Lesson 4 - Problems Solution

Mario Bonilla

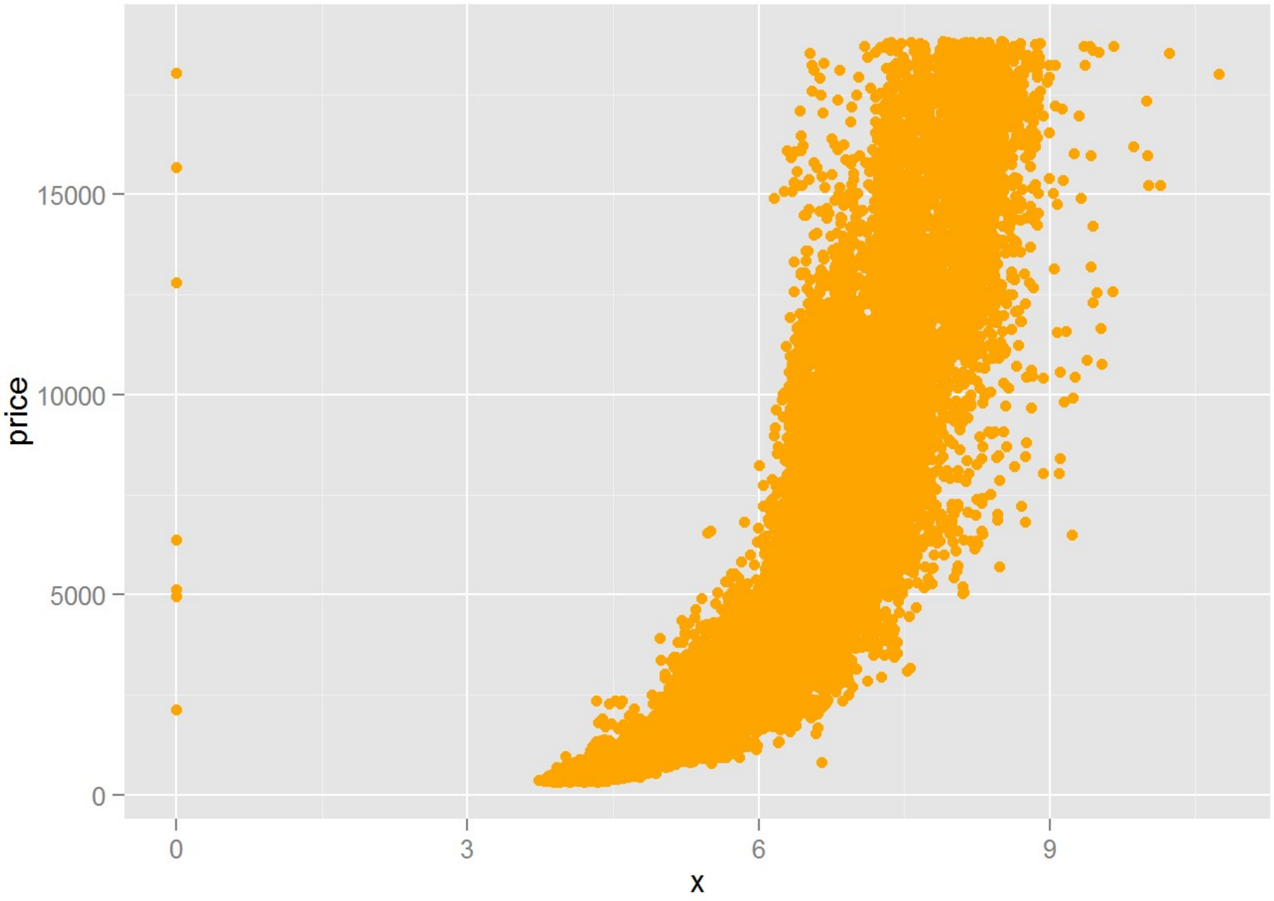
February 4, 2016

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
# diamonds data set.

