Problem Set: Explore Two Variables

Hugo Jal

2024-06-10

This project is an assignment from the Data Analysis with R course.

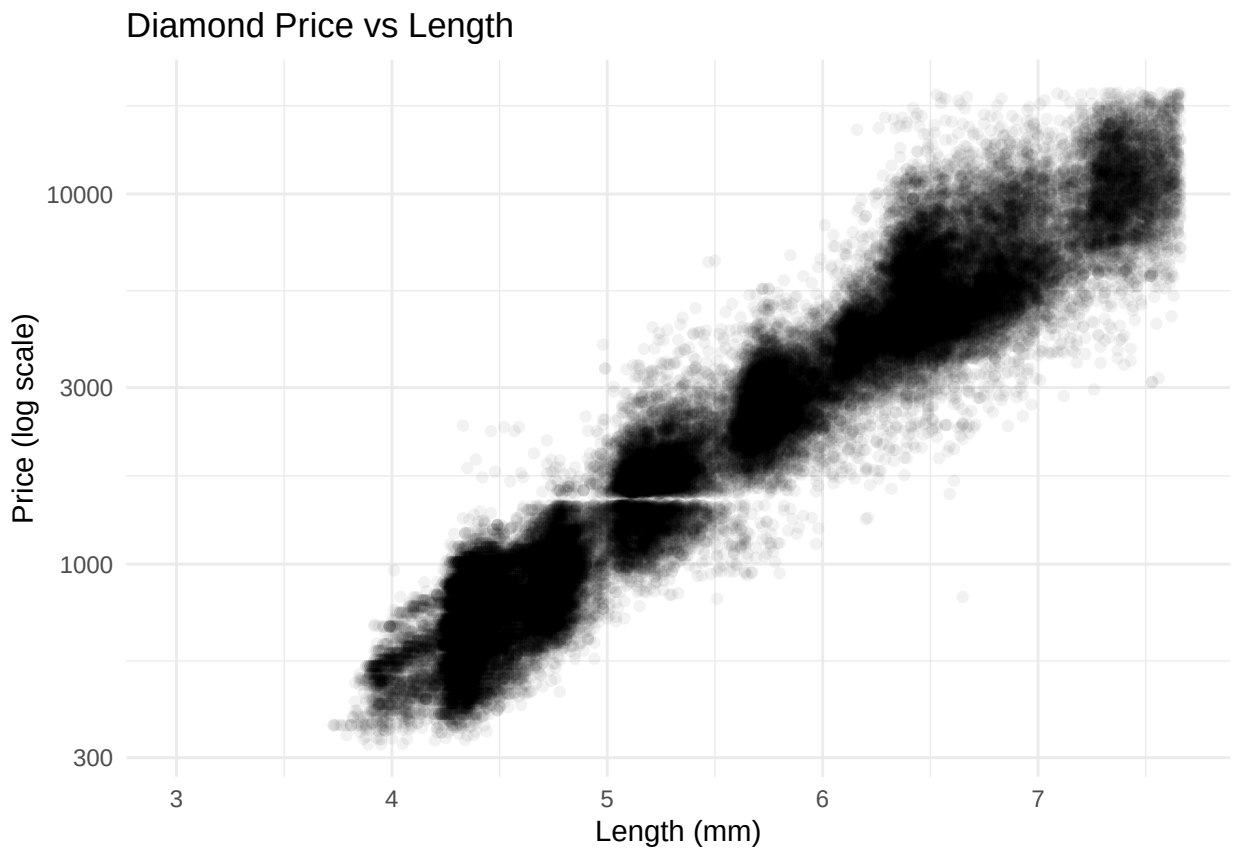
```
library(ggplot2)
data(diamonds)
```

Scatterplots of price v. x

```
p1 <- ggplot(data = diamonds, aes(x = x, y = price)) +
  geom_point(alpha = 1/20) +
  scale_y_log10() +
  scale_x_continuous(limits = c(3, quantile(diamonds$x, 0.95))) +
  labs(x = "Length (mm)", y = "Price (log scale)", title = "Diamond Price vs Length") +
  theme_minimal()

print(p1)
```

```
## Warning: Removed 2702 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Observations: By implementing a log scale to the price, we can clearly see the relationship between the length of the diamond (x) and the price. Prior to the transformation, an exponential increase was displayed as opposed to the apparently linear increase of the current plot.

