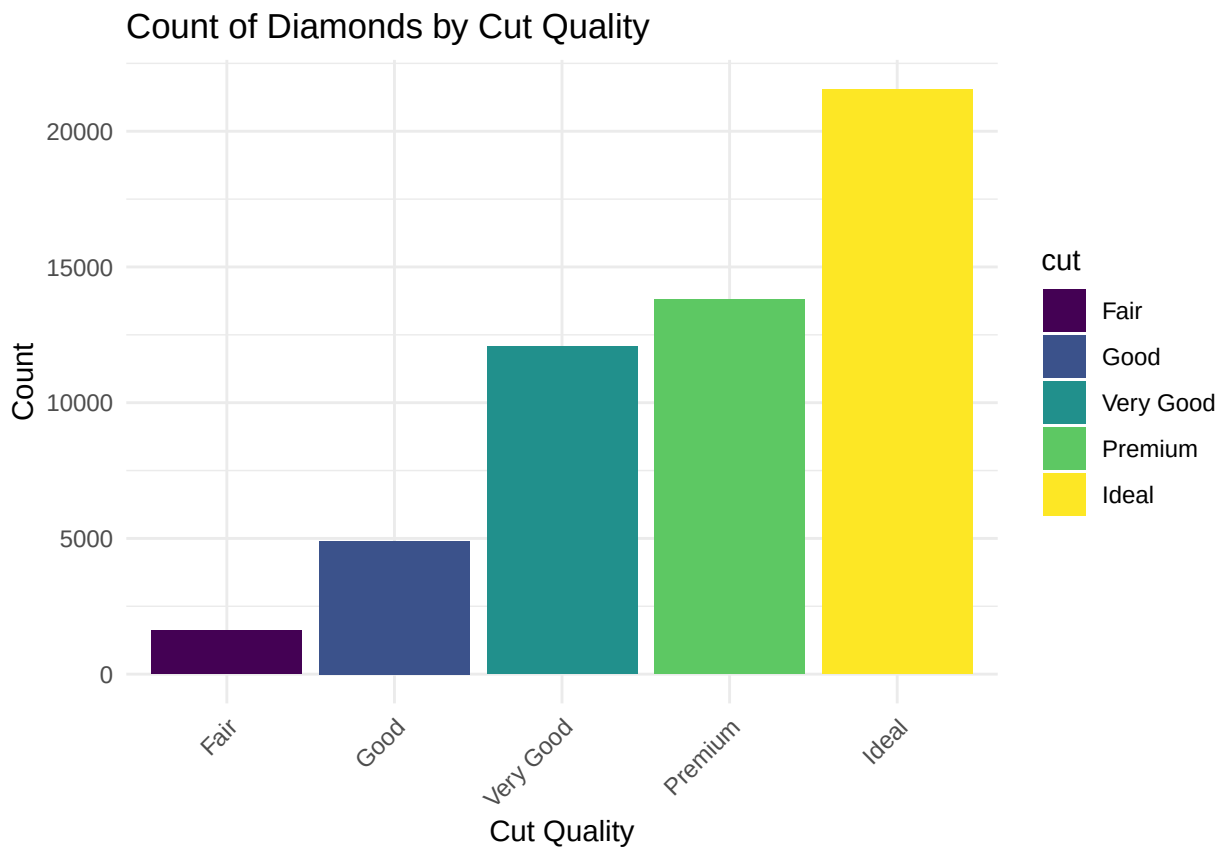
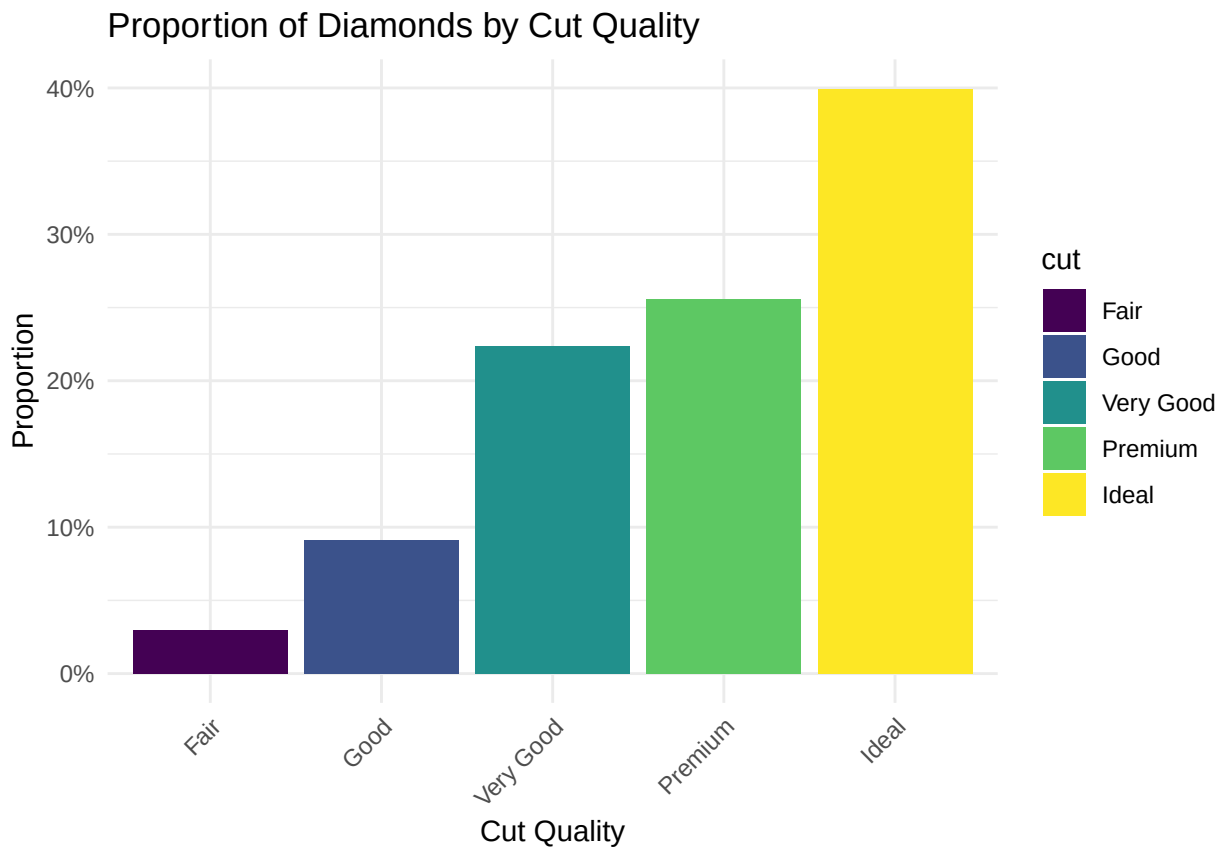
Untitled

