Data Analysis with R

Problem Set 2

Pramod Duvvuri

4/8/2019

Diamonds Data

```
suppressMessages(library(tidyverse))
data(diamonds)
```

```
nrow(diamonds)
```

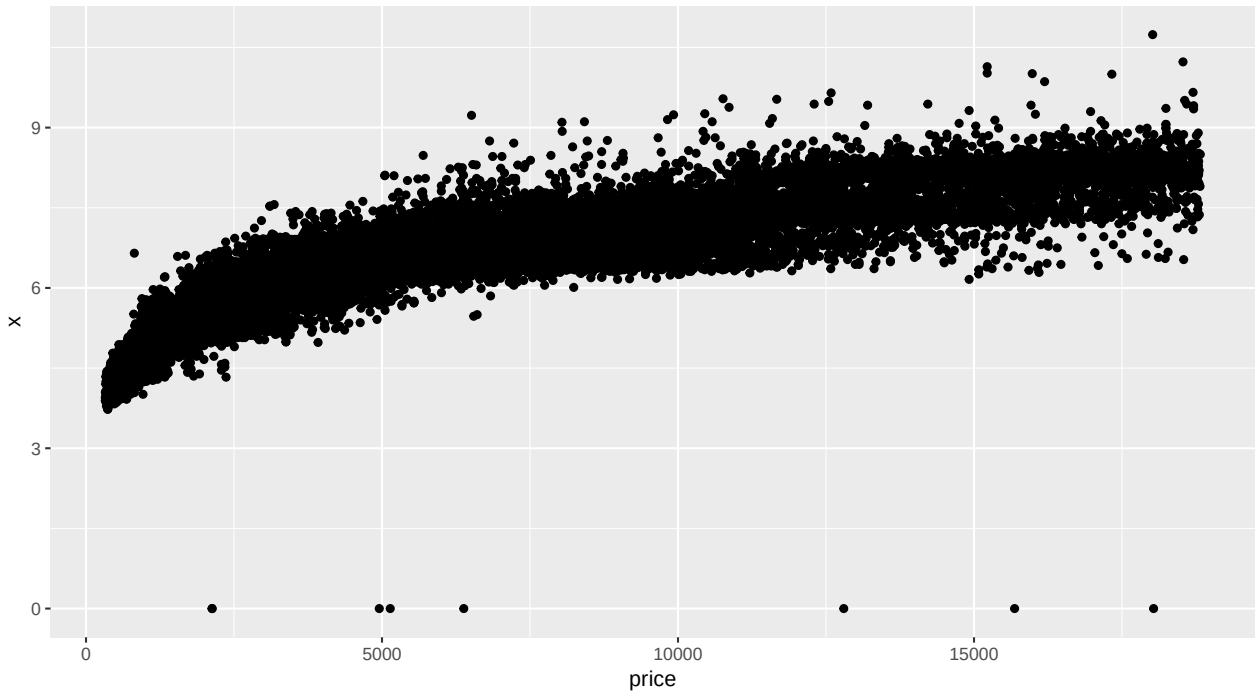
```
## [1] 53940
```

```
summary(diamonds)
```

```
##      carat               cut        color        clarity
##  Min.   :0.2000   Fair      : 1610   D: 6775   SI1    :13065
##  1st Qu.:0.4000   Good      : 4906   E: 9797   VS2    :12258
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194
##  Mean   :0.7979   Premium   :13791   G:11292   VS1    : 8171
##  3rd Qu.:1.0400   Ideal     :21551   H: 8304   VVS2   : 5066
##  Max.   :5.0100                     I: 5422   VVS1   : 3655
##                                     J: 2808   (Other): 2531
##      depth           table           price             x
##  Min.   :43.00   Min.   :43.00   Min.   :  326   Min.   : 0.000
##  1st Qu.:61.00   1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710
##  Median :61.80   Median :57.00   Median : 2401   Median : 5.700
##  Mean   :61.75   Mean   :57.46   Mean   : 3933   Mean   : 5.731
##  3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540
##  Max.   :79.00   Max.   :95.00   Max.   :18823   Max.   :10.740
##
##        y                z
##  Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 4.720   1st Qu.: 2.910
##  Median : 5.710   Median : 3.530
##  Mean   : 5.735   Mean   : 3.539
##  3rd Qu.: 6.540   3rd Qu.: 4.040
##  Max.   :58.900   Max.   :31.800
##
```

```
?diamonds
```

```
ggplot(data = diamonds, mapping = aes(x = price, y = x)) +
  geom_point()
```




```
# Correlation between price and x
cor.test(diamonds$price, diamonds$x)
```

```
##
##  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
```

```
# Correlation between price and y
cor.test(diamonds$price, diamonds$y)
```

