Lab 10

Exercise 1

We will continue using diamonds dataset

```
diamonds <- read.csv("diamonds.csv")

diamonds$price = as.numeric(diamonds$price)

## Select Numeric Columns
diamonds = diamonds[, sapply(diamonds, class) == "numeric"]
head(diamonds)
```

```
##   carat depth table price length.in.mm width.of.mm depth.in.mm
## 1  0.23  61.5    55   326         3.95        3.98        2.43
## 2  0.21  59.8    61   326         3.89        3.84        2.31
## 3  0.23  56.9    65   327         4.05        4.07        2.31
## 4  0.29  62.4    58   334         4.20        4.23        2.63
## 5  0.31  63.3    58   335         4.34        4.35        2.75
## 6  0.24  62.8    57   336         3.94        3.96        2.48
```

```
### Fitting a linear model
fit <- lm(price ~ ., data = diamonds)
summary(fit)
```

```
##
## Call:
## lm(formula = price ~ ., data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23878.2   -615.0    -50.7    347.9  12759.2
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20849.316    447.562  46.584  < 2e-16 ***
## carat        10686.309     63.201 169.085  < 2e-16 ***
## depth         -203.154      5.504 -36.910  < 2e-16 ***
## table         -102.446      3.084 -33.216  < 2e-16 ***
## length.in.mm -1315.668     43.070 -30.547  < 2e-16 ***
## width.of.mm     66.322     25.523   2.599  0.00937 **
## depth.in.mm     41.628     44.305   0.940  0.34744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1497 on 53933 degrees of freedom
## Multiple R-squared:  0.8592, Adjusted R-squared:  0.8592
## F-statistic: 5.486e+04 on 6 and 53933 DF,  p-value: < 2.2e-16
```

```
### Getting fitted values
fitted.vals = fit$fitted.values
```

(a)

Calculate the residuals, the residual sum of squares (RSS), and the total sum of squares (TSS) using the `fitted.value()`

```
# Insert you code here
residual <- residuals(fit)
RSS <- sum(residual^2)
TSS <- sum(((diamonds$price) - mean(diamonds$price))^2)
RSS
```

```
## [1] 1.20857e+11
```

```
TSS
```

```
## [1] 858473135517
```

(b)

Calculate the R-square (R^2) using RSS and TSS. What is the interpretation of R^2?

```
# Insert you code here, save your results as `Rsq`
Rsq <- 1 - (RSS/TSS)
Rsq
```

```
## [1] 0.8592187
```

R^2 indicates thetg goddness of the fit as it's the coefficient of determination.

(c)

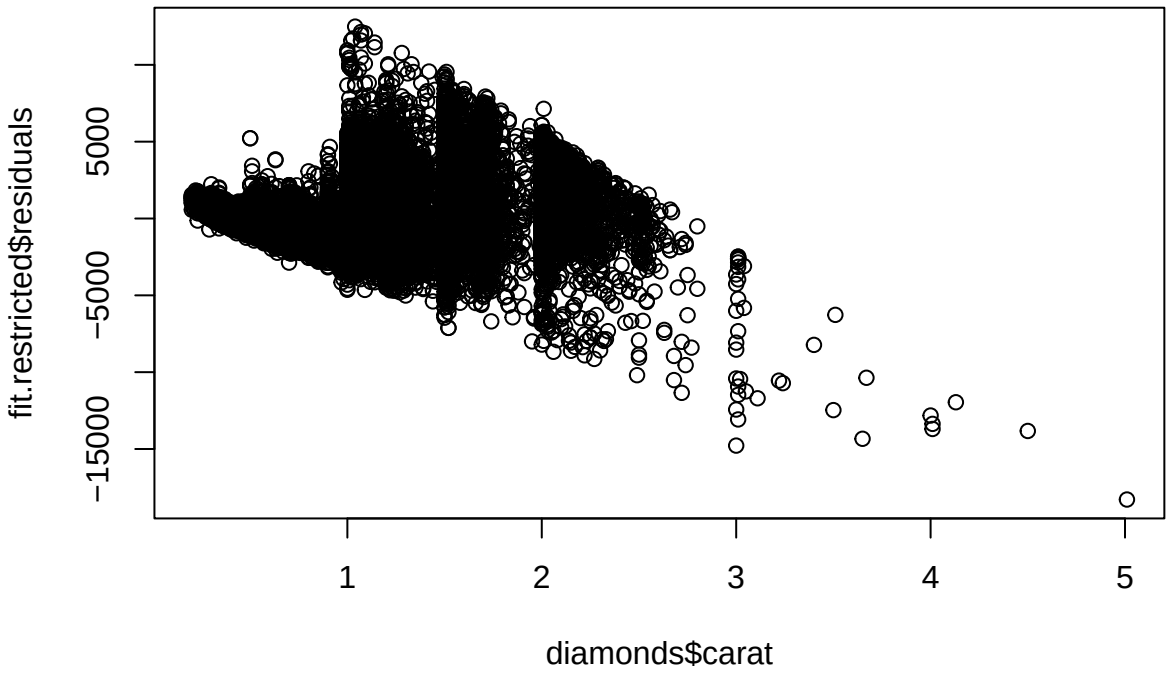
Fit another multivariate model (fit.restricted), but this time, drop `length.in.mm`, `width.of.mm` and `depth.in.mm`. Plot the residuals from this model against the variables added as covariates.

```
fit.restricted <- lm(price ~ carat + depth + table, data = diamonds)
summary(fit.restricted)
```