```
p2 <-
  ggplot(data = diamonds, aes(x = x, y = price, color = cut)) +
  geom_point(alpha = 1/20) +
  scale_y_log10() +
  scale_x_continuous(limits = c(3, quantile(diamonds$x, 0.95))) +
  scale_color_viridis_d() +
  labs(x = "Length (mm)", y = "Price (log scale)", title = "Diamond Price vs Length by Cut") +
  theme_minimal()

print(p2)
```

```
## Warning: Removed 2702 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


```
p3 <-
  ggplot(data = diamonds, aes(x = x, y = price)) +
  geom_point(alpha = 1/80) +
  scale_y_log10() +
  scale_x_continuous(limits = c(3, quantile(diamonds$x, 0.95))) +
  facet_wrap(~cut) +
  labs(x = "Length (mm)", y = "Price (log scale)", title = "Diamond Price vs Length by Cut")

print(p3)
```

```
## Warning: Removed 2702 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Correlations

```
cor.test(diamonds$price, diamonds$x,
         method = 'pearson')
```

```
##
##  Pearson's product-moment correlation
##
## data:  diamonds$price and diamonds$x
## t = 440.16, df = 53938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8825835 0.8862594
## sample estimates:
##       cor
## 0.8844352
```

```
cor.test(diamonds$price, diamonds$y,
         method = 'pearson')
```

```
##
##  Pearson's product-moment correlation
##
## data:  diamonds$price and diamonds$y
## t = 401.14, df = 53938, p-value < 2.2e-16
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8632867 0.8675241
## sample estimates:
##       cor
## 0.8654209
```

```
cor.test(diamonds$price, diamonds$z,
         method = 'pearson')
```

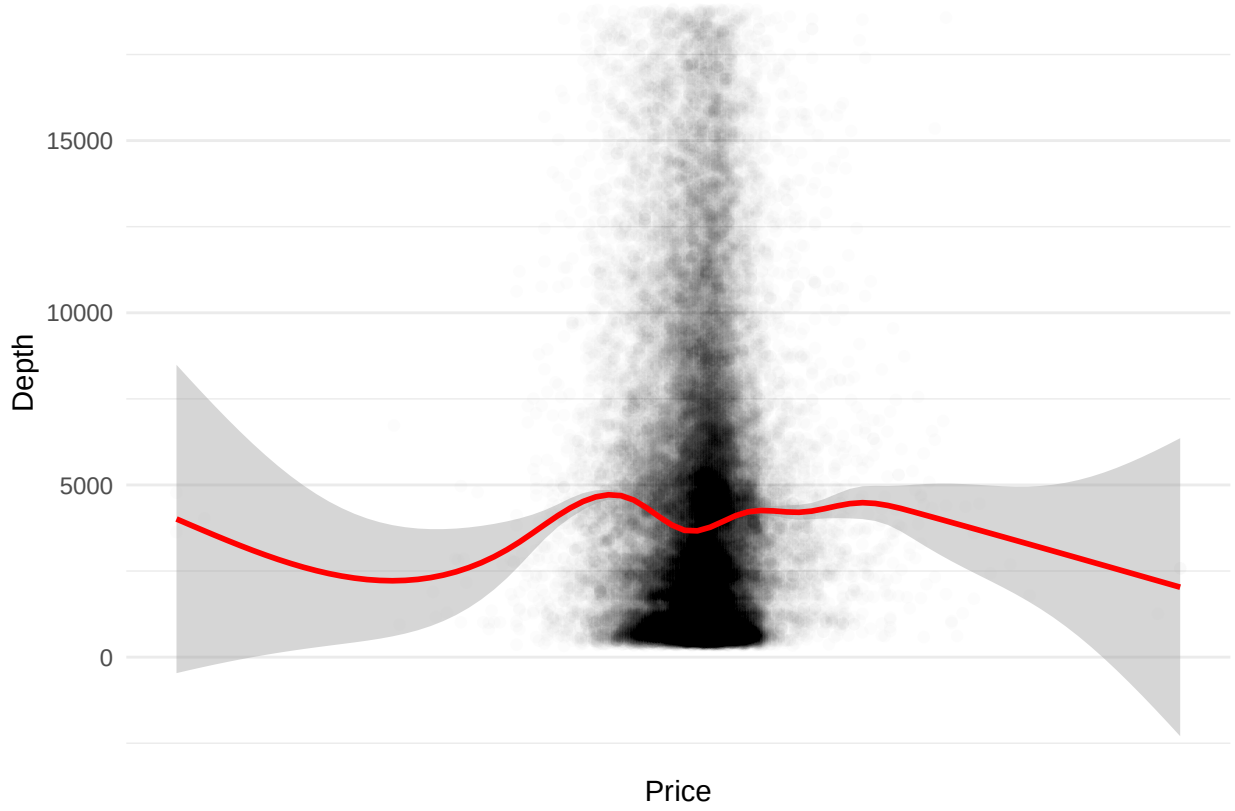
```
##
##  Pearson's product-moment correlation
##
## data:  diamonds$price and diamonds$z
## t = 393.6, df = 53938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8590541 0.8634131
## sample estimates:
##       cor
## 0.8612494
```

Scatterplots of price v. depth

