Final Project

William Su

2025-06-12

Data Description and Descriptive Statistics

1. Select a random sample.

```
diamonds_df <- read.csv("Diamonds Prices2022.csv")

set.seed(6122025)

library(tidyverse)

# select random sample of at least 1000 observations
diamonds_sample <- sample_n(diamonds_df, 1000)
```

2. Describe all the variables (call summary function on the dataset, see the structure, create histograms for continuous random variable, comment on their distribution, bar plots for categorical random variable).

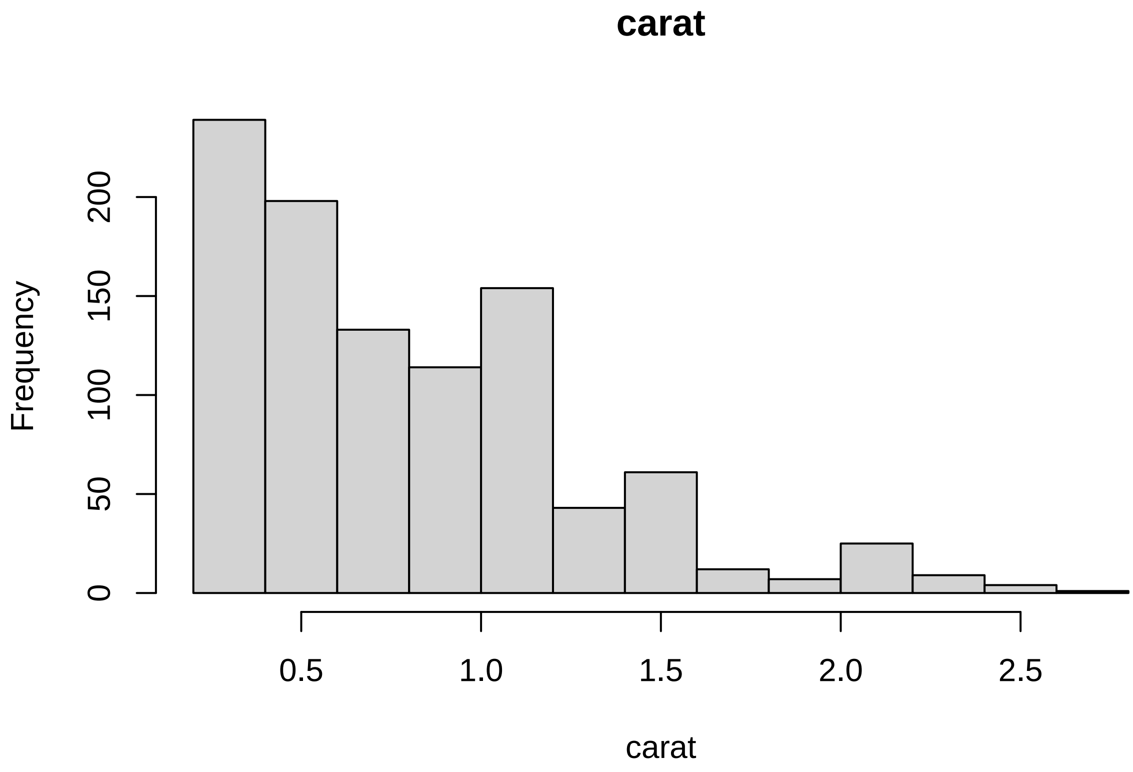
```
# summary function on the dataset
summary(diamonds_sample)
```

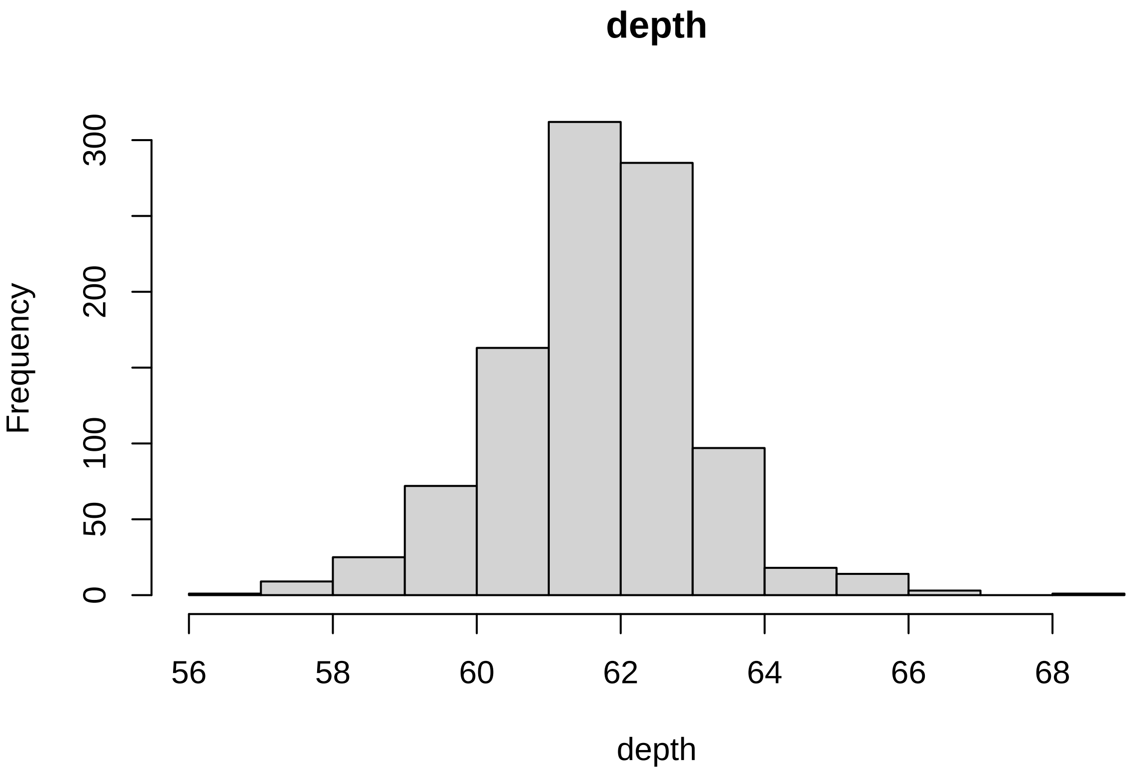
```
##        X             carat            cut               color
##  Min.   :   72   Min.   :0.2300   Length:1000        Length:1000
##  1st Qu.:13403   1st Qu.:0.4100   Class :character   Class :character
##  Median :26342   Median :0.7100   Mode  :character   Mode  :character
##  Mean   :26991   Mean   :0.8061
##  3rd Qu.:40915   3rd Qu.:1.0325
##  Max.   :53938   Max.   :2.7500
##    clarity              depth           table           price
##  Length:1000        Min.   :56.00   Min.   :50.00   Min.   :  361
##  Class :character   1st Qu.:60.98   1st Qu.:56.00   1st Qu.: 1006
##  Mode  :character   Median :61.90   Median :57.00   Median : 2444
##                     Mean   :61.75   Mean   :57.44   Mean   : 4037
##                     3rd Qu.:62.60   3rd Qu.:59.00   3rd Qu.: 5364
##                     Max.   :68.30   Max.   :68.00   Max.   :18760
##        x               y               z
##  Min.   :3.930   Min.   :3.970   Min.   :2.400
##  1st Qu.:4.758   1st Qu.:4.770   1st Qu.:2.950
##  Median :5.715   Median :5.740   Median :3.530
##  Mean   :5.762   Mean   :5.764   Mean   :3.557
##  3rd Qu.:6.520   3rd Qu.:6.500   3rd Qu.:4.020
##  Max.   :9.040   Max.   :8.980   Max.   :5.490
```

```
# see the structure
head(diamonds_sample)
```

```
##       X carat       cut color clarity depth table price    x    y    z
## 1 50753  0.70      Good     D     SI1  64.2    60  2298 5.59 5.62 3.60
## 2 33104  0.32     Ideal     E    VVS2  61.2    56   816 4.39 4.43 2.70
## 3 34100  0.36   Premium     D     VS2  61.0    58   852 4.59 4.62 2.81
## 4 15341  1.01 Very Good     F     VS2  61.5    57  6159 6.40 6.48 3.96
## 5 17010  1.52   Premium     I     VS2  61.7    61  6793 7.35 7.30 4.52
## 6 42435  0.53     Ideal     D     SI2  60.4    57  1314 5.26 5.30 3.19
```

```
cont_vars <- c("carat","depth","table","price","x","y","z")

for (var in cont_vars) {
  hist(diamonds_sample[[var]],
       main = var,
       xlab = var)
}
```




