MA334 Assignment Report

Nallapu Naveen

2026-01-07

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)

kable(data_pre, caption = "First 6 Observations")
```

Table 1: First 6 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
0.56	Very Good	E	VS1	62.7	60	1931	5.25	5.28	3.30
0.70	Very Good	I	VS1	63.3	55	1995	5.64	5.60	3.56
1.01	Premium	I	SI2	62.6	58	3818	6.43	6.38	4.01
1.02	Very Good	F	VS2	63.3	58	6007	6.36	6.38	4.03

```
# Number of observations and variables
no_of_observations <- nrow(data)
no_of_variables <- ncol(data)

# Creating a table to show the size of the dataset
dimensions_table <- data.frame(
  Item = c("Total Observations", "Total Variables"),
  Value = c(no_of_observations, no_of_variables)
)

kable(dimensions_table, caption = "Dataset Dimensions")
```

Table 2: Dataset Dimensions

Item	Value
Total Observations	1000
Total Variables	10

(Dataset Dimensions)

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

```r
# Variable names and types
variable_types <- data.frame(
  Variable = names(data),
  Type = sapply(data, class),
  row.names = NULL
)

kable(variable_types, 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 2 shows all the variables in the dataset and their data types. Some variables contain numbers (such as carat, depth, price, and size measurements), while others contain categories (cut, color, and clarity).

```r
# Identifying qualitative variables
categorical_vars <- names(data)[sapply(data, function(x)
  is.character(x) | is.factor(x))]

# Converting qualitative variables to factors
data[categorical_vars] <- lapply(data[categorical_vars], as.factor)
```

```r
# Create table for categories
categories_table <- do.call(rbind, lapply(categorical_vars, function(v) {
  data.frame(
    Variable = v,
    Categories = paste(levels(data[[v]]), collapse = ", "),
    stringsAsFactors = FALSE
  )
}))

kable(categories_table, caption = "Categories for Qualitative Variables")
```

Table 4: Categories for Qualitative Variables

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

(Categories)

Table 3 shows the categories for each qualitative variable.
The 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)),
  row.names = NULL
)

