Homework 3

2023-11-13

```
library(DataAnalytics)
library(ggplot2)
```

Question 1: Prediction from Multiple Regression

```
# Part A
data(multi)
mlr_multi <- lm(Sales~p1+p2,data=multi)
summary(mlr_multi)
```

```
##
## Call:
## lm(formula = Sales ~ p1 + p2, data = multi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -66.916 -15.663  -0.509  18.904  63.302
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  115.717      8.548   13.54   <2e-16 ***
## p1           -97.657      2.669  -36.59   <2e-16 ***
## p2           108.800      1.409   77.20   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.42 on 97 degrees of freedom
## Multiple R-squared:  0.9871, Adjusted R-squared:  0.9869
## F-statistic:  3717 on 2 and 97 DF,  p-value: < 2.2e-16
```

Part B This is because there is some relationship between p1 and p2 as they are competing products in this case. If we set P2 = Mean(P2), we are essentially claiming that they are not correlated, which is not true in this specific case. Also, for p1 = 7.5, if we set p2 = mean(p2), the potential effect of p2 on sales is not considered, which might result in biased prediction.

```
# Part C
p1_2 <- lm(p2~p1,data=multi)
pred_p2 <- predict(p1_2, new = data.frame(p1 = 7.5))
pred_p2
```

```
##        1
## 12.00116
```

p2 = 12.00116

```
# Part D

# Using Simple Linear Regression of Sales on p1
lm_p1 <- lm(Sales~p1, data = multi)
simple_sales <- predict(lm_p1, new = data.frame(p1 = 7.5))
simple_sales
```

```
##        1
## 689.0118
```

```
# Using the predicted value from Part c
lm_p1p2 <- lm(Sales~p1+p2, data = multi)
mul_sales <- predict(lm_p1p2, new = data.frame(p1 = 7.5, p2 = pred_p2))
mul_sales
```

```
##        1
## 689.0118
```

This must be true because by regressing p2 on p1, we investigate the effect of p2 on p1. Since such relationship is investigated, we are able to purge p1 of its relationship to p2 when estimating sales in our multiple linear regression. In other words, the predicted value of p2 from Part C would incorporate the relationship between p2 and p1 in our multiple linear regression model. Hence, the two regression would have the same prediction because it is now about the effect of p1 on sales for both cases.

Question 2: Interactions

```
# Part A
data(mvehicles)
cars=mvehicles[mvehicles$bodytype != "Truck",]
inter_model <- lm(log(emv)~luxury+sporty+luxury*sporty ,data=cars)
b_0 <- inter_model$coefficients[1]
b_1 <- inter_model$coefficients[2]
b_2 <- inter_model$coefficients[3]
b_3 <- inter_model$coefficients[4]

emv_change <- exp(0.1 * (b_2 + b_3 * 0.3))
emv_change
```

```
##    sporty
## 0.9978489
```

The change in emv for a 0.1 units increase in sporty is approximately 0.998 units, holding luxury constant at 0.3 units.

```
# Part B
change_emv2 <- exp(0.1 * (b_2 + b_3 * 0.7))
change_emv2
```