2025-08-01

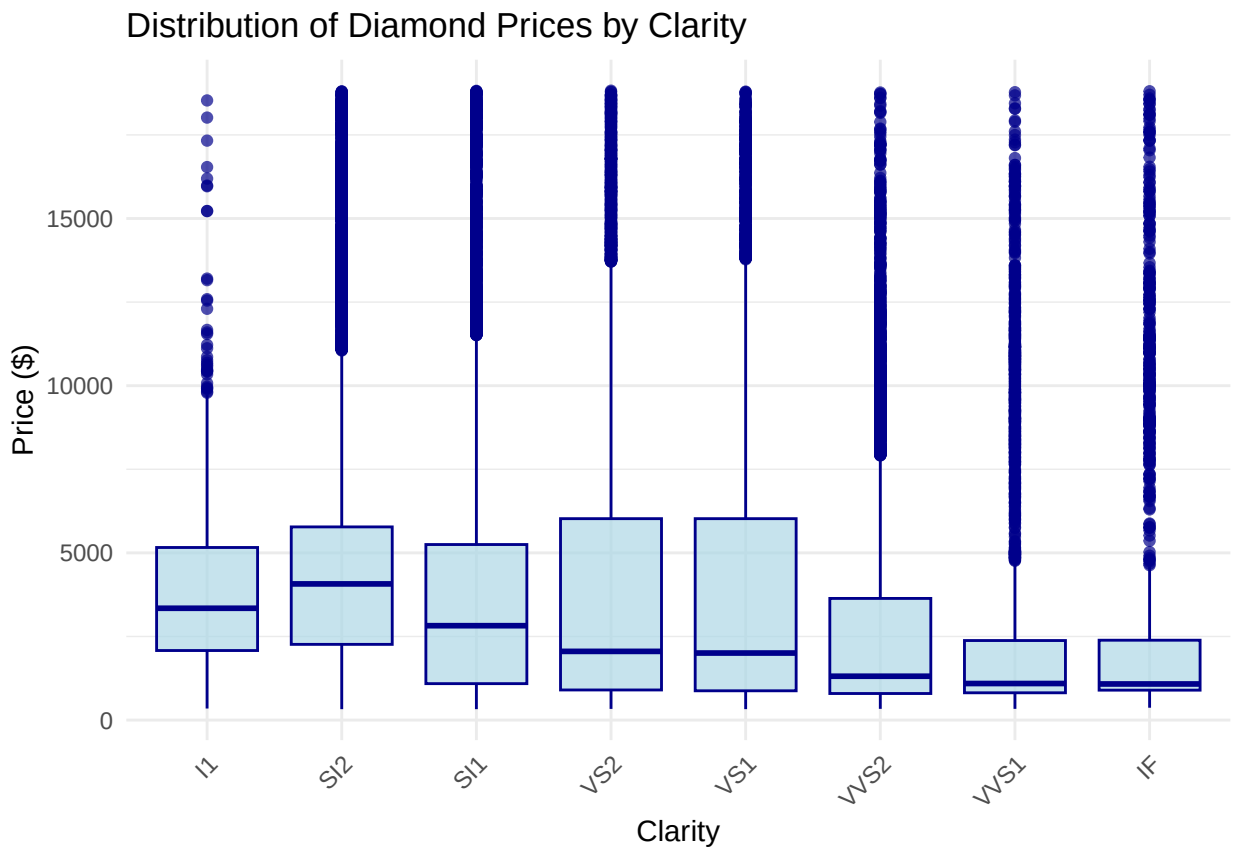
```
pacman :: p_load("dplyr","ggplot2","scales")

View(diamonds)

# Q.1: Bar chart showing count of diamonds for each cut with fill colors

a = ggplot(diamonds, aes(x = cut, fill = cut)) +
  geom_bar() +
  labs(title = "Count of Diamonds by Cut Quality",
       x = "Cut Quality",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

print(a)
```