kable(summary_table, caption = "Summary Statistics for Numeric Variables")
```

Table 5: Summary Statistics for Numeric Variables

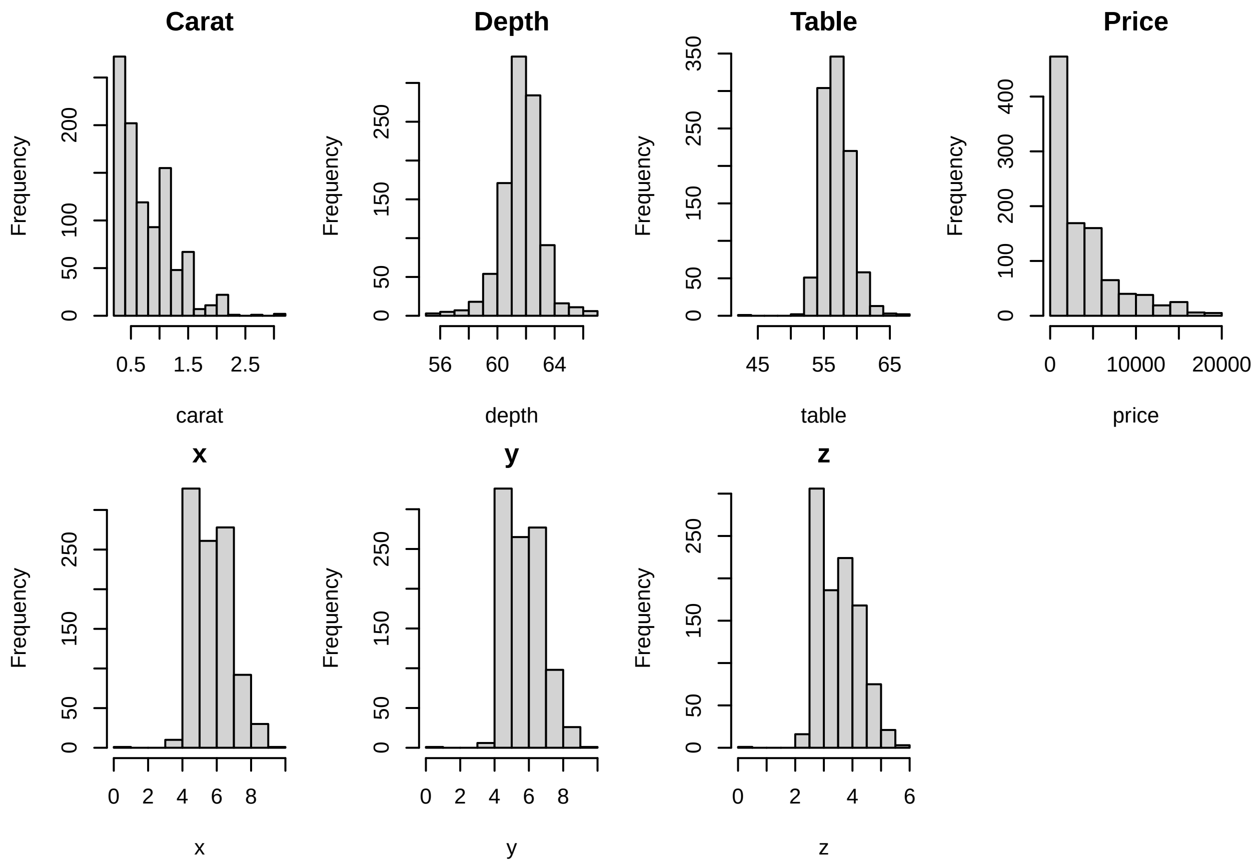
Variable	Mean	Median	Standard_Deviation	Range
carat	0.78173	0.70	0.4603453	2.88
depth	61.75350	61.90	1.3726660	11.80

Variable	Mean	Median	Standard_Deviation	Range
table	57.44450	57.00	2.1992604	24.00
price	3786.08000	2278.50	3833.2003865	18403.00
x	5.69129	5.64	1.1116064	9.15
y	5.69468	5.64	1.1025724	9.02
z	3.51608	3.46	0.6894836	5.98

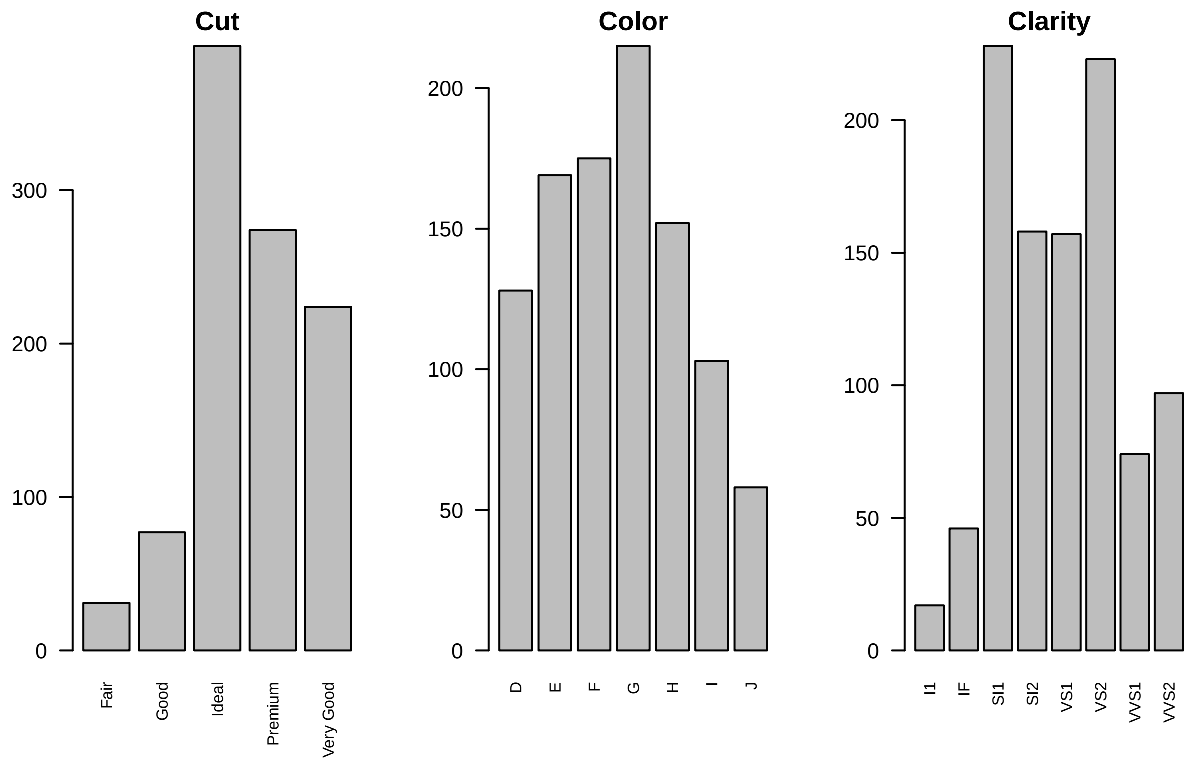
This table shows typical values (mean and median) and how spread out the values are (standard deviation and range) for each numeric variable.

(c) Visualisations of Variable Distributions

