

Homework 1

Rudy Martinez

10/04/2020

In this homework, you will use `diamonds` data set from `ggplot2` package. It is automatically loaded when you execute `library(ggplot2)` so you don't have to separately load it. Please create a duplicate of `diamonds` and use that for homework. This will avoid corrupting the original data set.

Make sure that you understand the variables in the data by checking the help file by executing `help(diamonds)` in the console. The data has five variables that measure the dimensions.

Instructions

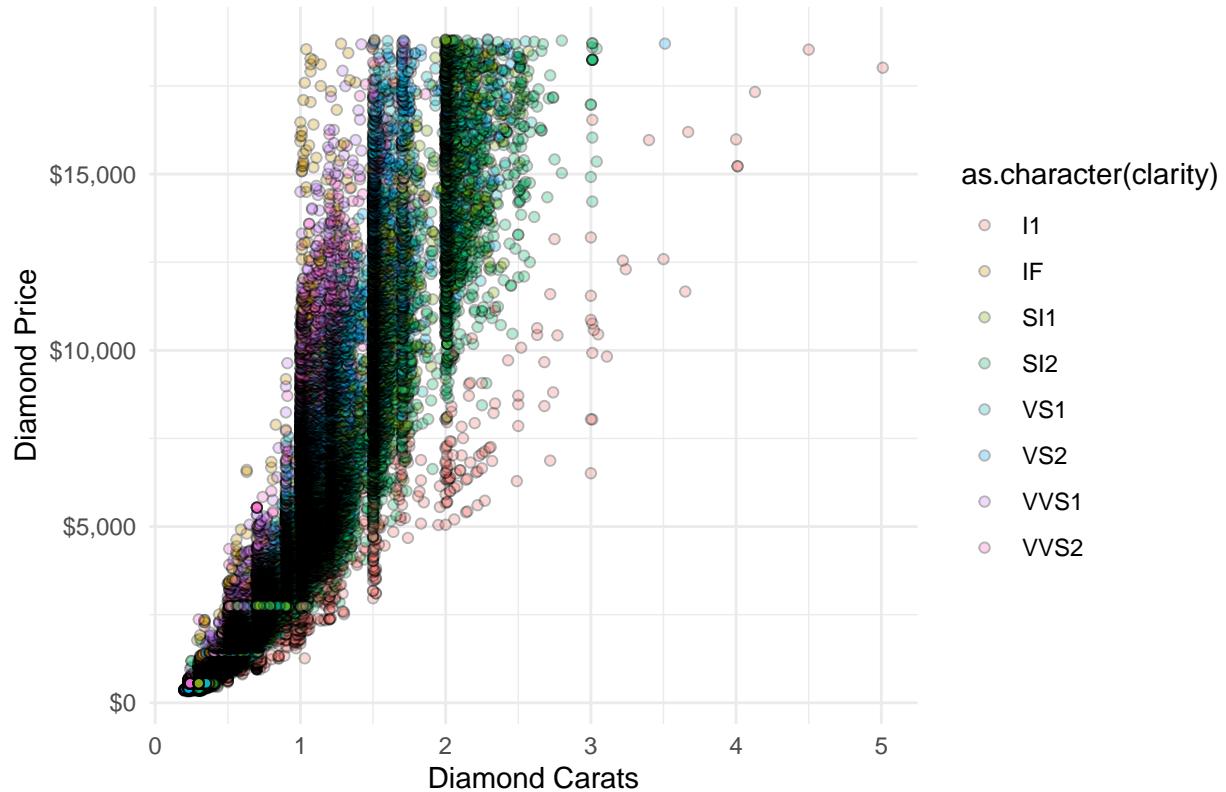
1. You are expected to recreate each plot **exactly** as shown in this homework.
2. The objective of this homework is to help you develop fine grain control over `ggplot2`. As such, please refrain from exercising artistic freedom!
3. All the plots use `theme_minimal()`. You can set this as your default theme by adding this line in the setup chunk **after** you load `ggplot2` library: `theme_set(theme_minimal())`