# create a scatterplot of price vs x.
# using the ggplot syntax.

library(ggplot2)
ggplot(aes(x = x, y = price), data = diamonds) + geom_point(color = "orange")
```



```
# correlations price vs x, y, z
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
```

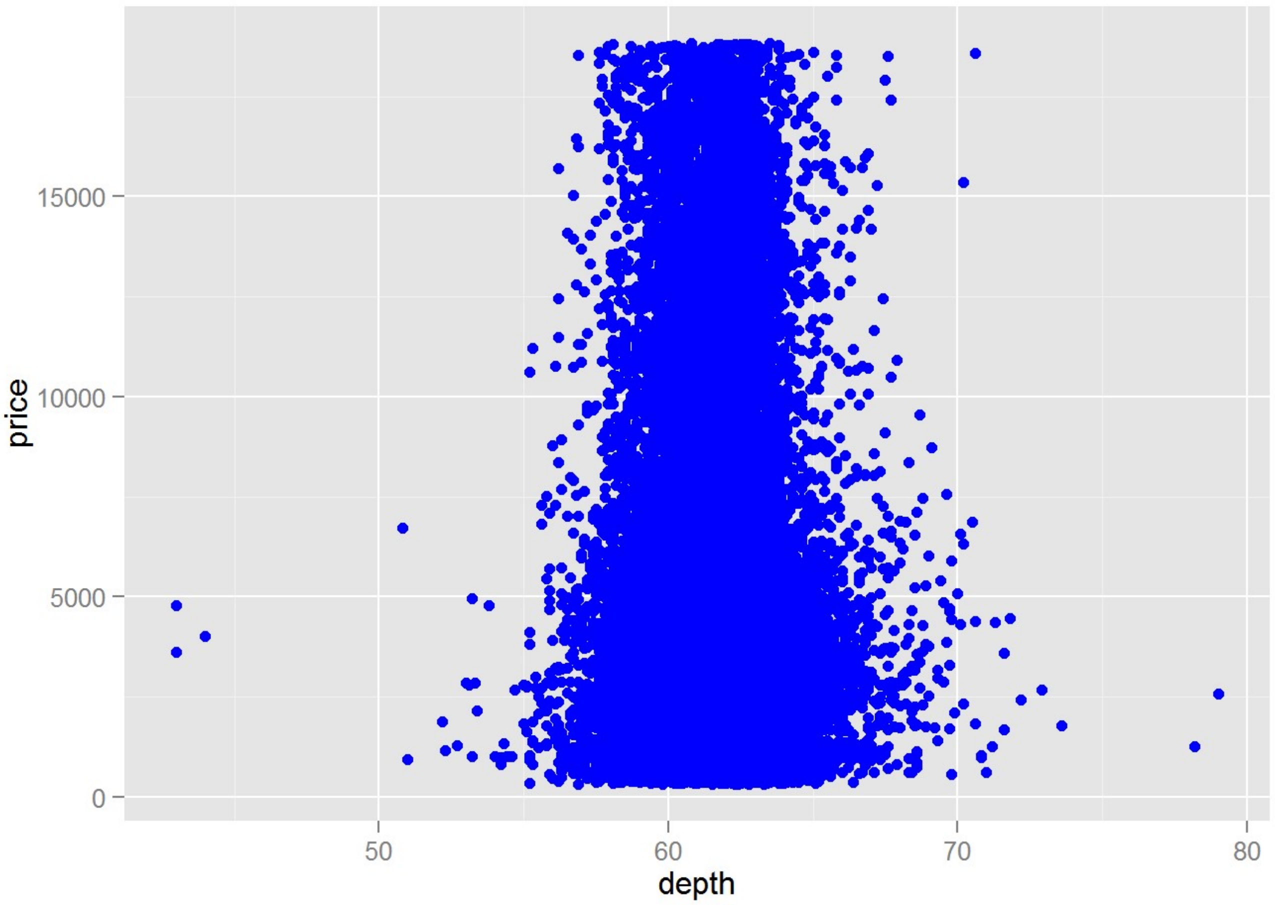
```
cor.test(diamonds$price, diamonds$y, method="pearson")
```