```r
# ----- Numeric variables: Histograms (including z) -----
par(mfrow = c(2, 4), mar = c(4, 4, 2, 1))

hist(data$carat, main = "Carat", xlab = "carat")
hist(data$depth, main = "Depth", xlab = "depth")
hist(data$table, main = "Table", xlab = "table")
hist(data$price, main = "Price", xlab = "price")
hist(data$x, main = "x", xlab = "x")
hist(data$y, main = "y", xlab = "y")
hist(data$z, main = "z", xlab = "z")

# Leave one panel empty (2x4 layout has 8 slots)
plot.new()
```




```r
# ----- Categorical variables: Bar plots (labels visible) -----
par(mfrow = c(1, 3), mar = c(8, 4, 2, 1))

barplot(table(data$cut), main = "Cut", las = 2, cex.names = 0.8)
barplot(table(data$color), main = "Color", las = 2, cex.names = 0.8)
barplot(table(data$clarity), main = "Clarity", las = 2, cex.names = 0.8)
```




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


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

# Select numeric variables only
numeric_data <- data[sapply(data, is.numeric)]

# Create correlation matrix
cor_matrix <- cor(numeric_data)

# Display correlation matrix
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

	carat	depth	table	price	x	y	z
z	0.96	0.09	0.18	0.84	0.99	0.99	1.00

```r
# Question 2 (Final)
# =========================

# (a) Probabilities for one randomly chosen diamond
p_price_gt_10000 <- mean(data$price > 10000)      # i)
p_ideal <- mean(data$cut == "Ideal")              # ii)

p_price_gt_10000
```

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

```r
p_ideal
```

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

```r
# (b) From a sample of 20 diamonds, P(more than 12 are Ideal)
# X ~ Binomial(n = 20, p = p_ideal)
prob_more_than_12_ideal <- 1 - pbinom(12, size = 20, prob = p_ideal)

prob_more_than_12_ideal
```

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

```r
# (c) 90% and 95% confidence intervals for mean carat weight (t-interval)
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_95 <- xbar + c(-1, 1) * qt(0.975, df = n - 1) * s / sqrt(n)

