Data Visualization Homework

author: Prommin Vutivivatchai

date: 2023-07-20

This homework contains

- ggplot2
  - histogram
  - bar
  - scatter
  - facet
- ggpubr (additional)

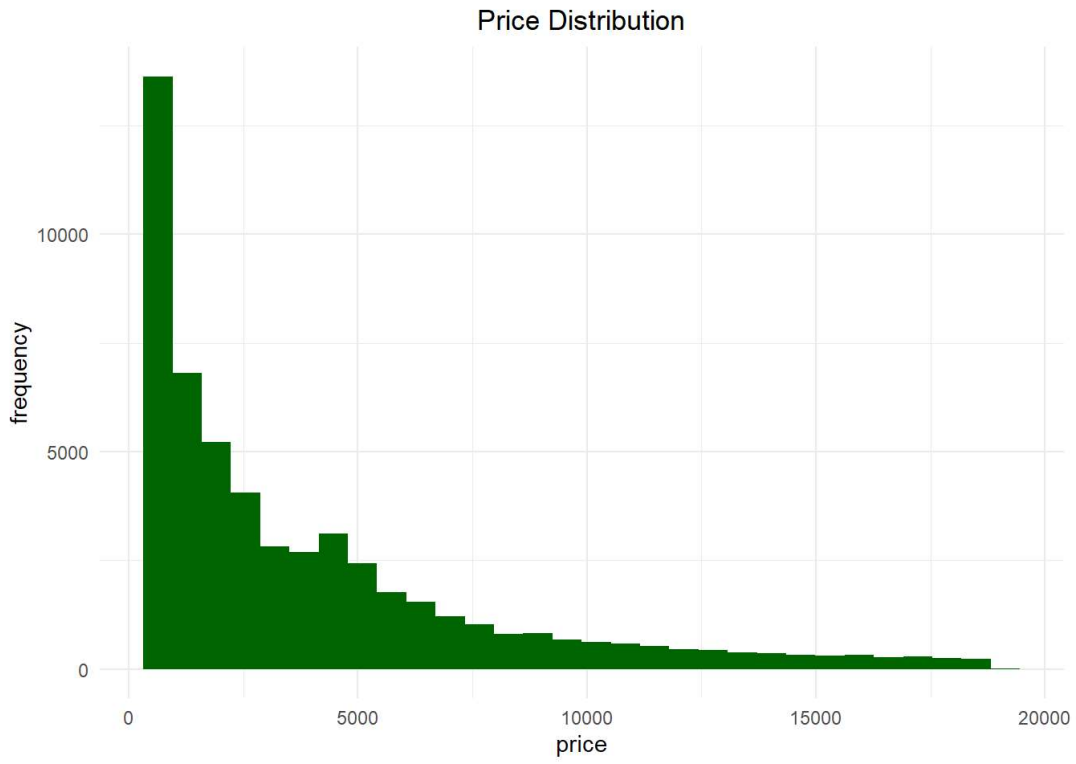
---

0. Load Library

```
library(tidyverse) # ggplot2 included
library(ggpubr) # use for combine multiple of ggplot graphs
```

