MA334: Project Report

Nallapu Naveen

2026-01-09

1. Data Exploration

(a) Summary of the Data Set

```
library(knitr)
# Load the data
data <- read.csv("MA334-AU-7_2501629.csv")
data_pre <- head(data, 2)
kable(data_pre, caption = "First 2 Observations")
```

Table 1: First 2 Observations

carat	cut	color	clarity	depth	table	price	x	y	z
0.53	Premium	D	SI1	61.4	58	1566	5.19	5.23	3.20
0.85	Ideal	G	SI1	62.2	56	3403	6.06	6.09	3.78

```
# Number of observations and variables
kable(data.frame(
  Item = c("Total Observations", "Total Variables"),
  Value = c(nrow(data), ncol(data))
), caption = "Dataset Dimensions")
```

Table 2: Dataset Dimensions

Item	Value
Total Observations	1000
Total Variables	10

(Dataset Dimensions)

Table 2 shows that the dataset contains 1,000 observations and 10 variables.

```
# Variable names and types
kable(data.frame(Variable = names(data), Type = unname(sapply(data, class))),
      caption = "Variable Names and Data Types")
```

Table 3: Variable Names and Data Types

Variable	Type
carat	numeric
cut	character
color	character
clarity	character
depth	numeric
table	numeric
price	integer
x	numeric
y	numeric
z	numeric

(Variable Types)

Table 3 shows all the variables in the dataset and their data types. Some variables contain numbers (such as carat, depth, price,table and x, y, z), and others contain categories (cut, color, and clarity). —

```
# Finding categorical variables
categorical_variables <- names(data)[sapply(data, is.character)]

# Converting them to factors
data[categorical_variables] <- lapply(data[categorical_variables], as.factor)
# Creating categories table
categories_table <- data.frame(
  Variable = categorical_variables,
  Categories = sapply(categorical_variables, function(v)
    paste(levels(data[[v]]), collapse = ", ")
  )
)
kable(categories_table, caption = "Categories for Qualitative Variables")
```

Table 4: Categories for Qualitative Variables

	Variable	Categories
cut	cut	Fair, Good, Ideal, Premium, Very Good
color	color	D, E, F, G, H, I, J
clarity	clarity	I1, IF, SI1, SI2, VS1, VS2, VVS1, VVS2

(Categories)

Table 4 shows the categories for each qualitative variable and variable **cut** has 5 categories, **color** has 7 categories, and **clarity** has 8 categories.
These categories represent different levels of diamond quality.

(b) Location and Spread of Numeric Variables

```r
# Select numeric variables only
numeric_data <- data[sapply(data, is.numeric)]
# Create a table of summary statistics
summary_table <- data.frame(
  Variable = names(numeric_data),
  Mean = sapply(numeric_data, mean),
  Median = sapply(numeric_data, median),
  Standard_Deviation = sapply(numeric_data, sd),
  Range = sapply(numeric_data, function(x) max(x) - min(x))
)
kable(summary_table, caption = "Summary Statistics for Numeric Variables")
```

