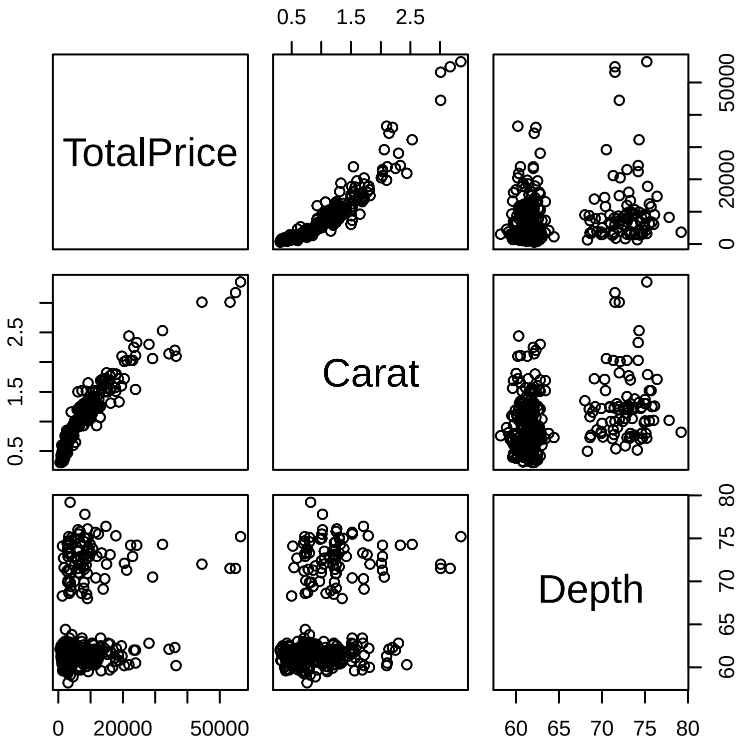
# HW 3 Solutions

## Part 1: EDA

### a)

*Carat* looks like it probably has a quadratic relationship with *TotalPrice*. *Depth* vs. *TotalPrice* is a very weird scatterplot: two groups, neither of which have a linear (or other obvious) relationship with *TotalPrice*.

```
pairs(~TotalPrice+Carat+Depth,data=Diamonds)
```




### b)

*Carat* has a much higher correlation with *TotalPrice* than *Depth* does, which is what we expected from the scatterplots. However, we should question how useful correlation is in this case, since neither variable seems to have a *linear* relationship with *TotalPrice*.

```
cor(~cbind(Carat,Depth,TotalPrice),data=Diamonds)
```

```
##                 Carat     Depth TotalPrice
## Carat       1.0000000 0.3202687  0.9291358
## Depth       0.3202687 1.0000000  0.2177839
## TotalPrice  0.9291358 0.2177839  1.0000000
```

## c)

*Carat* and *Depth* are absolutely NOT highly correlated with each other. $r = 0.32$, which is a weak correlation; and the scatterplot shows no real linear relationship between these two variables.

# Part 2: Book Problems

## 3.23: Diamonds

Code is below for each of the 4 models they ask us to fit, plus the models from Example 3.11. Here's a summary/comparison of the 6 models:

| Terms | Rsq | Adj Rsq | significant terms (10% level) |
|---|---|---|---|
| $Depth, Depth^2$ | 4.7% | 4.2% | none |
| $Carat, Depth$ | 87.0% | 87.0% | $Carat, Depth$ |
| $Carat, Depth, Carat * Depth$ | 89.0% | 89.0% | $Carat, Depth, Carat * Depth$ |
| $Carat, Depth, Carat * Depth, Carat^2, Depth^2$ | 93.1% | 93.0% | $Carat, Carat^2$ |
| $Carat, Carat^2$ | 92.6% | 92.5% | $Carat, Carat^2$ |
| $Carat, Carat^2, Carat^3$ | 92.6% | 92.5% | $Carat^2$ |