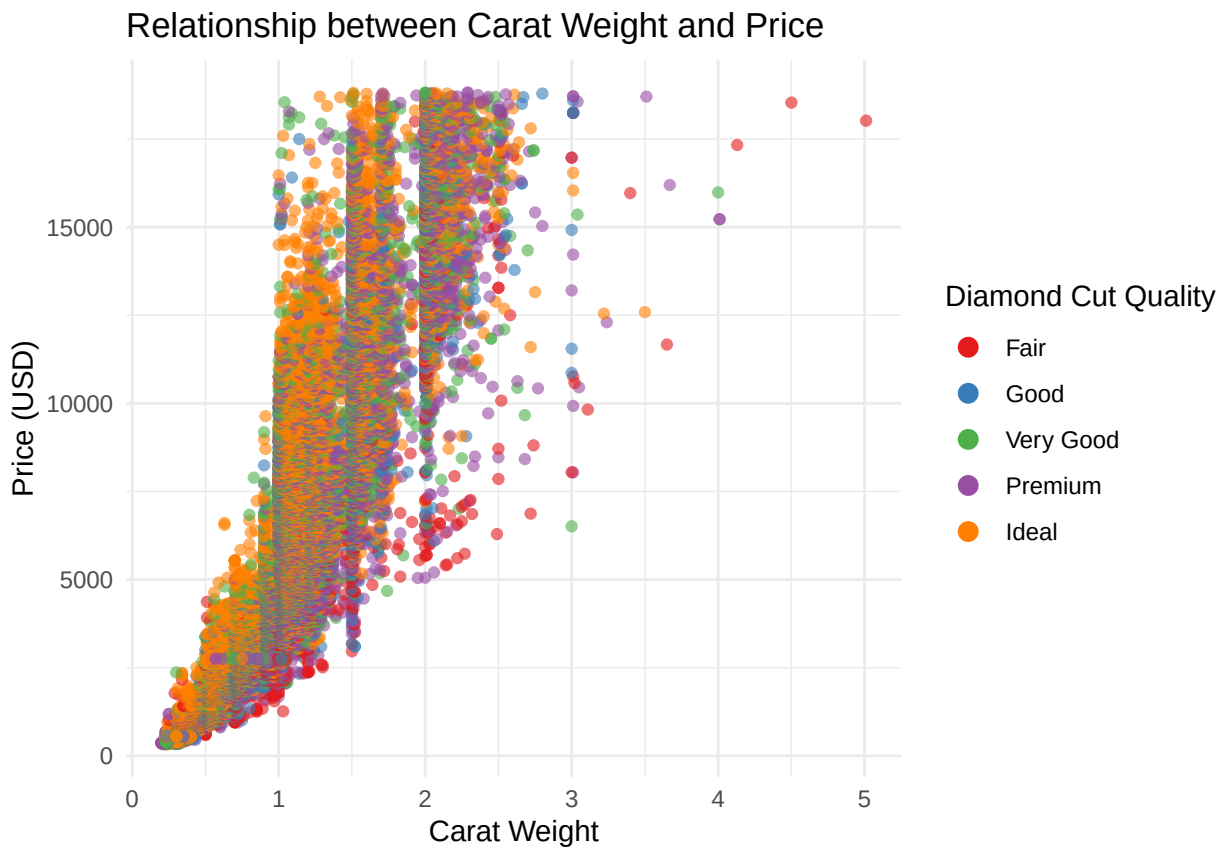
```r
# Q.2: Modified bar chart to show proportions

b = ggplot(diamonds, aes(x = cut, fill = cut)) +
  geom_bar(aes(y = after_stat(count/sum(count)))) +
  labs(title = "Proportion of Diamonds by Cut Quality",
       x = "Cut Quality",
       y = "Proportion") +
  scale_y_continuous(labels = percent_format()) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

print(b)
```