ci_90
```

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

```r
ci_95
```

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

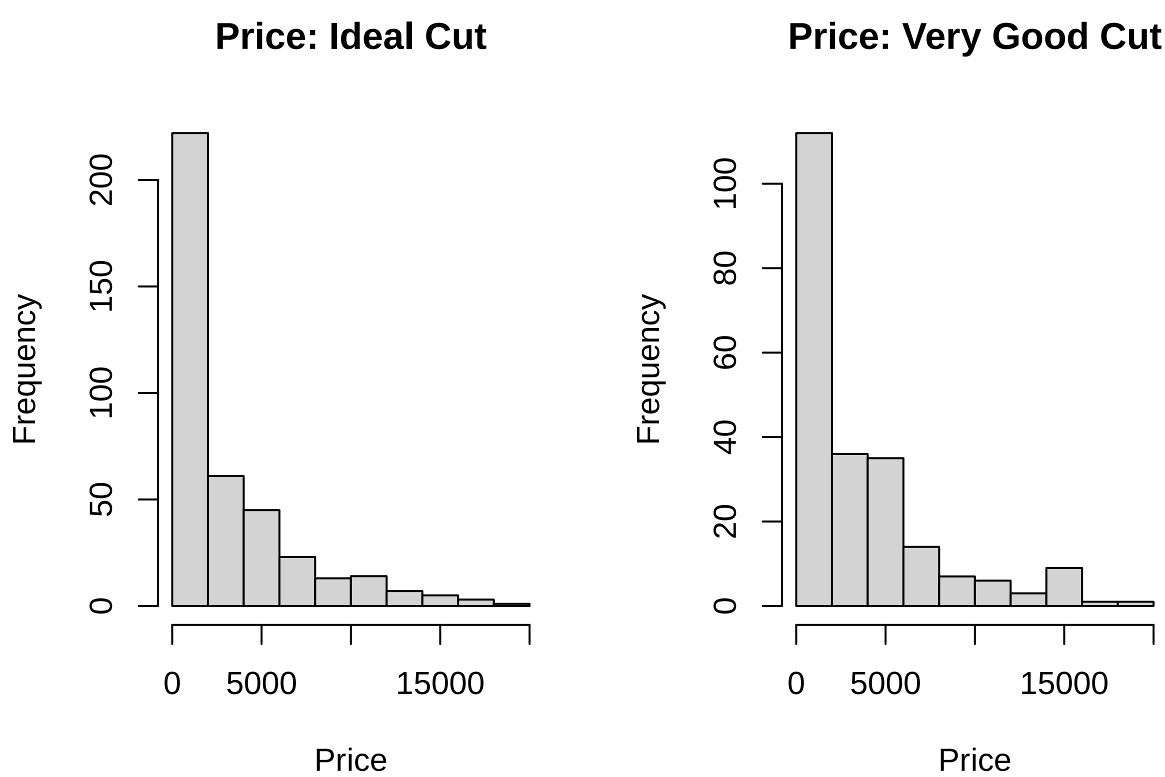
```r
# Which interval is wider?
width_90 <- ci_90[2] - ci_90[1]
width_95 <- ci_95[2] - ci_95[1]