If we use just adjusted Rsq as a criteria, the best model among these would be the complete second-order model (d). However, 3 of the predictors in that model (all involving *Depth*) are insignificant. So the better model is the quadratic model based on *Carat* (from Example 3.11), which has nearly as large an adjusted Rsq, small p-values for the coefficients of each predictor, plus fewer predictors. (If we were not limited to just these models, we might also try a model that adds just Depth to the quadratic model based on Carat to see if it would be significant without the other two strongly related predictors.) Residual plots for these two models are equivalent, and conditions are generally met. Residual plots for all other models are worse (except for the cubic model).

### a. Depth and Depth$^2$

```
model.a <- lm(TotalPrice~Depth+I(Depth^2),data=Diamonds); summary(model.a)
```

```
##
## Call:
## lm(formula = TotalPrice ~ Depth + I(Depth^2), data = Diamonds)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9323  -4251  -2676   2134  45513
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28406.783 112211.790  -0.253    0.800
## Depth          766.369   3353.222   0.229    0.819
## I(Depth^2)      -3.233     24.869  -0.130    0.897
##
## Residual standard error: 7616 on 348 degrees of freedom
## Multiple R-squared:  0.04748,    Adjusted R-squared:  0.042
## F-statistic: 8.673 on 2 and 348 DF,  p-value: 0.0002111
```

### b. Carat and Depth

```
model.b <- lm(TotalPrice~Depth+Carat,data=Diamonds); summary(model.b)
```

```
##
## Call:
## lm(formula = TotalPrice ~ Depth + Carat, data = Diamonds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9234.7 -1223.7  -274.3  1161.0 16368.6
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1059.24    1918.36   0.552    0.581
## Depth        -134.94      30.92  -4.364 1.68e-05 ***
## Carat       15087.01     320.96  47.006  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2809 on 348 degrees of freedom
## Multiple R-squared:  0.8704, Adjusted R-squared:  0.8696
## F-statistic:  1168 on 2 and 348 DF,  p-value: < 2.2e-16
```

### c. Carat, Depth, CaratxDepth

```
model.c <- lm(TotalPrice~Carat*Depth,data=Diamonds); summary(model.c)
```

```
##
## Call:
## lm(formula = TotalPrice ~ Carat * Depth, data = Diamonds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8254.4 -1311.5  -157.2  1131.8 14513.9
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  31171.41    4219.58   7.387 1.13e-12 ***
## Carat       -11827.73    3436.47  -3.442 0.000648 ***
## Depth         -598.18      65.47  -9.137  < 2e-16 ***
## Carat:Depth    408.45      51.96   7.861 4.84e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2592 on 347 degrees of freedom
## Multiple R-squared:   0.89,  Adjusted R-squared:  0.889
## F-statistic: 935.7 on 3 and 347 DF,  p-value: < 2.2e-16
```

### d. Carat^2, Depth^2, CaratxDepth

```r
model.d <- lm(TotalPrice~I(Carat^2) + I(Depth^2) + Carat*Depth,data=Diamonds); summary(model.d)
```

```
##
## Call:
## lm(formula = TotalPrice ~ I(Carat^2) + I(Depth^2) + Carat * Depth,
##     data = Diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12196.1   -652.7    -38.5    485.7  10582.2
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24338.820  30297.912   0.803   0.4223
## I(Carat^2)   4761.592    330.246  14.418   <2e-16 ***
## I(Depth^2)      5.276      6.727   0.784   0.4333
## Carat        7573.620   3040.787   2.491   0.0132 *
## Depth        -728.700    904.439  -0.806   0.4210
## Carat:Depth   -83.891     53.530  -1.567   0.1180
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2053 on 345 degrees of freedom
## Multiple R-squared:  0.9313, Adjusted R-squared:  0.9304
## F-statistic: 936.1 on 5 and 345 DF,  p-value: < 2.2e-16
```