```
##
##  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
```

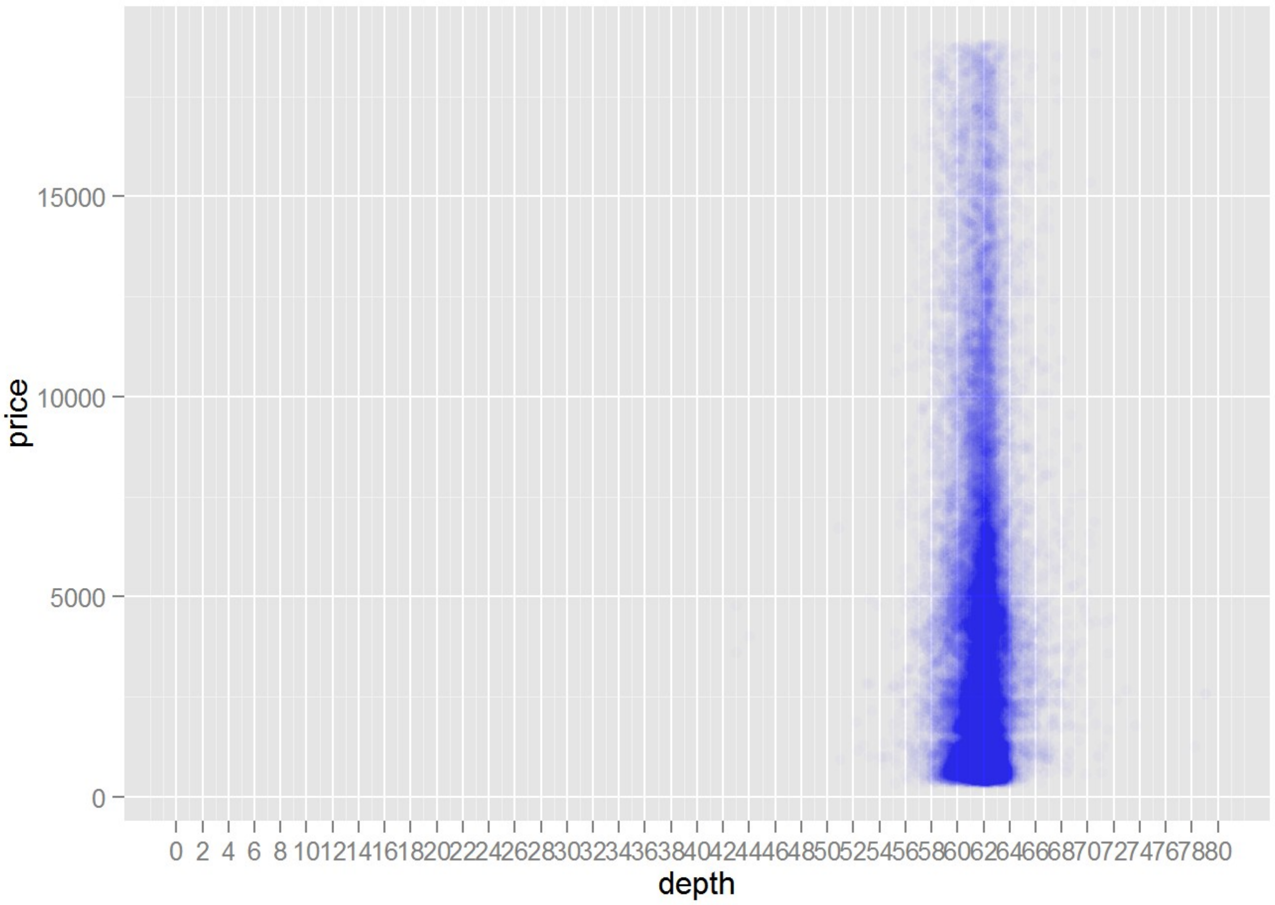
```
cor.test(diamonds$price, diamonds$z, method="pearson")
```

```
##
##  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
```

```
# create a scatterplot of price vs depth.
library(ggplot2)
ggplot(aes(x = depth, y = price), data = diamonds) + geom_point(color = "blue")
```