depth
Frequency
0
50
100
200
300
56
58
60
62
64
66
68
depth

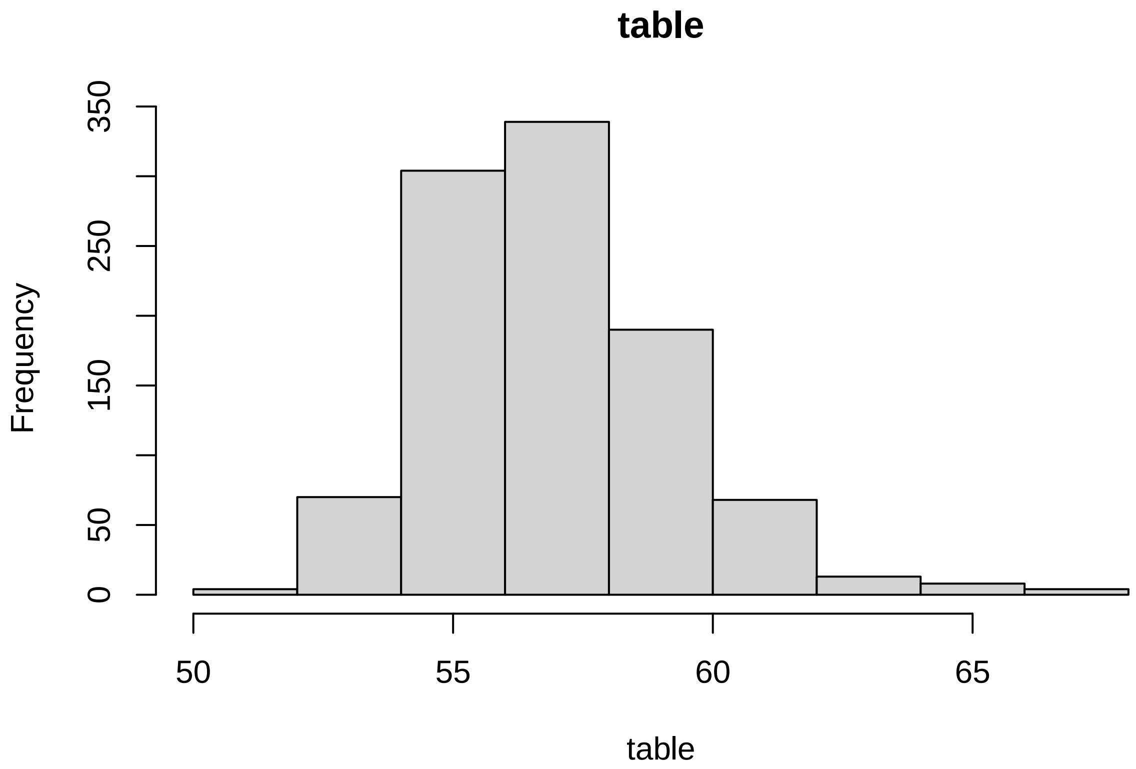
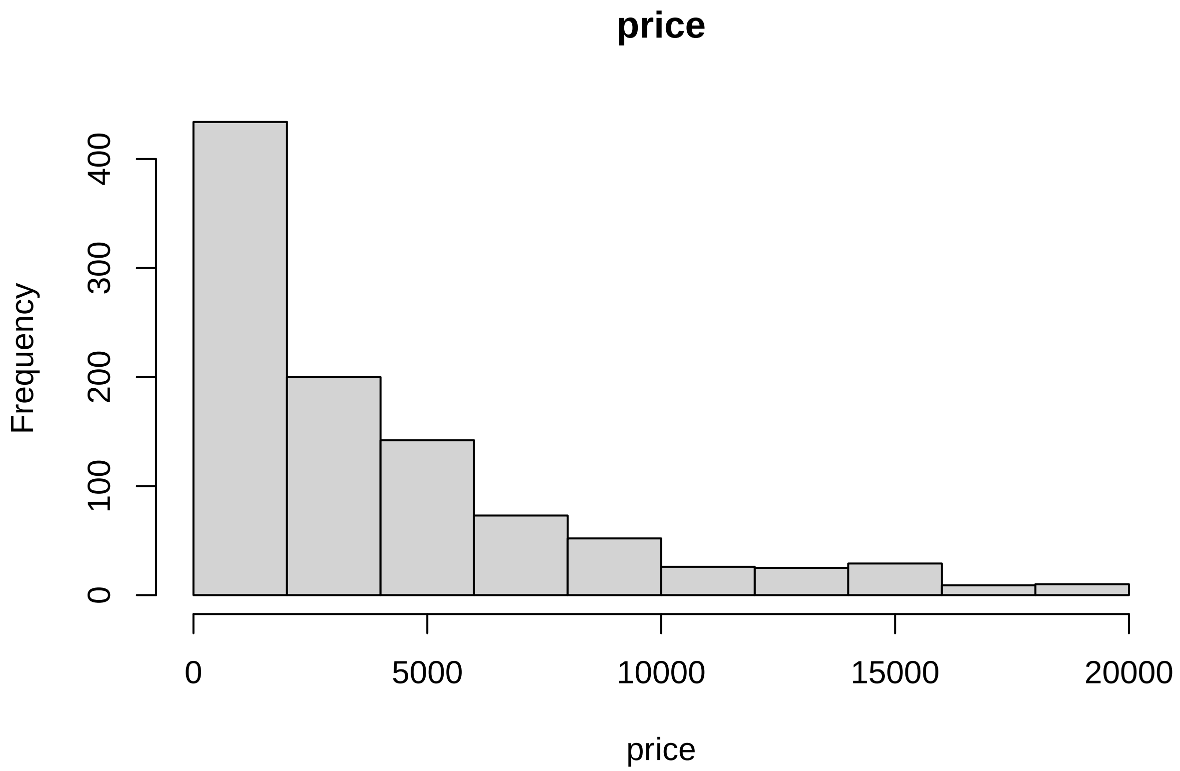
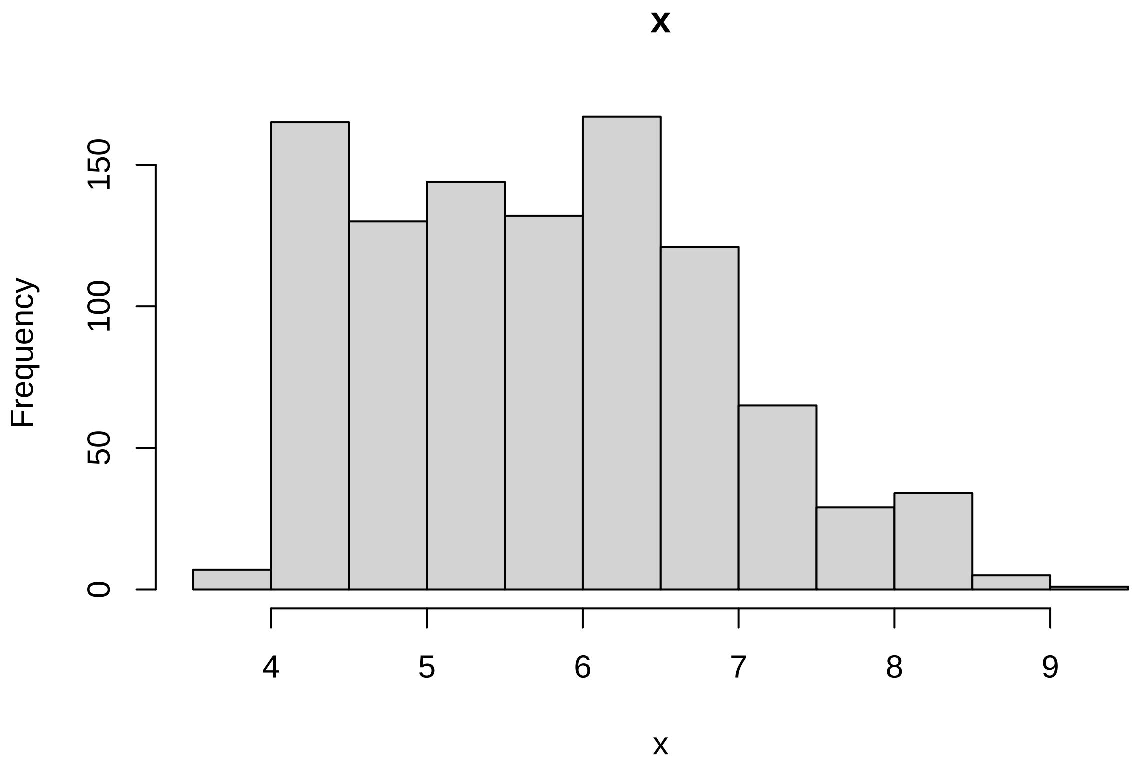
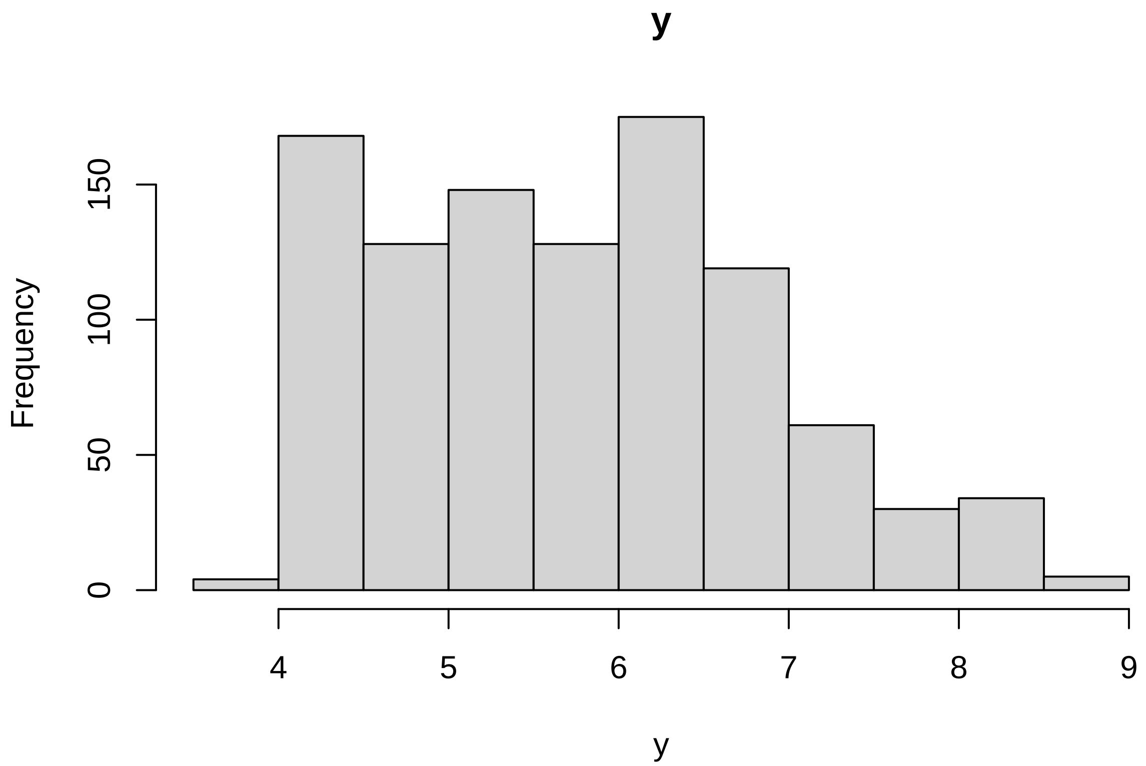
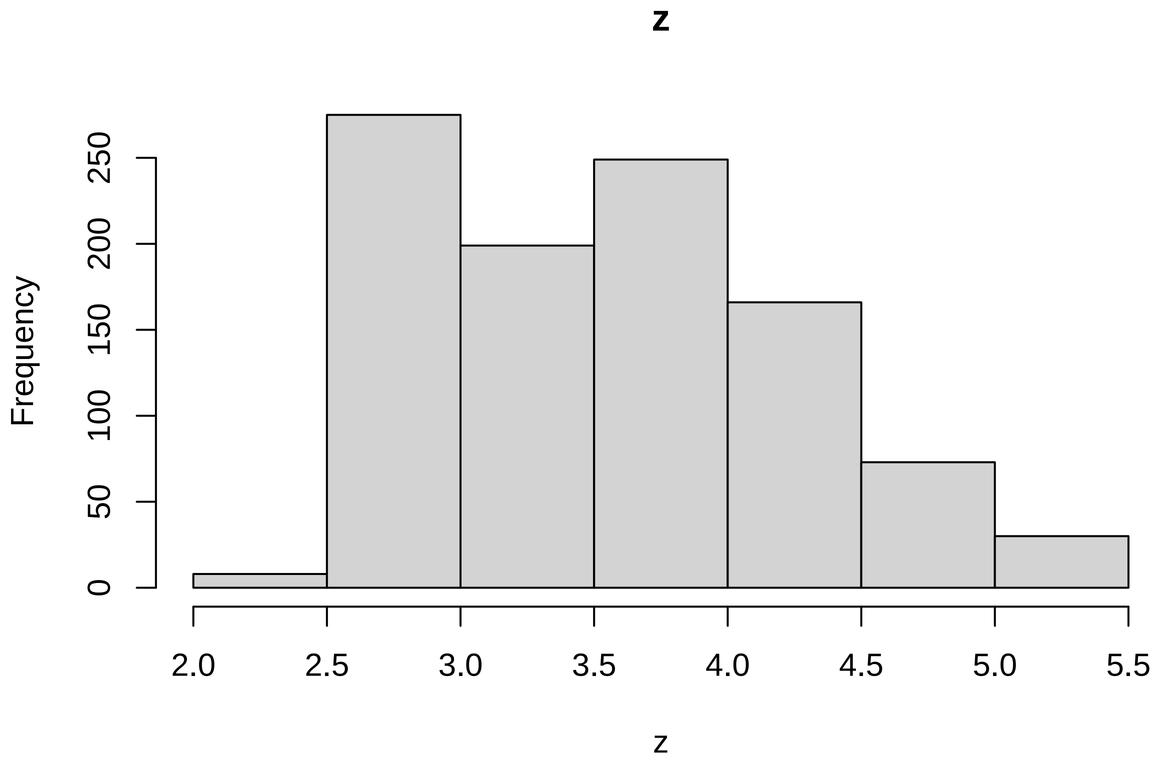


table
Frequency
0
50
150
250
350
50
55
60
65
table


price
Frequency
0
100
200
300
400
0
5000
10000
15000
20000
price


x
Frequency
0
50
100
150
4
5
6
7
8
9
x


y
Frequency
0
50
100
150
4
5
6
7
8
9
y



Carat

- Right-skewed: most diamonds in our sample are under 1 carat, with a long tail stretching past 2 carats.
- Spikes at “round” numbers (0.5, 1.0, etc.), probably due to how dealers often cut to standard weights.

Depth (%)

- Roughly bell-shaped, centered around 61%, with most depths between 59% and 63%.
- Very few extreme flats (<58%) or very deep stones (>65%).

Table (%)

- Also approximately normal, peaked at about 57–59%.
- Tight spread, most tables fall within a narrow 55–61% band.

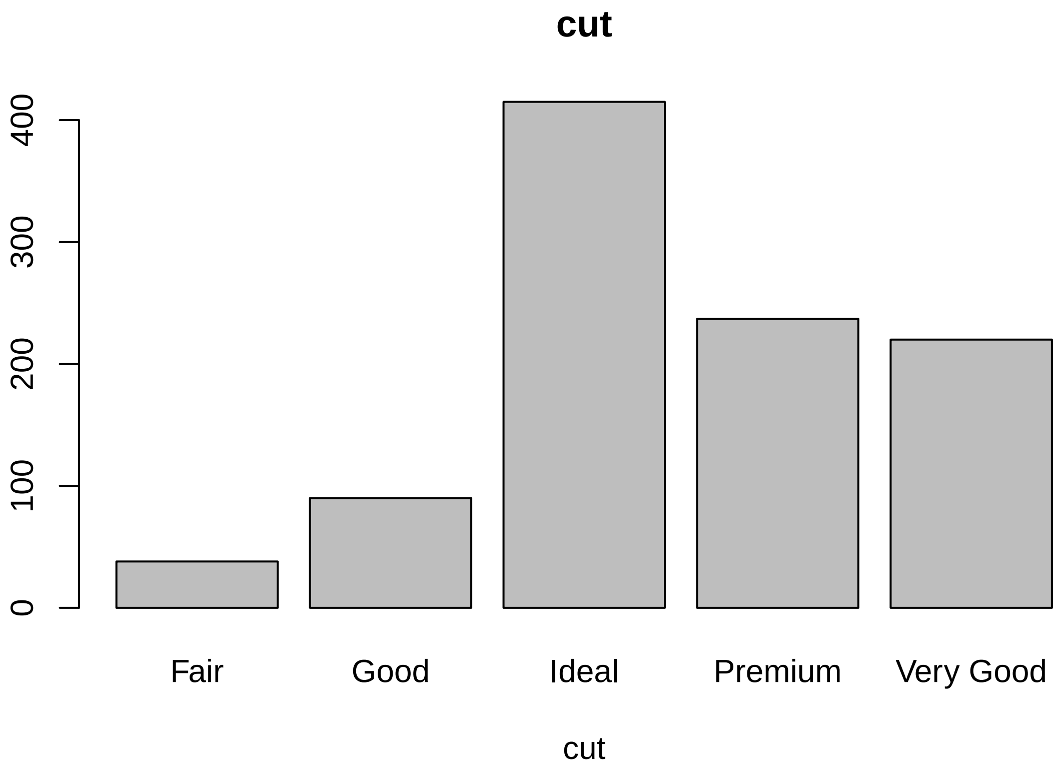
Price (USD)