Table 5: Summary Statistics for Numeric Variables

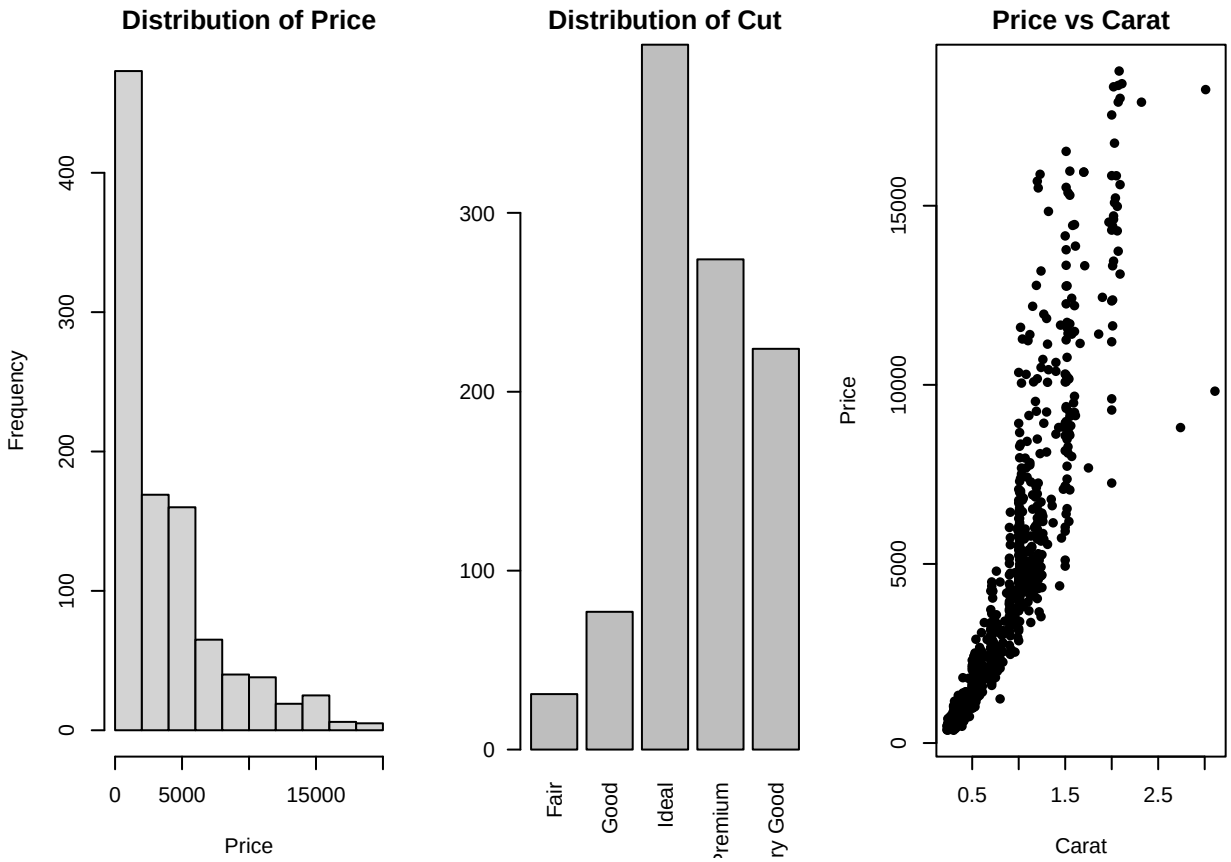
	Variable	Mean	Median	Standard_Deviation	Range
carat	carat	0.78173	0.70	0.4603453	2.88
depth	depth	61.75350	61.90	1.3726660	11.80
table	table	57.44450	57.00	2.1992604	24.00
price	price	3786.08000	2278.50	3833.2003865	18403.00
x	x	5.69129	5.64	1.1116064	9.15
y	y	5.69468	5.64	1.1025724	9.02
z	z	3.51608	3.46	0.6894836	5.98

This table shows typical values (mean and median) and how spread out the values are (standard deviation and range) for each numeric variable.

(c) Visualisations of Variable Distributions

```r
par(mfrow = c(1, 3), mar = c(4, 4, 2, 1))

hist(data$price, main = "Distribution of Price", xlab = "Price")
barplot(table(data$cut), main = "Distribution of Cut", las = 2)
plot(data$carat, data$price,
     main = "Price vs Carat",
     xlab = "Carat", ylab = "Price",
     pch = 16)
```




```
par(mfrow = c(1, 1))
```

(d) Produce and interpret a correlation matrix between the numeric variables

Selecting numeric variables only

```
numeric_data <- data[sapply(data, is.numeric)]
cor_matrix <- cor(numeric_data)
knitr::kable(round(cor_matrix, 2),
             caption = "Correlation Matrix of Numeric Variables")
```

Table 6: Correlation Matrix of Numeric Variables

	carat	depth	table	price	x	y	z
carat	1.00	0.05	0.21	0.91	0.96	0.96	0.96
depth	0.05	1.00	-0.31	0.00	-0.03	-0.03	0.09
table	0.21	-0.31	1.00	0.15	0.23	0.22	0.18
price	0.91	0.00	0.15	1.00	0.85	0.85	0.84
x	0.96	-0.03	0.23	0.85	1.00	1.00	0.99
y	0.96	-0.03	0.22	0.85	1.00	1.00	0.99
z	0.96	0.09	0.18	0.84	0.99	0.99	1.00

==========

Question 2: Probability, distributions, and confidence intervals

==========

(a) A diamond is chosen at random

(i) Probability price exceeds $10,000

```
probability_price_get_10000 <- mean(data$price > 10000)
probability_price_get_10000
```

```
## [1] 0.093
```

Interpretation: About 9.3% of the diamonds in this dataset are priced above $10,000. So if we randomly select one, there's a 9.3% chance it'll be one of those expensive ones.