Q1 (3 points)

Recreate the following graph. The parameter that controls transparency is set at 0.3. You need not get exactly the same colors but they must be **discrete** and should not follow a color gradient.

```
ggplot(diam_1, aes(carat, price)) +
  geom_point(aes(fill = as.character(clarity)),
             color = "black",
             alpha = 0.3,
             shape = 21,
             size = 1.5) +
  labs(title = "Scatterplot of Diamond Prices", x = "Diamond Carats", y = "Diamond Price") +
  scale_y_continuous(labels = scales::dollar_format(prefix = "$"))
```

Scatterplot of Diamond Prices

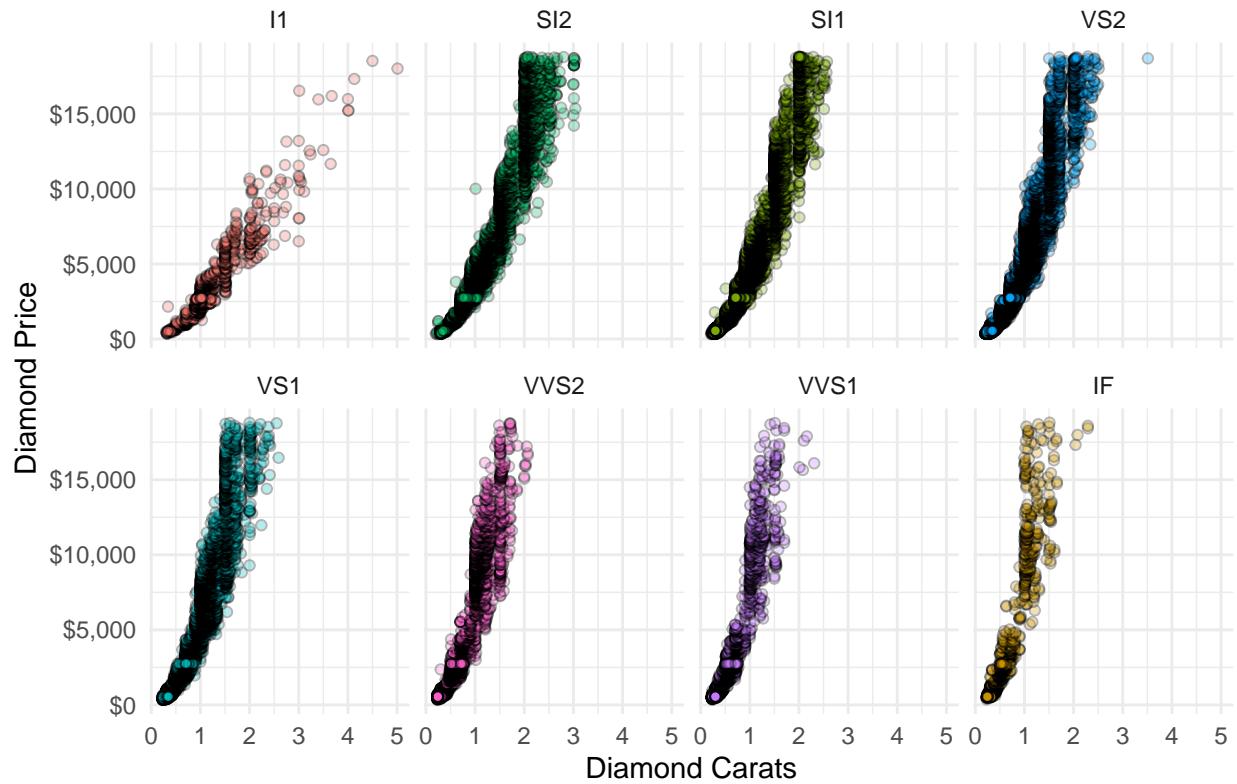


Q2 (2 points)