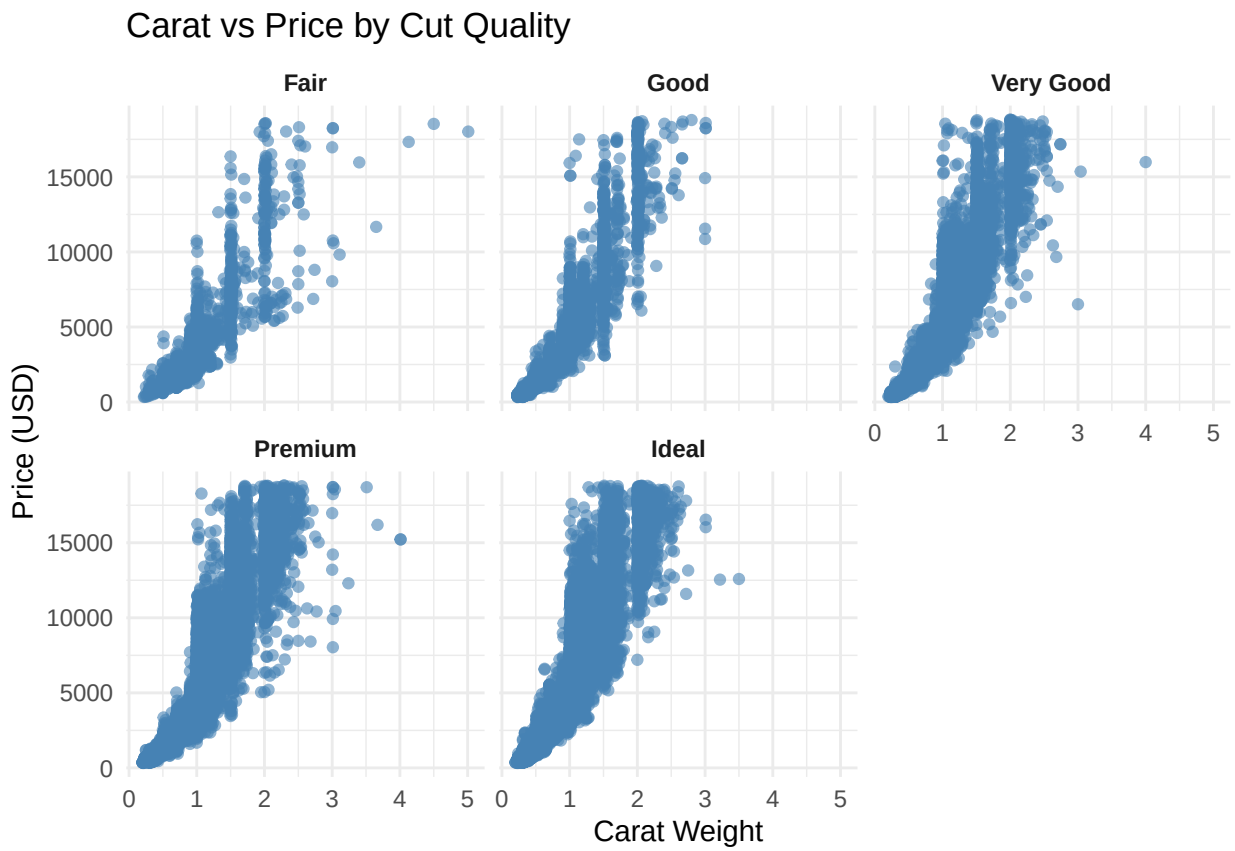

```r
#Q.3: Boxplot comparing diamond price across clarity levels with custom color


c = ggplot(diamonds, aes(x = clarity, y = price)) +
  geom_boxplot(fill = "lightblue", color = "darkblue", alpha = 0.7) +
  labs(title = "Distribution of Diamond Prices by Clarity",
       x = "Clarity",
       y = "Price ($)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

print(c)
```