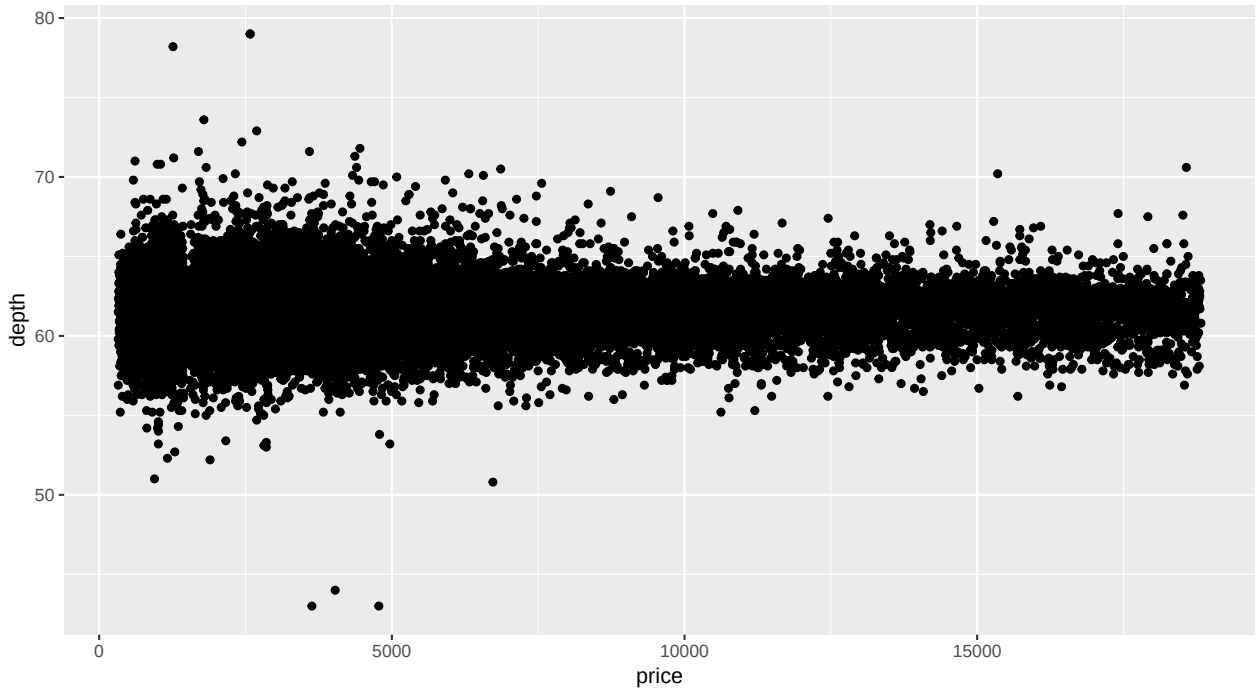
```
##
##  Pearson's product-moment correlation
##
## data:  diamonds$price and diamonds$y
## t = 401.14, df = 53938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8632867 0.8675241
## sample estimates:
##       cor
## 0.8654209
```

```
# Correlation between price and z
cor.test(diamonds$price, diamonds$z)
```