```
# Change the code to make the transparency of the
# points to be 1/100 of what they are now and mark
# the x-axis every 2 units. See the instructor notes
# for two hints.
ggplot(aes(x = depth, y = price), data = diamonds) +
  geom_point(alpha = 1/100, color = "blue") +
  scale_x_continuous(limits = c(0,80), breaks = seq(0, 80, 2))
```

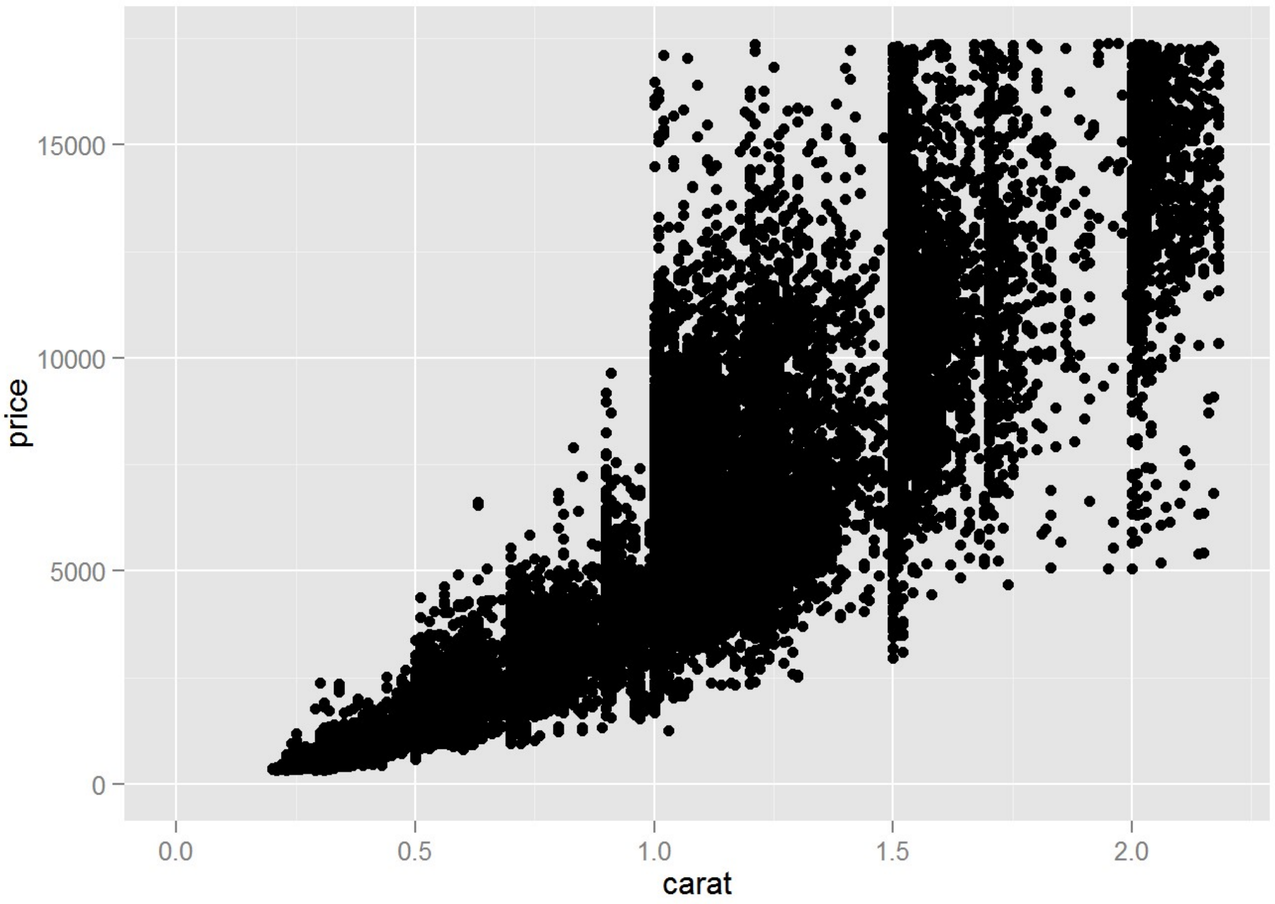


```
# correlations price vs depth
cor.test(diamonds$price, diamonds$depth, method="pearson")
```

