Homework 3
MSBA 400: Statistical Foundations for Data Analytics
UID 106082225, Sarvari Pidaparty

Question 1 : Prediction from Multiple Regressions

Q1, part A

Run the multiple regression of `Sales` on `p1` and `p2` using the dataset, `multi`.

```
library(DataAnalytics)
data(multi)
mlt_sales = lm(Sales ~ p1+p2, data = multi)
summary(mlt_sales)
```

```
##
## 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
```

Q1, part B

Suppose we wish to use the regression from part A to estimate sales of this firm's product with, `p1` = \$7.5. To make predictions from the multiple regression, we will have to predict what p2 will be given that `p1` =\$7.5.

Explain why setting `p2=mean(p2)` would be a bad choice. Be specific and comment on why this is true for this particular case (value of 'p1).

Answer: Using mean(p2) in the prediction model is not a good choice at all since we see that p1 and p2 are correlated (shown below - correlation coefficient = 0.78). Putting in mean(p2) directly disregards changes in p2 that happen due to changes in p1. For example, if p1 values are on the lower end, the conditional mean of p2 for this case would be different than the conditional mean of p2 when the p1 values are on the higher end.

```
cor(multi)
```

```
##              p1        p2     Sales
## p1    1.0000000 0.7833345 0.4425828
## p2    0.7833345 1.0000000 0.8996148
## Sales 0.4425828 0.8996148 1.0000000
```

For the specific example p1 = \$7.5, we notice that predicted value of p2 is 12.00116 (computed below in part C), which is different from mean(p2) which is equal to 8. Substituting this value is incorrect.

```
mean(multi$p2)
```

```
## [1] 8
```

Q1, part C

Use a regression of p2 on p1 to predict what p2 would be given that p1 = $7.5.

```
model = lm(p2 ~ p1, data = multi)
summary(model)
```

```
##
## Call:
## lm(formula = p2 ~ p1, data = multi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5921 -1.3602  0.0299  1.3851  5.5472
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.8773     0.6062   1.447    0.151
## p1            1.4832     0.1189  12.475   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.037 on 98 degrees of freedom
## Multiple R-squared:  0.6136, Adjusted R-squared:  0.6097
## F-statistic: 155.6 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
p2r = predict(model, newdata = data.frame(p1 = 7.5))
p2r
```

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

Q1, part D

Use the predicted value of p2 from part C, to predict Sales. Show that this is the same predicted value of sales as you would get from the simple regression of Sales on p1. Explain why this must be true.

```
predict(mlt_sales, newdata = data.frame(p1=7.5,p2=p2r))
```

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

```
lm.slr=lm(Sales~p1,data=multi)
predict(lm.slr, newdata = data.frame(p1=7.5))
```

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

We notice that the sales values we obtain from both methods are the same. This is because in method 1 where we use multiple regression to regress Sales on p1 and p2, we further regress p2 on p1 to predict p2 due to which we purge the effect of p1 on p2. This value is therefore equal to the result we obtain via the simple regression of Sales on p1.

Question 2: Interactions

An interaction term in a regression is formed by taking the product of two independent or predictor variables as in:

$$Y_i = \beta_0 + \beta_1 X1_i + \beta_2 X2_i + \beta_3 X1_i * X2_i + \varepsilon_i$$

This term has a non-linear effect, which allows the effect of variable $X1$ to be moderated by the level of $X2$. We can take the partial derivative of the conditional mean function to see this:

$$\frac{\partial}{\partial X1} E[Y|X1, X2] = \beta_1 + \beta 3 X2$$

Return to the regression in Chapter 6 of `log(emv)` on `luxury`, `sporty` and add the interaction term `luxury*sporty`.