width_90
```

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

```r
width_95
```

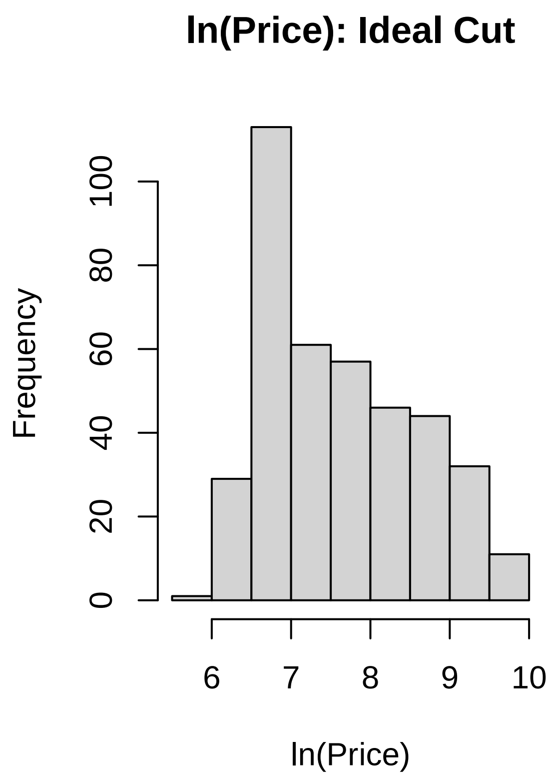
```
## [1] 0.05713317
```

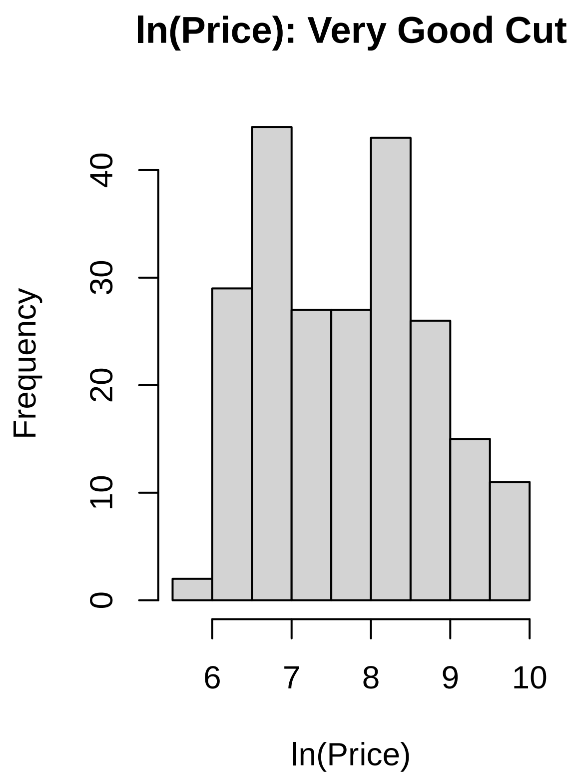
```r
# =========================
# Question 3: Hypothesis Tests (a-e) in ONE chunk
# =========================

# (a) i-ii: Distributions of price for Ideal and Very Good
ideal_price <- data$price[data$cut == "Ideal"]
vg_price <- data$price[data$cut == "Very Good"]

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))

# (a) iv: Create ln(price) and repeat plots
data$ln_price <- log(data$price)
ideal_ln <- data$ln_price[data$cut == "Ideal"]
vg_ln <- data$ln_price[data$cut == "Very Good"]

par(mfrow = c(1, 2))
```

```r
hist(ideal_ln, main = "ln(Price): Ideal Cut", xlab = "ln(Price)")
hist(vg_ln, main = "ln(Price): Very Good Cut", xlab = "ln(Price)")
```






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

# (b) t-test on price (5% level)
# H0: mean price (Ideal) = mean price (Very Good)
# H1: mean price (Ideal) != mean price (Very Good)
t_price <- t.test(ideal_price, vg_price, alternative = "two.sided", conf.level = 0.95)
t_price
```

```
##
##  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
```

```r
# (c) t-test on ln(price) (5% level)
# H0: mean ln(price) (Ideal) = mean ln(price) (Very Good)
# H1: 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
```

```
##
##  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
```

```r
# (d) Compare conclusions using p-values
t_price$p.value
```

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

```r
t_ln$p.value
```

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

```r
# (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
```