```
##
##  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
```

```
# Create a scatterplot of price vs carat
# and omit the top 1% of price and carat
# values.
ggplot(aes(x = carat, y = price), data = diamonds) +
  geom_point() +
  scale_x_continuous(limits = c(0, quantile(diamonds$carat, 0.99))) +
  scale_y_continuous(limits = c(0, quantile(diamonds$price, 0.99)))
```

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

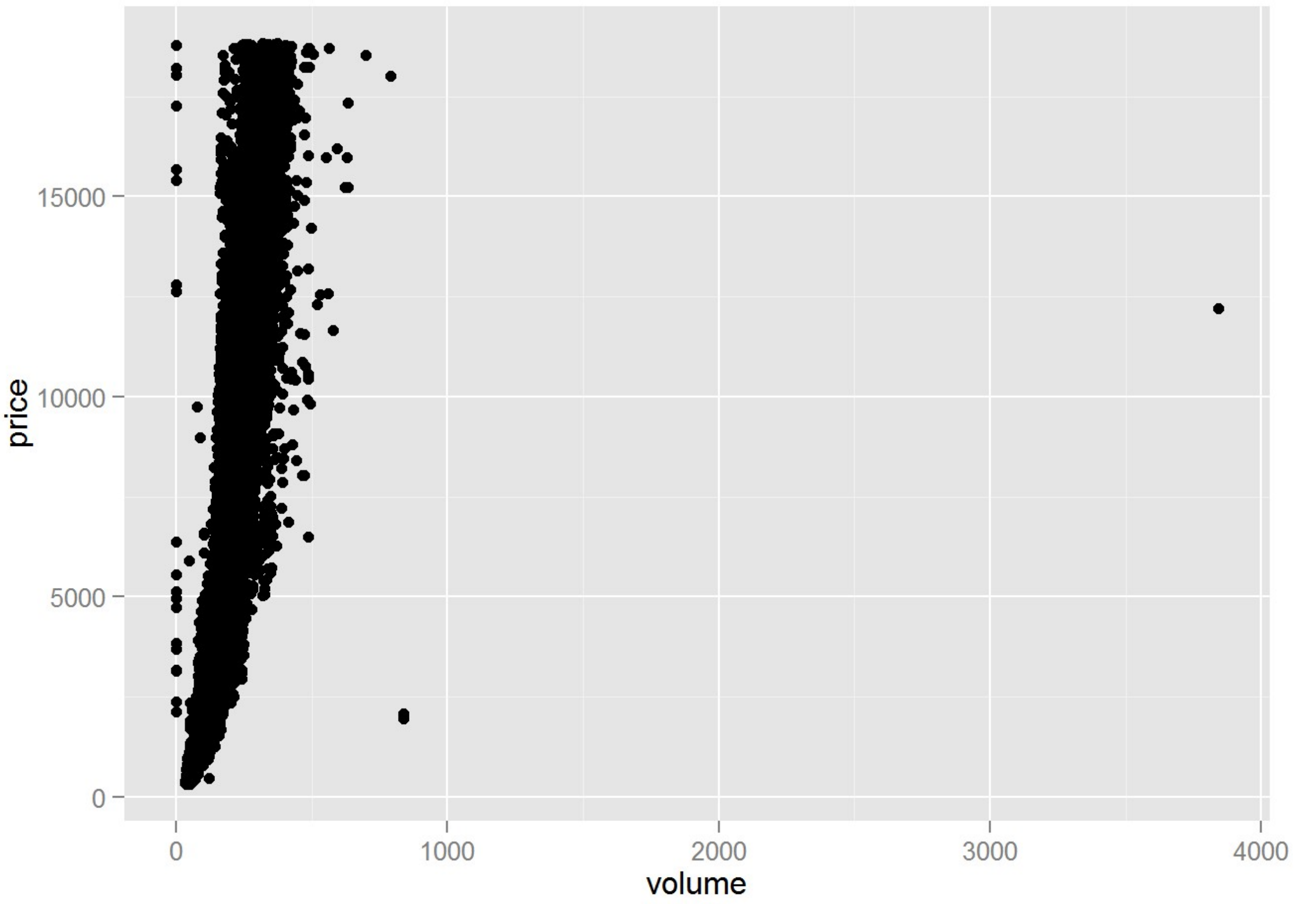


```
# create new variable volume (x * y * z)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
diamonds <- mutate(diamonds, volume = x * y * z)
```

```
# scatterplot price vs. volume
ggplot(aes(x = volume, y = price), data = diamonds) +
  geom_point()
```