(ii) Probability diamond is Ideal cut

```
probability_ideal <- mean(data$cut == "Ideal")
probability_ideal
```

```
## [1] 0.394
```

Interpretation: About 39.4% of the diamonds are of Ideal cut. Hence, the probability that a randomly chosen diamond is Ideal cut is 0.394.

(b) Sample of 20 diamonds

```
prob_more_than_12_ideal <- 1 - pbinom(12, size = 20, prob = probability_ideal)
prob_more_than_12_ideal
```

```
## [1] 0.01834124
```

Interpretation: If we randomly select 20 diamonds, there's less than a 1.83% chance that more than 12 of them will be Ideal cuts.

(c) Confidence intervals for mean carat weight

```r
n <- length(data$carat)
xbar <- mean(data$carat)
s <- sd(data$carat)

ci_90 <- xbar + c(-1, 1) * qt(0.95, df = n - 1) * s / sqrt(n)
ci_90
```

```
## [1] 0.757763 0.805697
```

```r
ci_95 <- xbar + c(-1, 1) * qt(0.975, df = n - 1) * s / sqrt(n)
ci_95
```

```
## [1] 0.7531634 0.8102966
```

Interpretation: We are 90% confident that the true mean carat weight lies between 0.758 and 0.806. We are 95% confident that the true mean carat weight lies between 0.753 and 0.810.

```r
# Comparing widths
width_90 <- ci_90[2] - ci_90[1]
width_95 <- ci_95[2] - ci_95[1]
width_90
```

```
## [1] 0.04793403
```

```r
width_95
```

```
## [1] 0.05713317
```

Interpretation: The 95% confidence interval is wider (0.057) than the 90% interval (0.048). This happens because to be more certain we've captured the true mean, we need a broader range.