The previous graph looks cluttered. So you decided to use facets instead. Recreate the following graph:

```
ggplot(diam_1, aes(carat, price)) +
  geom_point(aes(fill = as.character(clarity)),
             color = "black",
             alpha = 0.3,
             shape = 21,
             size = 1.5,
             show.legend = FALSE) +
  labs(title = "Scatterplot of Diamond Prices", x = "Diamond Carats", y = "Diamond Price") +
  scale_y_continuous(labels = scales::dollar_format(prefix = "$")) +
  facet_wrap(~clarity, nrow = 2, ncol = 4)
```

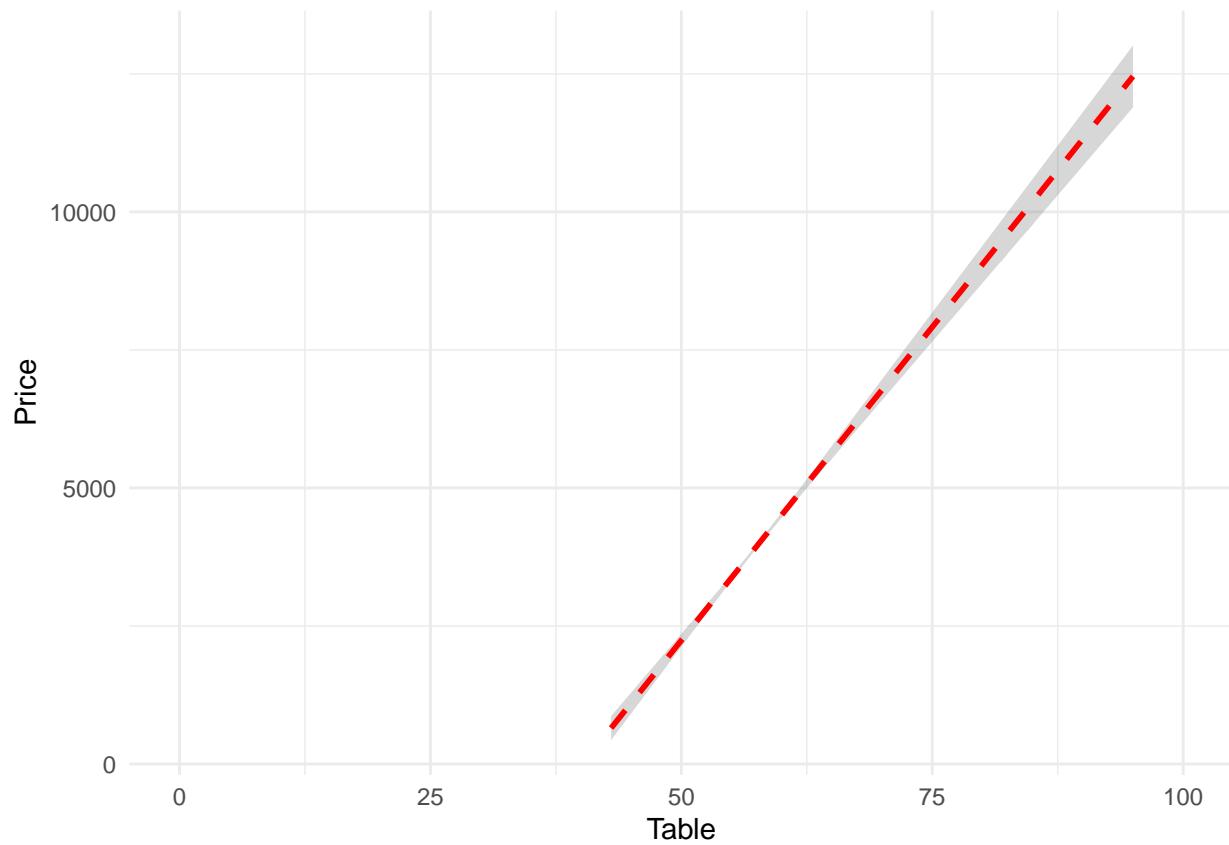
Scatterplot of Diamond Prices