Q2, part A

Compute the change in `emv` we would expect to see if sporty increased by .1 units, holding luxury constant at .30 units

```
data(mvehicles)
cars=mvehicles[mvehicles$bodytype != "Truck",]
lmout=lm(log(emv)~luxury*sporty,data=cars)
summary(lmout)
```

```
##
## Call:
## lm(formula = log(emv) ~ luxury * sporty, data = cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77690 -0.20474 -0.03719  0.19434  2.50271
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.73506    0.04385 222.016  < 2e-16 ***
## luxury         1.32184    0.10904  12.122  < 2e-16 ***
## sporty        -0.40956    0.11601  -3.530 0.000429 ***
## luxury:sporty  1.29343    0.22206   5.825  7.1e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3122 on 1391 degrees of freedom
## Multiple R-squared:  0.5883, Adjusted R-squared:  0.5874
## F-statistic: 662.5 on 3 and 1391 DF,  p-value: < 2.2e-16
```

```
predout=(predict(lmout,data.frame(luxury=.3,sporty=.7)))-(predict(lmout,data.frame(luxury=.3,sporty=.6)))
predout*100
```

```
##          1
## -0.2153429
```

Q2, part B

Compute the change in `emv` we would expect to see if sporty was increased by .1 units, holding luxury constant at .70 units.

```
predout1=(predict(lmout,data.frame(luxury=.7,sporty=.7)))-(predict(lmout,data.frame(luxury=.7,sporty=.6)
predout1*100
```

```
##        1
## 4.958395
```

Q2, part C

Why are the answers different in part A and part B? Does the interaction term make intuitive sense to you? Why?

Answer: We see that the percentage change in log(emv) for luxury = 0.7 units when sporty is increased by 0.1 units is much higher than when luxury = 0.3 units. This makes sense because the sportiness of the car has a greater (positive) effect on the price of the car when the car is also more luxurious. Our calculations in part A and B confirm this intuition.

Question 3: More on ggplot2 and regression planes

The classic dataset, `diamonds`, (you must load the `ggplot2` package to access this data) has about 50,000 prices of diamonds along with weight (`carat`) and quality of cut (`cut`).

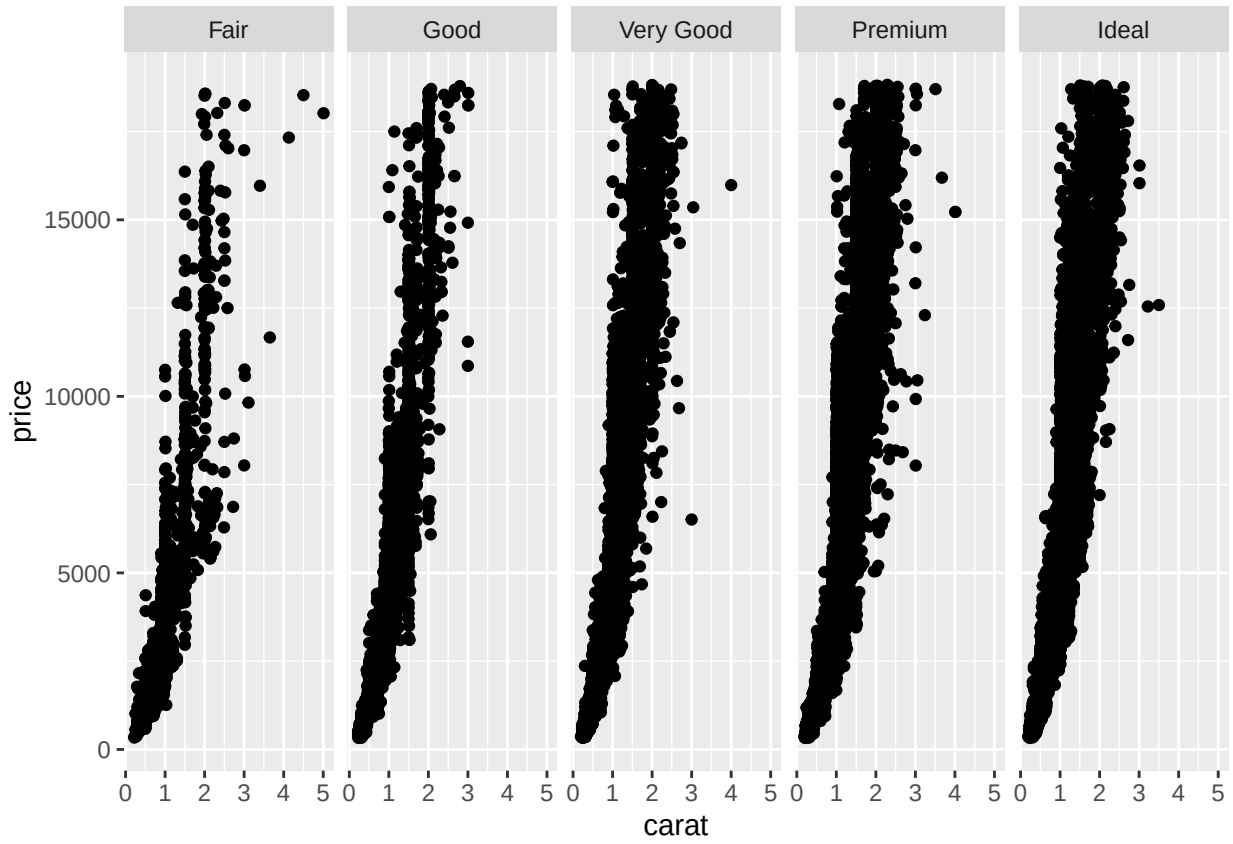
1. Use ggplot2 to visualize the relationship between price and carat and cut. 'price' is the dependent variable. Consider both the log() and sqrt() transformation of price.
2. Run a regression of your preferred specification. Perform residual diagnostics. What do you conclude from your regression diagnostic plots of residuals vs. fitted and residuals vs. carat?

