Diamond Prices Part 2

Audrey Ekuban

6 March 2016

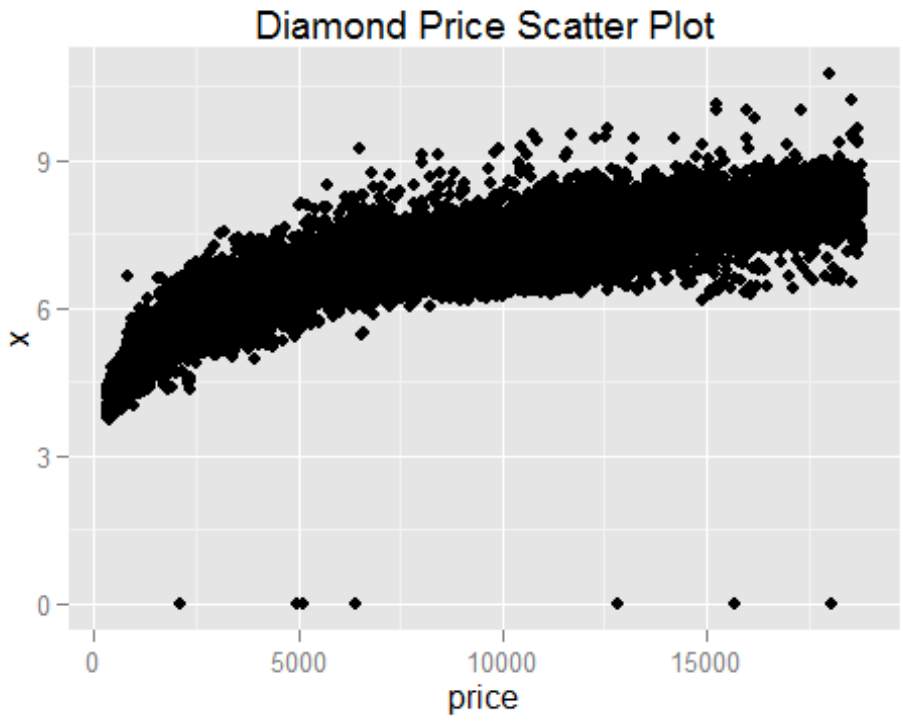
```
library(ggplot2)
data("diamonds")
summary(diamonds)
```

```
##      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
##
```

```
str(diamonds)
```

```
## 'data.frame':    53940 obs. of  10 variables:
##  $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
##  $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3
...
##  $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5
...
##  $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4
5 ...
##  $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
```

```
##  $ x       : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y       : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z       : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...

#?diamonds

ggplot(aes(x = price, y = x), data = diamonds) +
  geom_point() +
  ggtitle("Diamond Price Scatter Plot")
```



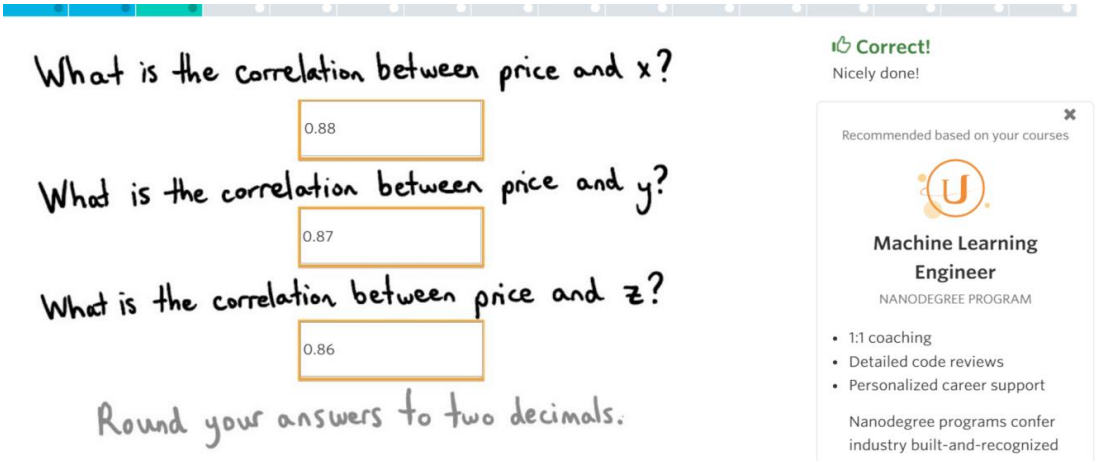
There appears to be a positive correlation and an exponentional relationship between price and x.There are some outliers (7)