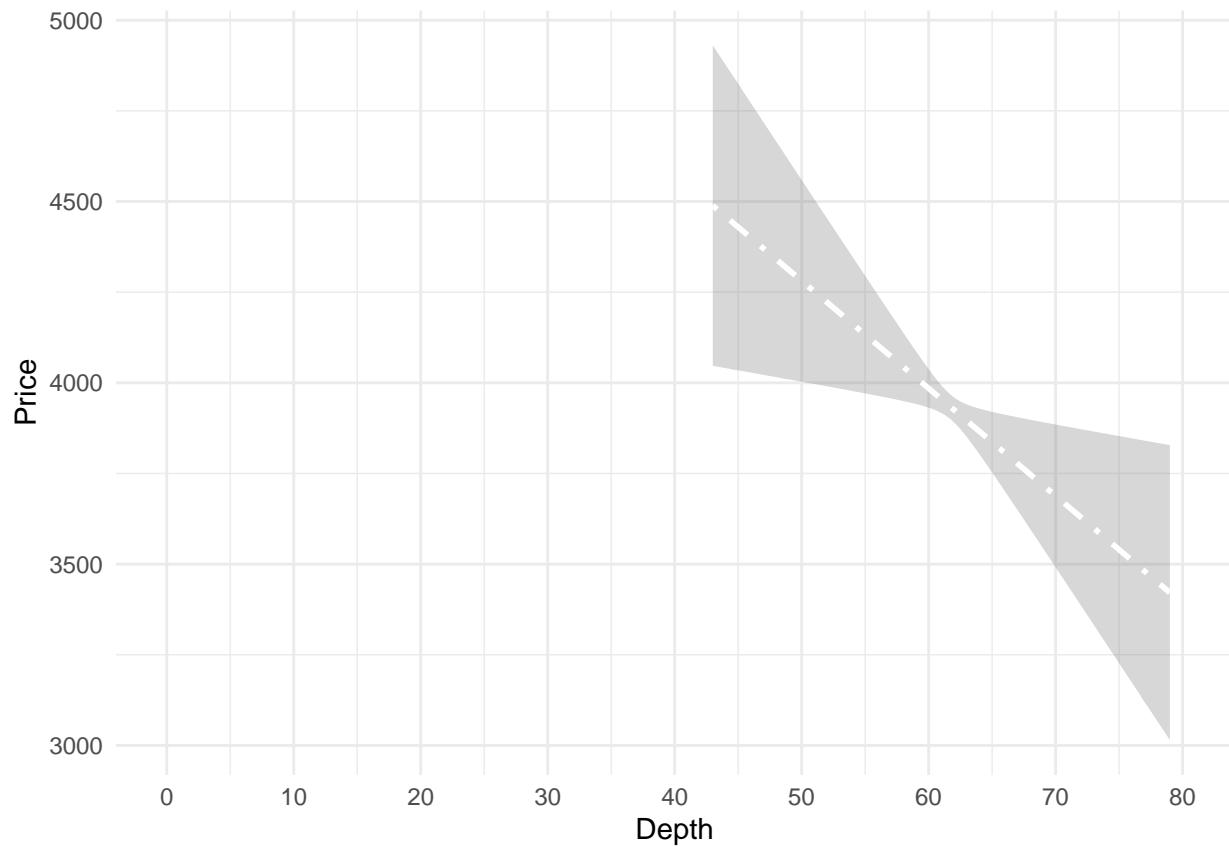
Q3 (5 points)

Next, you want to know whether the price of diamonds depends on table and depth. Note the line types. Recreate the following graphs:

```
ggplot(diam_1, aes(table, price)) +
  geom_smooth(method = "lm",
              formula = y ~ x,
              alpha = 0.3,
              color = "red",
              linetype = "dashed",
              fill = "grey47",
              show.legend = FALSE) +
  scale_x_continuous(breaks = seq(0,100, by = 25),
                     limits = c(0,100)) +
  scale_y_continuous(breaks = seq(0,10000, by = 5000)) +
  labs(x = "Table", y = "Price")
```