```
p4 <-
  ggplot(data = diamonds, aes(x = depth, y = price)) +
  geom_point(alpha = 1/100) +
  geom_smooth(color = "red") +
  scale_x_continuous(breaks = seq(326, 18823, 2)) +
  labs(x = "Price", y = "Depth") +
  theme_minimal()


print(p4)
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



Correlation of price vs. depth

```
cor.test(diamonds$price, diamonds$depth,
         method = 'pearson')
```

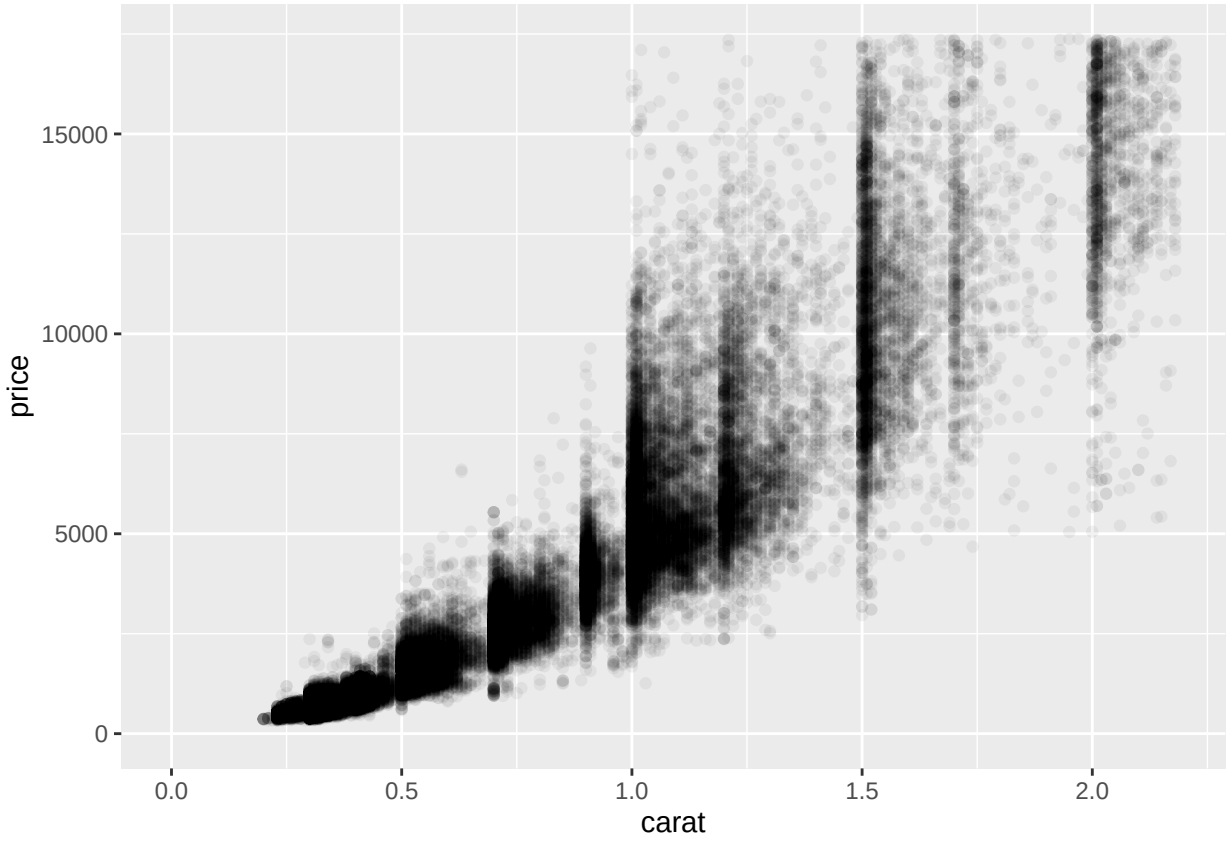
```
##
##  Pearson's product-moment correlation
##
## data:  diamonds$price and diamonds$depth
## t = -2.473, df = 53938, p-value = 0.0134
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.019084756 -0.002208537
## sample estimates:
##        cor
## -0.0106474
```