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

##
##  Pearson's product-moment correlation
##
## data:  diamonds$x and diamonds$price
## 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$y, diamonds$price)
```

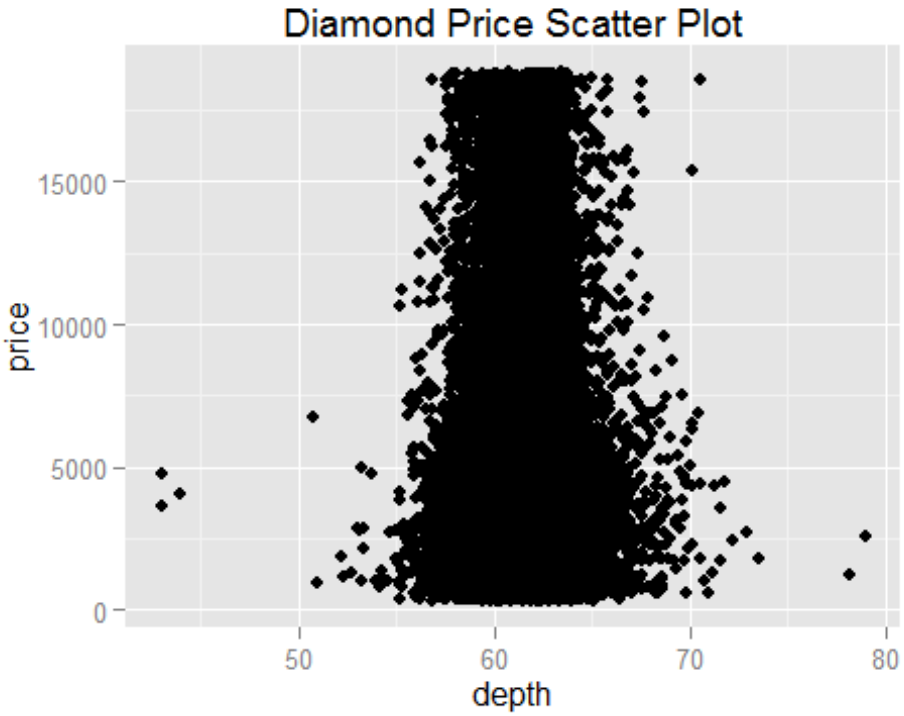
```
##
##  Pearson's product-moment correlation
##
## data:  diamonds$y and diamonds$price
## t = 401.14, df = 53938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8632867 0.8675241
## sample estimates:
##       cor
## 0.8654209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