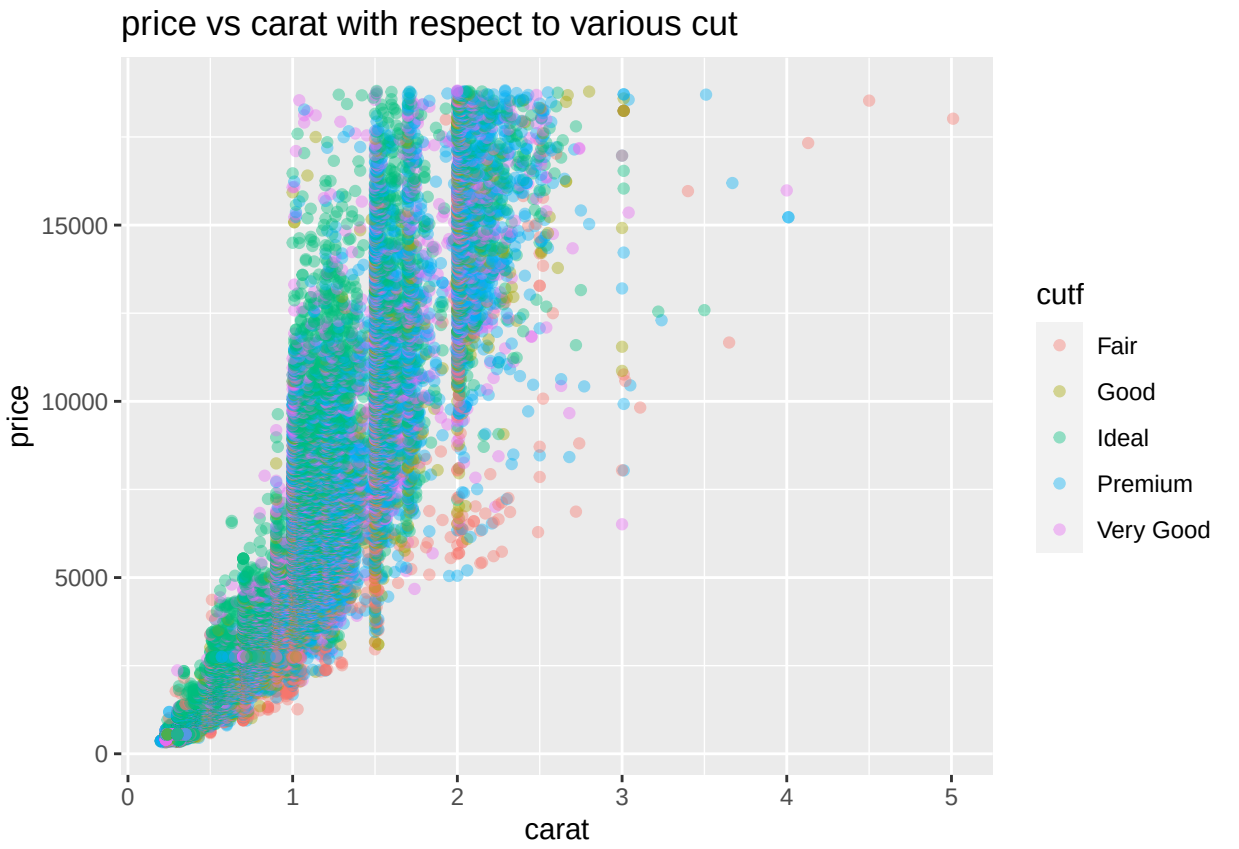
```
##   sporty
## 1.050834
```

The change in emv for a 0.1 units increase in sporty is approximately 1.051 units, holding luxury constant at 0.7 units.