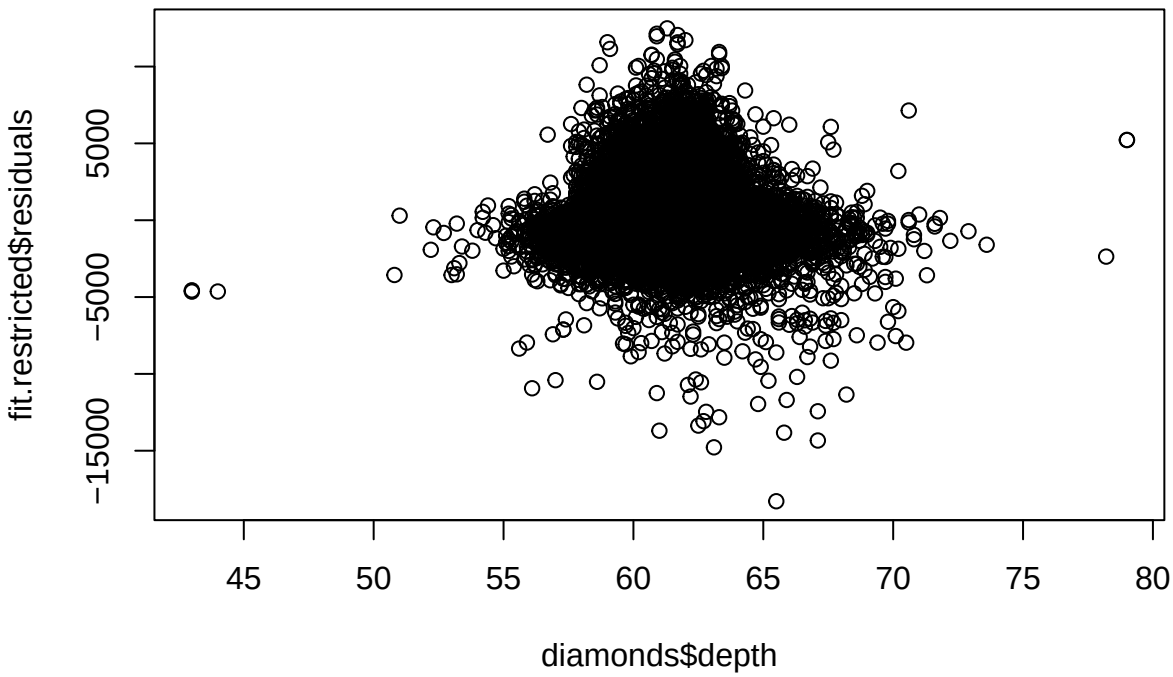
```
##
## Call:
## lm(formula = price ~ carat + depth + table, data = diamonds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18288.0   -785.9    -33.2    527.2  12486.7
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13003.441    390.918   33.26   <2e-16 ***
## carat        7858.771     14.151  555.36   <2e-16 ***
## depth        -151.236      4.820  -31.38   <2e-16 ***
## table        -104.473      3.141  -33.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1526 on 53936 degrees of freedom
```

```
## Multiple R-squared:  0.8537, Adjusted R-squared:  0.8537
## F-statistic: 1.049e+05 on 3 and 53936 DF,  p-value: < 2.2e-16
```