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

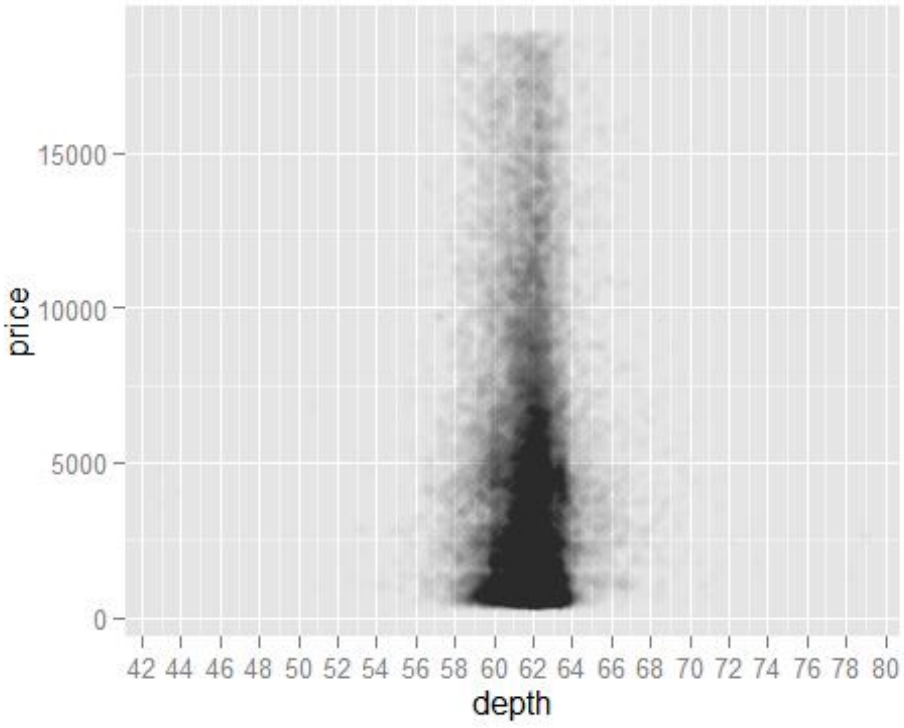
```
##
##  Pearson's product-moment correlation
##
## data:  diamonds$z and diamonds$price
## t = 393.6, df = 53938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8590541 0.8634131
## sample estimates:
##       cor
## 0.8612494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




```
ggplot(aes(x = depth, y = price), data = diamonds) +
  geom_point() +
  ggtitle("Diamond Price Scatter Plot")
```




```
ggplot(data = diamonds, aes(x = depth, y = price)) +
  geom_point(alpha = 1/100) +
  scale_x_continuous(breaks =  seq(40,80,2), labels =  seq(40,80,2))
```



Based on the scatterplot of depth vs. price, most diamonds are between what values of depth?

58 to 64

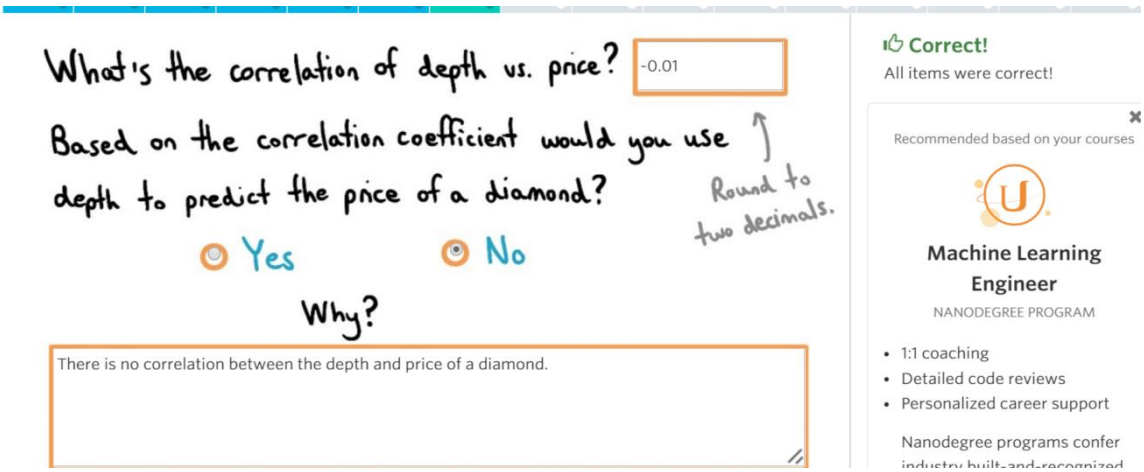
lower limit (a few numbers will work)

upper limit (a few numbers will work)

Correct!
Excellent!

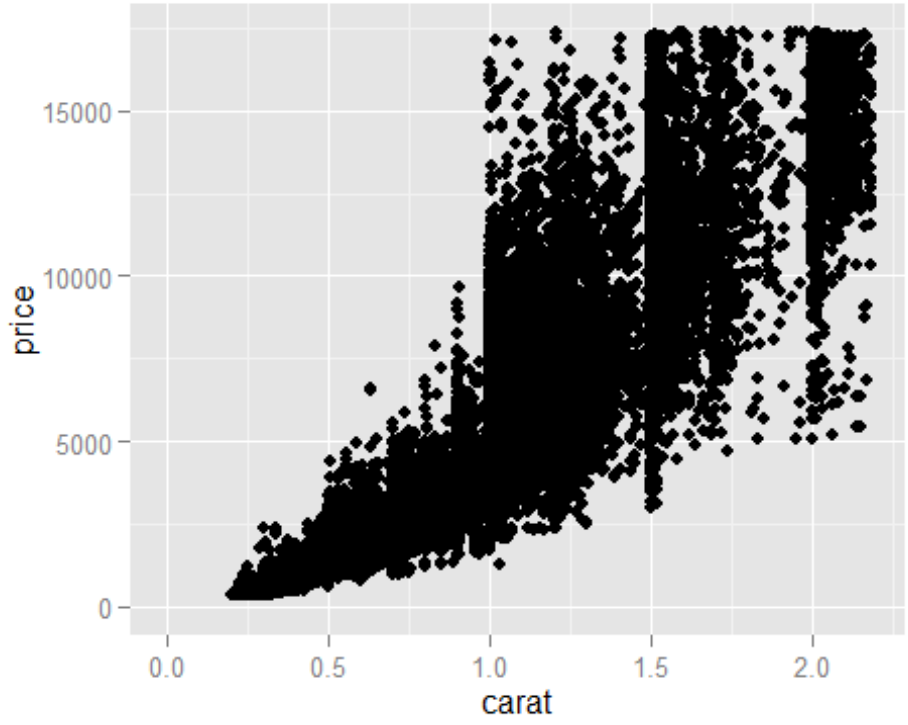