Scatterplot of price vs. carat

```
p5 <-
  ggplot(data = diamonds, aes(x = carat, y = price)) +
  geom_point(alpha = 1/20) +
  scale_x_continuous(limits = c(0, quantile(diamonds$carat, 0.99))) +
  scale_y_continuous(limits = c(0, quantile(diamonds$price, 0.99)))

print(p5)
```

```
## Warning: Removed 926 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



Scatterplot of price vs. volume

```
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
volume <- diamonds$x * diamonds$y * diamonds$z

diamonds <- diamonds %>%
  mutate(volume)
```

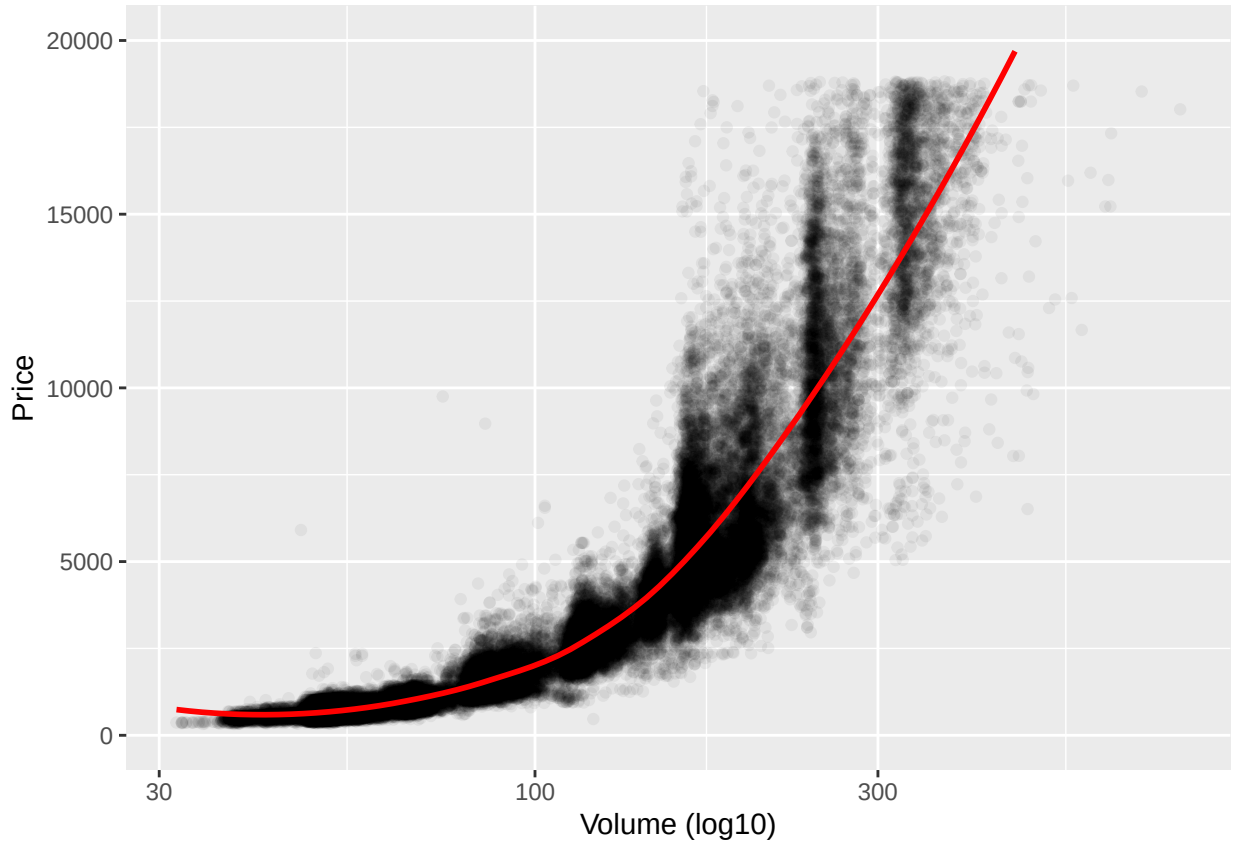
```r
diamonds <- diamonds %>%
  filter(diamonds$volume > 0 & diamonds$volume <= 800)