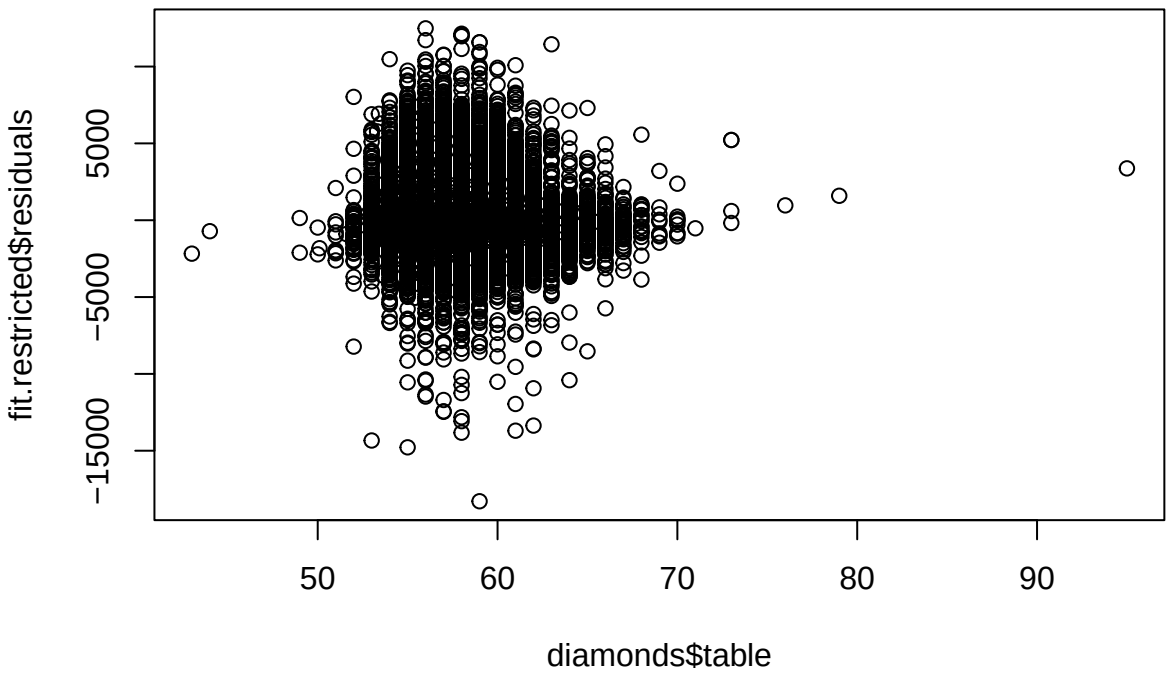
```
plot(diamonds$carat, fit.restricted$residuals)
```




```
plot(diamonds$depth, fit.restricted$residuals)
```




```
plot(diamonds$table, fit.restricted$residuals)
```



Regression dianosis

Red wine dataset

Reload the data.

```
wine<- read.csv("winequality-red.csv", sep = ";")
wine$quality <- wine$quality + rnorm(length(wine$quality))
```

Fit the model.

```
wine.fit <- lm(quality~volatile.acidity+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+pH+sulphates+
summary(wine.fit)
```