```
cor.test(diamonds$depth, diamonds$price)
```

```
##
##  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
```




```
ggplot(data = diamonds,aes(x = carat, y = price)) +
  xlim(0,quantile(diamonds$carat,0.99)) +
  ylim(0,quantile(diamonds$price,0.99)) +
  geom_point()
```

```
## Warning: Removed 926 rows containing missing values (geom_point).
```




```
diamonds$volume = diamonds$x*diamonds$y*diamonds$z
str(diamonds)
```

```
## 'data.frame':    53940 obs. of  11 variables:
##  $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
##  $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3
```

```
...
##  $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5
...
##  $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4
5 ...
##  $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
##  $ volume : num  38.2 34.5 38.1 46.7 51.9 ...
```

```
# There are some outliers. Some diamonds have a valume of 0.
length(which(diamonds$volume == 0))
```

```
## [1] 20
```

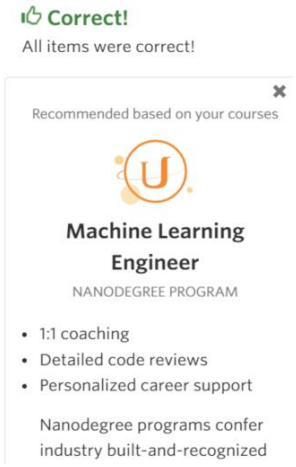
```
diamonds.set = subset(diamonds, volume != 0 & volume < 800 )
cor.test(diamonds.set$volume, diamonds.set$price)
```