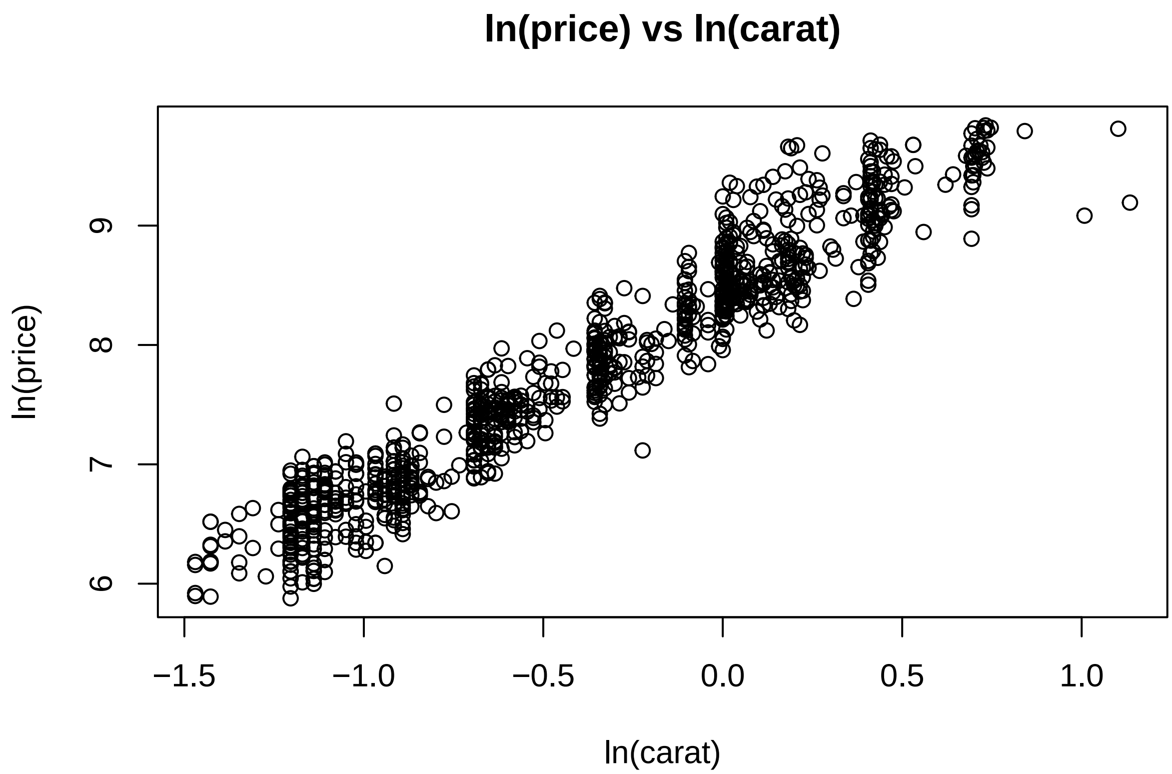
```
##
##         I1  IF SI1 SI2 VS1 VS2 VVS1 VVS2
##   High   6   8  64  37  41  76    7   13
##   Low   11  38 164 121 116 147   67   84
```

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

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

```r
# =========================
# Question 4: Linear Regression
# =========================
```

```r
# Create log variables
data$ln_price <- log(data$price)
data$ln_carat <- log(data$carat)

# (a) Scatter plot of ln(price) vs ln(carat)
plot(data$ln_carat, data$ln_price,
     xlab = "ln(carat)", ylab = "ln(price)",
     main = "ln(price) vs ln(carat)")
```




```r
# (b) Simple regression: ln(price) ~ ln(carat)
m1 <- lm(ln_price ~ ln_carat, data = data)
s1 <- summary(m1)

# (b)(i) slope
slope_m1 <- coef(m1)["ln_carat"]
slope_m1
```

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

```r
# (b)(ii) p-value for ln(carat)
pval_m1 <- s1$coefficients["ln_carat", "Pr(>|t|)"]
pval_m1
```

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

```r
# (b)(iii) R-squared
r2_m1 <- s1$r.squared
r2_m1
```

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

```r
# (c) Multiple regression: ln(price) ~ ln(carat) + cut
m2 <- lm(ln_price ~ ln_carat + cut, data = data)
s2 <- summary(m2)
s2
```

```
##
## 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
```

```r
# (d) Significant variables at 1% level
sig_1pct <- rownames(s2$coefficients)[s2$coefficients[, "Pr(>|t|)"] < 0.01]
sig_1pct
```

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

```r
# (e) Hypothesis test for ln(carat) in model (c)
# H0: beta_ln_carat = 0
# H1: beta_ln_carat != 0

# (f) Result of the test (t-statistic and p-value for ln(carat) in model (c))
s2$coefficients["ln_carat", ]
```

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