==========

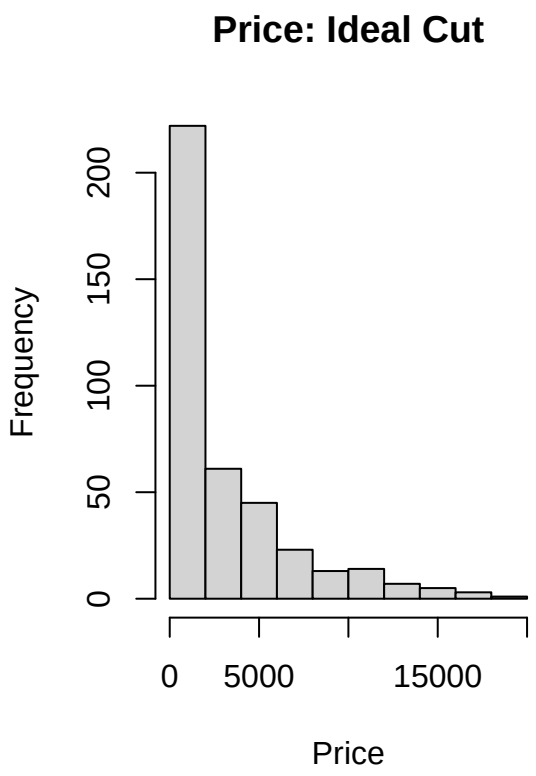
Question 3: Hypothesis Tests

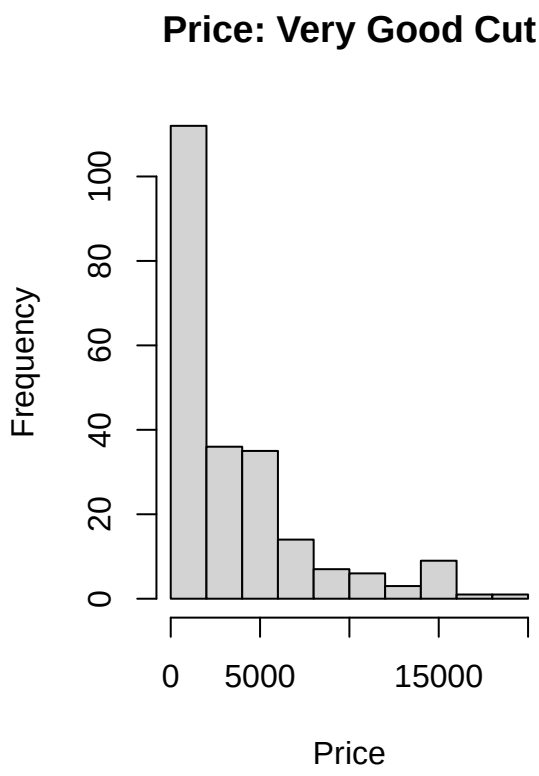
(a) i–ii: Distributions of price for Ideal and Very Good

```r
ideal_price <- data$price[data$cut == "Ideal"]
vg_price <- data$price[data$cut == "Very Good"]
```

iii) Based on the plots, do you believe they are normally distributed?

```r
par(mfrow = c(1, 2))
hist(ideal_price, main = "Price: Ideal Cut", xlab = "Price")
hist(vg_price, main = "Price: Very Good Cut", xlab = "Price")
```



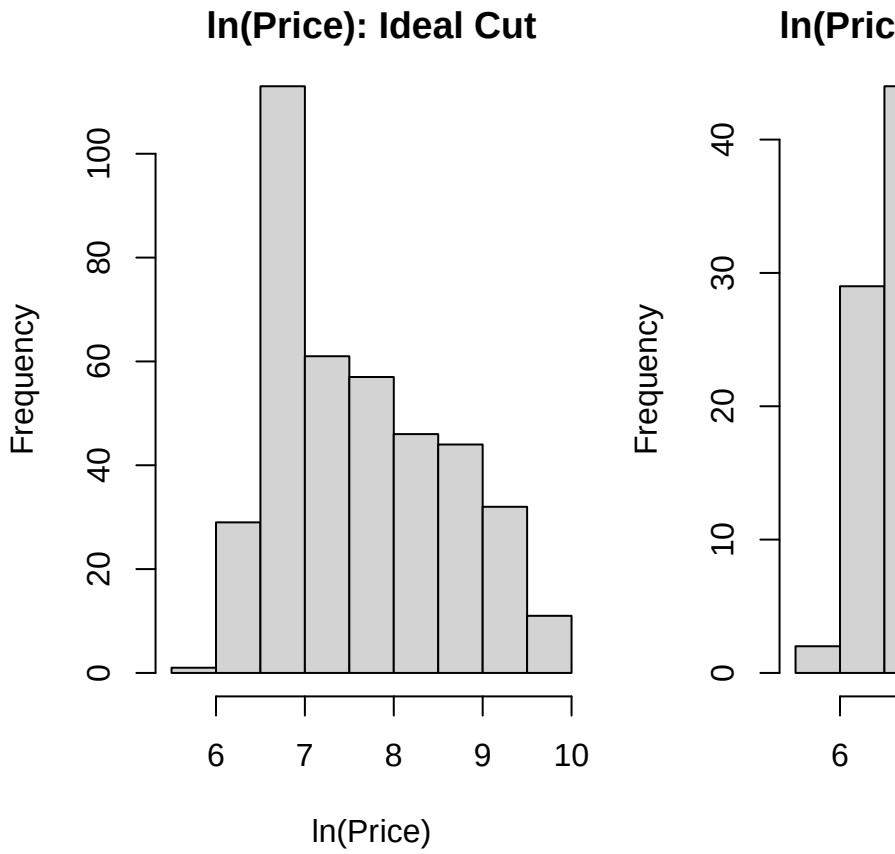



```r
par(mfrow = c(1, 1))
```

The price distributions for both Ideal and Very Good cut diamonds are right-skewed and not symmetric. Therefore, the prices do not appear to be normally distributed.

```r
data$ln_price <- log(data$price)
ideal_ln <- data$ln_price[data$cut == "Ideal"]
vg_ln <- data$ln_price[data$cut == "Very Good"]
```

```r
par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))
hist(ideal_ln, main = "ln(Price): Ideal Cut", xlab = "ln(Price)")
hist(vg_ln,    main = "ln(Price): Very Good Cut", xlab = "ln(Price)")
```



(a) iv: Create ln(price) and repeat plots

```
par(mfrow = c(1, 1))
```

After applying ln(price), both plots look more balanced and less skewed. So, the ln(price) values appear more normally distributed than the original prices for both cuts.