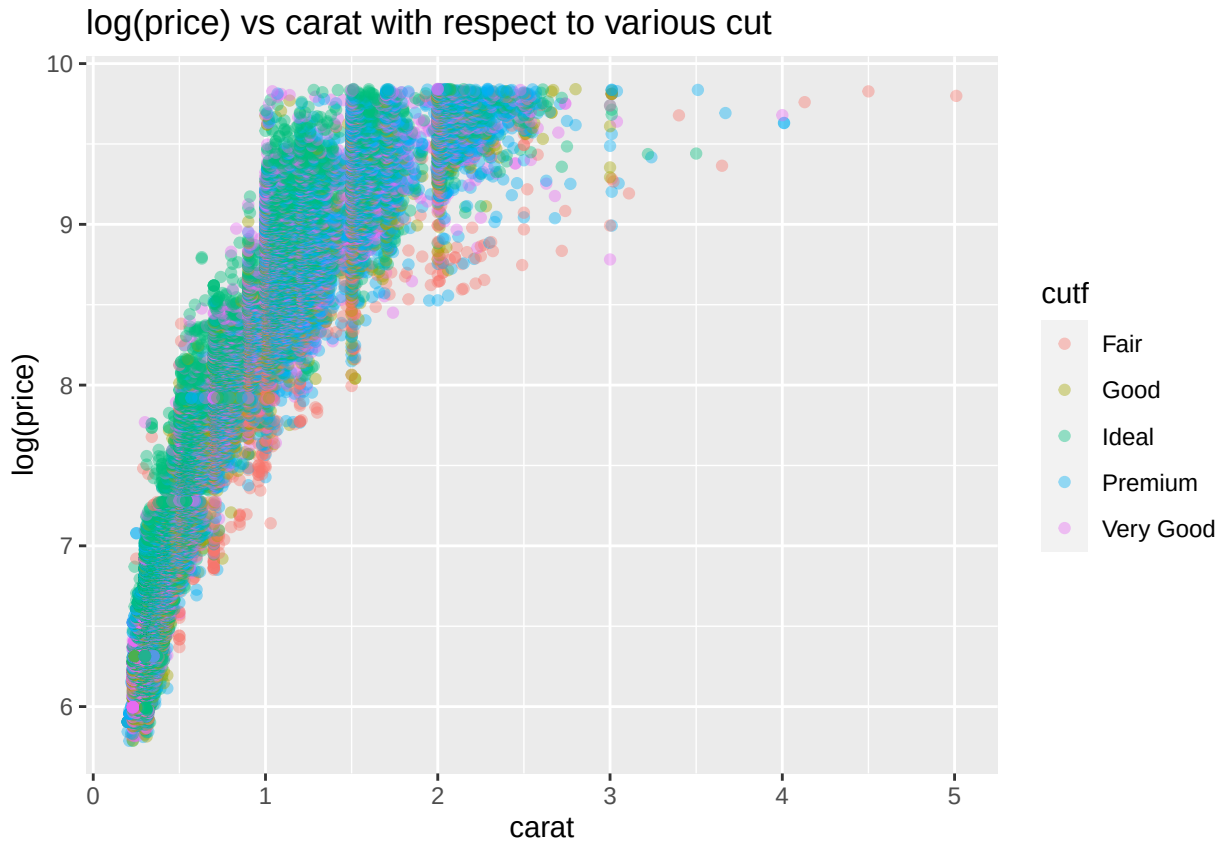
Part C This is because of the interaction term sporty*luxury. Since this interaction term allows luxury to have an effect on the relationship between sporty and emv, the change in emv will account for the interaction effect with luxury at different level. When the level of luxury changes from 0.3 to 0.7, such interaction effect will also have different level of impact on our regression even for the same level of increase in sporty, which then results in different prediction of emv. This intuitively makes sense because the impact of sporty on emv is not constant, as it depends on the level of luxury.