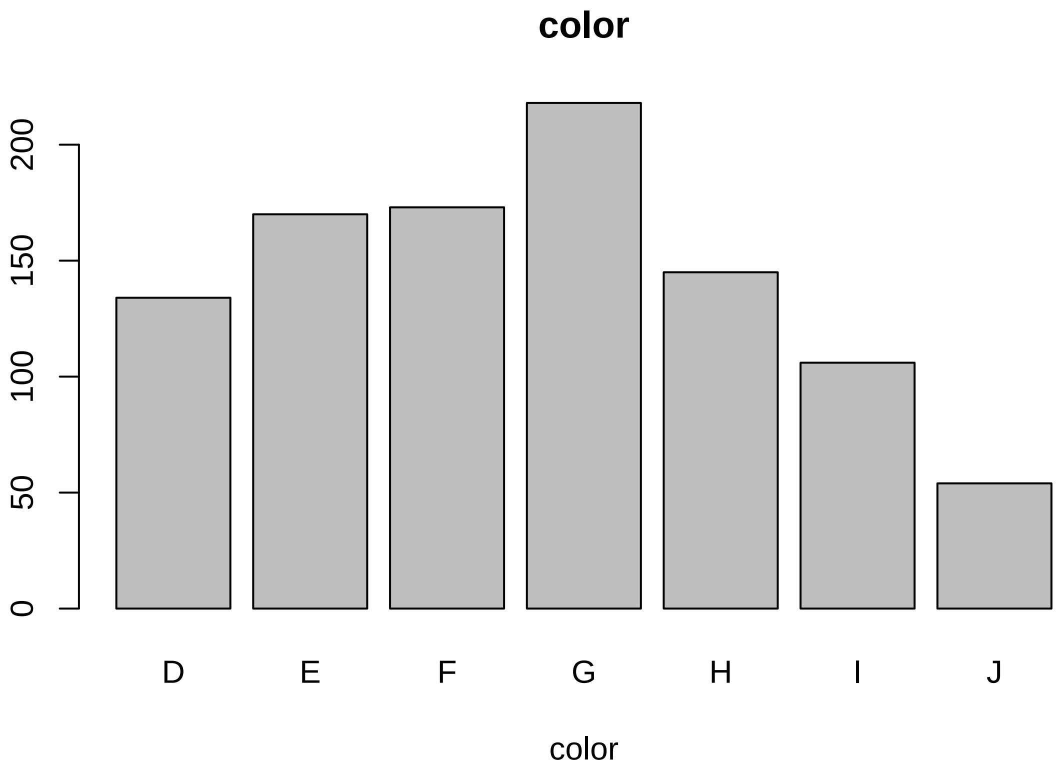
- Heavily right-skewed: a cluster under $4,000, then a long tail out to $15,000+.
- A few very expensive outliers push the mean above the bulk of the data.

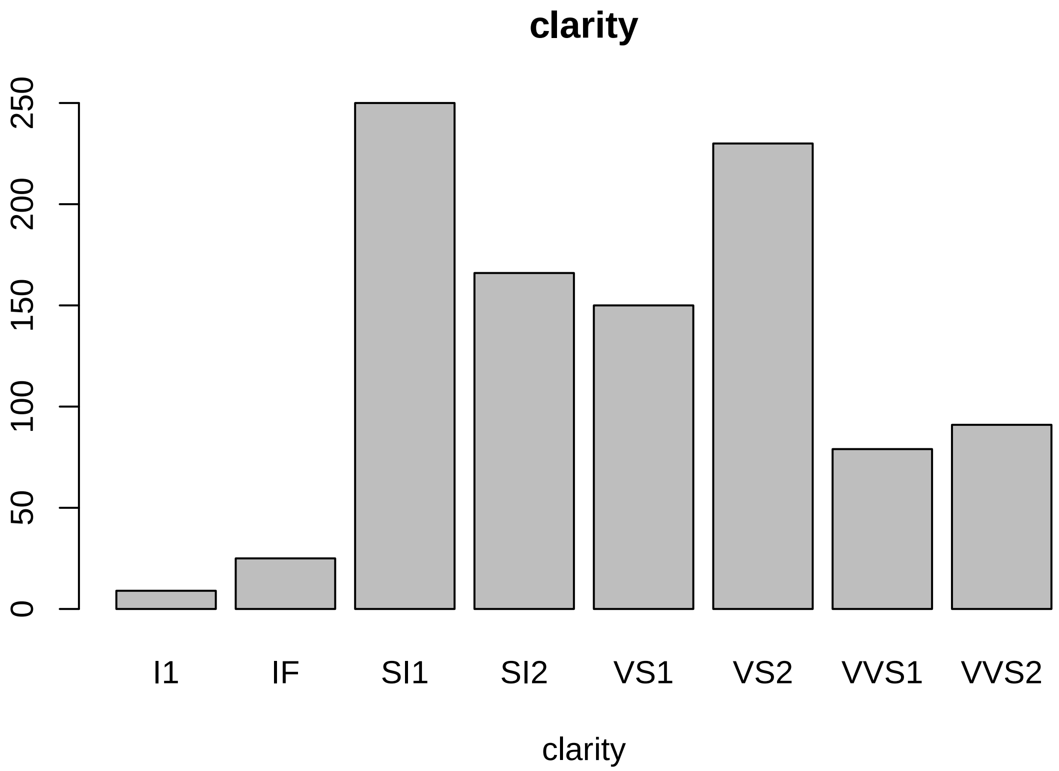
X, Y, Z (mm dimensions)

- Each dimension is right-skewed, with a bulk in the mid-ranges (x: ~5–7 mm; y: ~5–7 mm; z: ~3–4 mm) and fewer very large stones.
- You can see a taller "bar" at the lower end (around 4–5 mm for x and y), again reflecting standard small cuts.
- The z-dimension (height) is a bit more spread out but still clustered around 3.5–4 mm

```
cat_vars <- c("cut","color","clarity")

for (var in cat_vars) {
  barplot(table(diamonds_sample[[var]]),
          main = var,
          xlab = var)
}
```




color
0
50
100
150
200
D
E
F
G
H
I
J
color



3. Determine if there is any correlation between these variables.

```r
cor(diamonds_sample[, cont_vars])
```

```
##             carat       depth       table       price           x           y
## carat  1.00000000 -0.03096596  0.15198157  0.93065636  0.97733050  0.97712196
## depth -0.03096596  1.00000000 -0.32939615 -0.06566565 -0.08353924 -0.08590045
## table  0.15198157 -0.32939615  1.00000000  0.09530257  0.17311616  0.16794151
## price  0.93065636 -0.06566565  0.09530257  1.00000000  0.89351985  0.89568291
## x      0.97733050 -0.08353924  0.17311616  0.89351985  1.00000000  0.99895220
## y      0.97712196 -0.08590045  0.16794151  0.89568291  0.99895220  1.00000000
## z      0.97645962  0.03787166  0.13089193  0.88862672  0.99215123  0.99182162
##                z
## carat 0.97645962
## depth 0.03787166
## table 0.13089193
## price 0.88862672
## x     0.99215123
## y     0.99182162
## z     1.00000000
```

The continuous predictors exhibit very high intercorrelation: carat and price correlate at about 0.93, and carat with each of the physical dimensions (x, y, z) at roughly 0.98. Likewise, x, y, and z correlate nearly perfectly with one another ($p \approx 0.99$), and all three also correlate strongly with price ($p \approx 0.89$˘0.90). By contrast, depth and table show only weak associations ($|p| < 0.33$ with size and price.

4. Run the multiple linear regression model using all these variables and observe the summary statistics. (DO NOT EXPLAIN HYPOTHESIS TESTING OR ANYTHING ELSE)

```r
model <- lm(price ~ ., data = diamonds_sample)

summary(model)
```

```
##
## Call:
## lm(formula = price ~ ., data = diamonds_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7677.2  -571.7  -141.6   425.0  9235.0
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.898e+04  9.217e+03  -3.144  0.00172 **
## X             3.995e-03  2.369e-03   1.686  0.09208 .
## carat         1.187e+04  3.649e+02  32.532  < 2e-16 ***
## cutGood       3.177e+02  2.172e+02   1.463  0.14384
## cutIdeal      4.589e+02  2.106e+02   2.179  0.02959 *
## cutPremium    5.425e+02  2.030e+02   2.673  0.00765 **
## cutVery Good  3.824e+02  2.059e+02   1.858  0.06354 .
## colorE       -3.933e+02  1.242e+02  -3.167  0.00159 **
## colorF       -3.333e+02  1.249e+02  -2.669  0.00774 **
## colorG       -5.710e+02  1.202e+02  -4.750 2.34e-06 ***
## colorH       -9.727e+02  1.307e+02  -7.445 2.12e-13 ***
## colorI       -1.630e+03  1.432e+02 -11.380  < 2e-16 ***
## colorJ       -2.270e+03  1.788e+02 -12.695  < 2e-16 ***
## clarityIF     4.844e+03  4.295e+02  11.278  < 2e-16 ***
## claritySI1    2.764e+03  3.735e+02   7.401 2.91e-13 ***
## claritySI2    1.776e+03  3.748e+02   4.739 2.46e-06 ***
## clarityVS1    3.726e+03  3.790e+02   9.830  < 2e-16 ***
## clarityVS2    3.409e+03  3.740e+02   9.116  < 2e-16 ***
## clarityVVS1   4.099e+03  3.910e+02  10.481  < 2e-16 ***
## clarityVVS2   3.972e+03  3.880e+02  10.236  < 2e-16 ***
## depth         4.777e+02  1.449e+02   3.297  0.00101 **
## table        -3.541e+01  1.930e+01  -1.835  0.06685 .
## x            -7.293e+02  1.054e+03  -0.692  0.48897
## y             5.328e+03  1.055e+03   5.049 5.29e-07 ***
## z            -9.421e+03  2.350e+03  -4.010 6.54e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1062 on 975 degrees of freedom
## Multiple R-squared:  0.9334, Adjusted R-squared:  0.9317
## F-statistic:   569 on 24 and 975 DF,  p-value: < 2.2e-16
```

Observations: Carat, cut, color, and clarity are the most statistically significant. Other parameters are not as important.

5. Comment on anything of interest that occurred in this part. Were the data approximately what you expected, or did some of the results surprise you?

Overall, the diamonds data behaved as expected; most stones are small and inexpensive, so both carat and price are heavily right-skewed, while depth and table cluster around their “ideal” ranges. Size truly drives value. In our regression, carat dominates with an increase of nearly $12,000 per carat, clarity adds a substantial premium (roughly $4,800 from I1 to IF), and depth has a small positive effect. Surprisingly, once carat is accounted for, table and the x-dimension aren’t significant. With an R^2 of about 0.93, the model explains most of the variation in price.

SIMPLE LINEAR REGRESSION

1. Start with one predictor and one response from the variables in Part I. For instance, you can start with the predictor ’carat’ and the response ’price’, and conduct a simple linear regression analysis on it.

```r
model1 <- lm(price ~ carat, data = diamonds_sample)
```

2. Run the model and examine the summary statistics, interpreting everything (hypothesis testing, R^2_{adj} as discussed in class, confidence interval, prediction interval, plot, etc.).

```r
summary(model1)
```

```
##
## Call:
## lm(formula = price ~ carat, data = diamonds_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6479.0  -868.8   -20.5   620.2 11959.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2430.80      93.24  -26.07   <2e-16 ***
## carat        8023.92      99.86   80.35   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1488 on 998 degrees of freedom
## Multiple R-squared:  0.8661, Adjusted R-squared:  0.866
## F-statistic:  6457 on 1 and 998 DF,  p-value: < 2.2e-16
```

Residuals: five-number summary (Min, 1Q, Median, 3Q, Max) of the errors.

Coefficients:

- $\hat{\beta}_0$
  - Estimate: The price of the diamond is -2430.80 dollars when the diamond's carat is zero.
  - Std. Error (uncertainty): $\hat{\beta}_0$ is approximately 93.24 deviations away from its true value.
  - t value: $\hat{\beta}_0$ is -26.07 SE's away from zero.
  - Pr(>|t|) (p-value): The p-value from testing H_0 (null hypothesis): $\hat{\beta}_0$ is < 0.0001.
- $\hat{\beta}_1$
  - Estimate: The price of the diamond is 8023.92 dollars when the diamond's carat is zero.
  - Std. Error (uncertainty): $\hat{\beta}_1$ is approximately 99.86 deviations away from its true value.
  - t value: $\hat{\beta}_1$ is 80.35 SE's away from zero.
  - Pr(>|t|) (p-value): The p-value from testing H_0 (null hypothesis): $\hat{\beta}_1$ is < 0.0001.