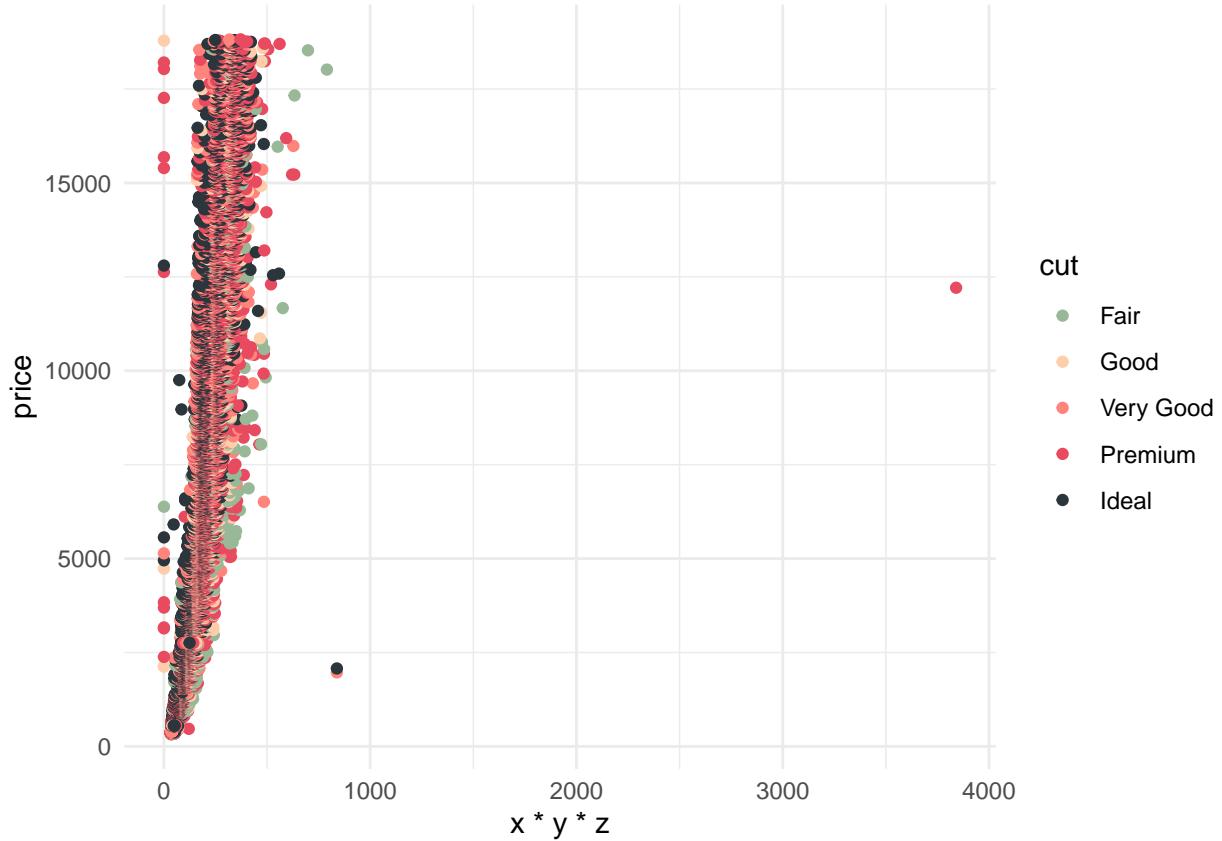
```
ggplot(diam_1, aes(depth, price)) +
  geom_smooth(method = "lm",
              formula = y ~ x,
              alpha = 0.3,
              color = "white",
              linetype = "dotdash",
              fill = "grey47",
              show.legend = FALSE) +
  scale_x_continuous(breaks = seq(0,80, by = 10),
                     limits = c(0,80)) +
  scale_y_continuous(breaks = seq(3000,5000, by = 500)) +
  labs(x = "Depth", y = "Price")
```