Question 3

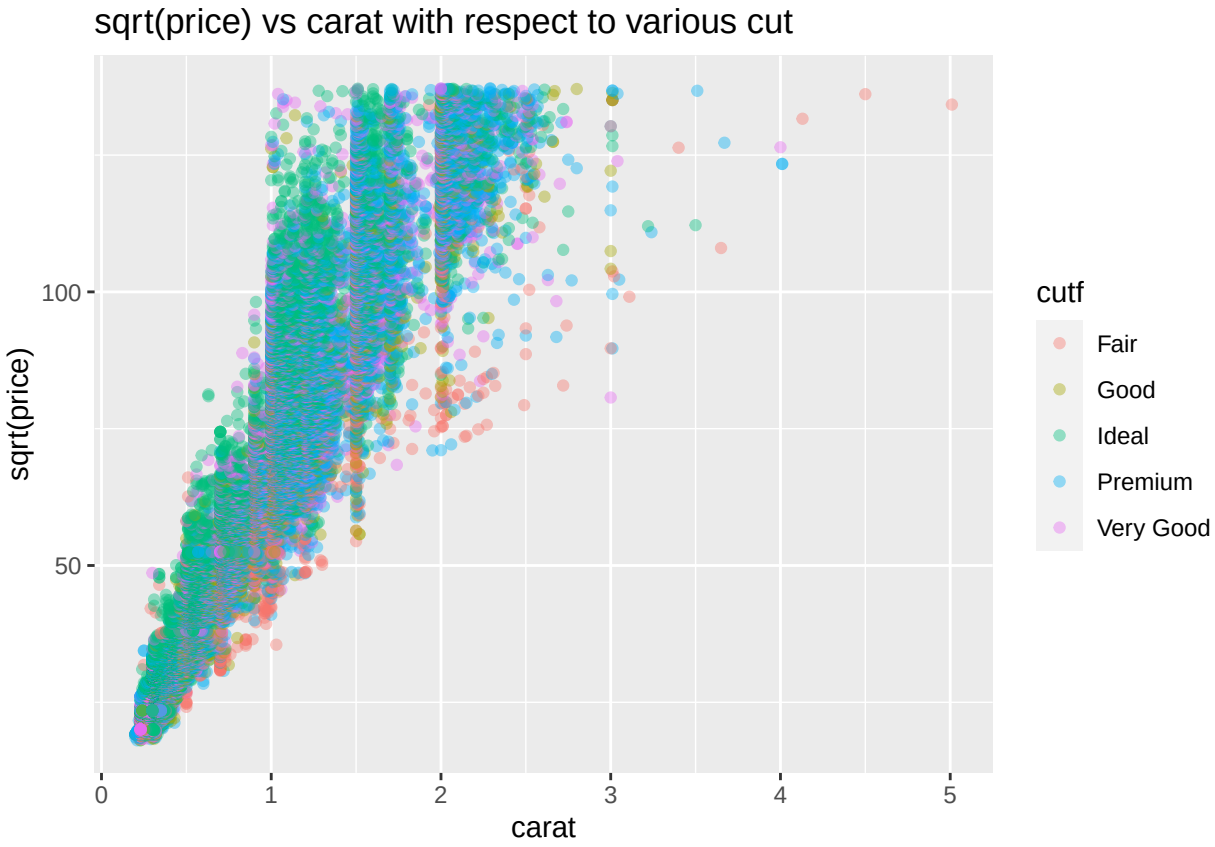
```
#1
data(diamonds)
cutf=as.character(diamonds$cut)
cutf=as.factor(cutf)

ggplot(diamonds, aes(x = carat, y = price, z = cutf)) +
  geom_point(aes(color = cutf), alpha = 0.4) +
  labs(title = 'price vs carat with respect to various cut', x = "carat", y = "price")
```



```
ggplot(diamonds, aes(x = carat, y = log(price), z = cutf)) +
  geom_point(aes(color = cutf), alpha = 0.4) +
  labs(title = 'log(price) vs carat with respect to various cut', x = "carat", y = "log(price)")
```




```
ggplot(diamonds, aes(x = carat, y = sqrt(price), z = cutf)) +
  geom_point(aes(color = cutf), alpha = 0.4) +
  labs(title = 'sqrt(price) vs carat with respect to various cut', x = "carat", y = "sqrt(price)")
```