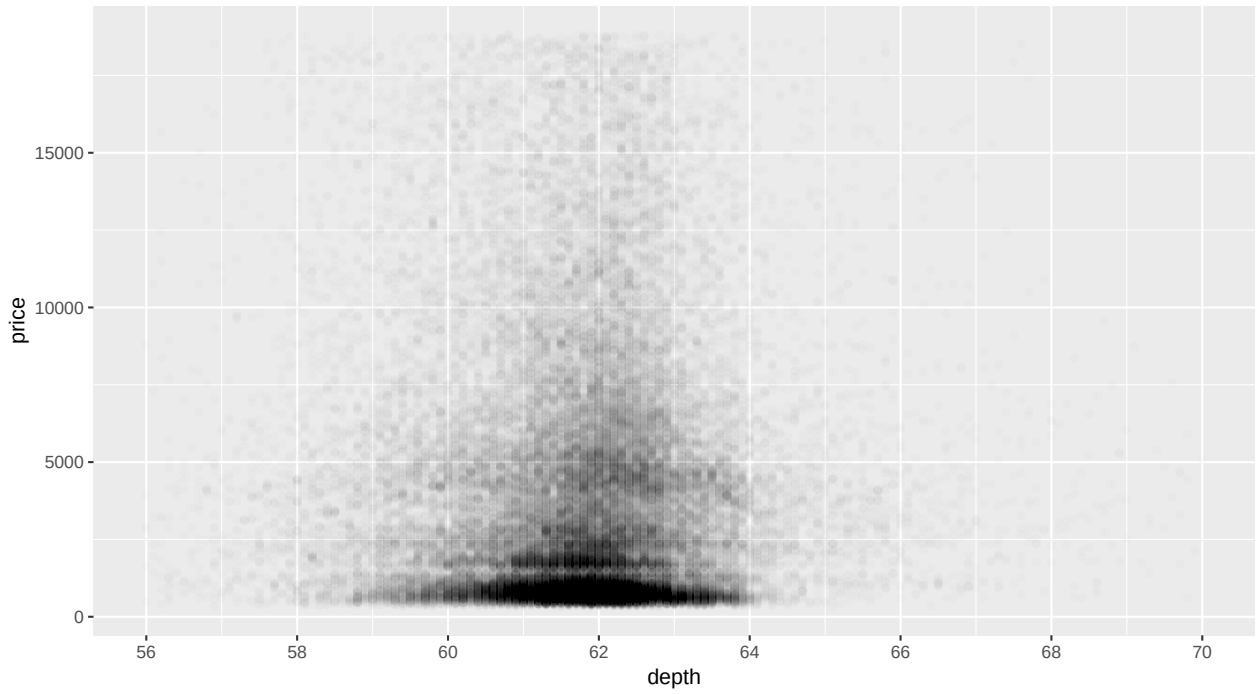
```
##
```

```
##  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
```

```
# Price vs Depth
ggplot(data = diamonds, mapping = aes(x = price, y = depth)) +
  geom_point()
```




```
ggplot(data = diamonds, aes(x = depth, y = price)) +
  geom_point(alpha = 1/100, na.rm = TRUE) +
  scale_x_continuous(limits = c(56,70), breaks = seq(50,70,2))
```