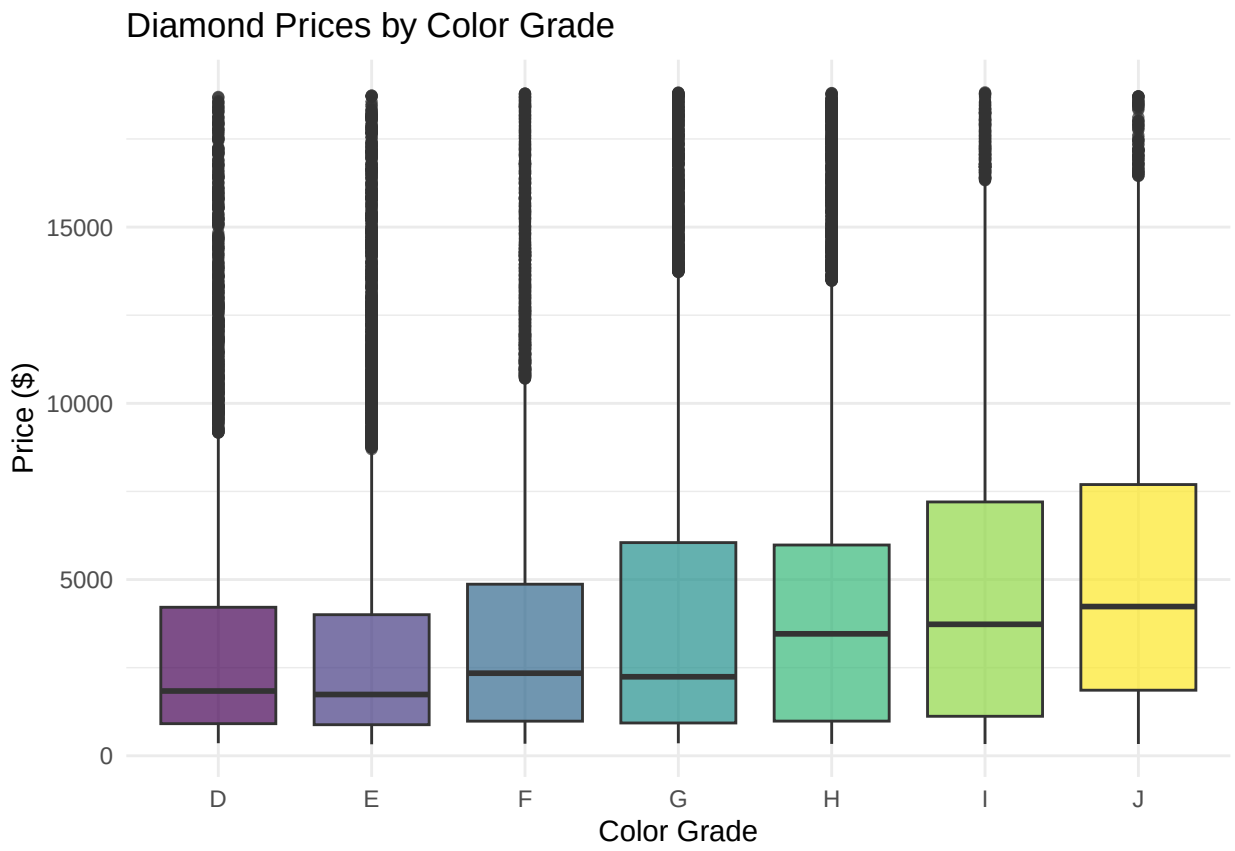

```
# Q.4: Scatter plot of carat vs price with cut differentiation and custom legend

d = ggplot(diamonds, aes(x = carat, y = price, color = cut)) +
  geom_point(alpha = 0.6) +
  labs(title = "Relationship between Carat Weight and Price",
       x = "Carat Weight",
       y = "Price (USD)",
       color = "Cut Quality") +
  scale_color_brewer(type = "qual", palette = "Set1") +
  theme_minimal() +
  guides(color = guide_legend(title = "Diamond Cut Quality",
                              override.aes = list(alpha = 1, size = 3)))

print(d)
```




```
# Q.5: Scatter plot with subplots for different cut qualities


e = ggplot(diamonds, aes(x = carat, y = price)) +
  geom_point(alpha = 0.6, color = "steelblue") +
  facet_wrap(~ cut, nrow = 2, ncol = 3) +
  labs(title = "Carat vs Price by Cut Quality",
       x = "Carat Weight",
       y = "Price (USD)") +
  theme_minimal() +
  theme(strip.text = element_text(face = "bold"))

print(e)
```



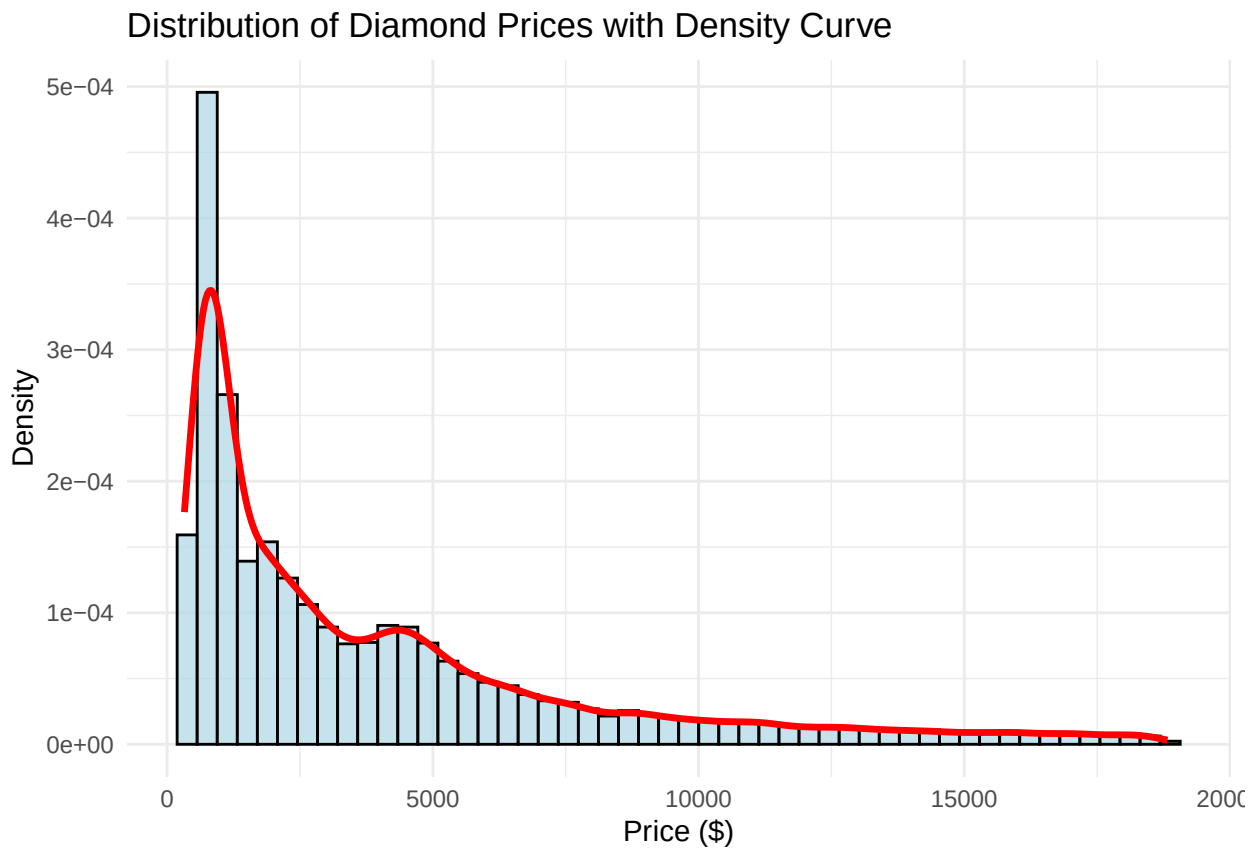

```r
# Q.6: Box plot showing prices for different diamond colors

f = ggplot(diamonds, aes(x = color, y = price, fill = color)) +
  geom_boxplot(alpha = 0.7) +
  labs(title = "Diamond Prices by Color Grade",
       x = "Color Grade",
       y = "Price ($)") +
  scale_fill_viridis_d() +
  theme_minimal() +
  theme(legend.position = "none")

print(f)
```