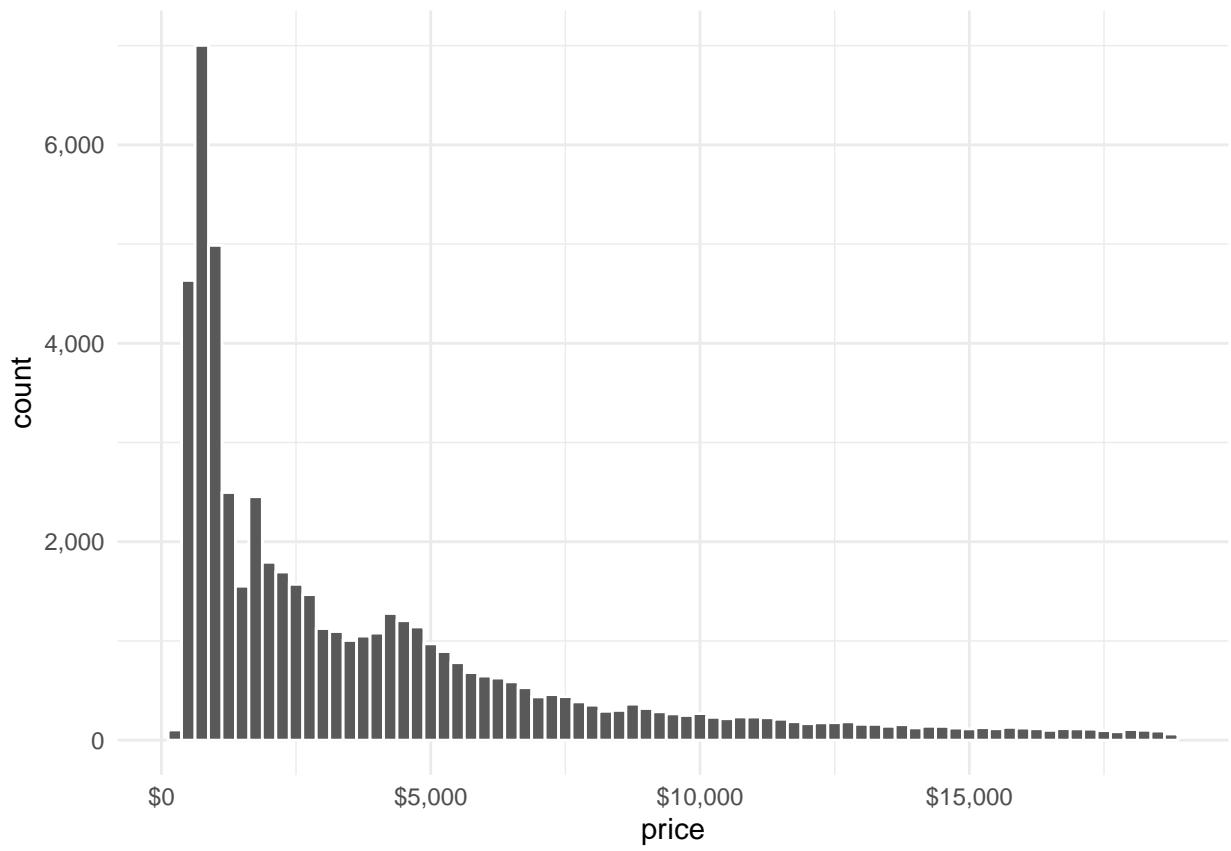
Q4 (5 points)

Recreate each of the following graphs for data exploration:

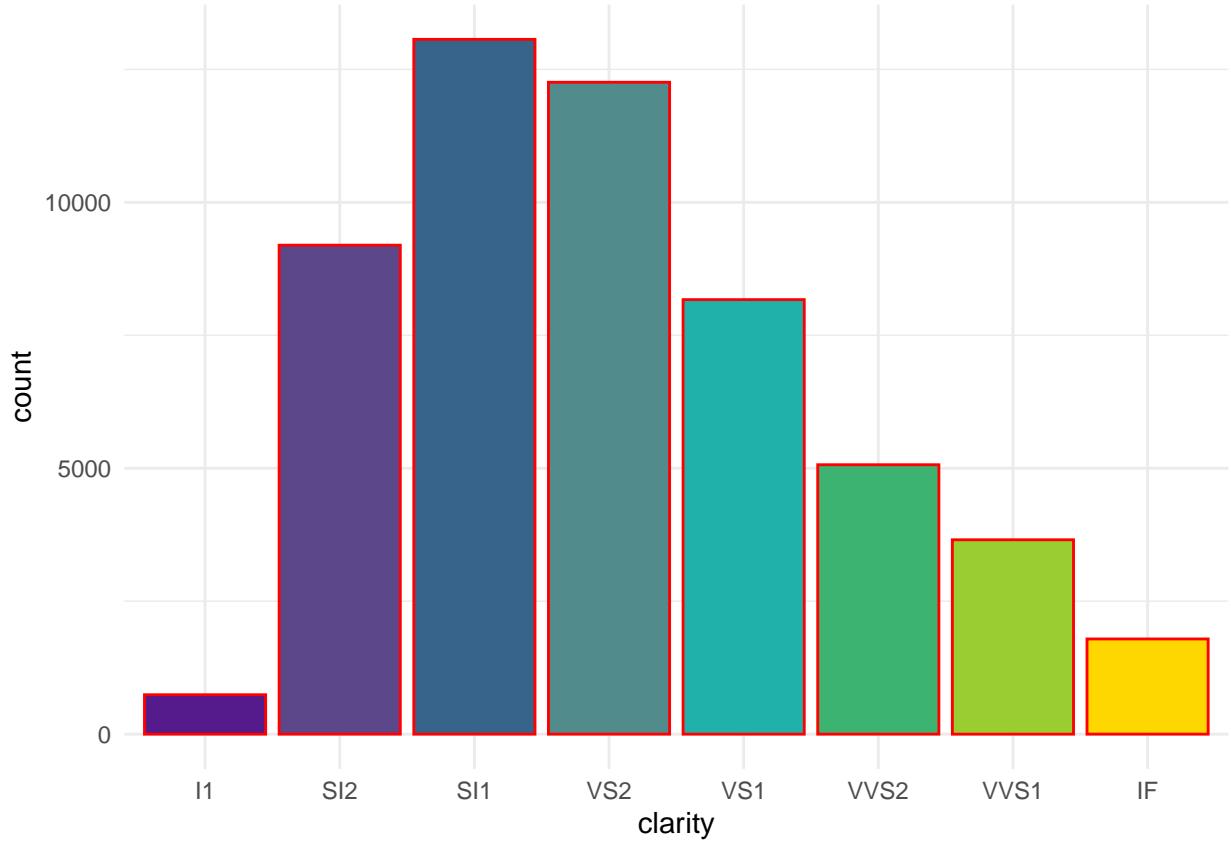
```
ggplot(diam_1, aes(x*y*z, price)) +  
  geom_point(aes(color = cut)) +  
  scale_fill_manual(values = colors_1) +  
  scale_color_manual(values = colors_1) +  
  labs(x = "x * y * z", y = "price")
```