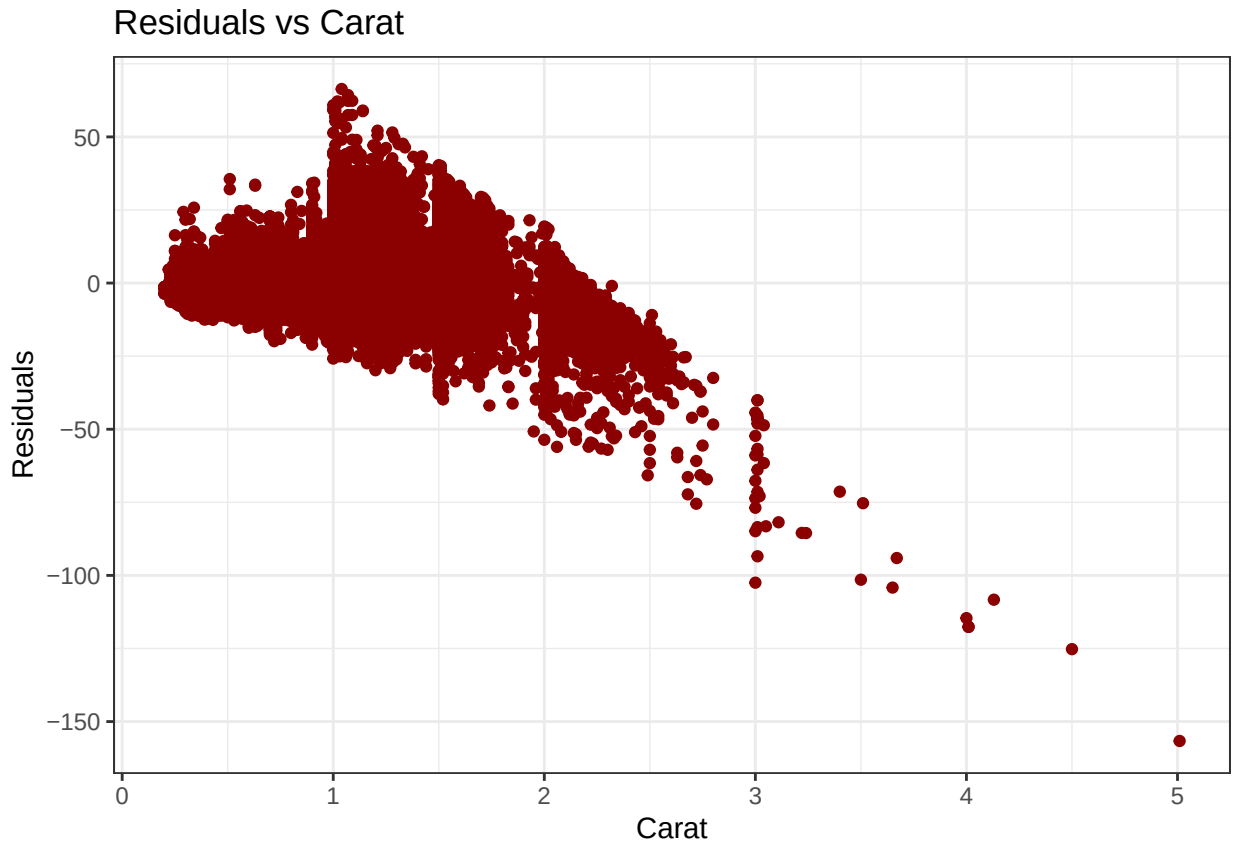



```
#2
# Using sqrt transformation on price to build the model
dia_mlr=lm(sqrt(price)~carat + cutf,data = diamonds)
lmSumm(dia_mlr)
```