1. Price Histrogram

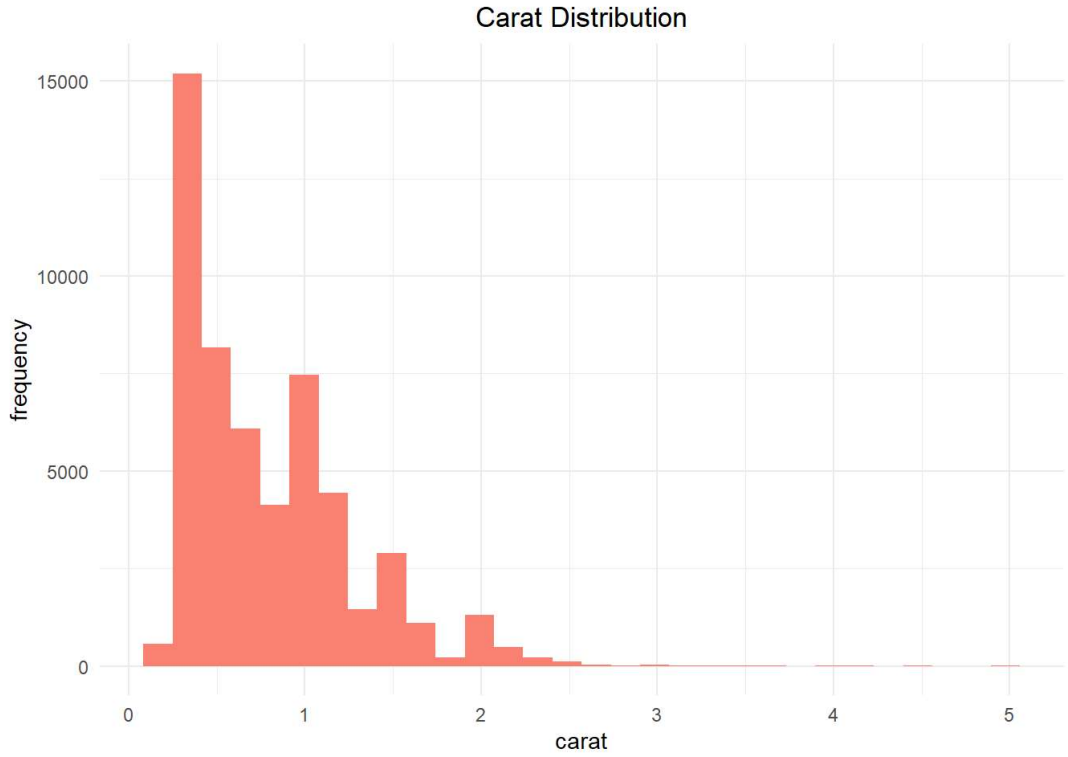
```
ggplot(diamonds, aes(price)) +
  geom_histogram(bins=30, fill="darkgreen") +
  labs(title="Price Distribution", y="frequency")+
  theme_minimal() + # remove background color
  theme(plot.title=element_text(hjust=0.5)) # align title to center
```



- As shown, we found a positively skewed distribution in the price data.

2. Carat Histogram

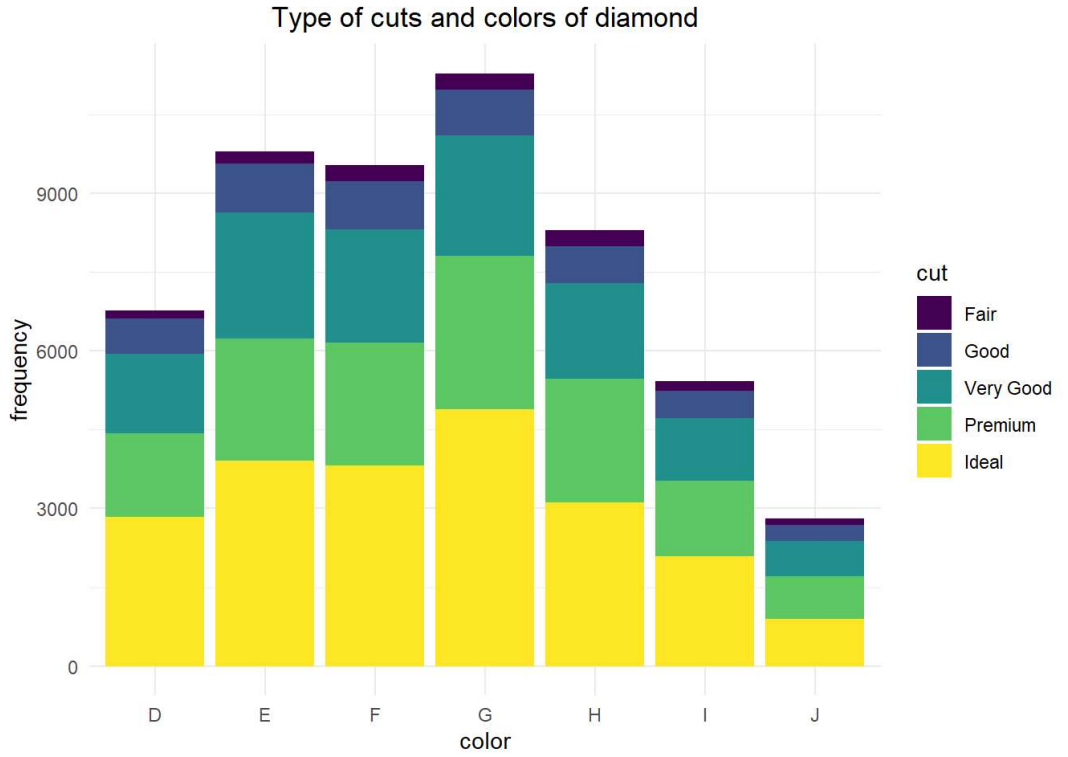
```
ggplot(diamonds, aes(carat)) +
  geom_histogram(bins=30, fill="salmon") +
  labs(title="Carat Distribution", y="frequency")+
  theme_minimal() +
  theme(plot.title=element_text(hjust=0.5)) # align title to center
```



