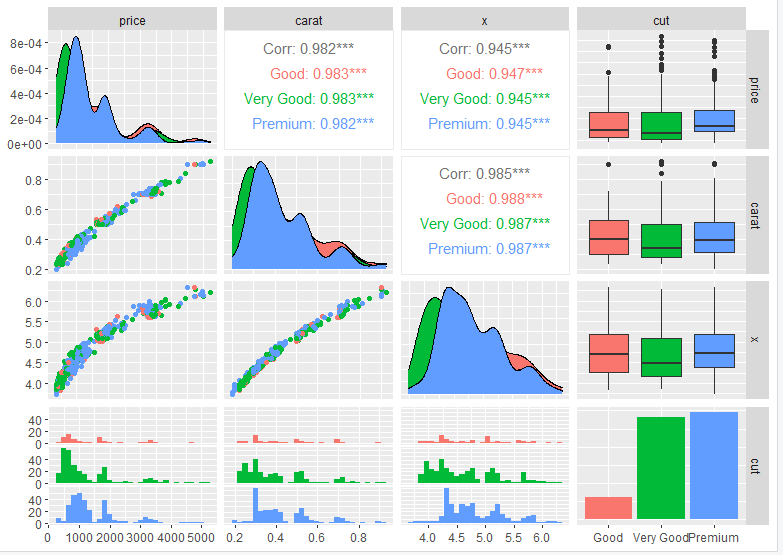
mutate(cut = factor(cut, levels = c("Good", "Very Good", "Premium"))) %>%

sample\_n(640)

## # question2

library(GGally)

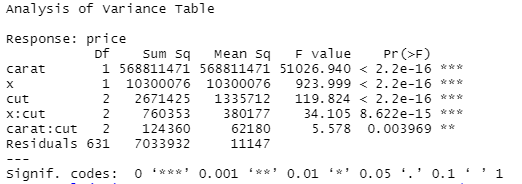
ggpairs(my\_diamonds, columns = c("price","carat", "x", "cut"), aes(colour = cut))



## #question3

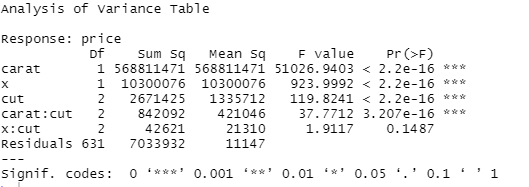
m1 <- lm(price ~ carat + x + cut + x:cut + carat:cut, data = my\_diamonds)

anova(m1)



m2 <- lm(price ~ carat + x + cut + carat:cut + x:cut, data = my\_diamonds)

anova(m2)

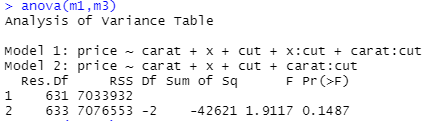


# the interaction between x and cut is 8.622 \* 10^-16 which is significant in m1 since p < 0.05, but in m2, the interaction between x and cut is 0.1487 which it is not significant since p > 0.05.

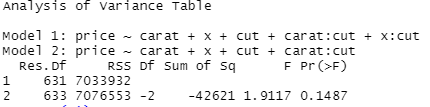
## #question4

m3 <- lm(price ~ carat + x + cut + carat:cut, data = my\_diamonds)

anova(m1,m3)



anova(m2,m3)



# m3 is the nested model of m1 and m2. Comparing anova output for (m1, m3) and (m2, m3), we can observe that the output is the same, which means m1 is the same as m2, only with different coefficient.

# H0: there is no interaction between x and cut

# And from the anova(m1, m3), the p value is 0.1487 which is greater than 0.05,so x:cut is not significant. We don’t reject H0, we accept H0, which means the interaction between x and cut is not making any difference.

## #question5

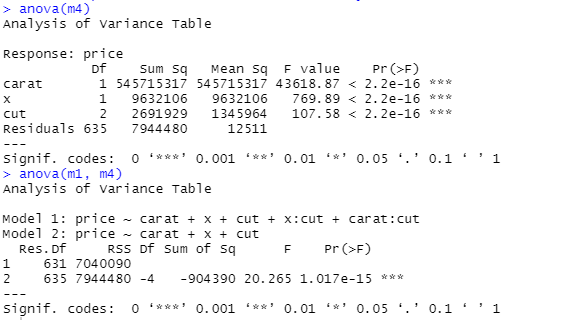
# if carat:cut is expressed earlier than x:cut, then there is no too much for x:cut to express, since caract:cut is sort of including x:cut. That’s why the p-value for x:cut in m1 is significant, but in m2 when carat:cut and x:cut swapped position, the p-value of x:cut is significant.