(b) t-test on price (5% level)

H0: mean price (Ideal) = mean price (Very Good)

```
t_price <- t.test(ideal_price, vg_price, alternative = "two.sided", conf.level = 0.95)
t_price
```

H1: mean price (Ideal) != mean price (Very Good)

```
##
##  Welch Two Sample t-test
##
## data:  ideal_price and vg_price
## t = -1.2234, df = 431.73, p-value = 0.2219
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -1015.3727   236.2943
## sample estimates:
## mean of x mean of y
##  3275.563  3665.103
```

We test whether the average price of Ideal cut diamonds is different from that of Very Good cut diamonds. The p-value is 0.2219, which is greater than 0.05, so we do not reject the null hypothesis. This means there is no significant difference in the mean prices of Ideal and Very Good cut diamonds at the 5% significance level.

(c) t-test on ln(price) (5% level)

H0: mean ln(price) (Ideal) = mean ln(price) (Very Good)

```
t_ln <- t.test(ideal_ln, vg_ln, alternative = "two.sided", conf.level =0.95)
t_ln
```

H1: mean ln(price) (Ideal) != mean ln(price) (Very Good)

```
##
##  Welch Two Sample t-test
##
## data:  ideal_ln and vg_ln
## t = -0.92072, df = 432.88, p-value = 0.3577
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.24054911  0.08707367
## sample estimates:
## mean of x mean of y
##  7.620497  7.697235
```

We checked if Ideal and Very Good diamonds cost different after using ln(price) and the test says they cost almost the same, so there is no real difference.

(d) Compare conclusions using p-values

```
t_price$p.value
```

```
## [1] 0.2218553
```

```
t_ln$p.value
```

```
## [1] 0.3577081
```

Both p-values are bigger than 0.05, so we do not see any real difference.This means Ideal and Very Good diamonds have similar prices, both before and after taking ln(price).

(e) Chi-square independence test: price group (>5000) vs clarity

```
data$price_group <- ifelse(data$price > 5000, "High", "Low")
cont_table <- table(data$price_group, data$clarity)
cont_table
```

```
##
##         I1  IF SI1 SI2 VS1 VS2 VVS1 VVS2
##   High   6   8  64  37  41  76    7   13
##   Low   11  38 164 121 116 147   67   84
```

```
chisq.test(cont_table)
```

```
##
##  Pearson's Chi-squared test
##
## data:  cont_table
## X-squared = 29.959, df = 7, p-value = 9.661e-05
```

The p-value is very small (much less than 0.05), so price group and clarity are linked and this means diamonds with higher prices tend to have different clarity levels than cheaper ones.