Hypothesis Testing:

- Partial Significance Test:
  - $\hat{\beta}_0$: p $< 0.0001 < \alpha$ (significance level) $= 0.05 \Rightarrow$ we should not drop $\hat{\beta}_0$ from the model, it is statistically significant.
  - $\hat{\beta}_1$: p $< 0.0001 < \alpha$ (significance level) $= 0.05 \Rightarrow$ we should not drop $\hat{\beta}_1$ from the model, it is statistically significant.

$R^2_{adj} = 0.866$: 86.6% of variance in diamond price is explained by carat. There is only one predictor so R^2_{adj} does not punish our model.

```r
confint(model1)
```

```
##                 2.5 %    97.5 %
## (Intercept) -2613.765 -2247.838
## carat        7827.964  8219.880
```

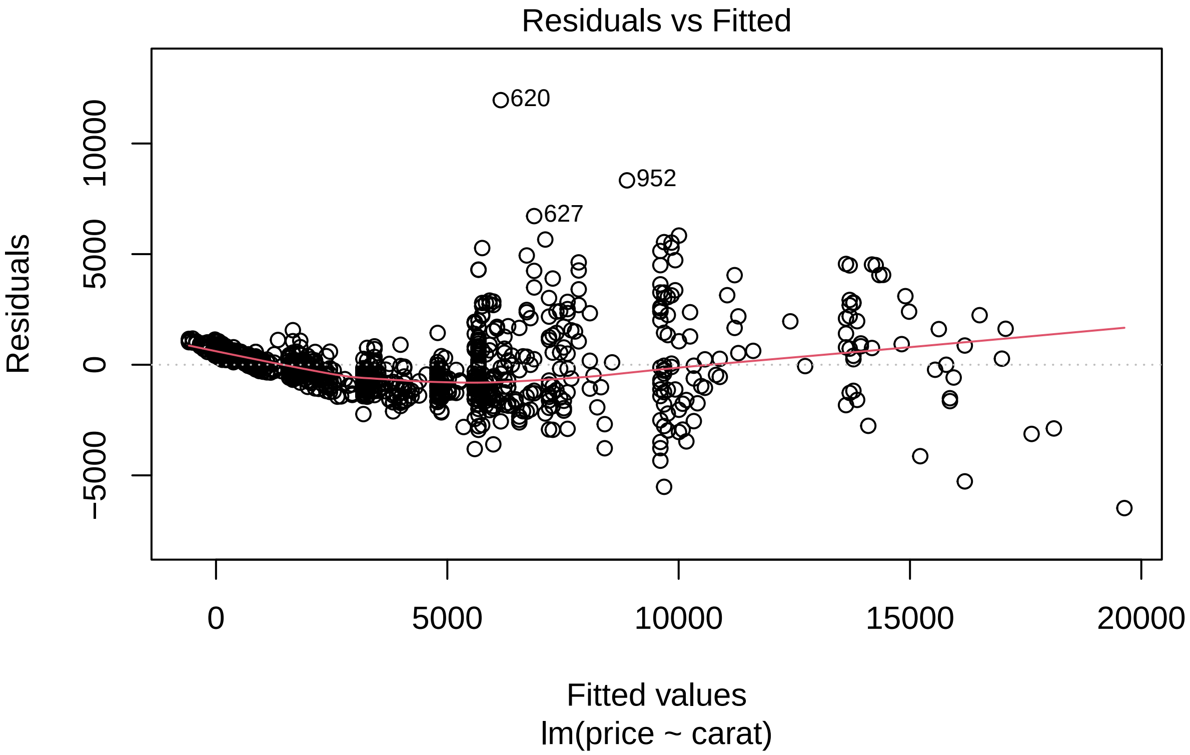
- $\hat{\beta}_0$
  - We are 95% confident prices will fall in this interval $[-2430.80 - t_{\alpha/2,998} * 93.24,\ -2430.80 + t_{\alpha/2,998} * 93.24] =$ [-2613.765, -2247.838] when carat = 0.
- $\hat{\beta}_1$
  - We are 95% confident that each additional carrat increases the average diamond price by some amount in this interval $[8023.92 - t_{\alpha/2,998} * 99.86,\ 8023.92 + t_{\alpha/2,998} * 99.86] =$ [7827.964, 8219.880].

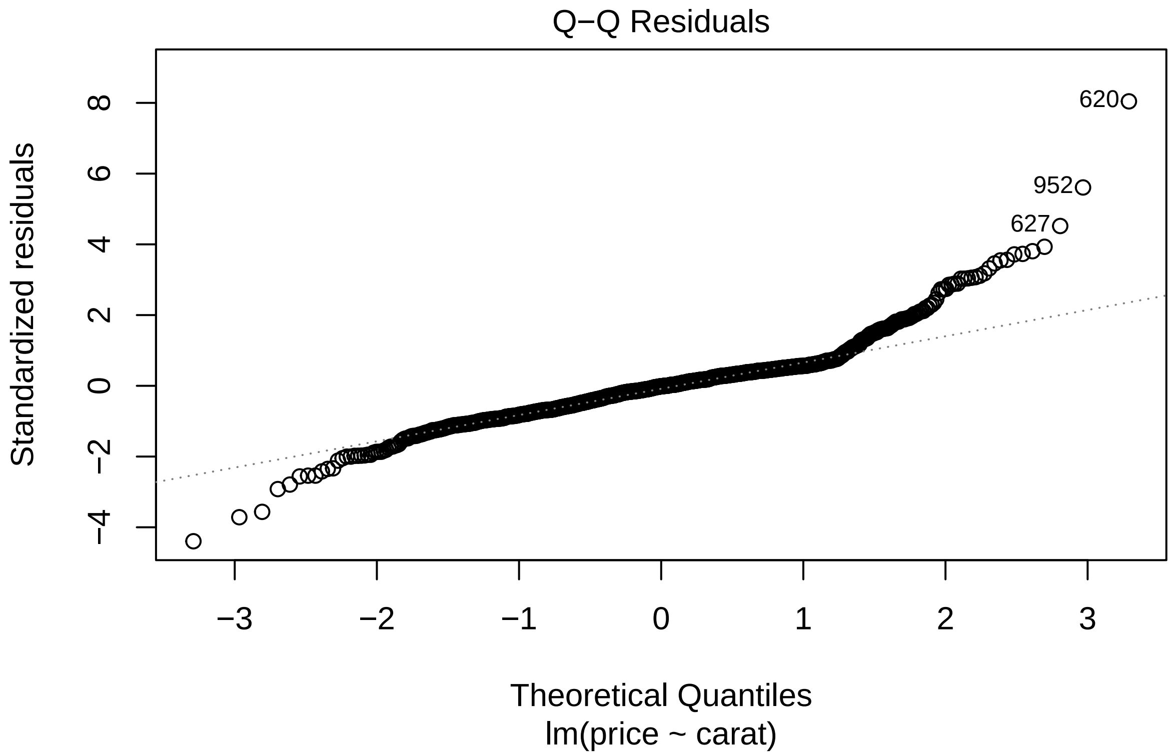
```r
# prediction interval of the sample's mean carat size
predict(model1,
        newdata=data.frame(carat=mean(diamonds_sample$carat)),
        interval="prediction",
        level=0.95)
```

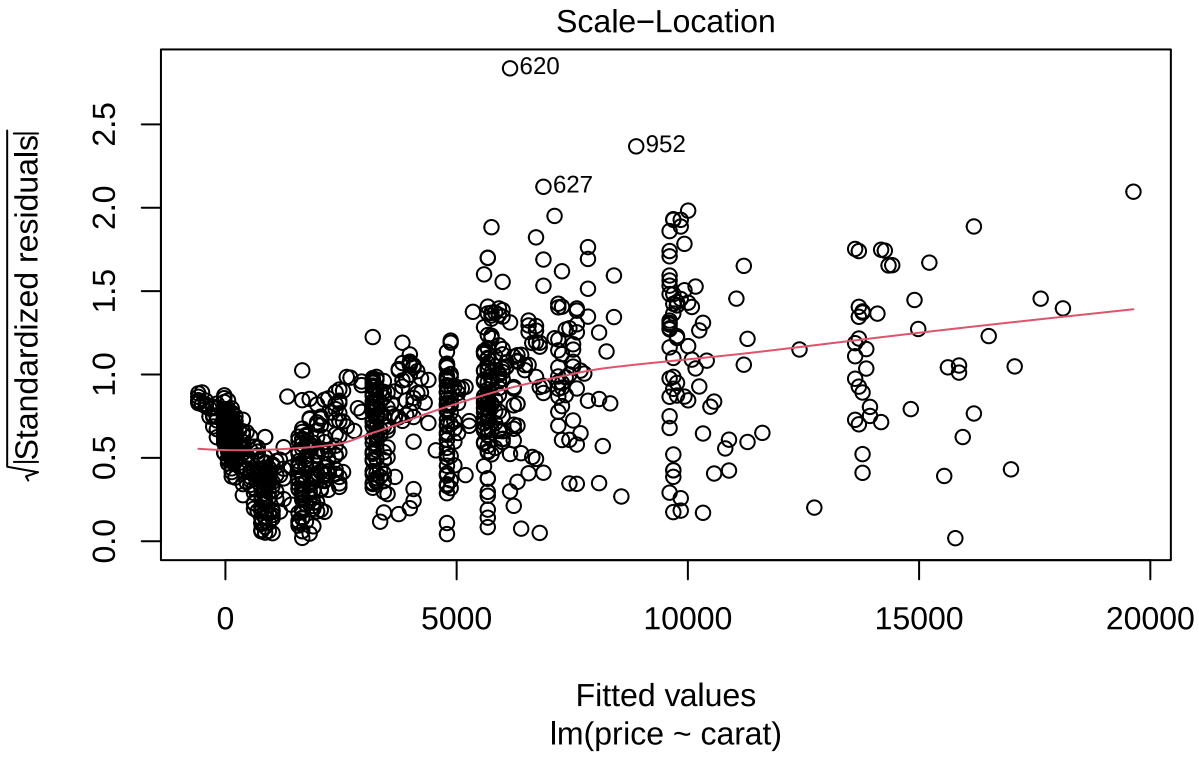
```
##        fit      lwr      upr
## 1 4037.282 1116.263 6958.301
```

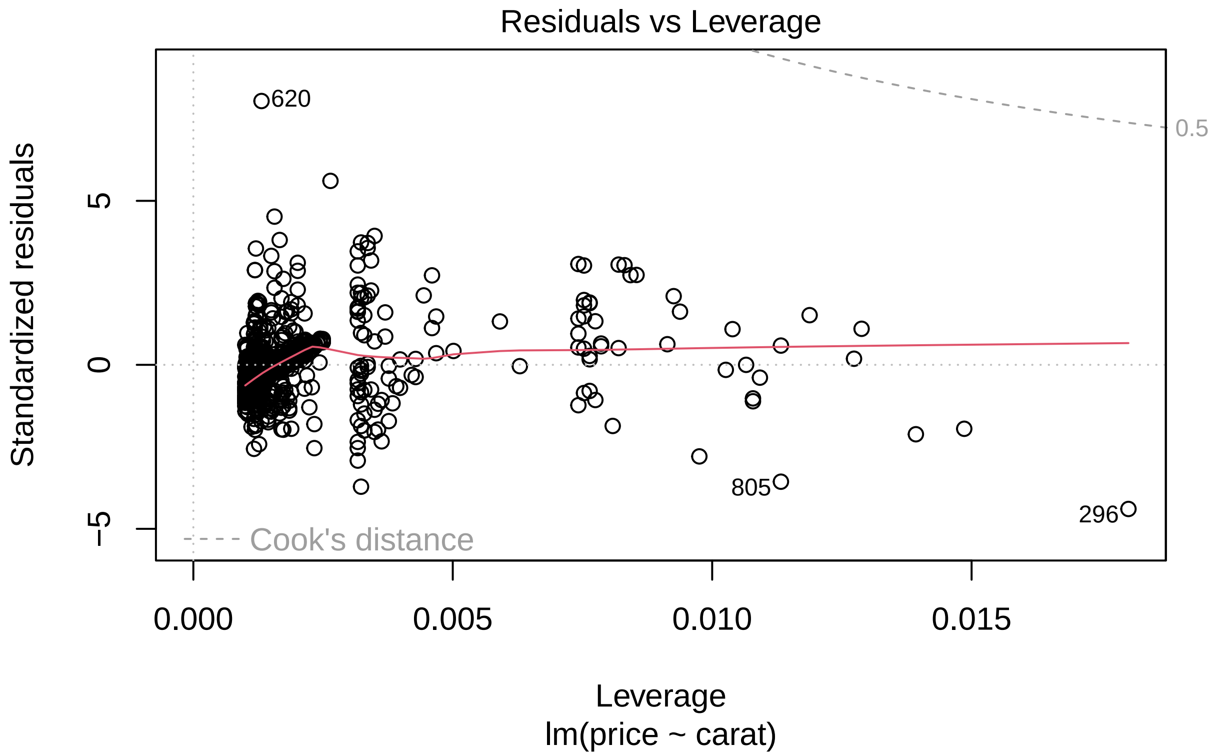
PI: The average price of a diamond falls in this interval [1111.847, 6964.689].

```
plot(model1)
```