**Compare to 2 models From Example 3.11:**

```r
#quadratic model
model.e <- lm(TotalPrice~Carat+I(Carat^2),data=Diamonds); summary(model.e)
```

```
##
## Call:
## lm(formula = TotalPrice ~ Carat + I(Carat^2), data = Diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10207.4   -711.6   -167.9    355.0  12147.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -522.7      466.3  -1.121  0.26307
## Carat         2386.0      752.5   3.171  0.00166 **
## I(Carat^2)    4498.2      263.0  17.101  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2127 on 348 degrees of freedom
## Multiple R-squared:  0.9257, Adjusted R-squared:  0.9253
## F-statistic:  2168 on 2 and 348 DF,  p-value: < 2.2e-16
```

```r
#cubic model
model.f <- lm(TotalPrice~Carat+I(Carat^2)+I(Carat^3),data=Diamonds); summary(model.f)
```

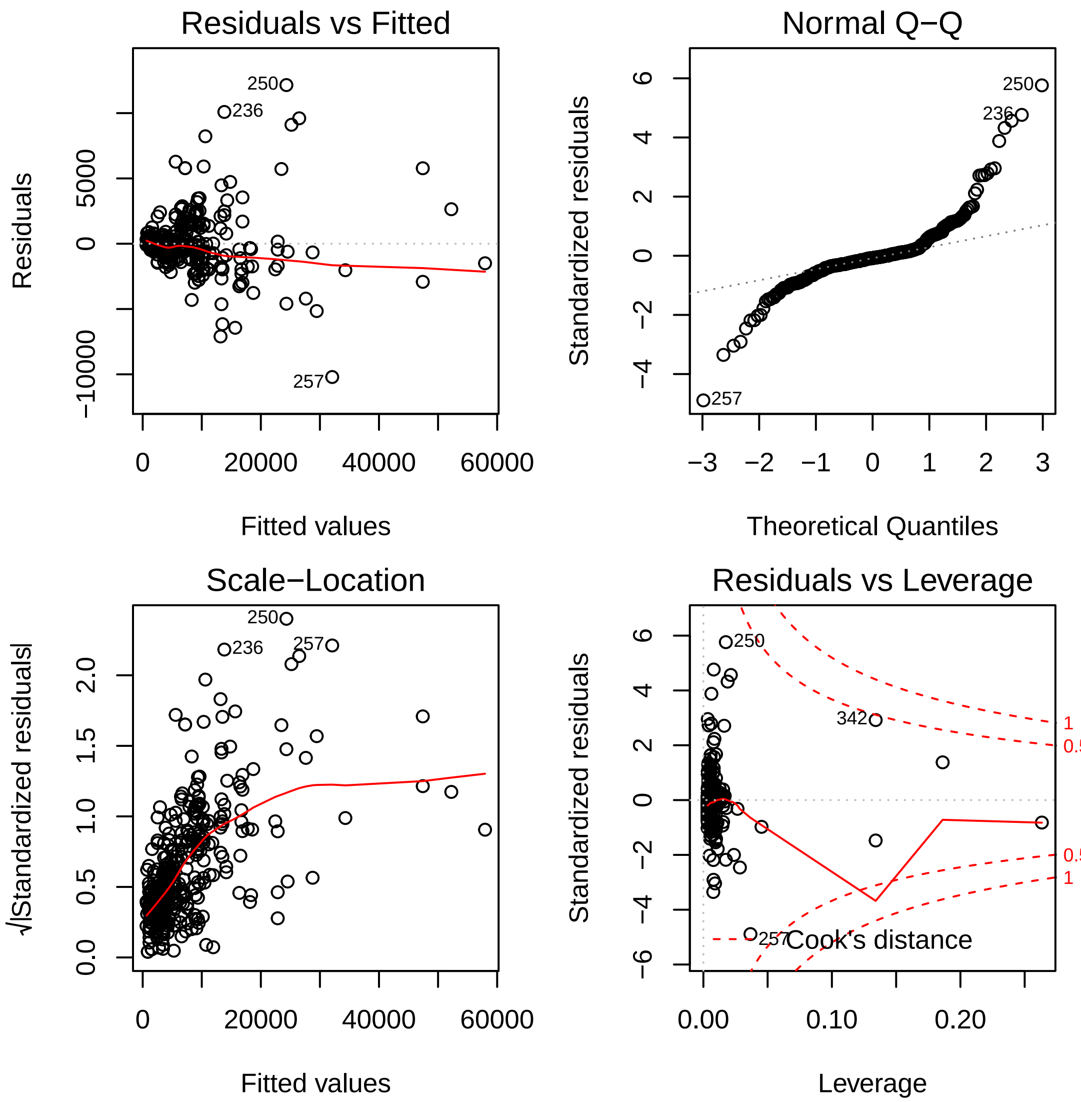
```
##
## Call:
## lm(formula = TotalPrice ~ Carat + I(Carat^2) + I(Carat^3), data = Diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10136.8   -725.2   -182.1    380.5  12220.8
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -723.44     875.50  -0.826  0.40919
## Carat        2942.02    2185.44   1.346  0.17912
## I(Carat^2)   4077.65    1573.80   2.591  0.00997 **
## I(Carat^3)     87.92     324.38   0.271  0.78652
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2130 on 347 degrees of freedom
## Multiple R-squared:  0.9257, Adjusted R-squared:  0.9251
## F-statistic:  1442 on 3 and 347 DF,  p-value: < 2.2e-16
```