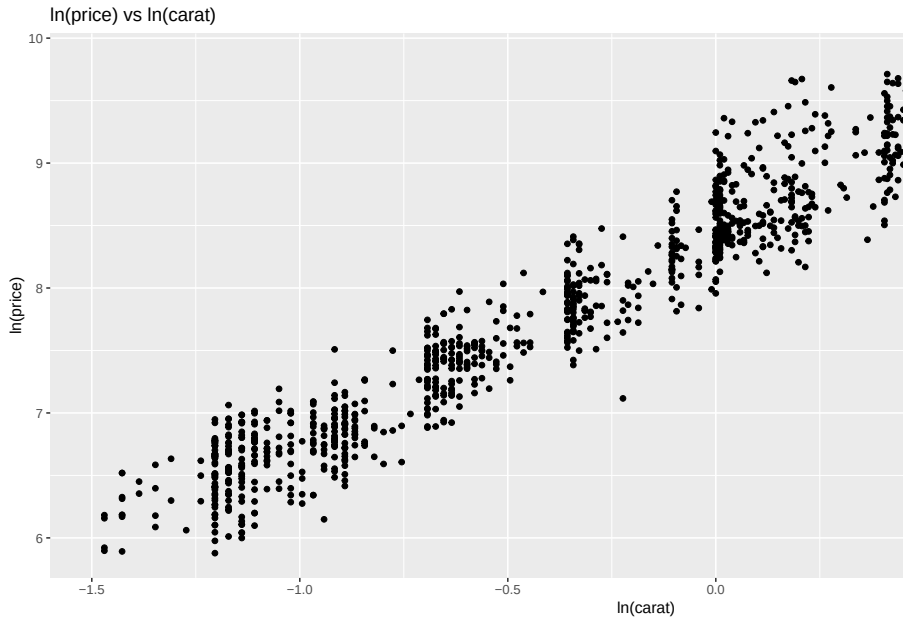
==========

Question 4: Linear Regression

Create log variables

```
data$ln_price <- log(data$price)
data$ln_carat <- log(data$carat)
```

```
library(ggplot2)
ggplot(data, aes(x = ln_carat, y = ln_price)) +
  geom_point() +
  labs(x = "ln(carat)", y = "ln(price)", title = "ln(price) vs ln(carat)")
```



(a) Scatter plot of ln(price) vs ln(carat)

The scatter plot shows a clear upward trend. This means that as ln(carat) increases, ln(price) also increases, suggesting a strong positive relationship between carat size and price.

```
m1 <- lm(ln_price ~ ln_carat, data = data)
s1 <- summary(m1)
```

(b) Simple regression: ln(price) ~ ln(carat)

```
slope_m1 <- coef(m1)["ln_carat"]
slope_m1
```

(b)(i) slope

```
## ln_carat
## 1.656087
```

The slope is 1.656, which means that a 1% increase in carat size leads to about a 1.66% increase in price on average.

```
pval_m1 <- s1$coefficients["ln_carat", "Pr(>|t|)"]
pval_m1
```

(b)(ii) p-value for ln(carat)

```
## [1] 0
```

The p-value is 0, which is much smaller than 0.05. This shows that ln(carat) is a statistically significant predictor of ln(price).

```
r2_m1 <- s1$r.squared
r2_m1
```

(b)(iii) R-squared

```
## [1] 0.9279249
```

```
m2 <- lm(ln_price ~ ln_carat + cut, data = data)
s2 <- summary(m2)
s2
```

(c) Multiple regression: ln(price) ~ ln(carat) + cut

```
##
## Call:
## lm(formula = ln_price ~ ln_carat + cut, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.86330 -0.15805 -0.01857  0.16553  0.89623
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.15080    0.04588 177.644  < 2e-16 ***
## ln_carat      1.67895    0.01448 115.937  < 2e-16 ***
## cutGood       0.16828    0.05480   3.071  0.00219 **
## cutIdeal      0.36339    0.04851   7.491 1.51e-13 ***
## cutPremium    0.30738    0.04869   6.313 4.12e-10 ***
## cutVery Good  0.28626    0.04956   5.776 1.02e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2553 on 994 degrees of freedom
## Multiple R-squared:  0.9336, Adjusted R-squared:  0.9333
## F-statistic:  2797 on 5 and 994 DF,  p-value: < 2.2e-16
```

After adding cut to the model, ln(carat) is still highly significant. All cut categories are also significant, showing that both carat size and cut quality affect diamond prices.

```r
sig_1pct <- rownames(s2$coefficients)[s2$coefficients[, "Pr(>|t|)"] < 0.01]
sig_1pct
```

(d) Significant variables at 1% level

```
## [1] "(Intercept)"  "ln_carat"     "cutGood"      "cutIdeal"     "cutPremium"
## [6] "cutVery Good"
```

At the 1% significance level, ln(carat) and all cut categories are significant. This means these variables have a strong and reliable effect on ln(price).

(e) Hypothesis test for ln(carat) in model (c)

H0: beta_ln_carat = 0

H1: beta_ln_carat != 0

We test whether ln(carat) has an effect on ln(price). Since the p-value is effectively zero, we reject the null hypothesis and conclude that ln(carat) is a significant predictor, even after including cut.

```r
s2$coefficients["ln_carat", ]
```

(f) Result of the test (t-statistic and p-value for ln(carat) in model (c))

```
##     Estimate   Std. Error      t value     Pr(>|t|)
##   1.67895120   0.01448158 115.93701166   0.00000000
```

The t-statistic is very large (115.94) and the p-value is 0, confirming strong evidence that ln(carat) has a significant positive effect on ln(price).