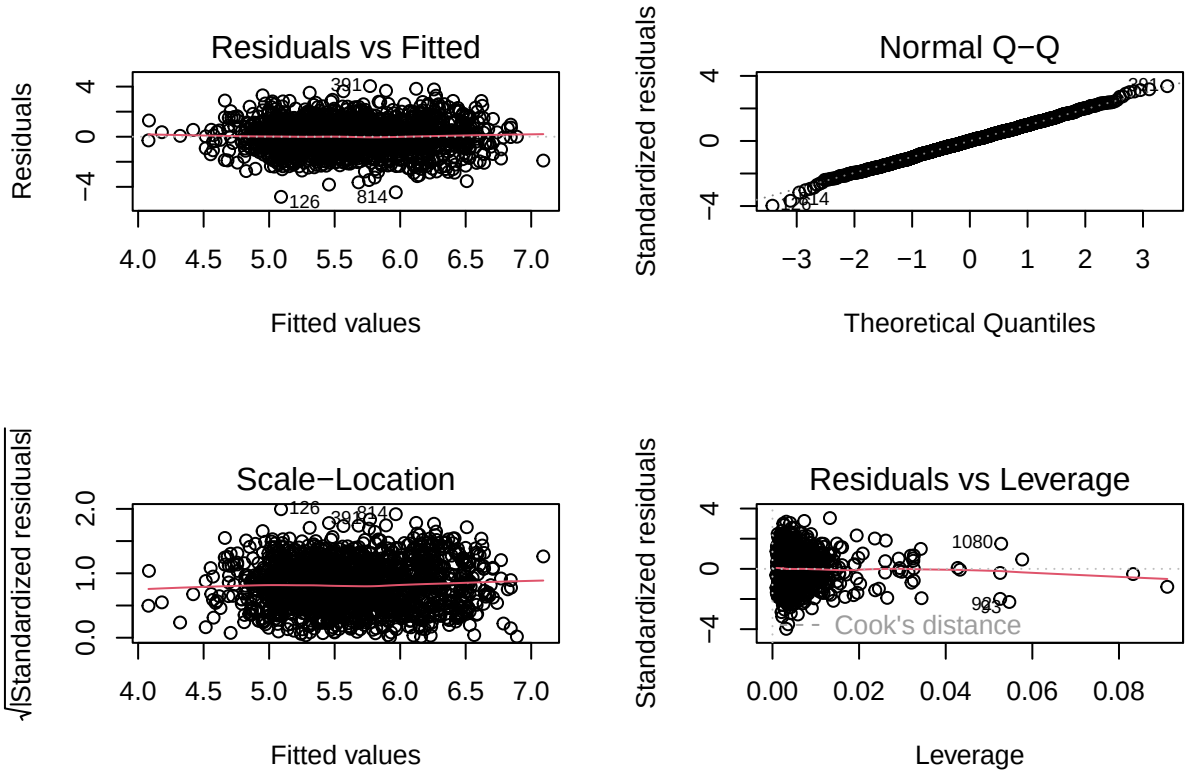
```
##
## Call:
## lm(formula = quality ~ volatile.acidity + chlorides + free.sulfur.dioxide +
##     total.sulfur.dioxide + pH + sulphates + alcohol, data = wine)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7979 -0.7996  0.0143  0.7904  4.0408
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.025143   0.751167   6.690 3.09e-11 ***
## volatile.acidity     -1.259405   0.188004  -6.699 2.91e-11 ***
```

```
## chlorides            -2.882170   0.741146  -3.889 0.000105 ***
## free.sulfur.dioxide   0.005742   0.003963   1.449 0.147504
## total.sulfur.dioxide -0.003983   0.001280  -3.111 0.001897 **
## pH                   -0.412464   0.219166  -1.882 0.060022 .
## sulphates             0.795293   0.204905   3.881 0.000108 ***
## alcohol               0.235885   0.031313   7.533 8.26e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.208 on 1591 degrees of freedom
## Multiple R-squared:  0.1367, Adjusted R-squared:  0.1329
## F-statistic: 35.98 on 7 and 1591 DF,  p-value: < 2.2e-16
```

Exercise 2

(a) Do regression diagnostics using the `plot` function.

```
# insert your code here to do regression diagnostics.
par(mfrow = c(2,2))
plot(wine.fit)
```



(b) Answer the following TRUE/FALSE questions based on the diagnostics plot. Uncomment your answer.

```
### I. The plot indicates heteroscedasticity.
# TRUE
FALSE
```

```
## [1] FALSE
```

```
### II. There are non-linearity between the explantory variable and response variable.
# TRUE
FALSE
```

```
## [1] FALSE
```

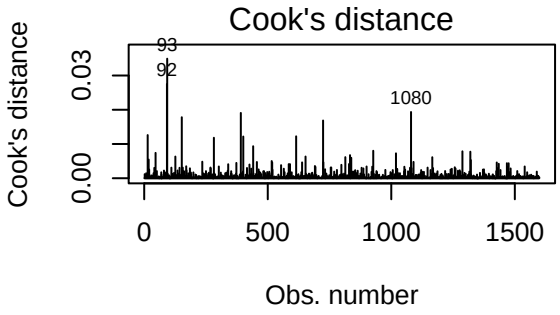
```
### III. The normal assumtion holds for this model.
TRUE
```

```
## [1] TRUE
```

```
# FALSE
```

(c) Identify at least two outliers from the data.

```
par(mfrow = c(2,2))
plot(wine.fit, which = 4)
```



I think the sample 107, 152, 282, 724, and 911 are outliers.