Residuals vs Fitted
620
952
627
Residuals
10000
5000
0
−5000
0
5000
10000
15000
20000
Fitted values
lm(price ~ carat)


Q−Q Residuals
Standardized residuals
−4
−2
0
2
4
6
8
620
952
627
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(price ~ carat)


Scale−Location
620
952
627
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
0
5000
10000
15000
20000
Fitted values
lm(price ~ carat)

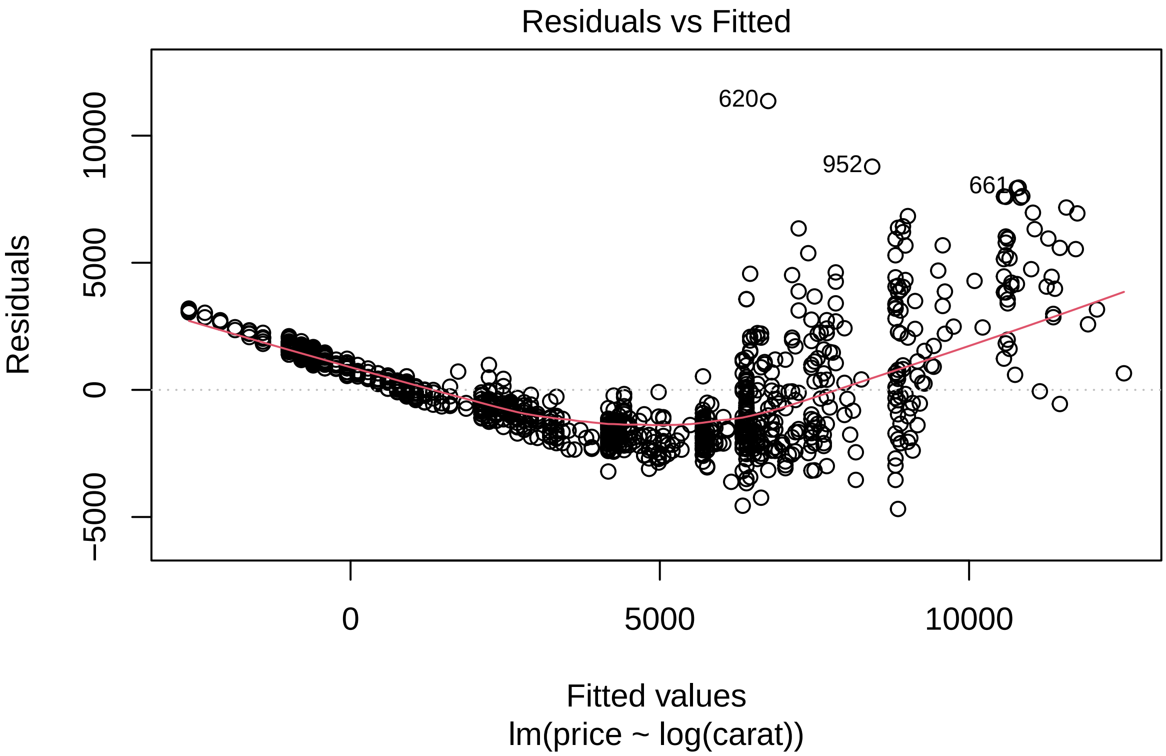


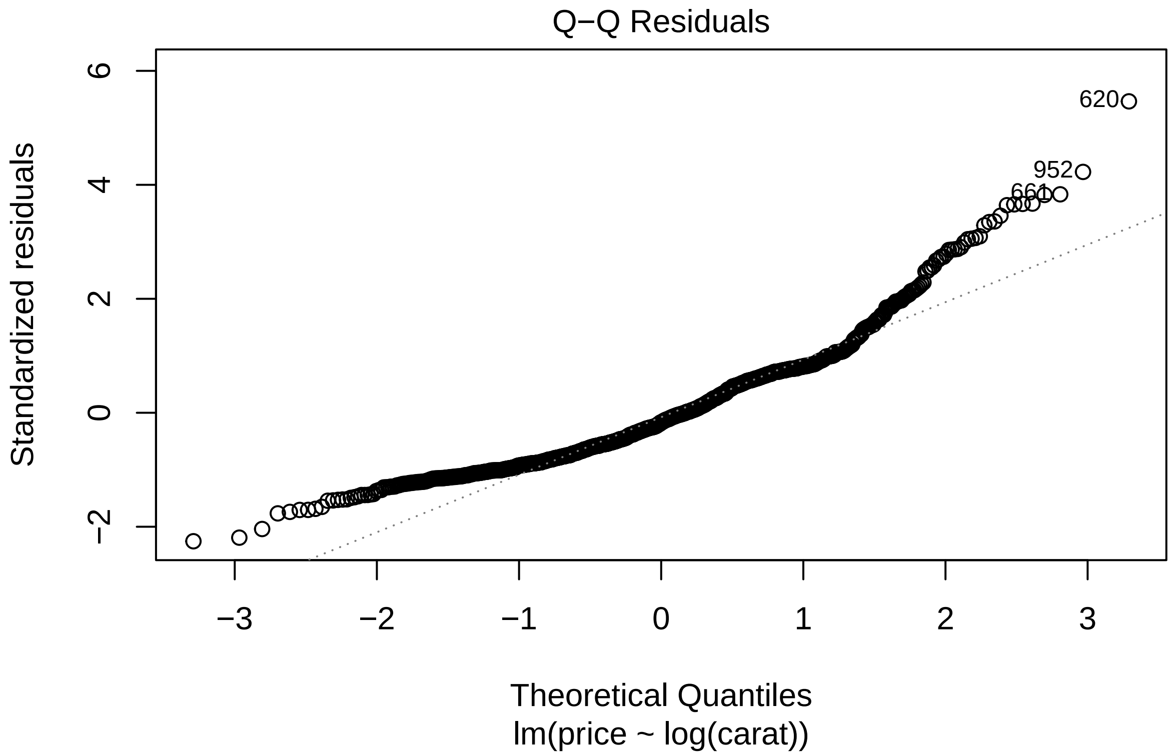
Normal Q–Q plot: some points stray from the reference line, suggesting a non-normal distribution.

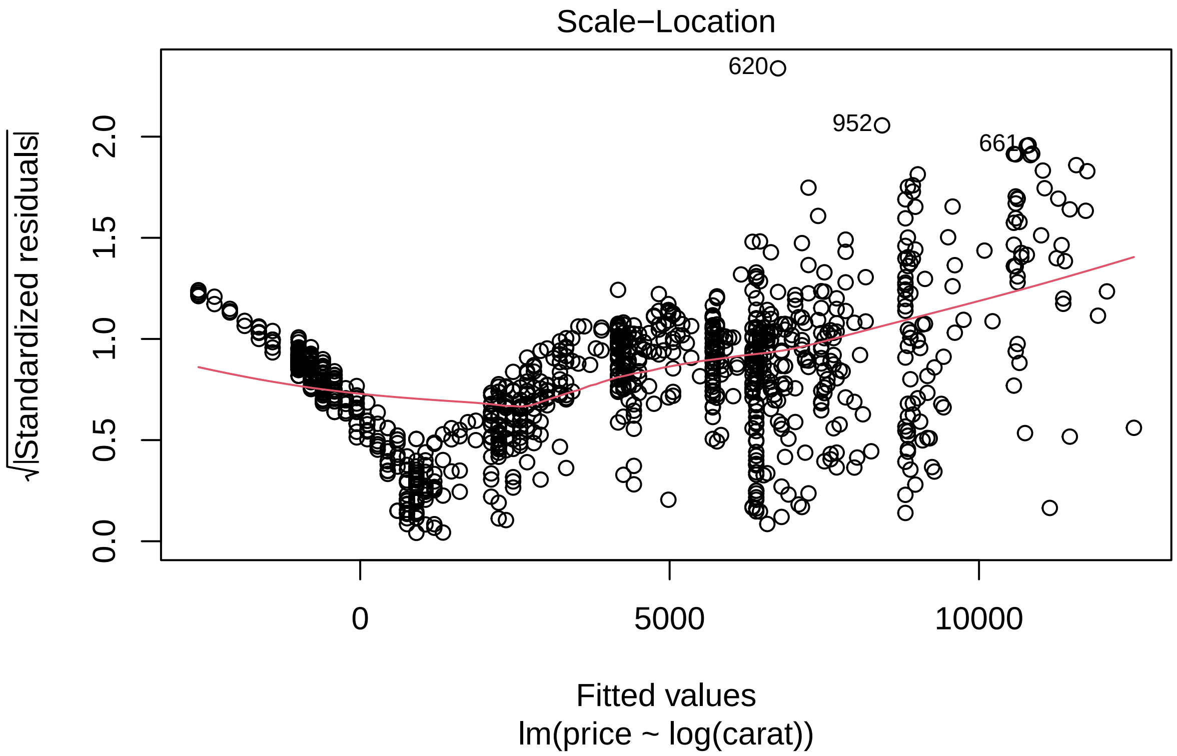
Residuals vs. fitted: there is an obvious funnel and spread does not look equal, suggesting non-constant variance and non-linearity.

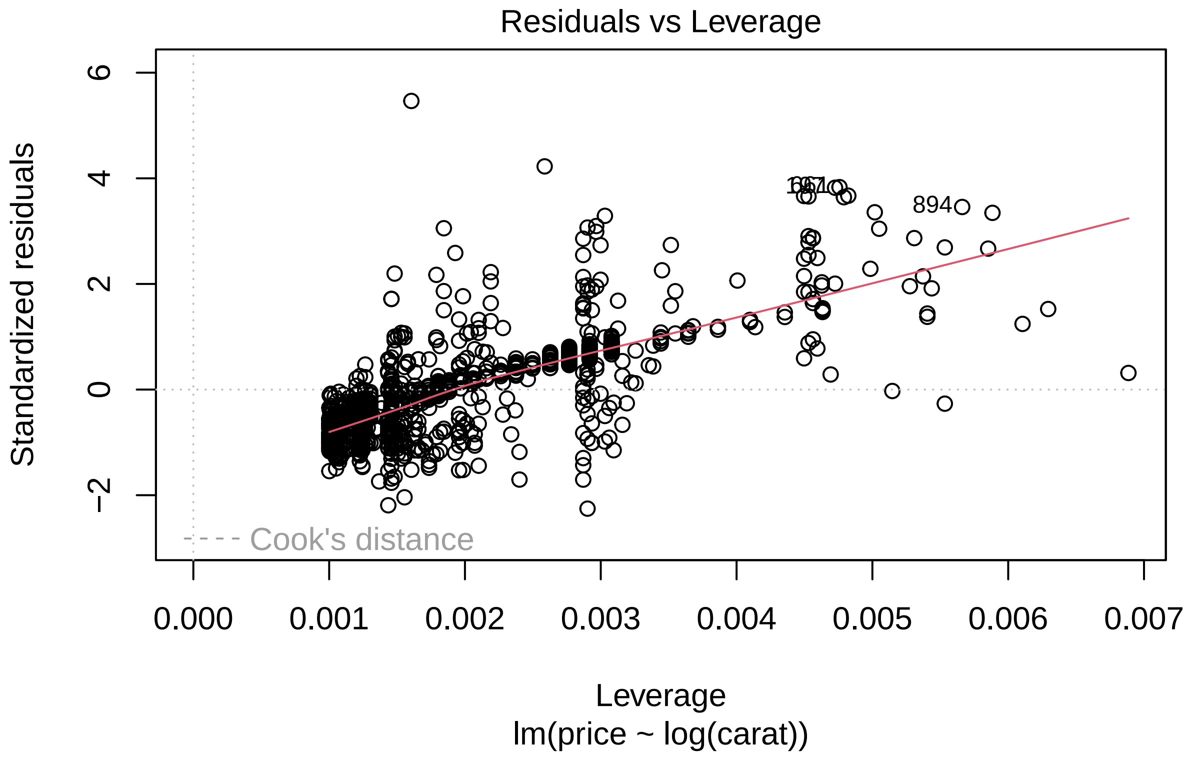
Therefore, we must transform the independent variable.

```r
model2 <- lm(price ~ log(carat), data = diamonds_sample)

plot(model2)
```


Residuals vs Fitted
Residuals
10000
5000
0
−5000
620
952
661
0
5000
10000
Fitted values
lm(price ~ log(carat))


Q−Q Residuals
Standardized residuals
6
4
2
0
−2
620
952
661
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(price ~ log(carat))


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
0
5000
10000
Fitted values
lm(price ~ log(carat))
620
952
661



```
summary(model2)
```

```
##
## Call:
## lm(formula = price ~ log(carat), data = diamonds_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4684.1 -1576.8  -358.5  1254.0 11361.6
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6340.13      78.83   80.42   <2e-16 ***
## log(carat)   6093.07     114.88   53.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2081 on 998 degrees of freedom
## Multiple R-squared:  0.7381, Adjusted R-squared:  0.7379
## F-statistic:  2813 on 1 and 998 DF,  p-value: < 2.2e-16
```