- As shown, we found a positively skewed distribution in the carat data.

3. (Stacked) Bar Chart : Color/Cutting of Diamonds

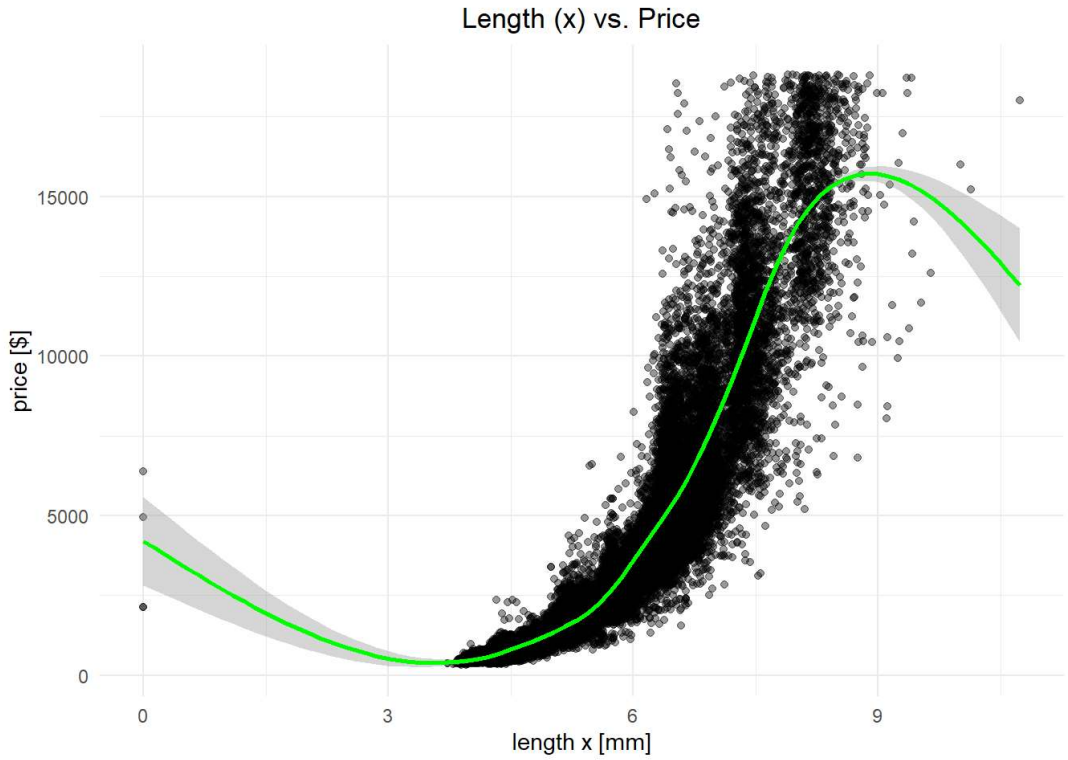
```
ggplot(diamonds, aes(color, fill=cut)) +
  geom_bar() +
  labs(title="Type of cuts and colors of diamond", y="frequency") +
  theme_minimal() +
  theme(plot.title=element_text(hjust=0.5))
```



- We found that the top rank of cutting is ideal cutting regardless of colors.
- Very-Good and Good cutting are quite the same proportion regardless of colors.
- G-color diamonds have the highest frequency, and J-color diamonds have the lowest frequency.