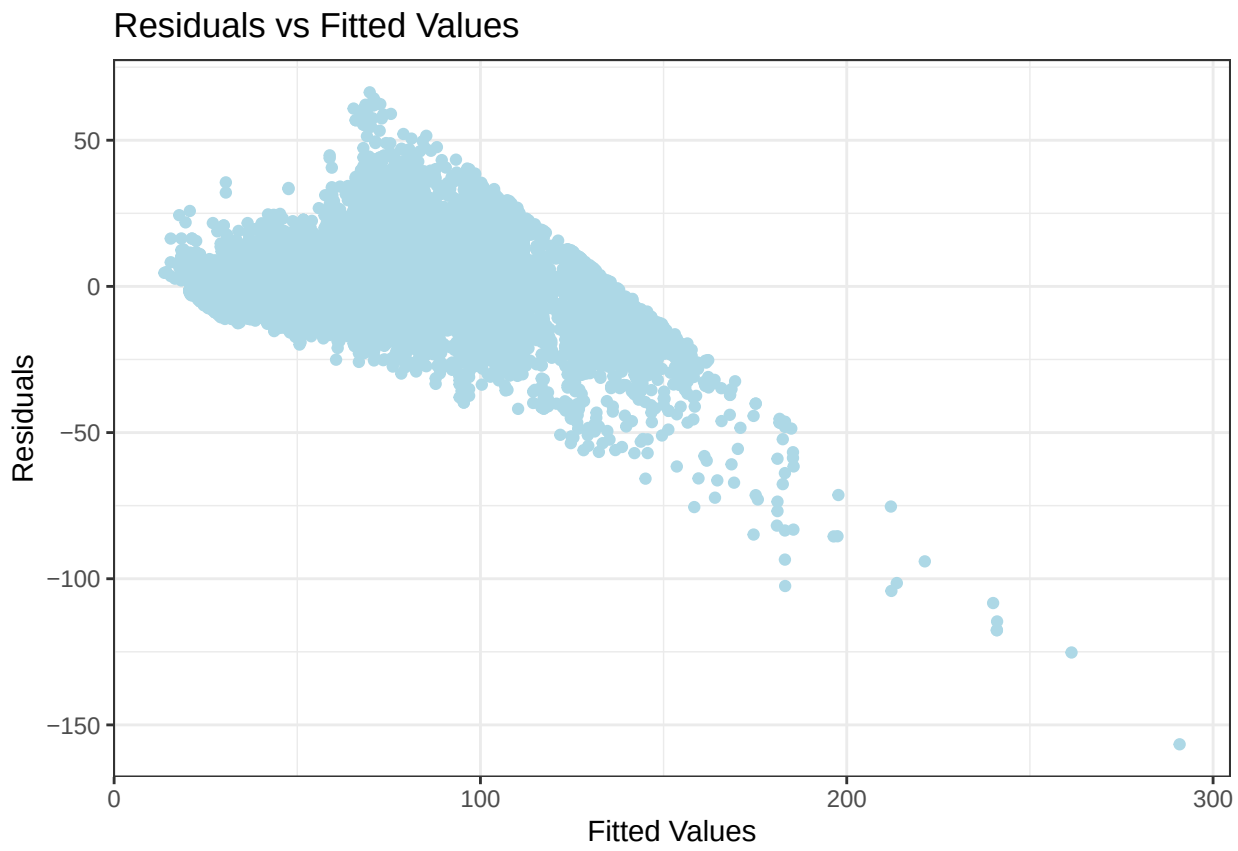
```
## Multiple Regression Analysis:
##     6 regressors(including intercept) and 53940 observations
##
## lm(formula = sqrt(price) ~ carat + cutf, data = diamonds)
##
## Coefficients:
##                Estimate Std Error t value p value
## (Intercept)      0.9977   0.24180    4.13       0
## carat           57.8600   0.08366  691.59       0
## cutfGood         6.5190   0.26030   25.04       0
## cutfIdeal       10.1700   0.23550   43.19       0
## cutfPremium      7.9710   0.23860   33.41       0
## cutfVery Good    8.6100   0.24080   35.76       0
## ---
## Standard Error of the Regression:  9.045
## Multiple R-squared:  0.9  Adjusted R-squared:  0.9
## Overall F stat: 97401.63 on 5 and 53934 DF, pvalue= 0
```

```
# Checking
qplot(carat,dia_mlr$residuals,data = dia_mlr$model, color=I("darkred")) + theme_bw() +
  labs(title = 'Residuals vs Carat', x = 'Carat', y = 'Residuals')
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
qplot(dia_mlr$fitted,dia_mlr$residuals,color=I("lightblue")) + theme_bw() +
  labs(title = 'Residuals vs Fitted Values', x = 'Fitted Values', y = 'Residuals')
```


Residuals vs Fitted Values
Residuals
50
0
−50
−100
−150
0
100
200
300
Fitted Values