## #question6

m4 <- lm(price ~ carat + x + cut, data = my\_diamonds)

anova(m4)

anova(m1, m4)



# p-value is 1.017 \* 10^(-15) is significant.

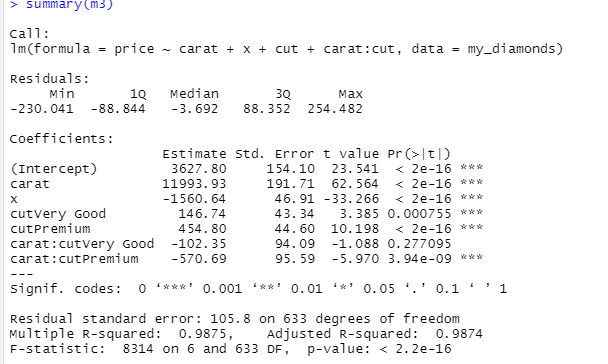
# in nested model F-terst of m1 or m2 against m3, we found the p-value is 0.1487 which is less than 0.05, which means x:cut is not significant.

# m3 and m4 are both reduced models. they cannot compare with each other since we could miss some important interaction that could be significant.

# a group of regressors could each individually have p-values indicating that individually they are not significant but dropping all of them could still be significant.

## # question7

summary(m3)



# when cut = good

# price = 3678.22 + 12091.88 \* carat - 1578.65 \* x

# when cut = very good

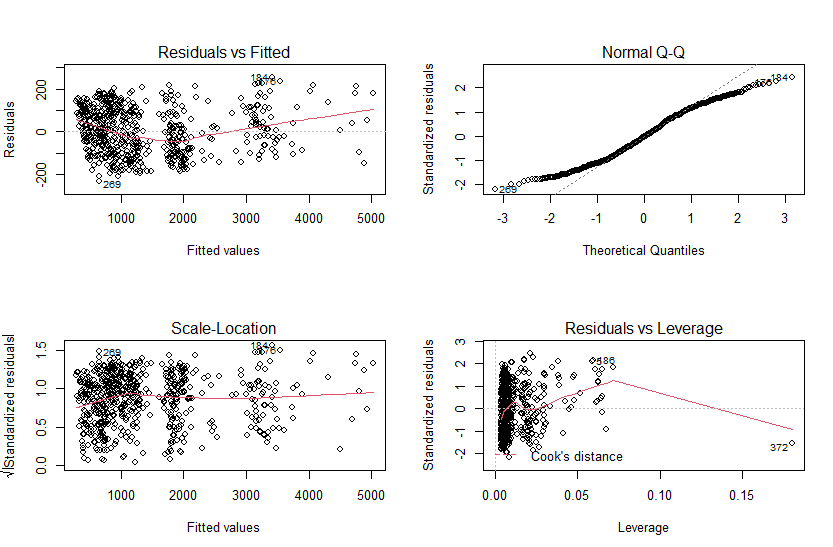
# price = 3678.22 + 153.40 + (12091.88 -134.83) \* carat - 1578.65 \* x

# when cut = premium

# price = 3678.22 + 455.85 + (12091.88 -582.46 ) \* carrot - 1578.65 \* x

par(mfrow = c(2, 2))

plot(m3)

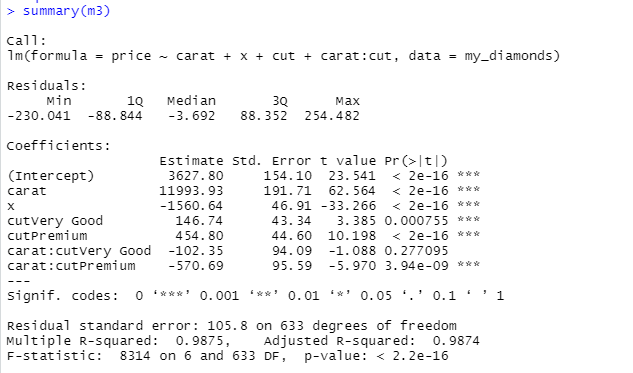


# From the Residuals vs Fitted, we can see the red line does not fit the red dash line, so we can say there is too much deviation from linearity and it does not fit the normality assumption.