note: `cut` is a special type of variable called an ordered factor in R. For ease of interpretation, convert the ordered factor into a "regular" or non-ordinal factor.

```
library(ggplot2)
data(diamonds)
cutf=as.character(diamonds$cut)
cutf=as.factor(cutf)
```

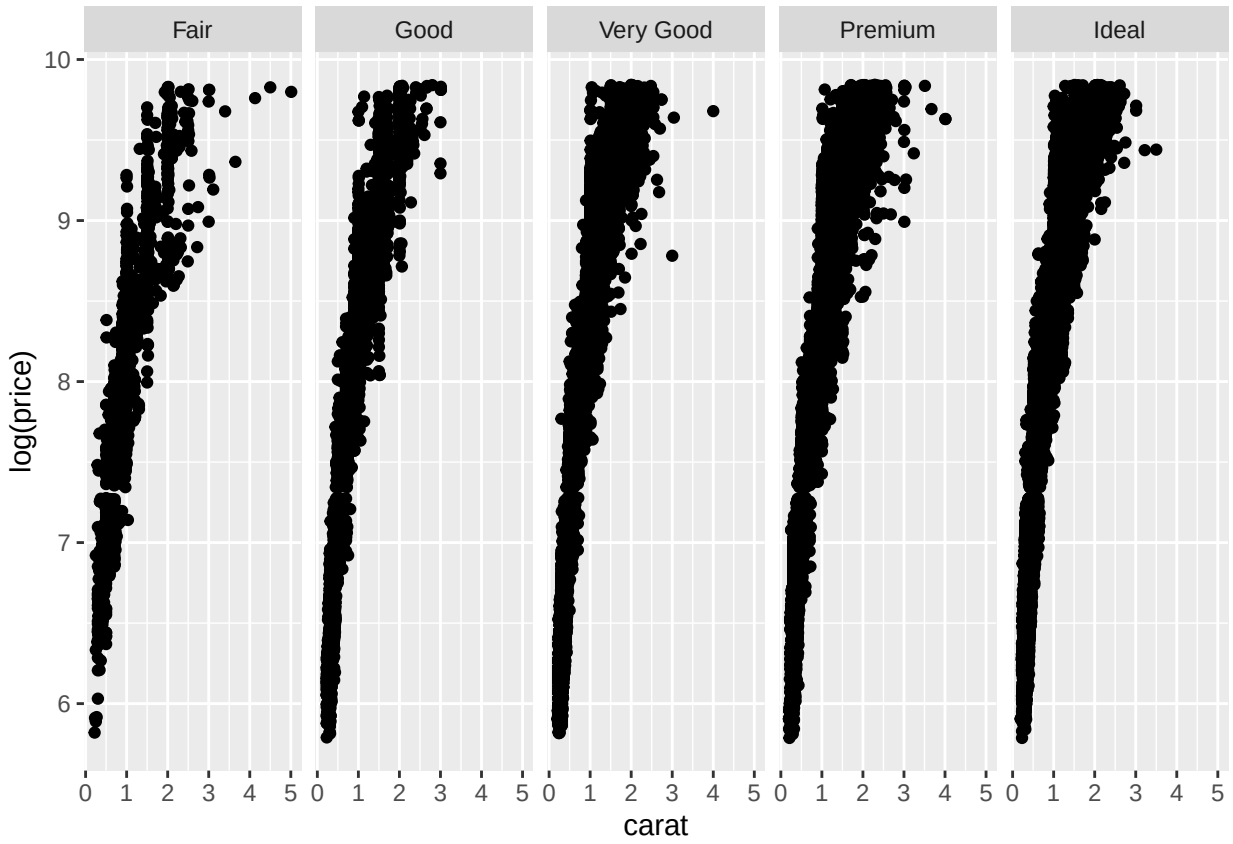
1.

```
ggplot(data=diamonds, mapping = aes(x = carat, y = price)) +
  geom_point() +
  facet_grid(~cut)
```