```
# correlations price vs depth, with subset
cor.test(subset(diamonds, volume >0 & volume <= 800)$volume, subset(diamonds, volu
me >0 & volume <= 800)$price, method="pearson")
```

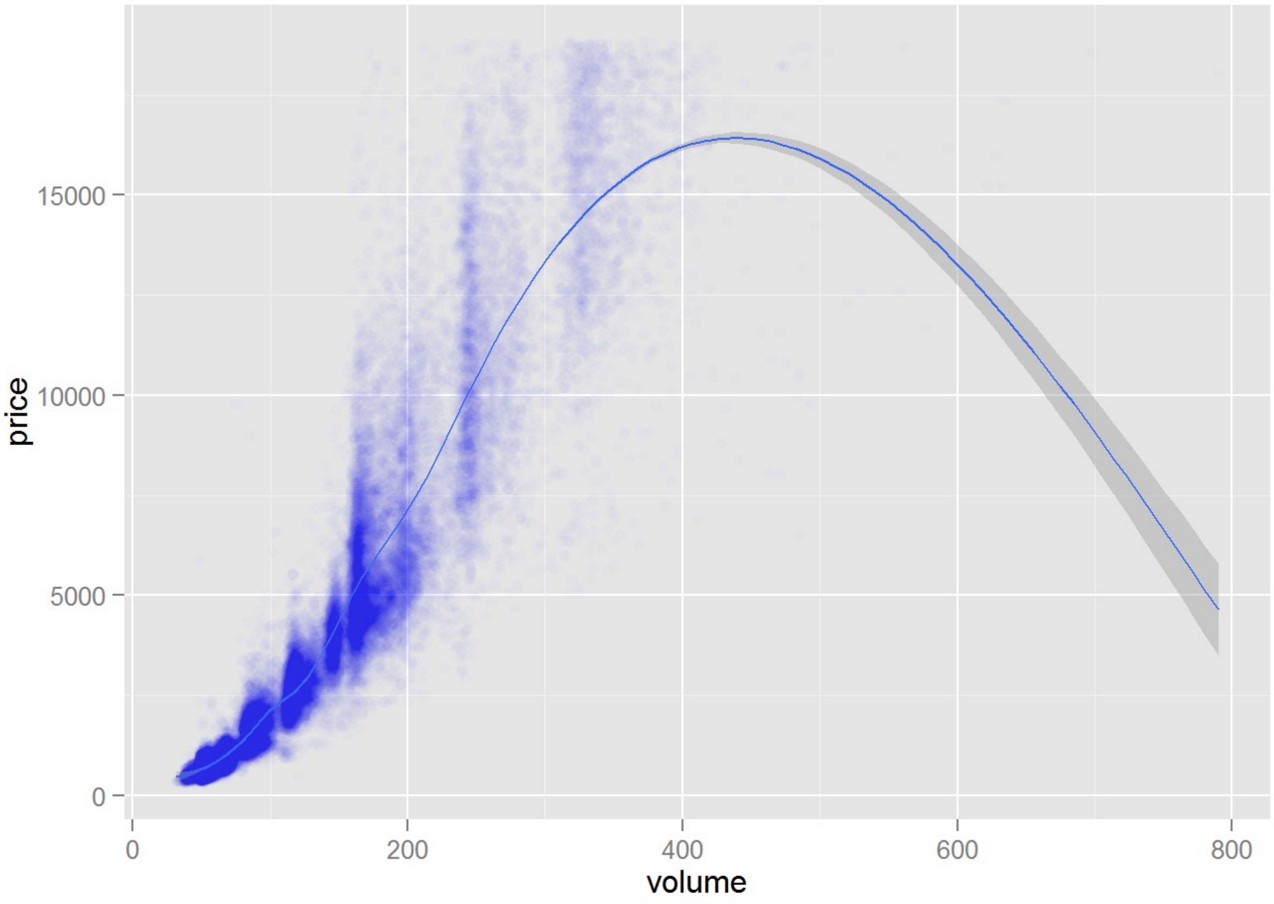
```
##
##  Pearson's product-moment correlation
##
## data:  subset(diamonds, volume > 0 & volume <= 800)$volume and subset(diamond
s, volume > 0 & volume <= 800)$price
## t = 559.19, df = 53915, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9222944 0.9247772
## sample estimates:
##       cor
## 0.9235455
```