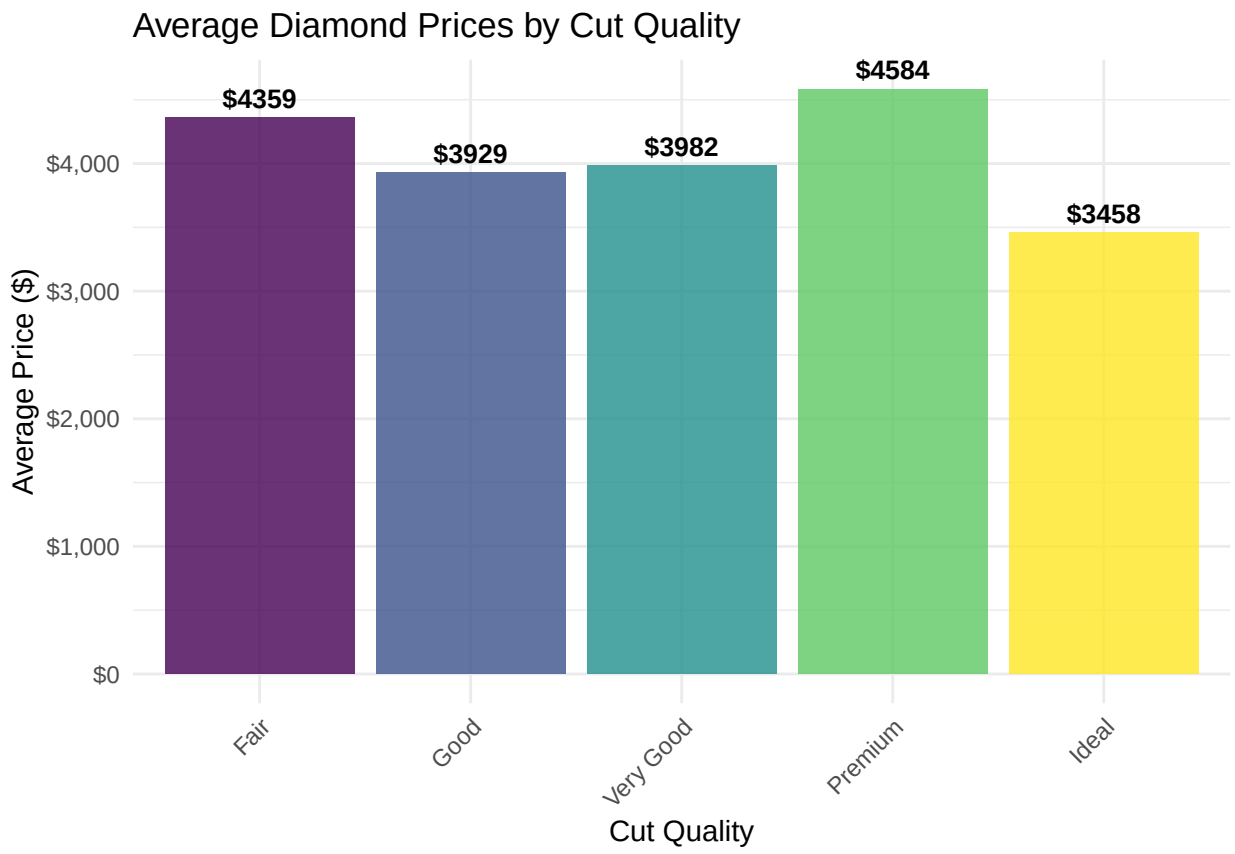

```
# Q.7: Histogram of price with density curve


g = ggplot(diamonds, aes(x = price)) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = 50, fill = "lightblue", alpha = 0.7, color = "black") +
  geom_density(color = "red", size = 1.2) +