## 3.24 Diamonds (continued)

### a. Residual plots

Using the $Carat, Carat^2$ model from Example 3.11 (which is my chosen model), the standard regression conditions do NOT appear to be satisfied. The tails of the normal probability plot fall off very harshly, indicating non-normality of the residuals. Their appears to be a megaphone pattern to the residual plot, indicating non-constant variance in the residuals.

```r
par(mar=c(4,4,2,2)); par(mfrow=c(2,2)); plot(model.e)
```




**b. Predicting ln(totalPrice)**

Code is below for each of the 4 models they ask us to fit, plus the models from Example 3.11, using Y=ln(totalPrice). Summary/comparison of the 6 models:

| Terms | Rsq | Adj Rsq | significant terms (10% level) |
|---|---|---|---|
| $Depth, Depth^2$ | 6.3% | 5.7% | none |
| $Carat, Depth$ | 85.8% | 85.7% | $Carat, Depth$ |
| $Carat, Depth, Carat * Depth$ | 88.1% | 88.0% | $Carat, Depth, Carat * Depth$ |
| $Carat, Depth, Carat * Depth, Carat^2, Depth^2$ | 93.0% | 92.9% | $Carat, Carat^2$ |
| $Carat, Carat^2$ | 92.5% | 92.5% | $Carat, Carat^2$ |
| $Carat, Carat^2, Carat^3$ | 93.3% | 93.3% | $Carat, Carat^2, Carat^3$ |

I would still argue that the quadratic model with $Carat, Carat^2$ is the best, with high $R^2$ and only two terms.

### Depth and Depth$^2$

```
logmodel.a <- lm(log(TotalPrice)~Depth+I(Depth^2),data=Diamonds); summary(logmodel.a)
```

```
##
## Call:
## lm(formula = log(TotalPrice) ~ Depth + I(Depth^2), data = Diamonds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1323 -0.6091 -0.1150  0.6217  2.1351
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.2516568 12.3981256   0.746    0.456
## Depth       -0.0605669  0.3704928  -0.163    0.870
## I(Depth^2)   0.0007626  0.0027477   0.278    0.782
##
## Residual standard error: 0.8415 on 348 degrees of freedom
## Multiple R-squared:  0.06262,    Adjusted R-squared:  0.05723
## F-statistic: 11.62 on 2 and 348 DF,  p-value: 1.298e-05
```