```
# alternate method
with(subset(diamonds, volume >0 & volume <= 800), cor.test(volume, price, method
="pearson"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  volume and 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
```

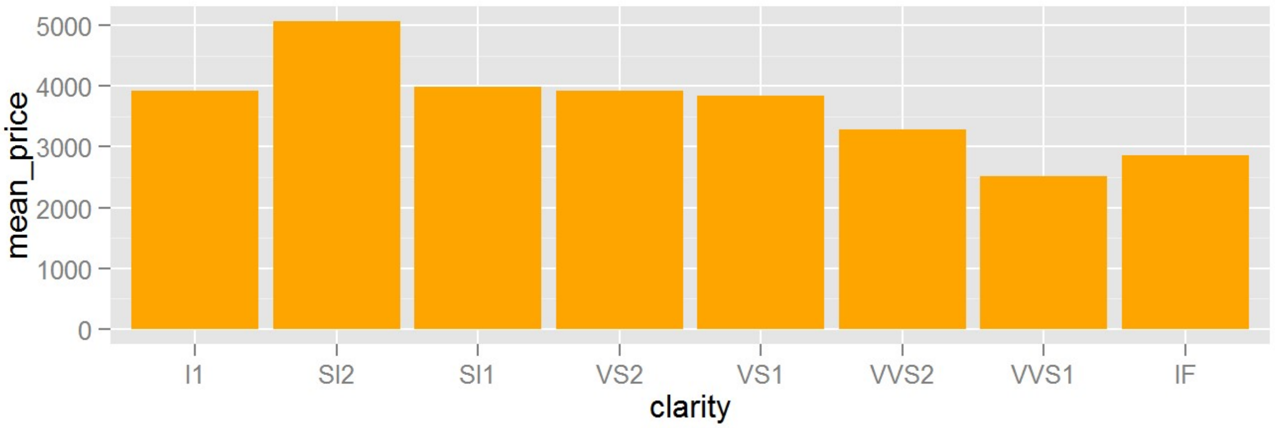
```
# Subset the data to exclude diamonds with a volume
# greater than or equal to 800. Also, exclude diamonds
# with a volume of 0. Adjust the transparency of the
# points and add a linear model to the plot.

ggplot(subset(diamonds, volume > 0 & volume < 800), aes(x = volume, y = price)) +
         geom_point(alpha = 1/100, color = "blue") +
         geom_smooth()
```

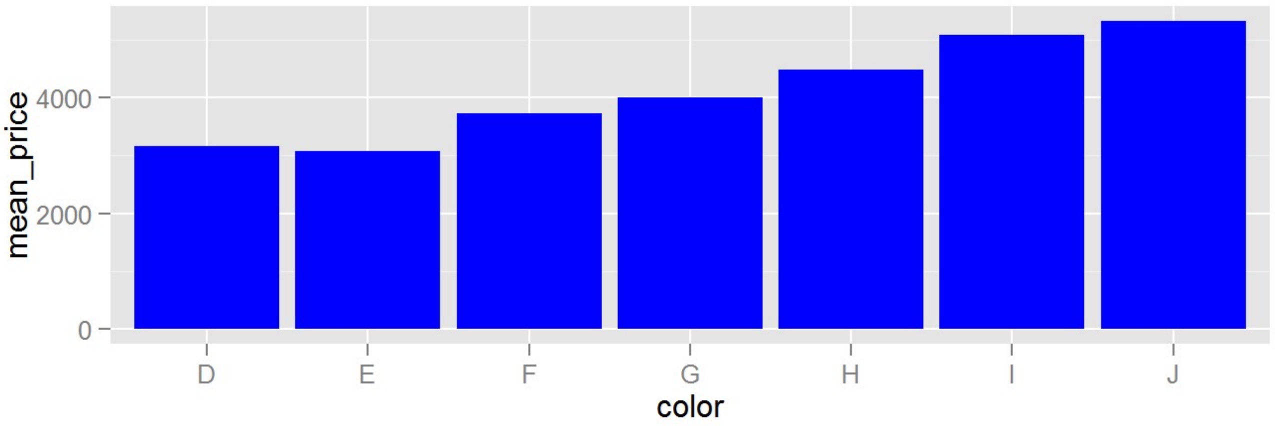
```
## geom_smooth: method="auto" and size of largest group is >=1000, so using gam wi
th formula: y ~ s(x, bs = "cs"). Use 'method = x' to change the smoothing method.
```



```r
# Use the function dplyr package
# to create a new data frame containing
# info on diamonds by clarity.
# Name the data frame diamondsByClarity

# The data frame should contain the following
# variables in this order.

#       (1) mean_price
#       (2) median_price
#       (3) min_price
#       (4) max_price
#       (5) n

# where n is the number of diamonds in each
# level of clarity.

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

```r
data(diamonds)
library(dplyr)
library(gridExtra)
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(aes(x = clarity, y = mean_price), data = diamonds_mp_by_clarity) +
geom_bar(stat = 'identity', fill ='orange')

p2 <- ggplot(aes(color, mean_price), data = diamonds_mp_by_color) +
geom_bar(stat = 'identity', fill ='blue')

grid.arrange(p1, p2)
```


mean_price
5000
4000
3000
2000
1000
0
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF
clarity


mean_price
4000
2000
0
D
E
F
G
H
I
J
color