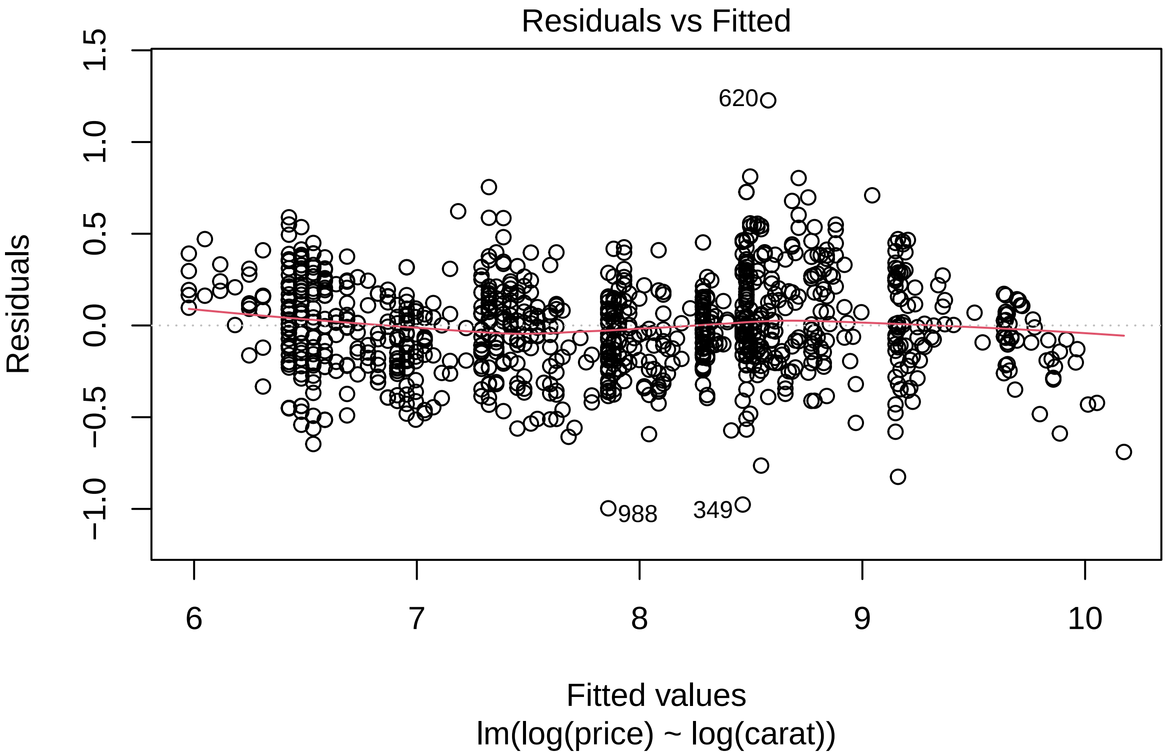
Normal Q–Q plot: some points stray from the reference line, suggesting a non-normal distribution.

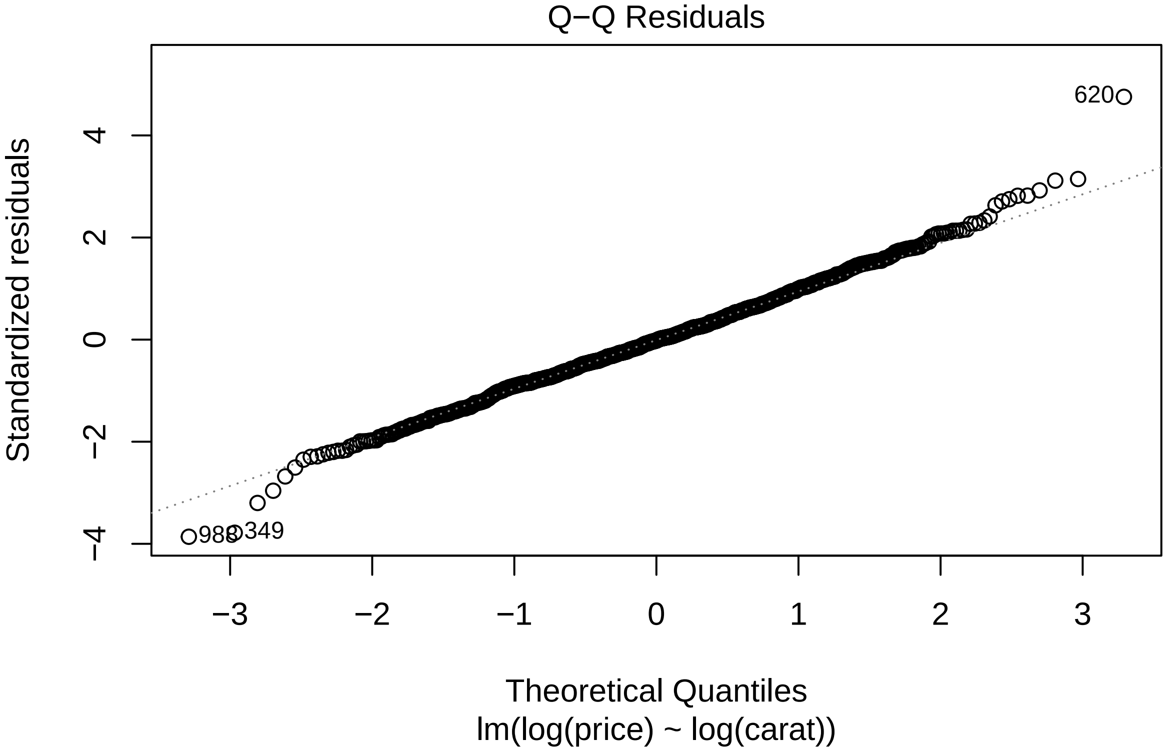
Residuals vs. fitted: again, there is an obvious funnel and spread does not look equal, suggesting non-constant variance and non-linearity.

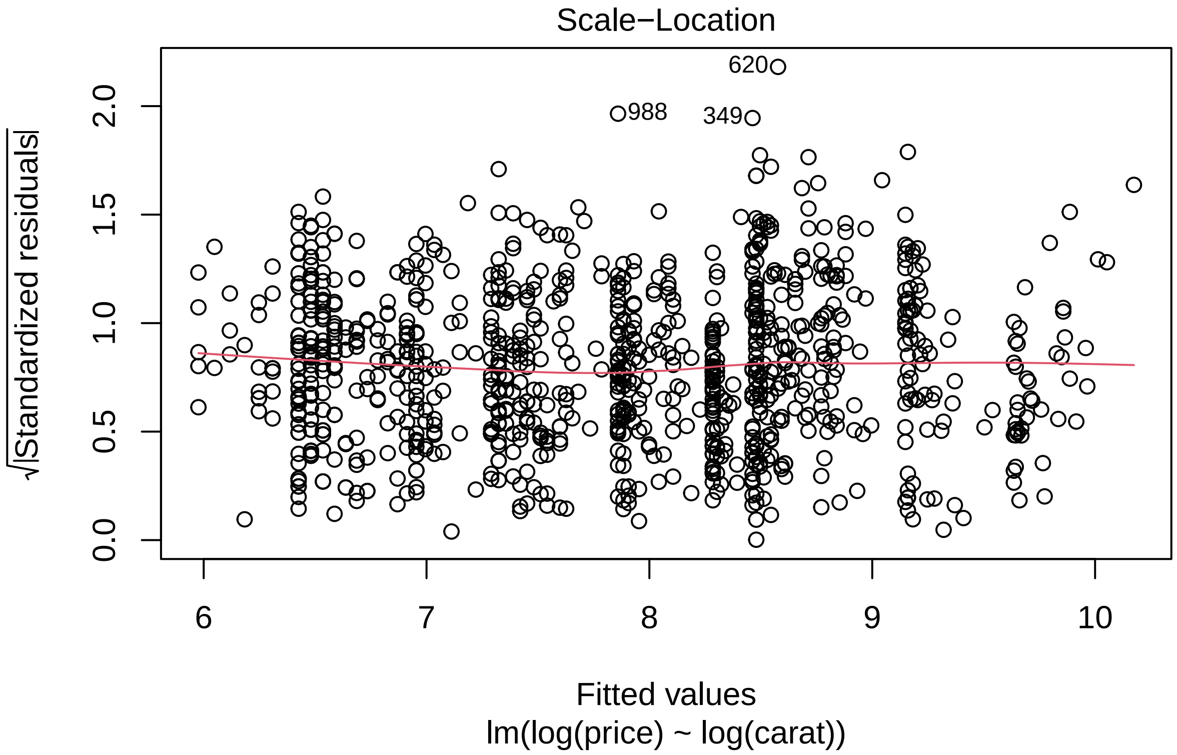
Summary: R^2 decreased which means our model weakened.

Thus, we must transform the dependent variable as well.

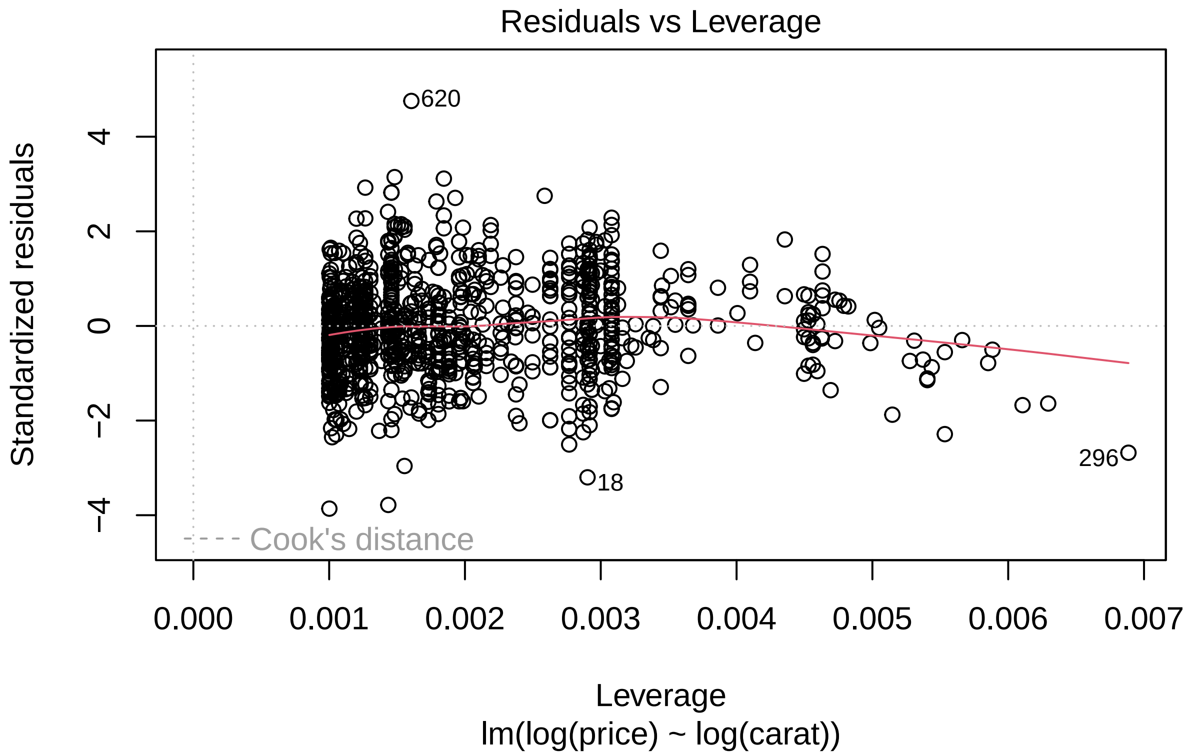
```
model3 <- lm(log(price) ~ log(carat), data = diamonds_sample)

plot(model3)
```




Q−Q Residuals
Standardized residuals
4
2
0
−2
−4
620
988
349
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(log(price) ~ log(carat))



Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
6
7
8
9
10
620
988
349
Fitted values
lm(log(price) ~ log(carat))



```
summary(model3)
```

```
##
## Call:
## lm(formula = log(price) ~ log(carat), data = diamonds_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.99667 -0.16829 -0.00472  0.16314  1.22711
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.462866   0.009784   865.0   <2e-16 ***
## log(carat)  1.691833   0.014258   118.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2582 on 998 degrees of freedom
## Multiple R-squared:  0.9338, Adjusted R-squared:  0.9337
## F-statistic: 1.408e+04 on 1 and 998 DF,  p-value: < 2.2e-16
```

Normal Q–Q plot: points stay close to the reference line, suggesting approximate normality.

Residuals vs. fitted: there are no obvious funnels and spread looks equal, suggesting constant variance and linearity.

Summary: R^2 increased which means our model strengthened.

In conclusion, our transformed model follows all assumptioms.