Multiple regression with continuous and categorical variables

Exercise 3

(a) Fit a linear regression model with explanatory variable `carat`, `depth`, `table`, `clarity`, `color` and `cut`.

```
diamonds <- read.csv("diamonds.csv")
head(diamonds,2)
```

```
##   carat     cut color clarity depth table price length.in.mm width.of.mm
```

```
## 1  0.23   Ideal     E     SI2  61.5    55   326         3.95        3.98
## 2  0.21 Premium     E     SI1  59.8    61   326         3.89        3.84
##   depth.in.mm
## 1        2.43
## 2        2.31
```

```
# Insert you code here, save your model as `fit.categorical`
fit.categorical <- lm(price ~ carat + depth + table + clarity + color + cut, data = diamonds)
fit.categorical$coefficients
```

```
##  (Intercept)        carat        depth        table    clarityIF   claritySI1
##  -4555.17136   8895.19401    -21.02361    -24.80274   5404.23650   3567.79380
##   claritySI2   clarityVS1   clarityVS2  clarityVVS1  clarityVVS2       colorE
##   2619.00402   4525.40005   4210.19430   5061.73441   4957.31037   -210.84903
##       colorF       colorG       colorH       colorI       colorJ      cutGood
##   -304.28757   -506.96369   -977.97368  -1438.27734  -2322.56486    614.42393
##     cutIdeal   cutPremium cutVery Good
##    877.56912    806.02434    778.42817
```

(b) Write the equation when

i. Clarity is VS2, color is H, and cut is Premium. Replace ??? with numerical values.

$$\text{price} = -9837.52239+5789.57454-766.83969+1003.89107+8670.72073\cdot\text{carat}+-5.27080\cdot\text{depth}+18.73633\cdot\text{table}$$

ii. clarity is I1, color is D and cut is Fair. Replace ??? by numerical values.

$$\text{price} = -9837.52239 + 8895.19401 \cdot \text{carat} - 5.27080 \cdot \text{depth} + 18.73633 \cdot \text{table}$$

Diamond dataset

We will include categorical variables in the following analysis. Read the data.

```
diamonds <- read.csv("diamonds.csv")
diamonds <- diamonds[sample(1:nrow(diamonds), 1000), ]
head(diamonds)
```

```
##       carat       cut color clarity depth table price length.in.mm width.of.mm
## 7982   1.12     Ideal     J     SI2  62.5    57  4325         6.63        6.59
## 7813   1.13   Premium     H     SI2  62.3    59  4294         6.68        6.65
## 36484  0.32     Ideal     F      IF  61.3    57   943         4.42        4.46
## 32707  0.31     Ideal     F     VS2  61.5    56   802         4.37        4.34
## 5767   0.90 Very Good     F     SI1  63.4    56  3898         6.06        6.12
## 23829  1.56   Premium     F     SI1  60.7    59 11901         7.55        7.45
##       depth.in.mm
## 7982         4.13
## 7813         4.15
## 36484        2.72
## 32707        2.68
## 5767         3.86
## 23829        4.55
```

Fit a linear regression.

```
diamond.fit <- lm(price ~ carat + cut + color + clarity + depth + table, data = diamonds)
summary(diamond.fit)
```