```
##
##  Pearson's product-moment correlation
##
## data:  diamonds.set$volume and diamonds.set$price
## 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
```

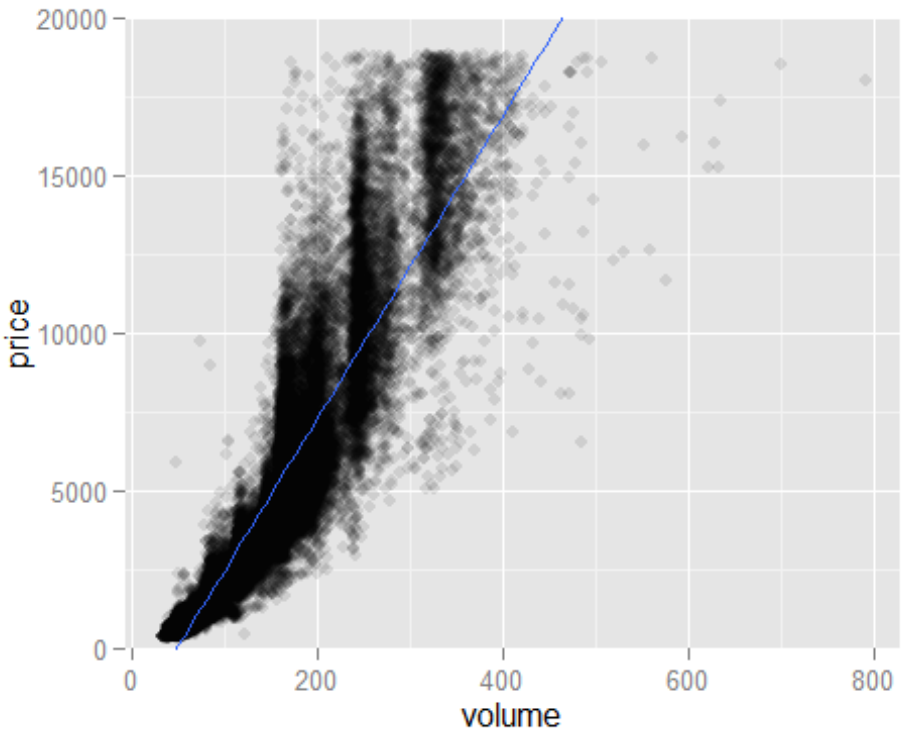
What's the correlation of price and volume?
Exclude diamonds that have a volume of 0 or that are greater than or equal to 800.

0.92 ← round to two decimals

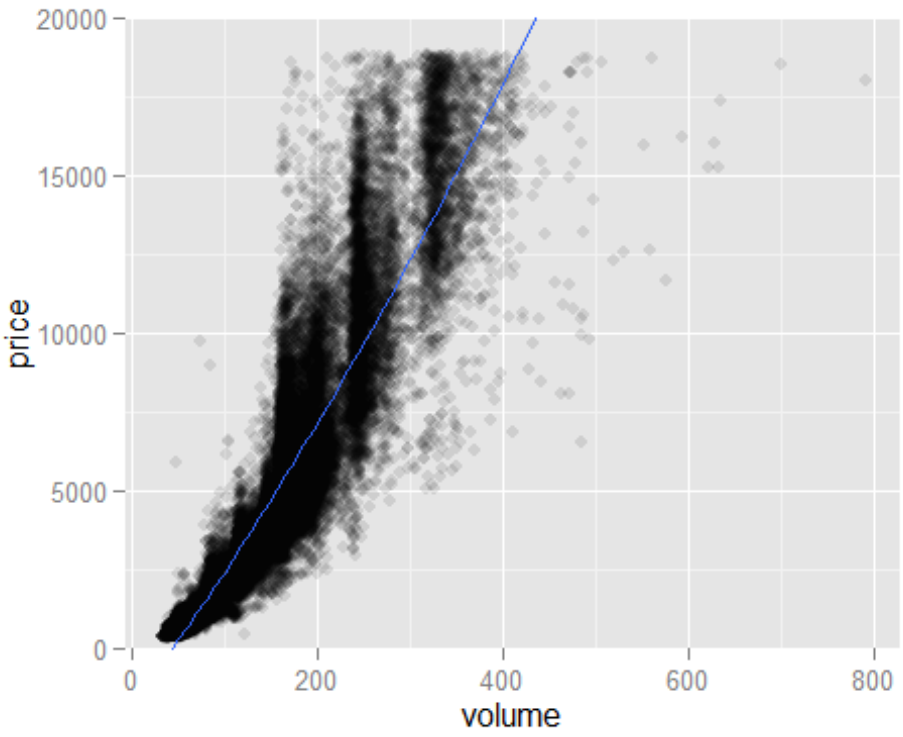
See the Instructor Notes for two hints.