### Carat and Depth

```
logmodel.b <- lm(log(TotalPrice)~Depth+Carat,data=Diamonds); summary(logmodel.b)
```

```
##
## Call:
## lm(formula = log(TotalPrice) ~ Depth + Carat, data = Diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.38177 -0.13236  0.01812  0.21550  0.92948
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.445406   0.223436   33.32   <2e-16 ***
## Depth       -0.008752   0.003601   -2.43   0.0156 *
## Carat        1.652443   0.037383   44.20   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3272 on 348 degrees of freedom
## Multiple R-squared:  0.8583, Adjusted R-squared:  0.8574
## F-statistic:  1054 on 2 and 348 DF,  p-value: < 2.2e-16
```

**Carat, Depth, CaratxDepth**

```
logmodel.c <- lm(log(TotalPrice)~Carat*Depth,data=Diamonds); summary(logmodel.c)
```

```
##
## Call:
## lm(formula = log(TotalPrice) ~ Carat * Depth, data = Diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.36271 -0.14008  0.03185  0.18673  0.93288
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.846674   0.489146   7.864 4.75e-14 ***
## Carat        4.869049   0.398366  12.223  < 2e-16 ***
## Depth        0.046610   0.007589   6.142 2.24e-09 ***
## Carat:Depth -0.048814   0.006023  -8.105 9.15e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3005 on 347 degrees of freedom
## Multiple R-squared:  0.8808, Adjusted R-squared:  0.8798
## F-statistic: 854.8 on 3 and 347 DF,  p-value: < 2.2e-16
```

**Carat^2, Depth^2, CaratxDepth**

```
logmodel.d <- lm(log(TotalPrice)~I(Carat^2) + I(Depth^2) + Carat*Depth,data=Diamonds); summary(logmodel
```

```
##
## Call:
## lm(formula = log(TotalPrice) ~ I(Carat^2) + I(Depth^2) + Carat *
##     Depth, data = Diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85021 -0.13209  0.01441  0.13613  0.79710
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.5049624  3.4020467   3.970 8.76e-05 ***
## I(Carat^2)  -0.5714071  0.0370821 -15.409  < 2e-16 ***
## I(Depth^2)   0.0013384  0.0007553   1.772   0.0773 .
## Carat        2.5863485  0.3414393   7.575 3.33e-13 ***
## Depth       -0.2027689  0.1015563  -1.997   0.0467 *
## Carat:Depth  0.0095943  0.0060107   1.596   0.1114
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2306 on 345 degrees of freedom
## Multiple R-squared:  0.9302, Adjusted R-squared:  0.9292
## F-statistic: 919.9 on 5 and 345 DF,  p-value: < 2.2e-16
```

**Compare to logged versions of 2 models From Example 3.11:**

```
#quadratic model
logmodel.e <- lm(log(TotalPrice)~Carat+I(Carat^2),data=Diamonds); summary(logmodel.e)
```

```
##
## Call:
## lm(formula = log(TotalPrice) ~ Carat + I(Carat^2), data = Diamonds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8215 -0.1313  0.0003  0.1391  0.8615
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.13042    0.05218  117.48   <2e-16 ***
## Carat        3.05963    0.08422   36.33   <2e-16 ***
## I(Carat^2)  -0.52730    0.02944  -17.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.238 on 348 degrees of freedom
## Multiple R-squared:  0.925,  Adjusted R-squared:  0.9246
## F-statistic:  2146 on 2 and 348 DF,  p-value: < 2.2e-16
```