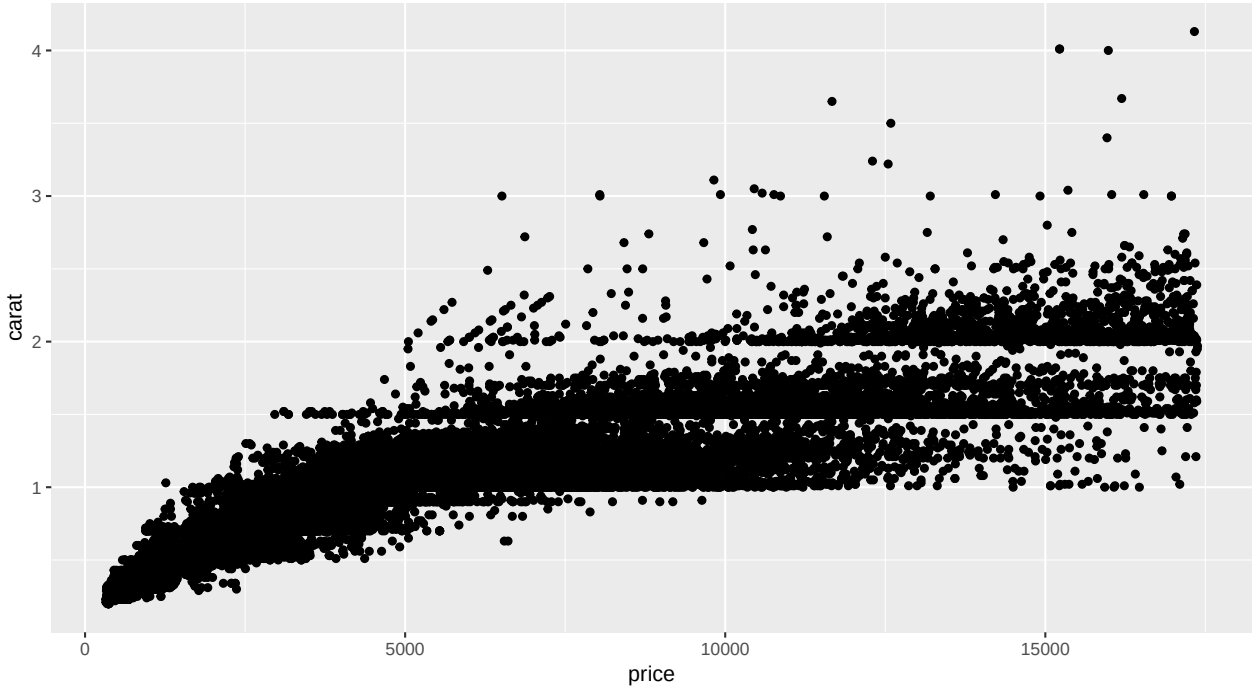

```
cor.test(diamonds$depth, diamonds$price)
```

```
##
##  Pearson's product-moment correlation
##
## data:  diamonds$depth and diamonds$price
## 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
```

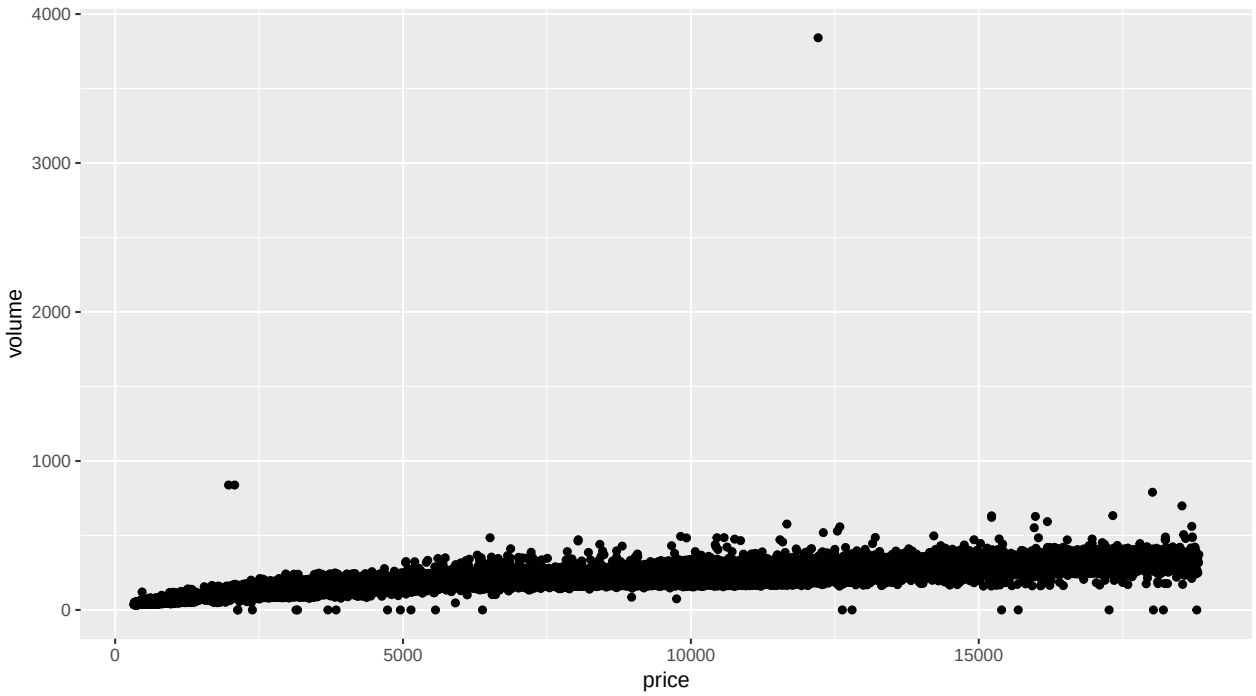
```
# Omitting top 1% in Price
quantile(diamonds$price, probs = 0.99)
```

```
##      99%
## 17378.22
```

```
ggplot(data = subset(diamonds, diamonds$price < 17378.22),
            mapping = aes(x = price, y = carat)) + geom_point()
```




```
# Price vs Volume
diamonds$volume = diamonds$x * diamonds$y * diamonds$z
ggplot(data = diamonds,
             mapping = aes(x = price, y = volume)) +
  geom_point()
```




```
table(diamonds$volume == 0)
```