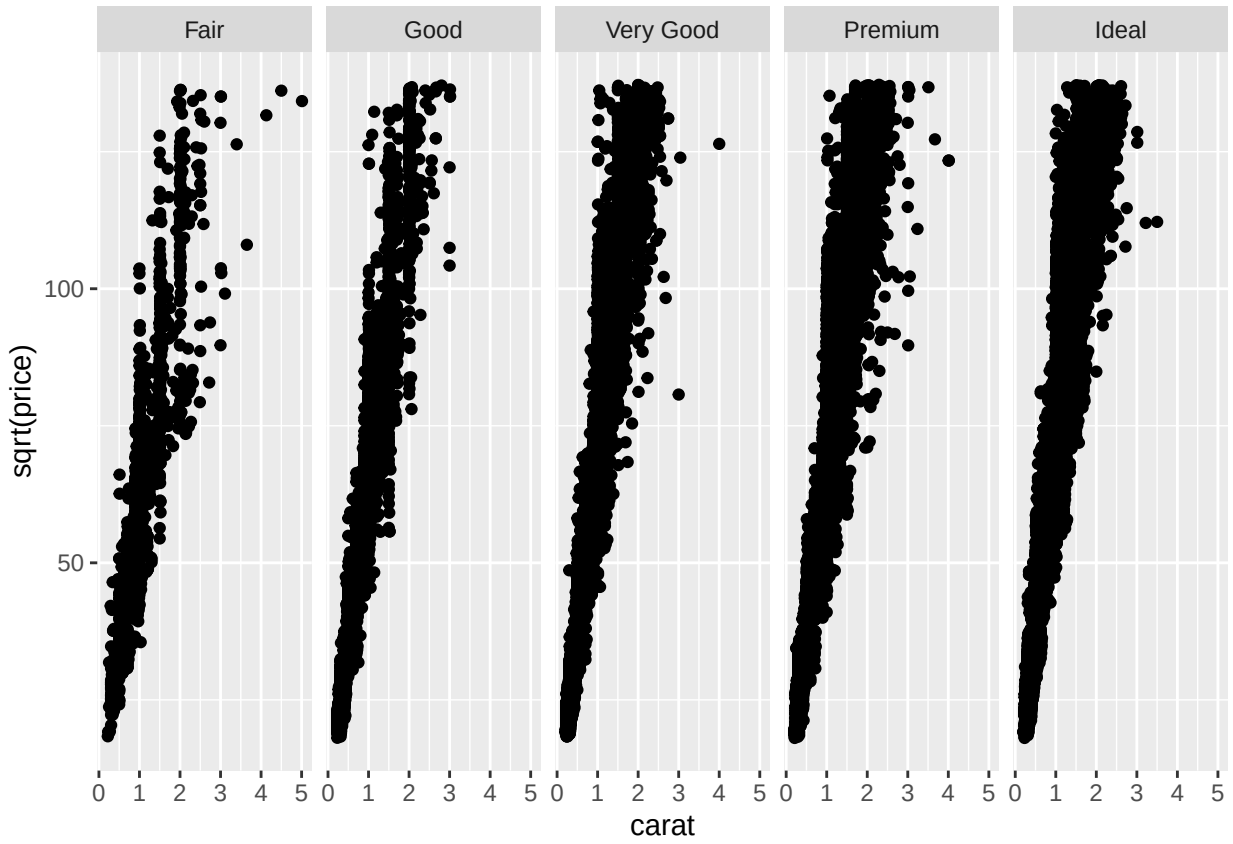


We use the log and sqrt transformations to obtain a more linear graph.

```
ggplot(data=diamonds, mapping = aes(x = carat, y = log(price))) +
  geom_point() +
  facet_grid(~cut)
```




```
ggplot(data=diamonds, mapping = aes(x = carat, y = sqrt(price))) +
  geom_point() +
  facet_grid(~cut)
```