  labs(title = "Distribution of Diamond Prices with Density Curve",
       x = "Price ($)",
       y = "Density") +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

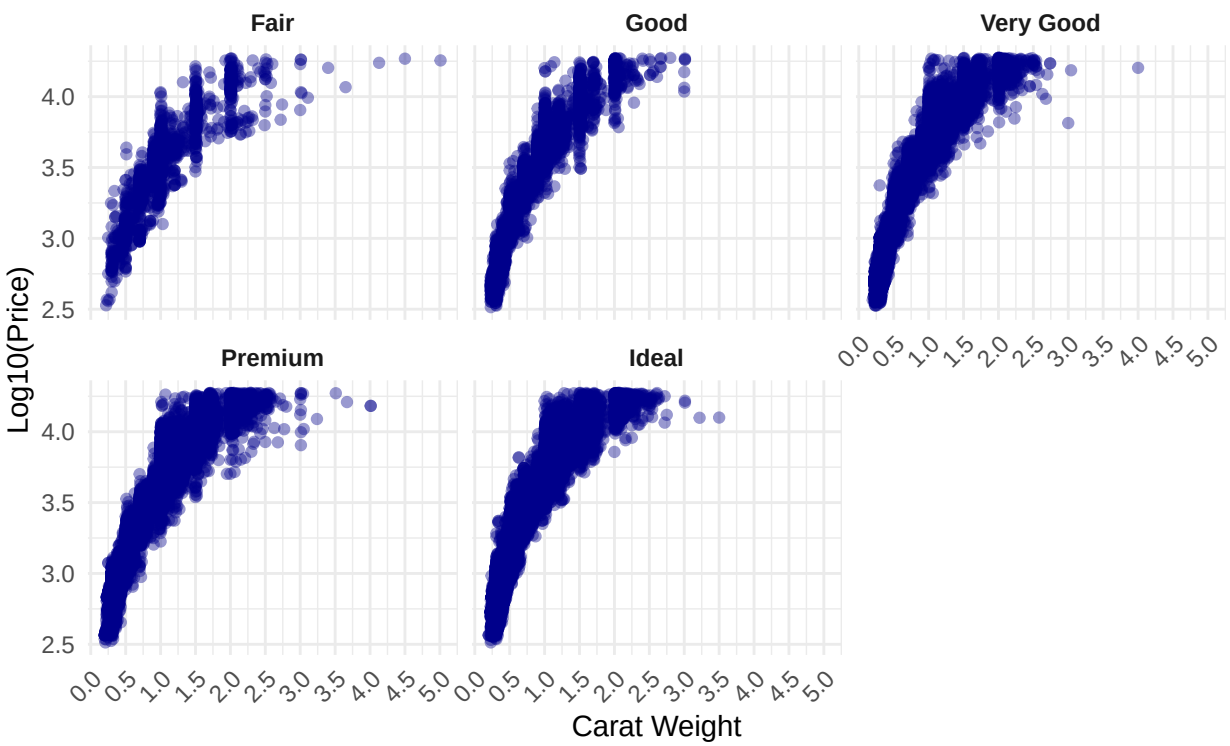
```
print(g)
```



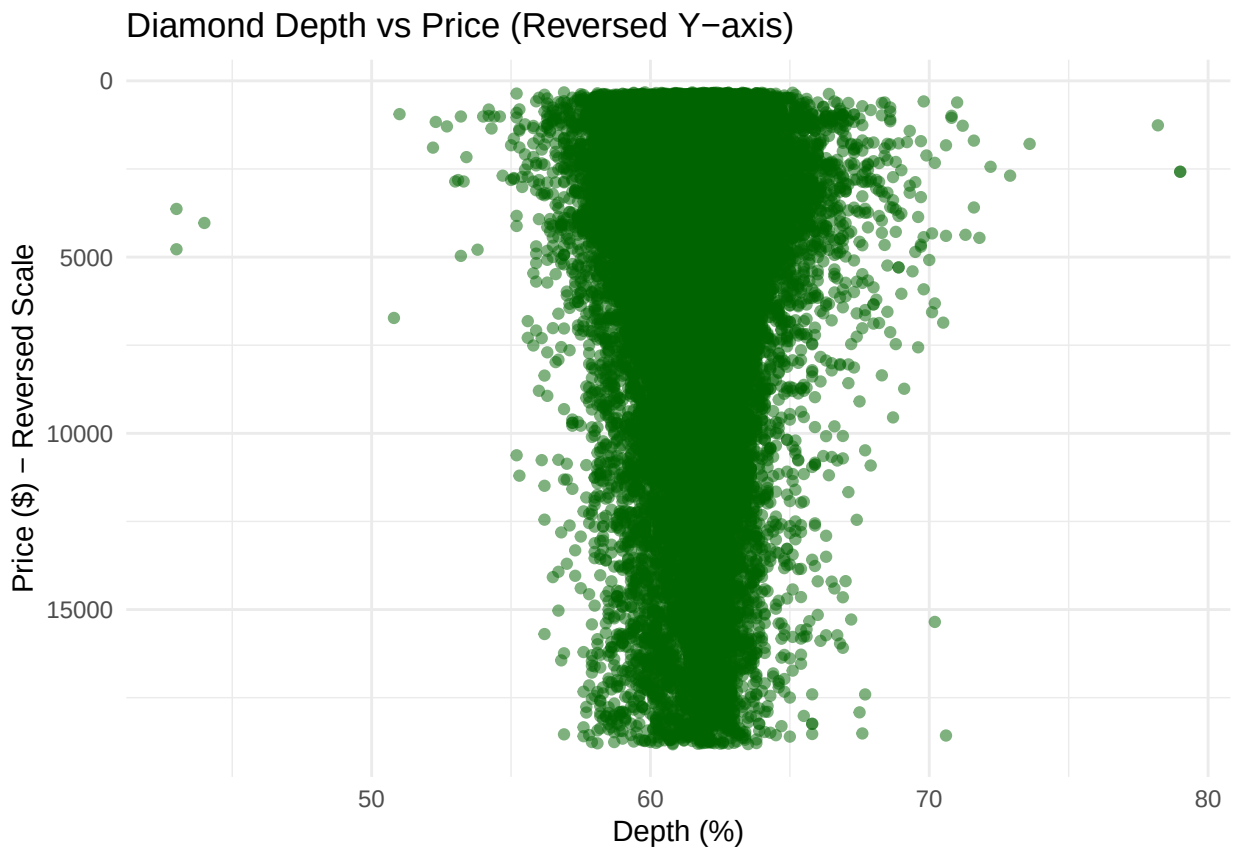