4. Scatter Plot: Relationship between Length (x) and Price

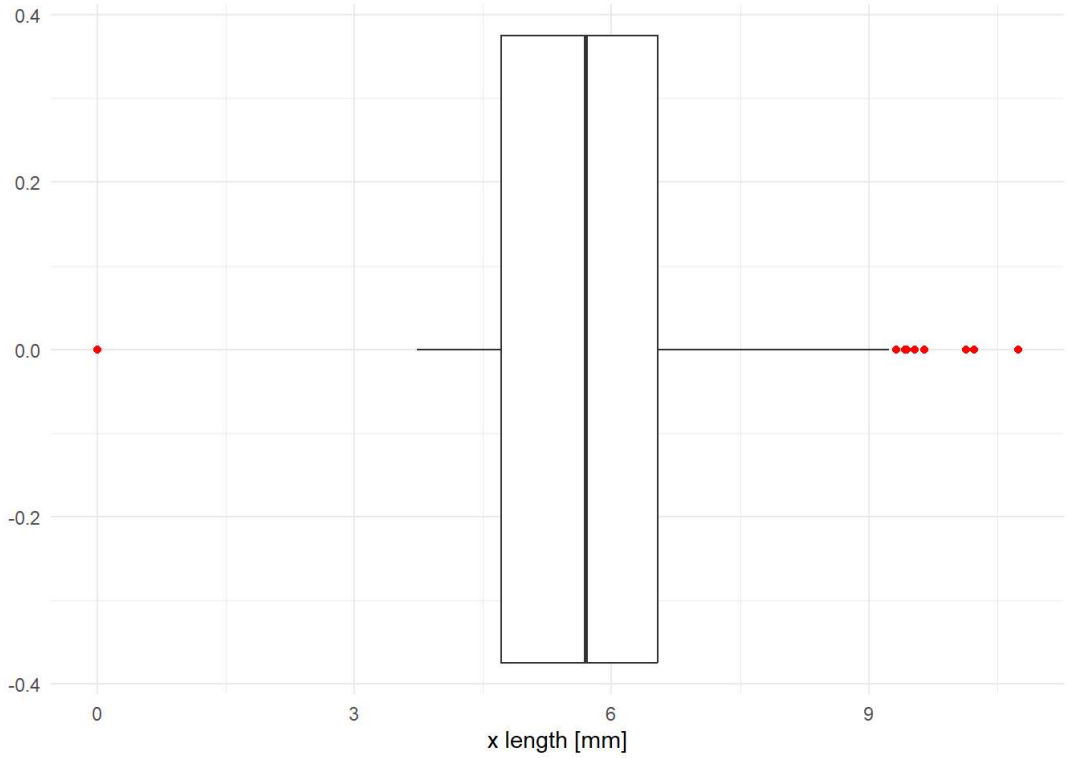
```
ggplot(diamonds %>% sample_frac(0.5), aes(x, price)) +
  geom_point(alpha=0.4) +
  geom_smooth(color="green") +
  labs(title = "Length (x) vs. Price", x = "length x [mm]", y = "price [$]") +
  theme_minimal() +
  theme(plot.title=element_text(hjust=0.5))
```



- Informally, we found outliers however we can deal with it.