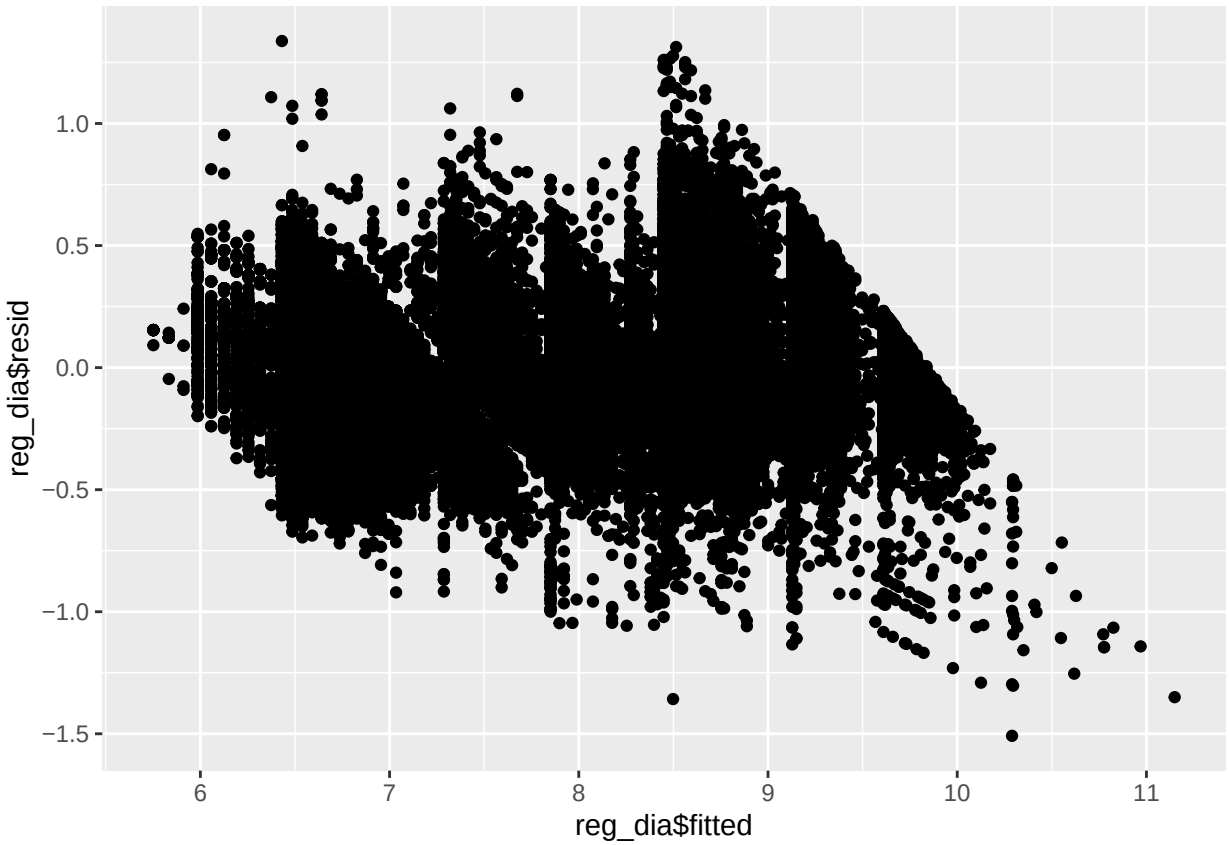


2.

```
reg_dia = lm(log(price)~log(carat), data = diamonds)
summary(reg_dia)
```

```
##
## Call:
## lm(formula = log(price) ~ log(carat), data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.50833 -0.16951 -0.00591  0.16637  1.33793
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.448661   0.001365  6190.9   <2e-16 ***
## log(carat)  1.675817   0.001934   866.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2627 on 53938 degrees of freedom
## Multiple R-squared:  0.933,  Adjusted R-squared:  0.933
## F-statistic: 7.51e+05 on 1 and 53938 DF,  p-value: < 2.2e-16
```