Add other variables to the model and assess if the model improves. For step 5, run the code in the background and include all interpretations in the file. For instance, if adding depth to the simple linear regression model (carat and price) increases the adjusted R^2, include it in the model; if it decreases, exclude it. Do not include the code for step 5 in the submitted file; only write the conclusions.

After comparing many versions of the model, we noticed adding predictors increased the adjusted R^2 (improved the model's fit). The biggest improvements came from including cut, clarity, and color. Other variables such as depth, table, x, y, and z contributed slightly to the model. In general, adding predictors strengthened the model's fit

6. Comment on anything of interest that occurred while doing this part.

To be completely honest, I had no idea how to do the transformation portion. I had to contact a friend for help, but I was very impressed it fixed the problems with the assumptions. I was also intrigued by how all the predictors added something helpful to the model. Prior to the analysis, I thought some of the predictors would be useless.

PART II continuation...

1. In class we saw different techniques and criterion to find best model. You can use any method and technique you prefer (e.g., backward elimination using AIC or stepwise regression using AIC or backward elimination using BIC criterion) to find the best model and document your observations.

```r
model4 <- lm(log(price) ~ log(carat) + cut + color + clarity + depth + table + x + y + z,
             data = diamonds_sample)

summary(model4)
```

```
##
## Call:
## lm(formula = log(price) ~ log(carat) + cut + color + clarity +
##     depth + table + x + y + z, data = diamonds_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52672 -0.08442 -0.00190  0.08124  0.55612
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.000900   1.154888   5.196 2.48e-07 ***
## log(carat)    1.723859   0.072720  23.705  < 2e-16 ***
## cutGood       0.047250   0.026679   1.771   0.0769 .
## cutIdeal      0.137352   0.025919   5.299 1.44e-07 ***
## cutPremium    0.121302   0.024995   4.853 1.41e-06 ***
## cutVery Good  0.101883   0.025344   4.020 6.27e-05 ***
## colorE       -0.038837   0.015254  -2.546   0.0110 *
## colorF       -0.072237   0.015316  -4.716 2.75e-06 ***
## colorG       -0.157407   0.014753 -10.670  < 2e-16 ***
## colorH       -0.230315   0.016055 -14.346  < 2e-16 ***
## colorI       -0.359501   0.017577 -20.453  < 2e-16 ***
## colorJ       -0.500217   0.021823 -22.922  < 2e-16 ***
## clarityIF     1.118186   0.052765  21.192  < 2e-16 ***
## claritySI1    0.556933   0.045992  12.109  < 2e-16 ***
## claritySI2    0.404334   0.046152   8.761  < 2e-16 ***
## clarityVS1    0.774623   0.046642  16.608  < 2e-16 ***
## clarityVS2    0.707105   0.046026  15.363  < 2e-16 ***
## clarityVVS1   0.989568   0.048058  20.591  < 2e-16 ***
## clarityVVS2   0.890967   0.047691  18.682  < 2e-16 ***
## depth         0.024097   0.017779   1.355   0.1756
## table        -0.002493   0.002402  -1.038   0.2996
## x             0.183775   0.129481   1.419   0.1561
## y             0.126002   0.129449   0.973   0.3306
## z            -0.362298   0.290422  -1.247   0.2125
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.1304 on 976 degrees of freedom
## Multiple R-squared:  0.9835, Adjusted R-squared:  0.9831
## F-statistic:  2530 on 23 and 976 DF,  p-value: < 2.2e-16
```

```
model_step_AIC <- step(model4, direction="backward")
```

```
## Start:  AIC=-4051.17
## log(price) ~ log(carat) + cut + color + clarity + depth + table +
##     x + y + z
##
##              Df Sum of Sq    RSS     AIC
## - y           1    0.0161 16.603 -4052.2
## - table       1    0.0183 16.605 -4052.1
## - z           1    0.0264 16.613 -4051.6
## - depth       1    0.0312 16.618 -4051.3
## <none>                    16.587 -4051.2
## - x           1    0.0342 16.621 -4051.1
## - cut         4    0.7885 17.375 -4012.7
## - log(carat)  1    9.5498 26.136 -3598.4
## - color       6   15.9586 32.545 -3389.1
## - clarity     7   30.2710 46.858 -3026.6
##
## Step:  AIC=-4052.2
## log(price) ~ log(carat) + cut + color + clarity + depth + table +
##     x + z
##
##              Df Sum of Sq    RSS     AIC
## - z           1    0.0109 16.613 -4053.5
## - depth       1    0.0152 16.618 -4053.3
## - table       1    0.0213 16.624 -4052.9
## <none>                    16.603 -4052.2
## - x           1    0.0367 16.639 -4052.0
## - cut         4    0.7730 17.376 -4014.7
## - log(carat)  1    9.6175 26.220 -3597.2
## - color       6   15.9525 32.555 -3390.8
## - clarity     7   30.4387 47.041 -3024.7
##
## Step:  AIC=-4053.54
## log(price) ~ log(carat) + cut + color + clarity + depth + table +
##     x
##
##              Df Sum of Sq    RSS     AIC
## - depth       1    0.0061 16.620 -4055.2
## - table       1    0.0202 16.634 -4054.3
## <none>                    16.613 -4053.5
## - x           1    0.1034 16.717 -4049.3
## - cut         4    0.7693 17.383 -4016.3
## - log(carat)  1    9.9341 26.548 -3586.8
## - color       6   16.0799 32.693 -3388.6
## - clarity     7   30.5130 47.126 -3024.9
##
## Step:  AIC=-4055.17
## log(price) ~ log(carat) + cut + color + clarity + table + x
##
##              Df Sum of Sq    RSS     AIC
## <none>                    16.620 -4055.2
## - table       1    0.0404 16.660 -4054.7
## - x           1    0.1031 16.723 -4051.0
## - cut         4    0.8145 17.434 -4015.3
## - log(carat)  1   12.6888 29.308 -3489.9
## - color       6   16.3081 32.928 -3383.4
## - clarity     7   30.5116 47.131 -3026.8
```

```
model_step_AIC
```

```
##
## Call:
## lm(formula = log(price) ~ log(carat) + cut + color + clarity +
##     table + x, data = diamonds_sample)
##
## Coefficients:
##  (Intercept)    log(carat)       cutGood      cutIdeal    cutPremium
##     7.572254      1.734108      0.048286      0.132907      0.117351
## cutVery Good        colorE        colorF        colorG        colorH
##     0.099975     -0.040377     -0.073462     -0.157893     -0.230390
##       colorI        colorJ     clarityIF    claritySI1    claritySI2
##    -0.358819     -0.502147      1.117081      0.556554      0.404051
##   clarityVS1    clarityVS2   clarityVVS1   clarityVVS2         table
##     0.774388      0.706665      0.988436      0.891367     -0.003269
##            x
##     0.080797
```

```
summary(model_step_AIC)
```

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

```
## lm(formula = log(price) ~ log(carat) + cut + color + clarity +
##     table + x, data = diamonds_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51880 -0.08469 -0.00215  0.08128  0.55659
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.572254   0.241346  31.375  < 2e-16 ***
## log(carat)    1.734108   0.063429  27.340  < 2e-16 ***
## cutGood       0.048286   0.026038   1.854  0.06397 .
## cutIdeal      0.132907   0.024547   5.414 7.74e-08 ***
## cutPremium    0.117351   0.024257   4.838 1.52e-06 ***
## cutVery Good  0.099975   0.024129   4.143 3.72e-05 ***
## colorE       -0.040377   0.015204  -2.656  0.00804 **
## colorF       -0.073462   0.015266  -4.812 1.73e-06 ***
## colorG       -0.157893   0.014734 -10.716  < 2e-16 ***
## colorH       -0.230390   0.016004 -14.396  < 2e-16 ***
## colorI       -0.358819   0.017535 -20.463  < 2e-16 ***
## colorJ       -0.502147   0.021526 -23.327  < 2e-16 ***
## clarityIF     1.117081   0.052286  21.365  < 2e-16 ***
## claritySI1    0.556554   0.045679  12.184  < 2e-16 ***
## claritySI2    0.404051   0.045772   8.827  < 2e-16 ***
## clarityVS1    0.774388   0.046215  16.756  < 2e-16 ***
## clarityVS2    0.706665   0.045708  15.461  < 2e-16 ***
## clarityVVS1   0.988436   0.047637  20.749  < 2e-16 ***
## clarityVVS2   0.891367   0.047319  18.838  < 2e-16 ***
## table        -0.003269   0.002120  -1.542  0.12328
## x             0.080797   0.032793   2.464  0.01392 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.1303 on 979 degrees of freedom
## Multiple R-squared:  0.9835, Adjusted R-squared:  0.9831
## F-statistic:  2913 on 20 and 979 DF,  p-value: < 2.2e-16
```

Due to our model’s small size, we chose Backward AIC. After running the model, we got an AIC value of -4055.17. Based on the Backward AIC analysis, the variables depth, y, and z were removed, meaning that they were not significant to the model. Looking at the summary post Backward AIC analysis, table is not statistically significant to the model so we removed it.

2. Detect multicollinearity among the variables using the variance inflation factor (VIF).

```r
library(car)

model5 <- lm(log(price) ~ log(carat) + cut + color + clarity + x,
            data = diamonds_sample)

summary(model5)
```

```
##
```

```
## Call:
## lm(formula = log(price) ~ log(carat) + cut + color + clarity +
##     x, data = diamonds_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52161 -0.08484 -0.00158  0.08019  0.55659
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.40550    0.21593  34.296  < 2e-16 ***
## log(carat)    1.74016    0.06335  27.468  < 2e-16 ***
## cutGood       0.05192    0.02595   2.001   0.0457 *
## cutIdeal      0.14461    0.02336   6.190 8.83e-10 ***
## cutPremium    0.12063    0.02418   4.989 7.19e-07 ***
## cutVery Good  0.10541    0.02389   4.413 1.13e-05 ***
## colorE       -0.04097    0.01521  -2.693   0.0072 **
## colorF       -0.07472    0.01525  -4.898 1.13e-06 ***
## colorG       -0.15823    0.01474 -10.733  < 2e-16 ***
## colorH       -0.23120    0.01601 -14.444  < 2e-16 ***
## colorI       -0.36009    0.01753 -20.544  < 2e-16 ***
## colorJ       -0.50238    0.02154 -23.322  < 2e-16 ***
## clarityIF     1.11353    0.05227  21.303  < 2e-16 ***
## claritySI1    0.55342    0.04567  12.119  < 2e-16 ***
## claritySI2    0.40088    0.04576   8.761  < 2e-16 ***
## clarityVS1    0.77097    0.04619  16.690  < 2e-16 ***
## clarityVS2    0.70286    0.04567  15.389  < 2e-16 ***
## clarityVVS1   0.98605    0.04765  20.695  < 2e-16 ***
## clarityVVS2   0.88919    0.04733  18.787  < 2e-16 ***
## x             0.07697    0.03272   2.352   0.0189 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.1304 on 980 degrees of freedom
## Multiple R-squared:  0.9834, Adjusted R-squared:  0.9831
## F-statistic:  3062 on 19 and 980 DF,  p-value: < 2.2e-16
```

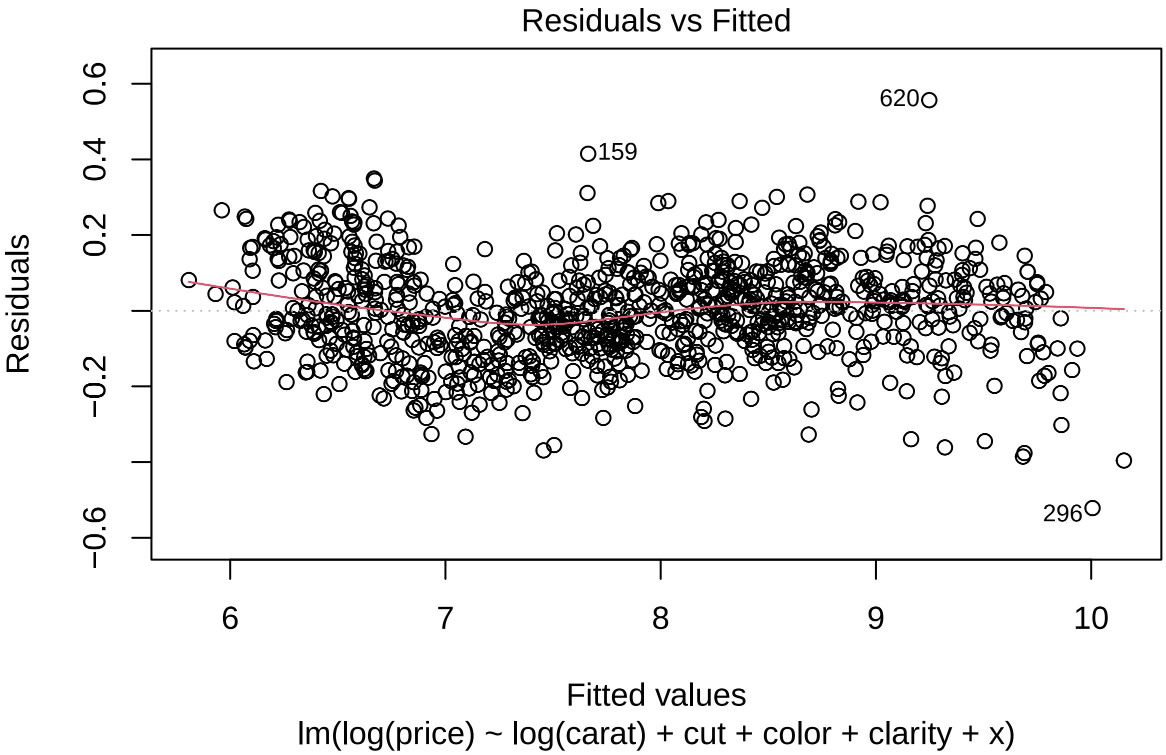
All predictors are statistically significant to the model.