4.1 Box Plot (Explore Outliers located out of +- 1.5*IQR)

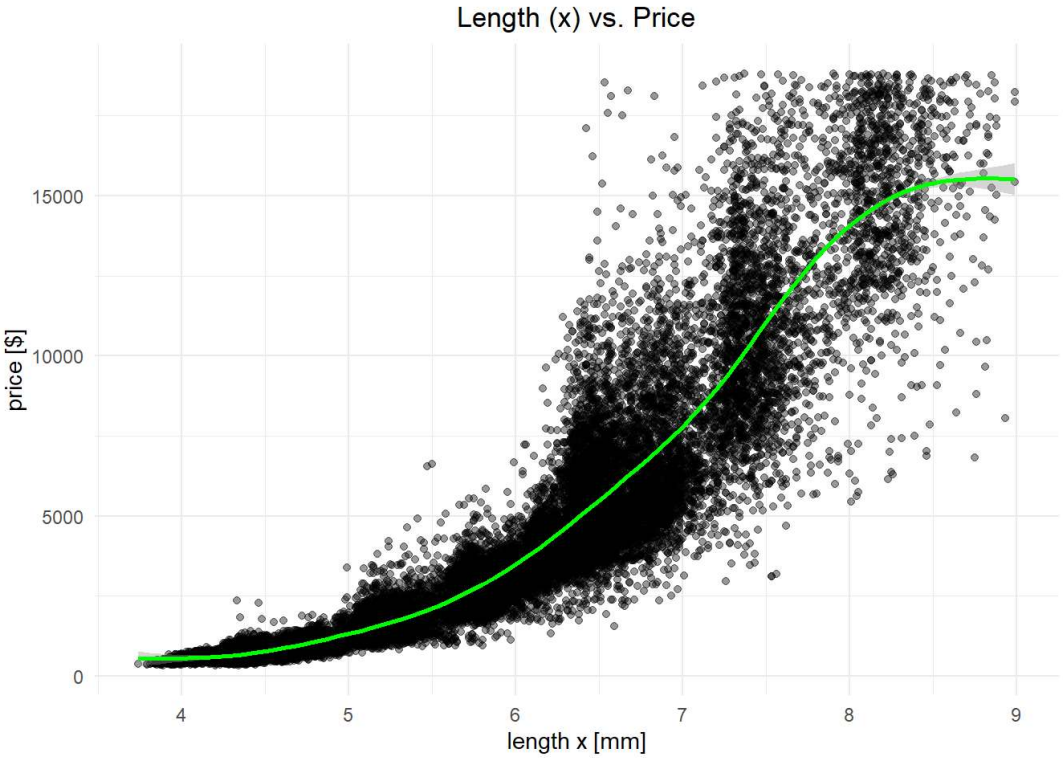
```
ggplot(diamonds %>% sample_frac(0.5), aes(x)) +
  geom_boxplot(outlier.color="red") +
  xlab("x length [mm]") +
  theme_minimal()
```



- Outliers located at x = 0 and x > 9 (approximately)

- There are methods to identify outlier values correctly but we don't mention here.
- We will filter out. See below