```
ggplot(diamonds.set, aes(x = volume, y = price)) +
  geom_point(alpha = 0.10) +
  geom_smooth(method = "lm") +
  coord_cartesian(ylim = c(0,20000))
```




```
# Looking at polynimoal functions of order 2
ggplot(diamonds.set, aes(x = volume, y = price)) +
  geom_point(alpha = 0.10) +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2)) +
  coord_cartesian(ylim = c(0,20000))
```



In the absence of another model, probably yes due to the correlation. However, there does appear to be a lot of random scattering which suggests that there may be alternative models.

```
suppressMessages(library(ggplot2))
suppressMessages(library(dplyr))
data(diamonds)

diamondsByClarity = group_by(diamonds,clarity) %>%
                                summarise(
                                mean_price = mean(price),
                                median_price = median(as.numeric(price)),
                                min_price = min(price),
                                max_price = max(price),
                                n = n())
head(diamondsByClarity)
```

```
## Source: local data frame [6 x 6]
##
##   clarity mean_price median_price min_price max_price     n
##    (fctr)      (dbl)        (dbl)     (int)     (int) (int)
## 1      I1   3924.169         3344       345     18531   741
## 2     SI2   5063.029         4072       326     18804  9194
## 3     SI1   3996.001         2822       326     18818 13065
## 4     VS2   3924.989         2054       334     18823 12258
## 5     VS1   3839.455         2005       327     18795  8171
## 6    VVS2   3283.737         1311       336     18768  5066
```

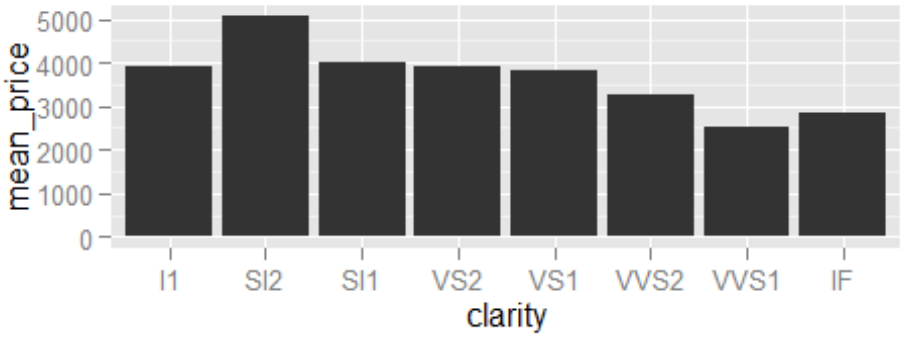
```r
data(diamonds)
library(dplyr)

#install.packages("gridExtra")
library(gridExtra)
```

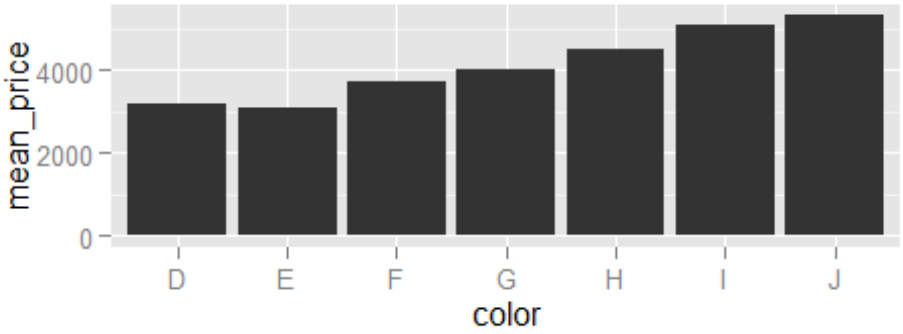
```
## Warning: package 'gridExtra' was built under R version 3.2.4
```

```
##
## Attaching package: 'gridExtra'
```

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

```r
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))

plot1 = ggplot(aes(x = clarity, y = mean_price), data =
diamonds_mp_by_clarity) +
        geom_bar(stat = "identity")

plot2 = ggplot(aes(x = color, y = mean_price), data = diamonds_mp_by_color) +
        geom_bar(stat = "identity")

grid.arrange(plot1,plot2, ncol = 1)
```





Mean price increases with color.