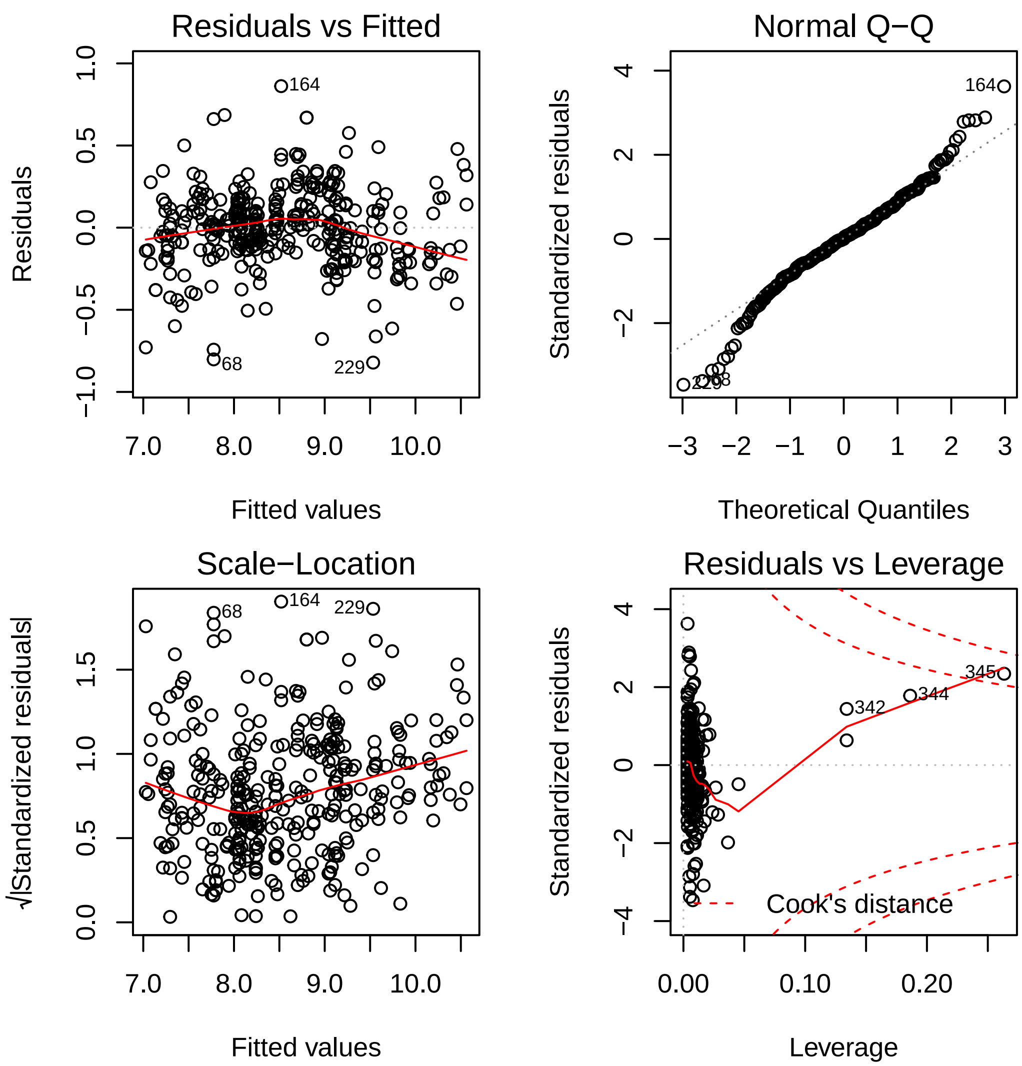
```
#cubic model
logmodel.f <- lm(log(TotalPrice)~Carat+I(Carat^2)+I(Carat^3),data=Diamonds); summary(logmodel.f)
```

```
##
## Call:
## lm(formula = log(TotalPrice) ~ Carat + I(Carat^2) + I(Carat^3),
##     data = Diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.80289 -0.12604 -0.00678  0.13125  0.80255
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.62372    0.09256  60.755  < 2e-16 ***
## Carat        4.46314    0.23106  19.316  < 2e-16 ***
## I(Carat^2)  -1.58885    0.16639  -9.549  < 2e-16 ***
## I(Carat^3)   0.22193    0.03430   6.471 3.32e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2252 on 347 degrees of freedom
## Multiple R-squared:  0.9331, Adjusted R-squared:  0.9325
## F-statistic:  1613 on 3 and 347 DF,  p-value: < 2.2e-16
```