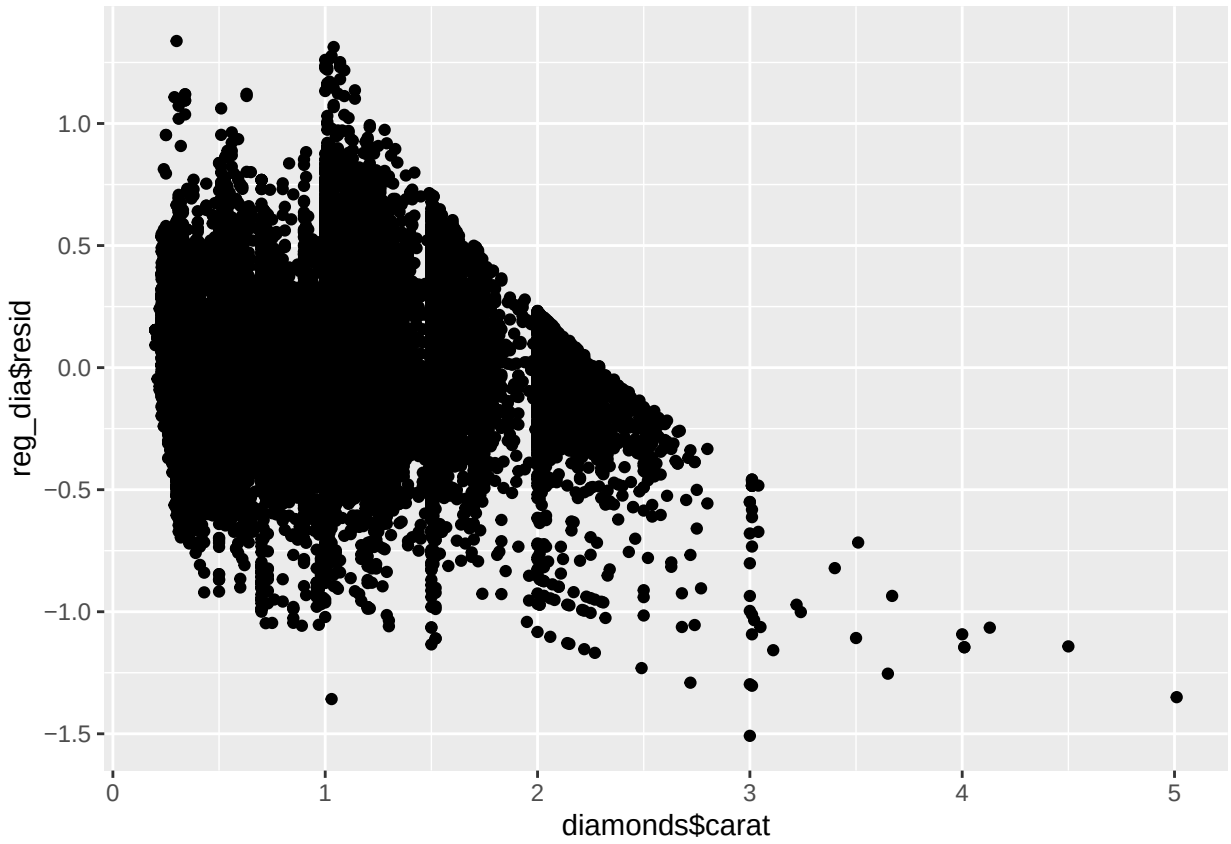
```
qplot(reg_dia$fitted, reg_dia$resid)
```




```
cor(reg_dia$fitted, reg_dia$resid)
```

```
## [1] 7.921523e-16
```

```
qplot(diamonds$carat, reg_dia$resid)
```




```
cor(diamonds$carat, reg_dia$resid)
```

```
## [1] -0.01388275
```

We do not see any correlation in both the graphs, i.e. no correlation is observed between the fitted and residual values as well as carat (X) and residual values. This means the basic regression property $[corr(X, e) = 0]$ is satisfied and the model is correct. This has also been confirmed by computing correlation coefficients with the `cor` command, and we obtain values that are computer speak for 0.