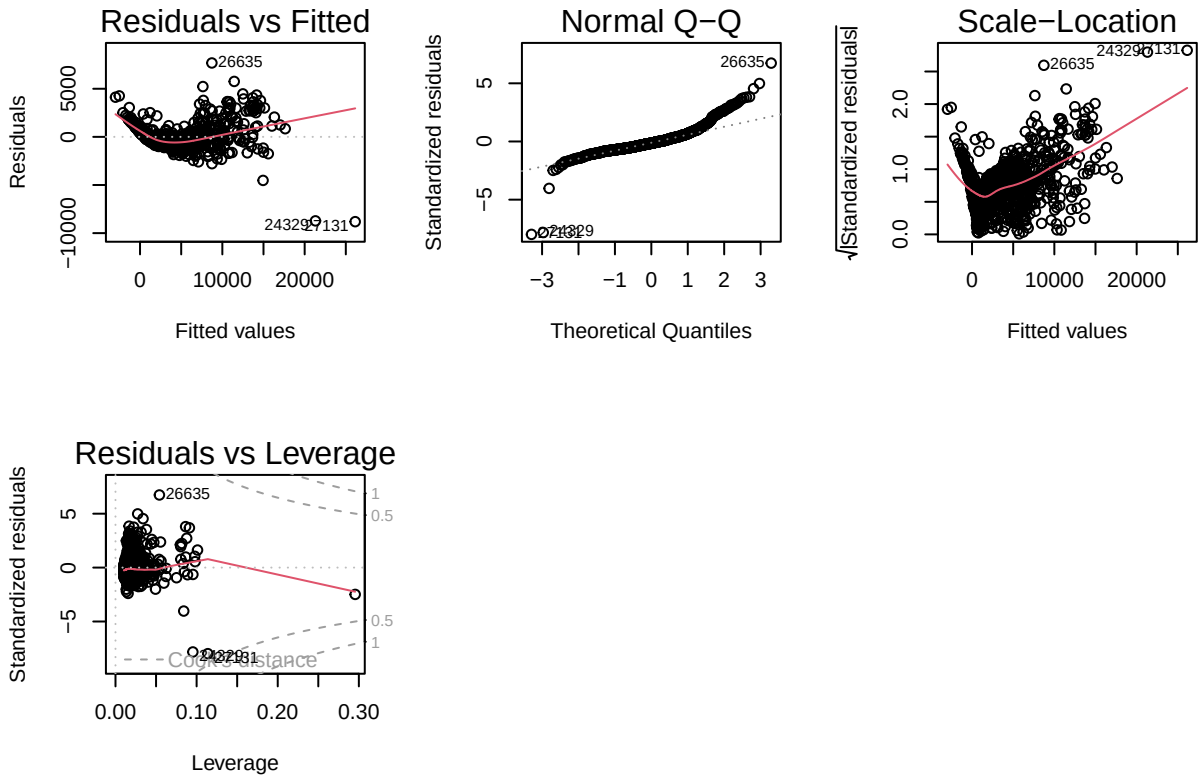
```
##
## Call:
```

```
## lm(formula = price ~ carat + cut + color + clarity + depth +
##     table, data = diamonds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8810.2  -659.7  -169.3   415.9  7678.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3750.37    2554.84  -1.468 0.142440
## carat         8687.82      89.13  97.473  < 2e-16 ***
## cutGood        513.51     244.50   2.100 0.035963 *
## cutIdeal       418.96     234.98   1.783 0.074909 .
## cutPremium     569.94     228.89   2.490 0.012939 *
## cutVery Good   434.24     229.04   1.896 0.058263 .
## colorE        -336.92     135.96  -2.478 0.013377 *
## colorF        -502.23     139.54  -3.599 0.000335 ***
## colorG        -582.16     134.08  -4.342 1.56e-05 ***
## colorH       -1142.14     145.19  -7.866 9.61e-15 ***
## colorI       -1461.90     167.53  -8.726  < 2e-16 ***
## colorJ       -2072.20     193.80 -10.693  < 2e-16 ***
## clarityIF     7096.66     379.46  18.702  < 2e-16 ***
## claritySI1    5228.88     328.06  15.939  < 2e-16 ***
## claritySI2    4459.12     328.51  13.574  < 2e-16 ***
## clarityVS1    6194.88     337.76  18.341  < 2e-16 ***
## clarityVS2    5837.43     332.43  17.560  < 2e-16 ***
## clarityVVS1   6649.43     359.60  18.491  < 2e-16 ***
## clarityVVS2   6591.04     346.66  19.013  < 2e-16 ***
## depth          -40.56      27.69  -1.465 0.143353
## table          -36.40      21.80  -1.670 0.095324 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1172 on 979 degrees of freedom
## Multiple R-squared:  0.9149, Adjusted R-squared:  0.9132
## F-statistic: 526.4 on 20 and 979 DF,  p-value: < 2.2e-16
```

Exercise 4

(a) Do regression diagnostics using the `plot` function.

```
# insert your code here to do regression diagnostics.
par(mfrow = c(2,3))
plot(diamond.fit)
```



(b) Answer the following TRUE/FALSE questions based on the diagnostics plot. Uncomment your answer.

```
### I. The plot indicates heteroscedasticity.
TRUE
```

```
## [1] TRUE
```

```
# FALSE
### II. There are non-linearity between the explantory variable and response variable.
TRUE
```

```
## [1] TRUE
```

```
# FALSE
### III. The normal assumtion holds for this model.
# TRUE
FALSE
```

```
## [1] FALSE
```