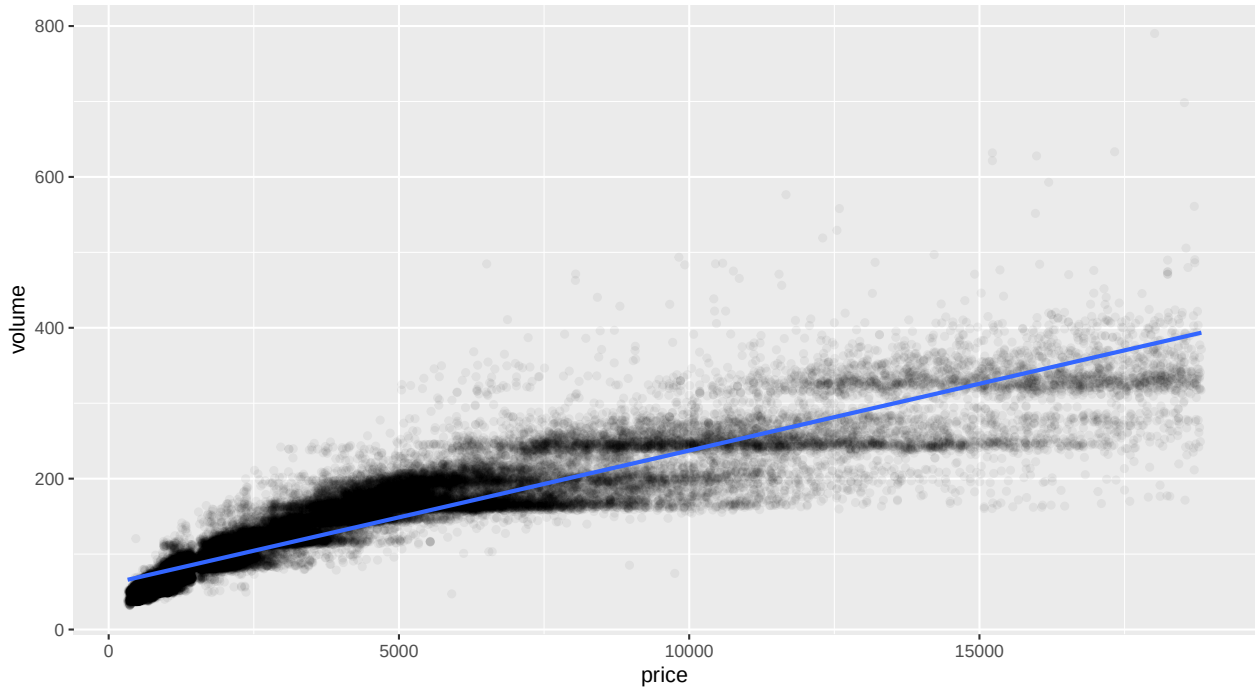
```
##
## FALSE  TRUE
```

```
## 53920    20
```

```
diamonds_subset <- subset(diamonds, diamonds$volume > 0 & diamonds$volume <= 800)
cor(diamonds_subset$price, diamonds_subset$volume)
```

```
## [1] 0.9235455
```