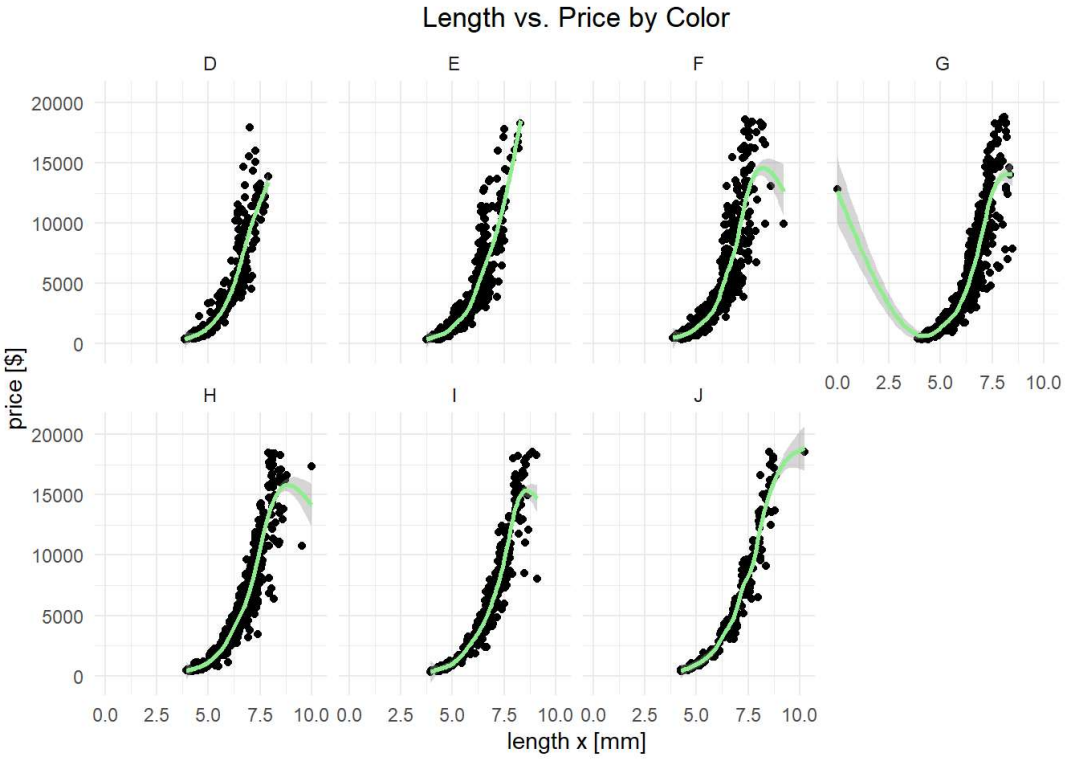
```
ggplot(diamonds %>% sample_frac(0.5) %>% filter(x > 0 & x < 9) ## this line
       , aes(x, price)) +
  geom_point(alpha=0.4) +
  geom_smooth(color="green") +
  labs(title = "Length (x) vs. Price", x = "length x [mm]", y = "price [$]") +
  theme_minimal() +
  theme(plot.title=element_text(hjust=0.5))
```



- It works! But a theoretical investigation is needed.

5. Length vs. Price + by Colors (using facet_wrap)

```
ggplot(diamonds %>% sample_n(5000), aes(x, price)) +
  geom_point() +
  facet_wrap(~color, ncol=4) +
  labs(title="Length vs. Price by Color", x="length x [mm]", y="price [$]") +
  geom_smooth(color="lightgreen") +
  theme_minimal() +
  theme(plot.title=element_text(hjust=0.5))
```



6. Relationship between Price and Dimensions (x,y,z) / showing color data into each point.

- Create each graph (length x to price, width y to price, depth z to price) as below.

```
## create each graphs

# x-price
g1 <- ggplot(diamonds %>% filter(x>2.5) %>% sample_frac(0.5), aes(x, price)) +
  geom_point(aes(color = color), alpha=0.6) +
  theme_minimal() +
  geom_smooth(color="red") +
  xlab("length x [mm]") +
  ylab("price [$]")

# y-price
g2 <- ggplot(diamonds %>% filter(y>2 & y<11) %>% sample_frac(0.5), aes(y, price)) +
  geom_point(aes(color = color), alpha=0.6) +
  theme_minimal() +
  geom_smooth(color="green") +
  xlab("width y [mm]") +
  ylab("price [$]")

# z-price
g3<- ggplot(diamonds %>% filter(z>2 & z<9) %>% sample_frac(0.5), aes(z, price)) +
  geom_point(aes(color = color), alpha=0.6) +
  theme_minimal() +
  geom_smooth(color="blue") +
  xlab("depth z [mm]") +
  ylab("price [$]")
```

- Let's combine them by 'ggpubr' library.

```
library(ggpubr)

combine <- ggarrange(g1,
                     g2,
                     g3,
                     nrow = 3,
                     common.legend=TRUE, # share color legend between 3 graphs
                     legend="right")

combine <- annotate_figure(combine, top=text_grob("Price vs. Dimensions (mm)",
                face = "bold", size = 14))

combine
```