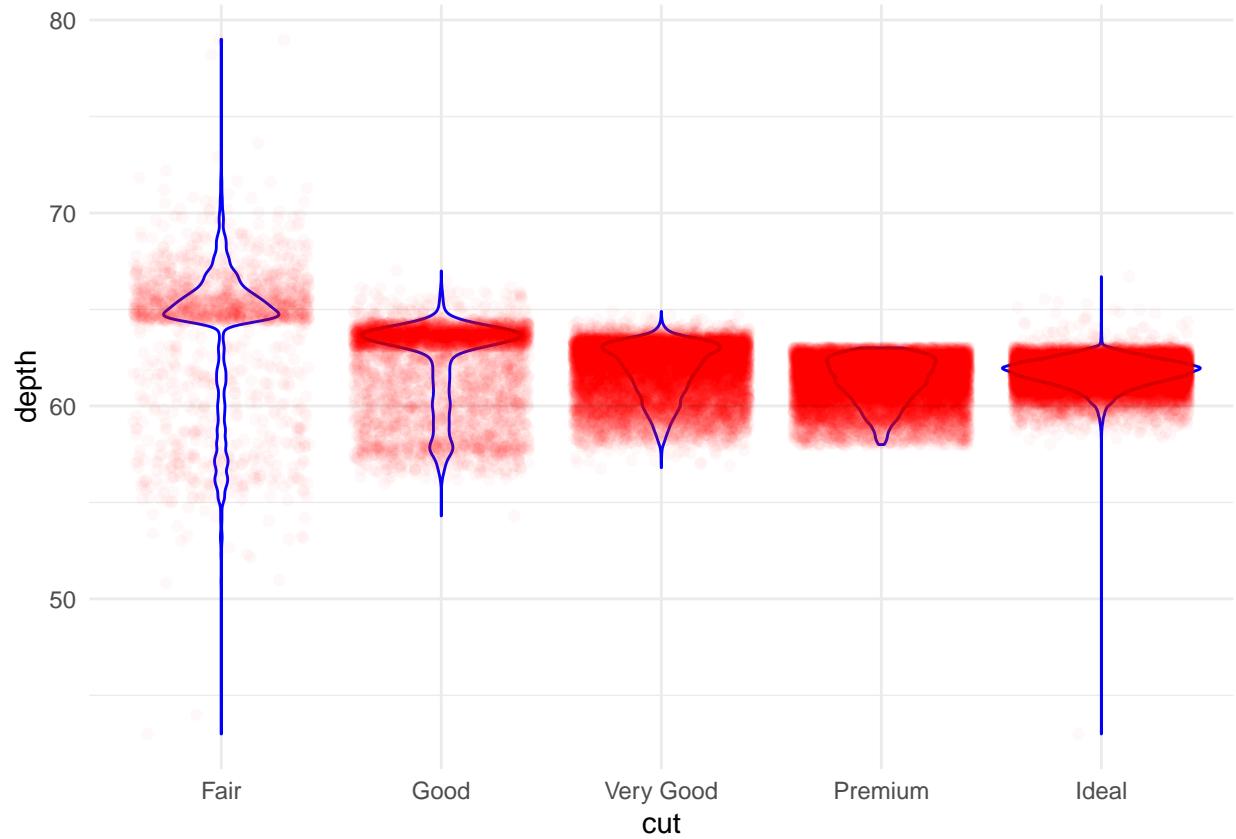
```
ggplot(diam_1, aes(price)) +
  geom_histogram(bins = 75, fill = "grey35", color = "white") +
  scale_x_continuous(labels = scales::dollar_format(prefix = "$"),
                     breaks = seq(0,15000, by = 5000)) +
  scale_y_continuous(labels = scales::comma_format(),
                     breaks = seq(0,7000, by = 2000))
```



```
ggplot(diam_1, aes(clarity)) +  
  geom_bar(fill = colors_2, color = "red") +  
  scale_y_continuous(breaks = seq(0,10000, by = 5000))
```



```
ggplot(diam_1, aes(cut, depth)) +  
  geom_violin(color = "blue") +  
  geom_jitter(color = "red", alpha = 0.03)
```



```
ggplot(diam_1, aes(x, price)) +  
  geom_smooth(method = "lm",  
             color = "green",  
             show.legend = FALSE) +  
  geom_smooth(fill = "white",  
             show.legend = FALSE)
```