**c. Residuals plots for chosen log model**

This is a vast improvement. There's still some tailing off on the normal probablity plot, but overall I think the residuals are close enough to normal. The resid vs fitted plot shows very nice random scatter with constant variance.

```
par(mar=c(4,4,2,2)); par(mfrow=c(2,2)); plot(logmodel.e)
```




## 3.25. Diamonds (continued)

The complete second-order model from 3.23d was called `model.d`. The reduced model without any of the *Depth* terms was called `model.e`. We want to do a nested F-test to compare these models.

The p-value is very close to zero, which gives strong evidence that at least one of the terms involving *Depth* should be included in the model and that dropping all three would significantly impair the effectiveness for predicting *TotalPrice*.

```
anova(model.e, model.d, test="F")
```

```
## Analysis of Variance Table
##
## Model 1: TotalPrice ~ Carat + I(Carat^2)
```

```
## Model 2: TotalPrice ~ I(Carat^2) + I(Depth^2) + Carat * Depth
##   Res.Df        RSS Df Sum of Sq      F   Pr(>F)
## 1    348 1574044410
## 2    345 1454702094  3 119342316 9.4345 5.24e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

### 3.26 Diamonds (continued)

Notice that this asks us to use the quadratic model with Y=$TotalPrice$ and the two predictors $Carat, Carat^2$ (which is `model.e` above).

```r
predict.lm(model.e, data.frame("Carat"=0.5),interval="confidence")
```

```
##        fit      lwr      upr
## 1 1794.843 1424.296 2165.389
```

```r
predict.lm(model.e, data.frame("Carat"=0.5),interval="prediction")
```

```
##        fit       lwr      upr
## 1 1794.843 -2404.462 5994.147
```

**a.** The model predicts that the average total price for a 0.5-carat diamond is \$1795.

**b.** We are 95% confident that the average price of *all* 0.5-carat diamonds is between \$1424 and \$2165. But this does not mean your particular 0.5-carat diamond will be in this range, only that the average of *all* such diamonds are in this range.

**c.** We expect that 95% of all 0.5-carat diamonds will cost between \$0 and \$5994. (Since a diamond can not cost a negative amount of money.) So we are 95% confident that your particular 0.5-carat diamond will be in this range.

**d.** Using the model I thought was best in 3.24b, which I called `logmodel.e`,

```r
predict.lm(logmodel.e, data.frame("Carat"=0.5),interval="confidence")
```

```
##        fit      lwr     upr
## 1 7.528412 7.486943 7.56988
```

```r
exp(predict.lm(logmodel.e, data.frame("Carat"=0.5),interval="confidence"))
```

```
##        fit      lwr      upr
## 1 1860.149 1784.588 1938.908
```

```r
predict.lm(logmodel.e, data.frame("Carat"=0.5),interval="prediction")
```

```
##        fit      lwr      upr
## 1 7.528412 7.058458 7.998366
```

```r
exp(predict.lm(logmodel.e, data.frame("Carat"=0.5),interval="prediction"))
```

```
##        fit      lwr     upr
## 1 1860.149 1162.651 2976.09
```

We are 95% confident that the average price of *all* 0.5-carat diamonds is between \$1784.59 and \$1938.91. But this does not mean your particular 0.5-carat diamond will be in this range, only that the average of *all* such diamonds are in this range.

We expect that 95% of all 0.5-carat diamonds will cost between \$1162.65 and \$2976.09. So we are 95% confident that your particular 0.5-carat diamond will be in this range.