p6 <-
  ggplot(data = diamonds, aes(x = volume, y = price)) +
  geom_point(alpha = 1/20) +
  scale_x_log10() +
  scale_y_continuous(limits = c(0, 20000)) +
  labs(x = 'Volume (log10)', y = 'Price') +
  geom_smooth(method = 'loess', color = 'red', se = FALSE)

print(p6)
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 13 rows containing missing values or values outside the scale range
## (`geom_smooth()`).
```



Correlation of price vs. volume

```r
diamonds <- diamonds %>%
  filter(diamonds$volume > 0 & diamonds$volume <= 800)

cor.test(diamonds$price, diamonds$volume,
         method = 'pearson')
```

```
##
##  Pearson's product-moment correlation
##
## data:  diamonds$price and diamonds$volume
## t = 559.19, df = 53915, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9222944 0.9247772
## sample estimates:
##       cor
## 0.9235455
```

Mean Price by Clarity

```
# Use dyplr to create new data frame
library(dplyr)

# Clarity
clarity_groups <- group_by(diamonds, clarity)

diamondsByClarity <- summarise(clarity_groups,
                               mean_price = mean(price),
                               median_price = median(price),
                               min_price = min(price),
                               max_price = max(price),
                               n = n())

head(diamondsByClarity)
```

```
## # A tibble: 6 x 6
##   clarity mean_price median_price min_price max_price     n
##   <ord>        <dbl>        <dbl>     <int>     <int> <int>
## 1 I1           3926.         3346       345     18531   738
## 2 SI2          5060.         4072       326     18804  9184
## 3 SI1          3994.         2822       326     18818 13063
## 4 VS2          3923.         2052       334     18823 12254
## 5 VS1          3840.         2005       327     18795  8168
## 6 VVS2         3284.         1311       336     18768  5066
```

```
# Color
color_groups <- group_by(diamonds, color)

diamondsByColor <- summarise(color_groups,
                               mean_price = mean(price),
                               median_price = median(price),
                               min_price = min(price),
                               max_price = max(price),
                               n = n())

head(diamondsByColor)
```

```
## # A tibble: 6 x 6
##   color mean_price median_price min_price max_price     n
```

```
##   <ord>      <dbl>        <dbl>     <int>     <int> <int>
## 1 D          3168.        1836.       357     18693  6774
## 2 E          3077.        1738        326     18731  9795
## 3 F          3725.        2344.       342     18791  9538
## 4 G          3997.        2240        354     18818 11284
## 5 H          4480.        3453        337     18803  8297
## 6 I          5090.        3730        334     18823  5421
```

Bar Charts of Mean Price

```
library(gridExtra)
```

```
##
## Adjuntando el paquete: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(ggplot2)
library(viridis)
```

```
## Warning: package 'viridis' was built under R version 4.4.1
```

```
## Cargando paquete requerido: viridisLite
```

```
p7 <-
  ggplot(data = diamondsByClarity, aes(x = factor(clarity), y = mean_price)) +
  geom_col(aes(fill = clarity)) +
  scale_fill_viridis_d(option = "A") +
  labs(x = 'Clarity', y = 'Mean Price', title = 'Mean Price by Clarity') +
  theme_minimal()

p8 <-
  ggplot(data = diamondsByColor, aes(x = factor(color), y = mean_price)) +
  geom_col(aes(fill = color)) +
  scale_fill_viridis_d(option = "D") +
  labs(x = 'Color', y = 'Mean Price', title = 'Mean Price by Color') +
  theme_minimal()

p9 <- grid.arrange(grobs = list(p7, p8), nrow = 1, ncol = 2)
```


```
print(p9)
```

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```