# From Normal Q-Q, we can observe that there is a deviation from the normality when the theoretical quantities are greater than 1. The residuals increase as the quality increases.

## #question8

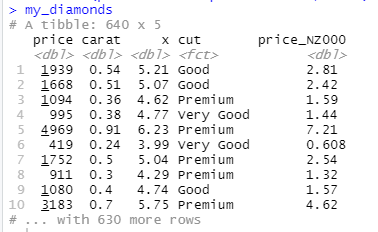
summary(m3)



my\_diamonds <- my\_diamonds %>%

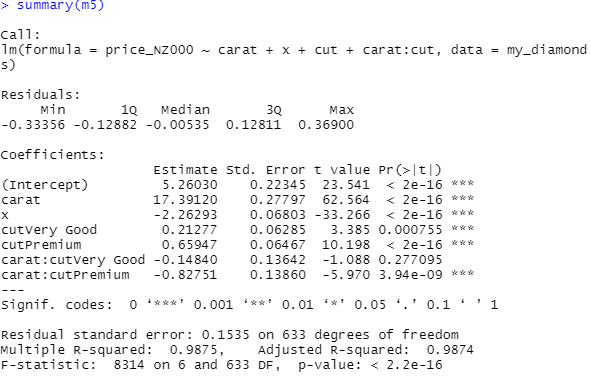
mutate(price\_NZ000 = price \* 1.45 / 1000)

my\_diamonds



m5 <- lm(price\_NZ000 ~ carat + x + cut + carat:cut, data = my\_diamonds)

summary(m5)



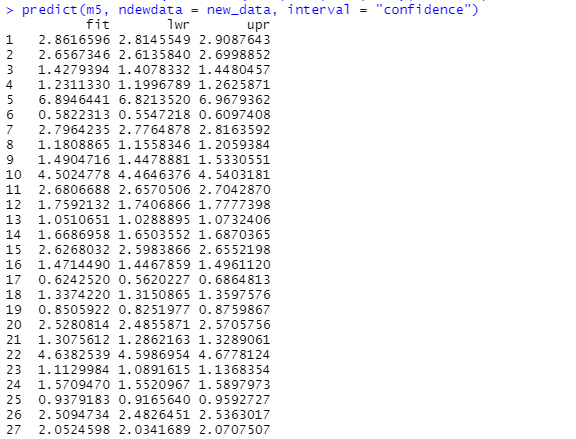
#the coefficient in m3 is 1000 times bigger than the coefficient in m5, since we give the price values in units of NZ$1000 in m5. We can see the coefficients decrease significantly. The lower the value of the coefficient of variation, the more precise the estimate, so new predictor variable makes the model fit better than before.

# we can also see the residual standard error goes down from 105.8 on 633 degrees of freedom to 0.1535 on 633 degrees of freedom. The smaller the residual standard deviation, the closer is the fit of the estimate to the actual data, so new predictor variable makes the model fit better.

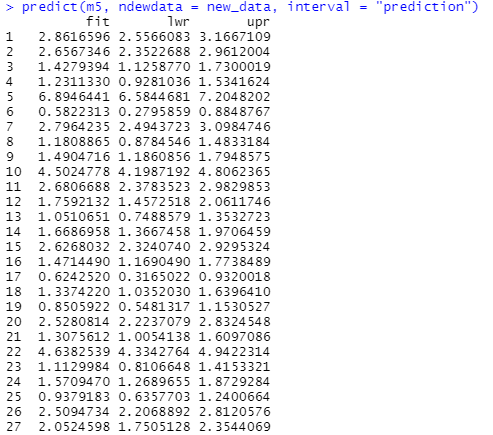
## #question9

new\_data = tibble(carat = c(0.25, 0.4, 0.6, 0.9), x = 4.7, cut = 'Premium')

predict(m5, ndewdata = new\_data, interval = "confidence")



predict(m5, ndewdata = new\_data, interval = "prediction")



# from the outputs of confidence and prediction, we can observe that the fir of each data is the same but lower and upper are different.

# Confidence interval for mean value of response variable given value of predictor variables. Confidence represents an interval estimate of the mean. By using m5 with variable x, carat, cut, and the interaction between carat and cut, we can find the estimated lower and upper interval of the mean value of price.

# The prediction interval predicts in what range a future individual observation will fall. Prediction represents the interval prediction of random variables. By using m5 with variable x, carat, cut, and the interaction between carat and cut, we can find the predicted lower and upper intervals of the mean value of price.