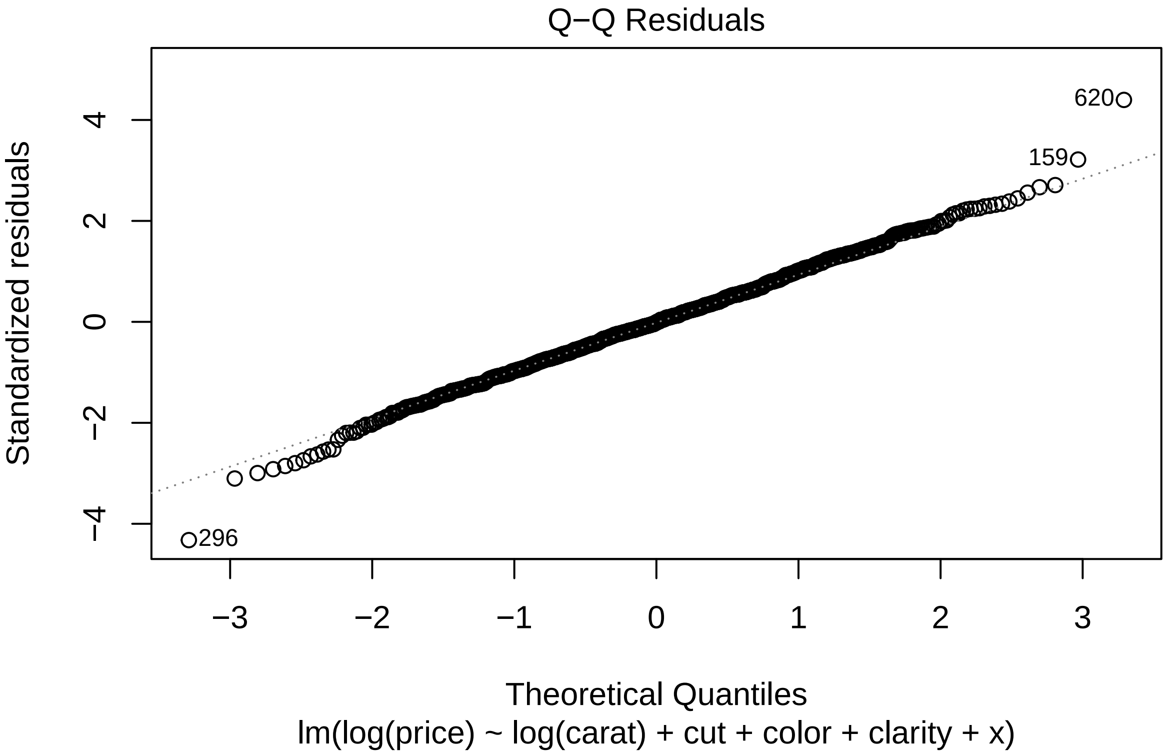
```
vif <- vif(model5)
vif
```

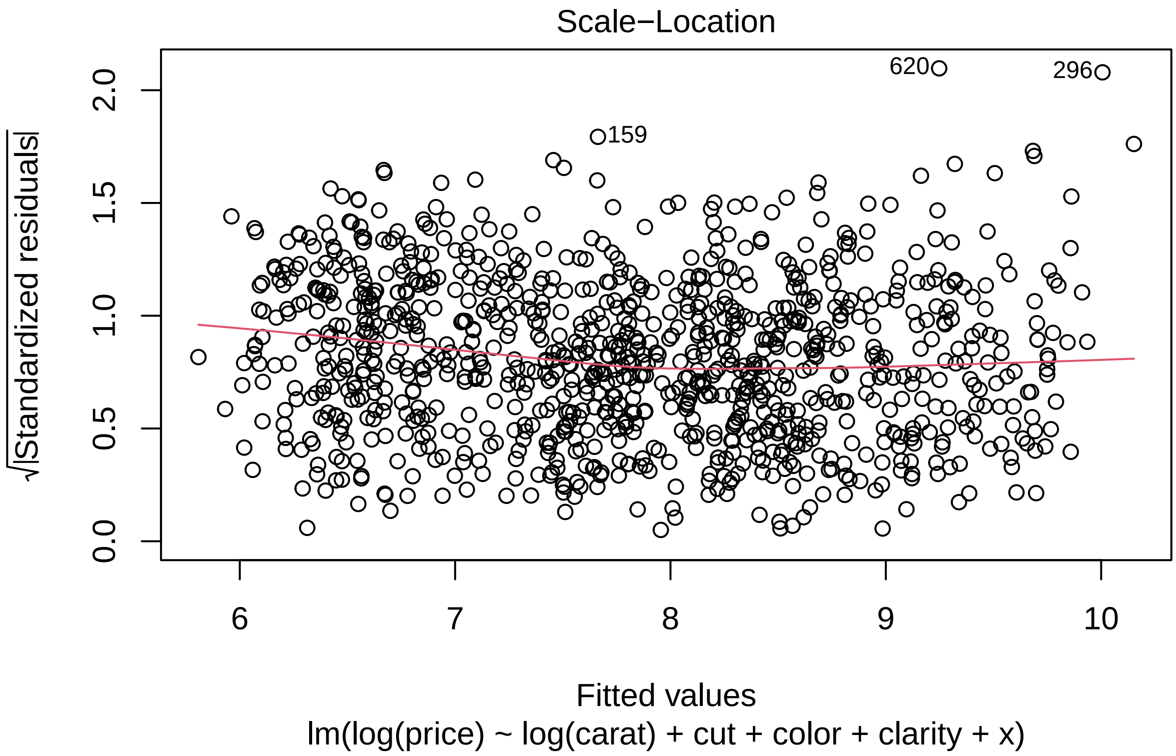
```
##                 GVIF Df GVIF^(1/(2*Df))
## log(carat) 77.452382  1        8.800704
## cut         1.250034  4        1.028289
## color       1.256651  6        1.019220
## clarity     1.446957  7        1.026741
## x          77.092020  1        8.780206
```

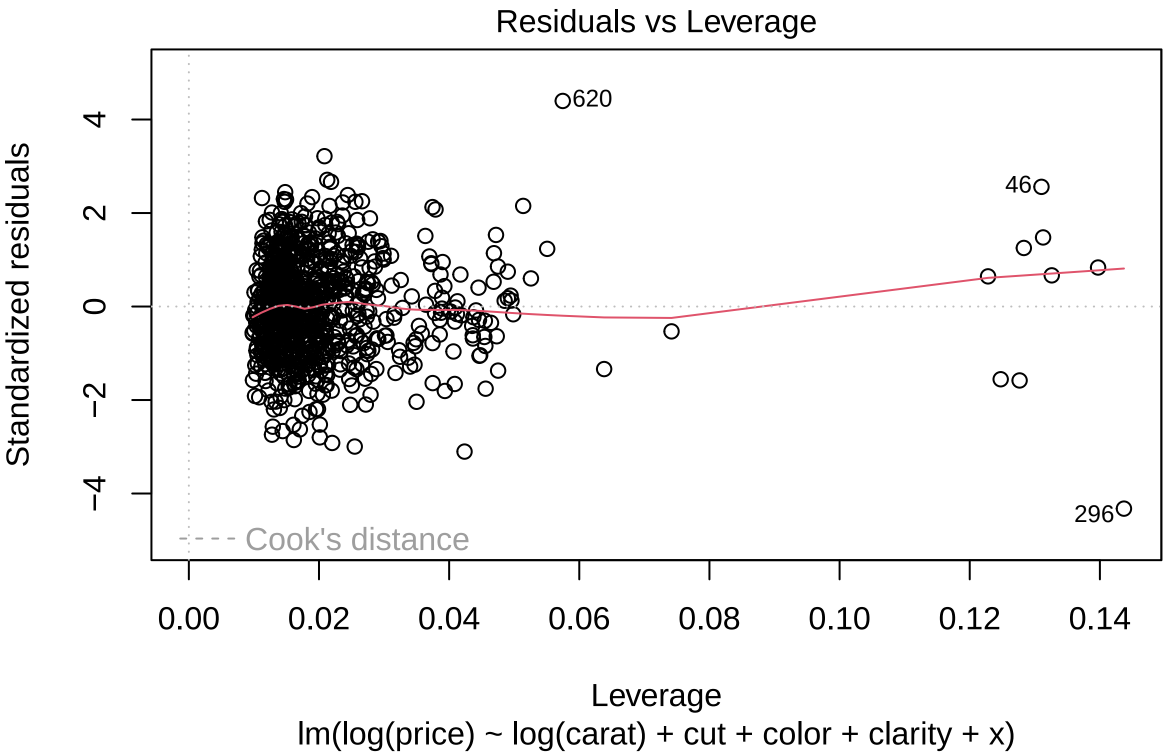
The VIF shows non-signficant multicollinearity among the variables, confirming that all predictors are significant. We will now check that the model assumptions are valid.

```r
plot(model5)
```




Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(log(price) ~ log(carat) + cut + color + clarity + x)
620
159
296
−4
−2
0
2
4
−3
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
620
296
159
6
7
8
9
10
Fitted values
lm(log(price) ~ log(carat) + cut + color + clarity + x)



Normal Q–Q plot: points stay close to the reference line, suggesting approximate normality.

Residuals vs. fitted: there are no obvious funnels and spread looks equal, suggesting constant variance and linearity.

3. Give CIs for a mean predicted value and the PIs of a future predicted value for at least one combination of X's (from your final linear model).

```
confint(model5, level=.95)
```

```
##                       2.5 %      97.5 %
## (Intercept)   6.9817612898  7.82923707
## log(carat)    1.6158401454  1.86448104
## cutGood       0.0009967574  0.10284159
## cutIdeal      0.0987662303  0.19045514
## cutPremium    0.0731767908  0.16808011
## cutVery Good  0.0585352946  0.15228545
## colorE       -0.0708131678 -0.01111713
## colorF       -0.1046530677 -0.04478249
## colorG       -0.1871608989 -0.12929828
## colorH       -0.2626134503 -0.19979007
## colorI       -0.3944856200 -0.32569300
## colorJ       -0.5446480976 -0.46010524
## clarityIF     1.0109475219  1.21610305
## claritySI1    0.4638043794  0.64303412
## claritySI2    0.3110867599  0.49067673
## clarityVS1    0.6803184558  0.86162084
## clarityVS2    0.6132338942  0.79249067
## clarityVVS1   0.8925515744  1.07955071
## clarityVVS2   0.7963043717  0.98206690
## x             0.0127538684  0.14118048
```

For each predictor, we are 95% sure the true predictor value lies between the 2.5% value and 97.5% value. For example, we are 95% sure the true population parameter log(carat) lies in [1.6158401454, 1.86448104].

```
predict_df <- data.frame(carat = mean(diamonds_sample$carat),
                         cut     = factor("Very Good", levels = c("Fair", "Good", "Ideal", "Premium", "
                         color   = factor("J", levels = c("D", "E", "F", "G", "H", "I", "J")),
                         clarity = factor("VVS2", levels = c("I1", "SI2", "SI1", "VS2", "VS1", "VVS2",
                         x = mean(diamonds_sample$x))

predict <- predict(model5, predict_df, level = 0.95, interval = "prediction")
exp(predict)
```

```
##        fit      lwr      upr
## 1 2881.732 2219.629 3741.337
```

We are 95% confident that a diamond with carat = 0.80605, cut = Very Good, color = J, clarity = VVS2 will, and x = 5.76242 will cost between $2219.63 and $3741.34.

4. Summarize your report (for the final deliverable).

The analysis showed that diamond price is overwhelmingly driven by size, with carat weight exhibiting a strong power-law relationship to price (log–log $R^2 \approx 0.93$), while quality grades (cut, color, clarity) contribute significant premiums. Exploratory histograms revealed right-skewed distributions for carat and price, and correlation analysis confirmed tight links among size measures (r > 0.97) and between size and price (r ≈ 0.9). A log-transformed regression model with carat, one physical dimension, and categorical quality predictors achieved an adjusted R^2 of 0.983 with minimal multicollinearity. Finally, inference on a typical diamond yielded a 95% prediction interval of $2,220 – $3,740.