```
# Q.8: Bar chart showing average prices by cut with values displayed


avg_price_by_cut <- diamonds %>%
  group_by(cut) %>%
  summarise(avg_price = mean(price), .groups = 'drop')

h = ggplot(avg_price_by_cut, aes(x = cut, y = avg_price, fill = cut)) +
  geom_bar(stat = "identity", alpha = 0.8) +
  geom_text(aes(label = paste0("$", round(avg_price, 0))),
            vjust = -0.5, size = 3.5, fontface = "bold") +
  labs(title = "Average Diamond Prices by Cut Quality",
       x = "Cut Quality",
       y = "Average Price ($)") +
  scale_y_continuous(labels = dollar_format()) +
  theme_minimal() +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 45, hjust = 1))

print(h)
```




```
# Q.9: Log-transformed price vs carat with facets and custom formatting


i = ggplot(diamonds, aes(x = carat, y = log10(price))) +
  geom_point(alpha = 0.4, color = "darkblue") +
  facet_wrap(~ cut, nrow = 2, ncol = 3) +
  labs(title = "Log-Transformed Price vs Carat Weight by Cut Quality",
       subtitle = "Log transformation reduces price skewness",
       x = "Carat Weight",
       y = "Log10(Price)") +
  scale_x_continuous(breaks = seq(0, 5, 0.5)) +
  theme_minimal() +
  theme(strip.text = element_text(face = "bold"),
        axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(size = 14, face = "bold"),
        plot.subtitle = element_text(size = 11))

print(i)
```




```
# Q.10: Scatter plot of depth vs price with reversed y-axis

j = ggplot(diamonds, aes(x = depth, y = price)) +
  geom_point(alpha = 0.5, color = "darkgreen") +
  scale_y_reverse() +  # This reverses the y-axis
  labs(title = "Diamond Depth vs Price (Reversed Y-axis)",
       x = "Depth (%)",
       y = "Price ($) - Reversed Scale") +
  theme_minimal()

print(j)
```




```
# Summary of dataset characteristics

eth = summary(diamonds)


print(eth)
```

```
##      carat               cut        color        clarity          depth
##  Min.   :0.2000   Fair      : 1610   D: 6775   SI1     :13065   Min.   :43.00
##  1st Qu.:0.4000   Good      : 4906   E: 9797   VS2     :12258   1st Qu.:61.00
##  Median :0.7000   Very Good:12082   F: 9542   SI2     : 9194   Median :61.80
##  Mean   :0.7979   Premium   :13791   G:11292   VS1     : 8171   Mean   :61.75
##  3rd Qu.:1.0400   Ideal     :21551   H: 8304   VVS2    : 5066   3rd Qu.:62.50
##  Max.   :5.0100                     I: 5422   VVS1    : 3655   Max.   :79.00
##                                     J: 2808   (Other): 2531
##      table           price             x                y
##  Min.   :43.00   Min.   :  326   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710   1st Qu.: 4.720
##  Median :57.00   Median : 2401   Median : 5.700   Median : 5.710
##  Mean   :57.46   Mean   : 3933   Mean   : 5.731   Mean   : 5.735
##  3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540
##  Max.   :95.00   Max.   :18823   Max.   :10.740   Max.   :58.900
##
##        z
##  Min.   : 0.000
##  1st Qu.: 2.910
```

```
##  Median : 3.530
##  Mean   : 3.539
##  3rd Qu.: 4.040
##  Max.   :31.800
##
```