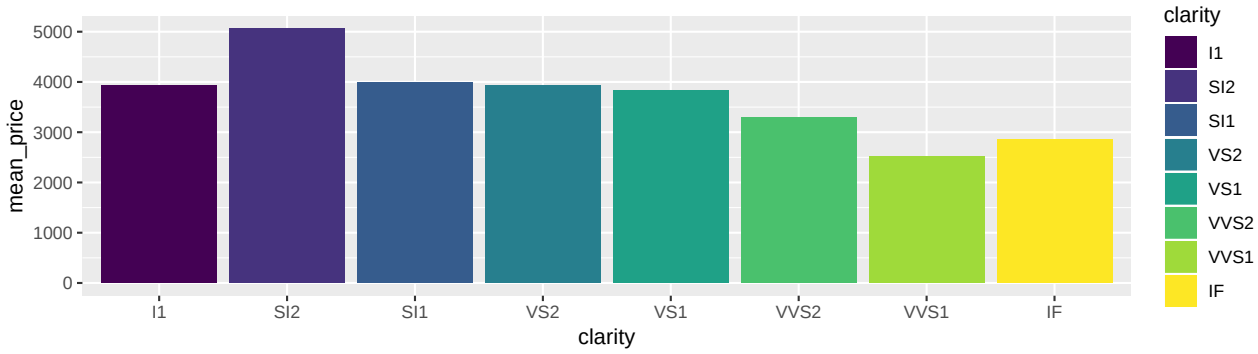
```
ggplot(data = diamonds_subset, mapping = aes(x = price, y = volume)) +
  geom_point(alpha = 1/20) +
  geom_smooth(method = 'lm')
```

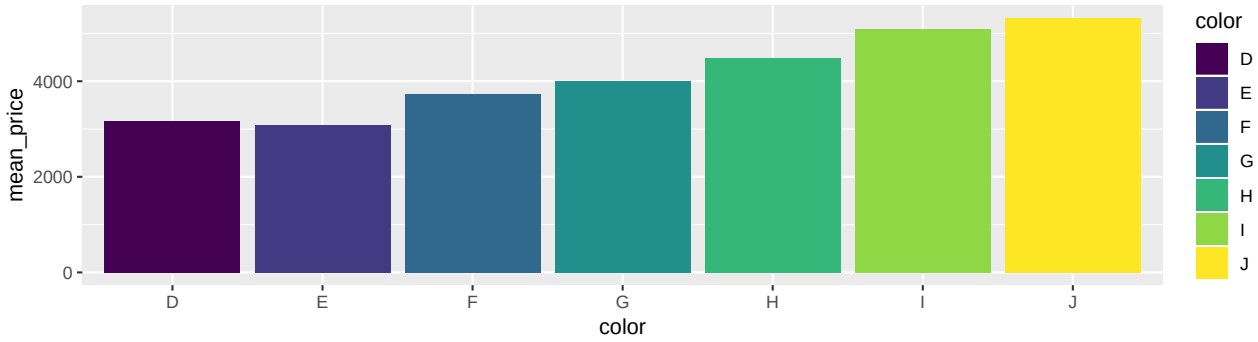



```
diamondsByClarity <- diamonds %>%
  group_by(clarity) %>%
  summarise(mean_price = mean(price),
            median_price = median(price),
            min_price = min(price),
            max_price = max(price),
            n = n()) %>%
   arrange(n)
```

```
diamonds_by_clarity <- group_by(diamonds, clarity)
diamonds_mp_by_clarity <- summarise(diamonds_by_clarity, mean_price = mean(price))

diamonds_by_color <- group_by(diamonds, color)
diamonds_mp_by_color <- summarise(diamonds_by_color, mean_price = mean(price))
```

```
p1 <- ggplot(data = arrange(diamonds_mp_by_clarity, mean_price)) +
  geom_col(mapping = aes(x = clarity, y = mean_price, fill = clarity))

p2 <- ggplot(data = arrange(diamonds_mp_by_color, mean_price)) +
  geom_col(mapping = aes(x = color, y = mean_price, fill = color))

gridExtra::grid.arrange(p1,p2, ncol = 1)
```

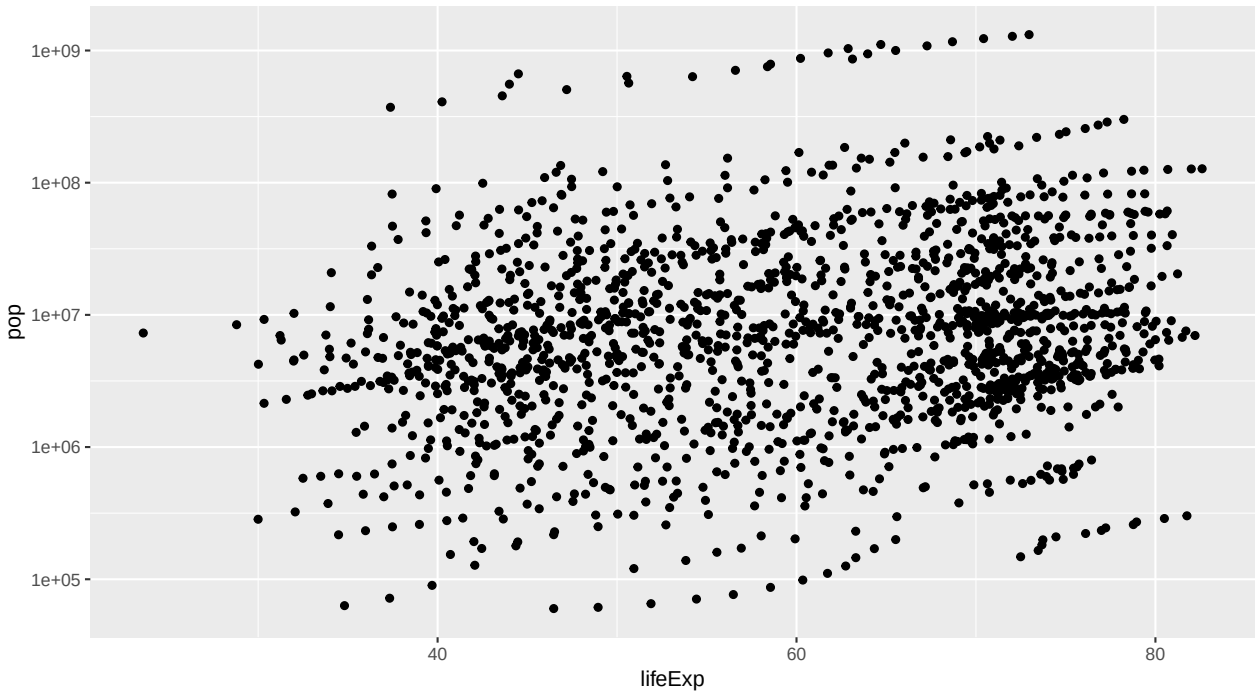





```
#install.packages('gapminder')
library(gapminder)
data("gapminder")
summary(gapminder)
```

```
##         country        continent        year         lifeExp
##  Afghanistan:  12   Africa  :624   Min.   :1952   Min.   :23.60
##  Albania    :  12   Americas:300   1st Qu.:1966   1st Qu.:48.20
##  Algeria    :  12   Asia    :396   Median :1980   Median :60.71
##  Angola     :  12   Europe  :360   Mean   :1980   Mean   :59.47
##  Argentina  :  12   Oceania : 24   3rd Qu.:1993   3rd Qu.:70.85
##  Australia  :  12                  Max.   :2007   Max.   :82.60
##  (Other)    :1632
##       pop              gdpPercap
##  Min.   :6.001e+04   Min.   :   241.2
##  1st Qu.:2.794e+06   1st Qu.:  1202.1
##  Median :7.024e+06   Median :  3531.8
##  Mean   :2.960e+07   Mean   :  7215.3
##  3rd Qu.:1.959e+07   3rd Qu.:  9325.5
##  Max.   :1.319e+09   Max.   :113523.1
##
```

```
ggplot(data = gapminder, mapping = aes(x = lifeExp, y = pop)) +
  geom_point() +
  scale_y_log10()
```




```
cor.test(gapminder$pop, gapminder$lifeExp)
```

```
##
##  Pearson's product-moment correlation
##
## data:  gapminder$pop and gapminder$lifeExp
## t = 2.6854, df = 1702, p-value = 0.007314
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.01752303 0.11209600
## sample estimates:
##       cor
## 0.06495537
```