Problem_set_2

Kyle

June 2, 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:

```r
library(ggplot2) #must load the ggplot package first
```

```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

```r
#detach("package:plyr", unload=TRUE)
#library(plyr)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.2.5
```

```
##
## 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
```

```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.2.5
```

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

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

```r
data(diamonds) #loads the diamonds data set since it comes with the ggplot package
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
##
```

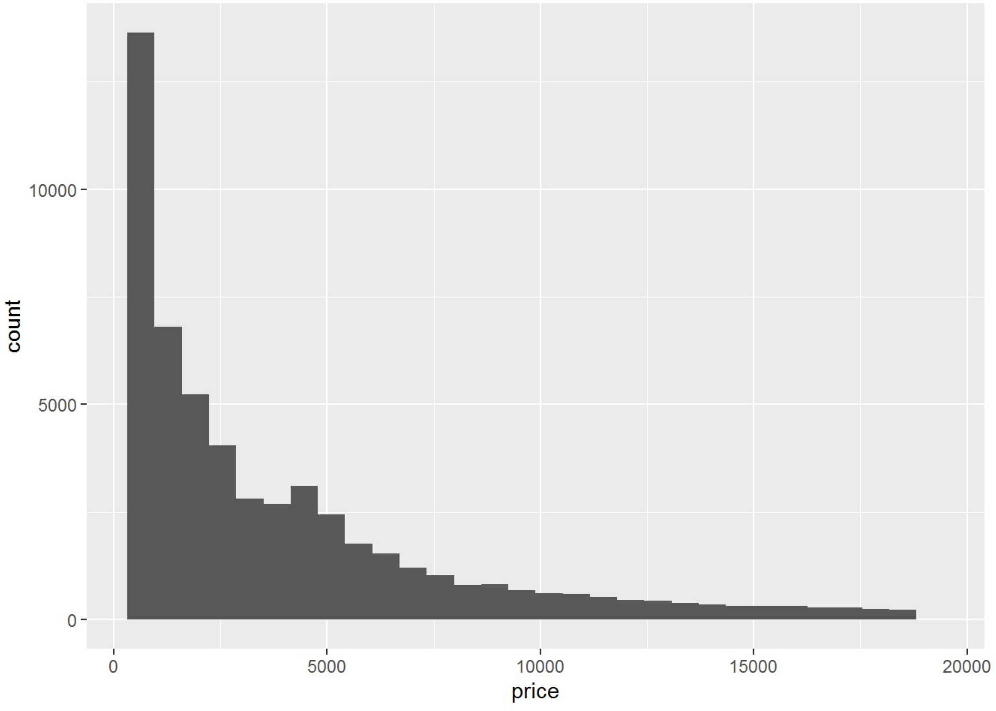
```
# ?diamonds
```

Including Plots

You can also embed plots, for example:

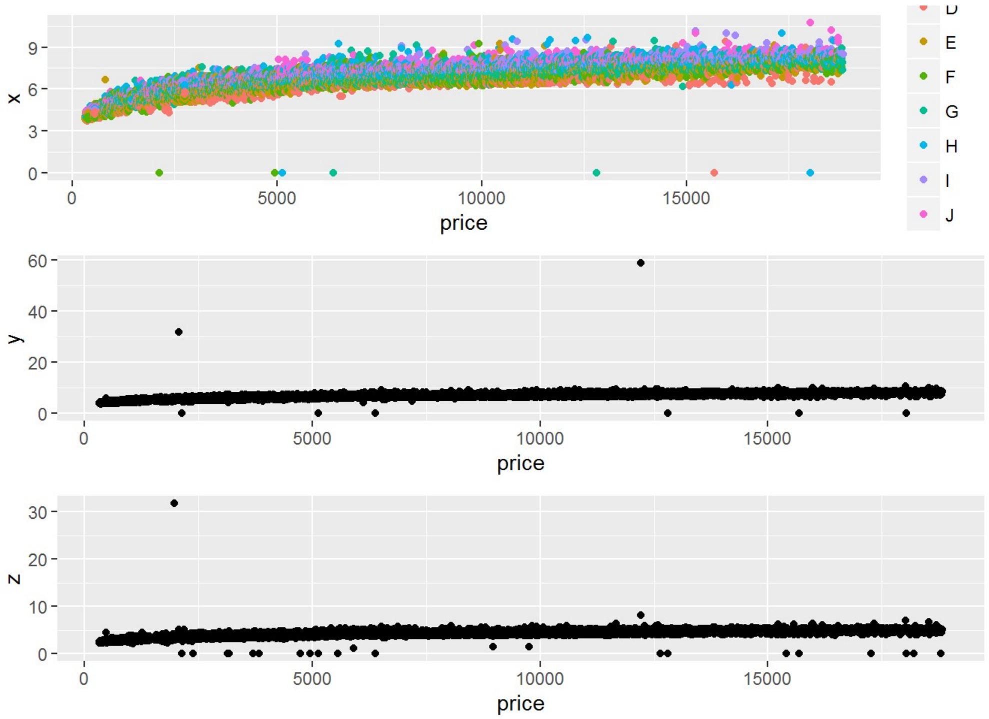
```
qplot(x = price, data = diamonds)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
p1 <- qplot(x = price, y = x, data = diamonds, colour=color)
p2 <-  qplot(x = price, y = y, data = diamonds)
p3 <-  qplot(x = price, y = z, data = diamonds)


grid.arrange(p1, p2, p3, ncol = 1)
```



```
# qplot docs:
# http://docs.ggplot2.org/0.9.3/qplot.html
```

```
#?cor.test

# formatted as: (data$column)

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
```

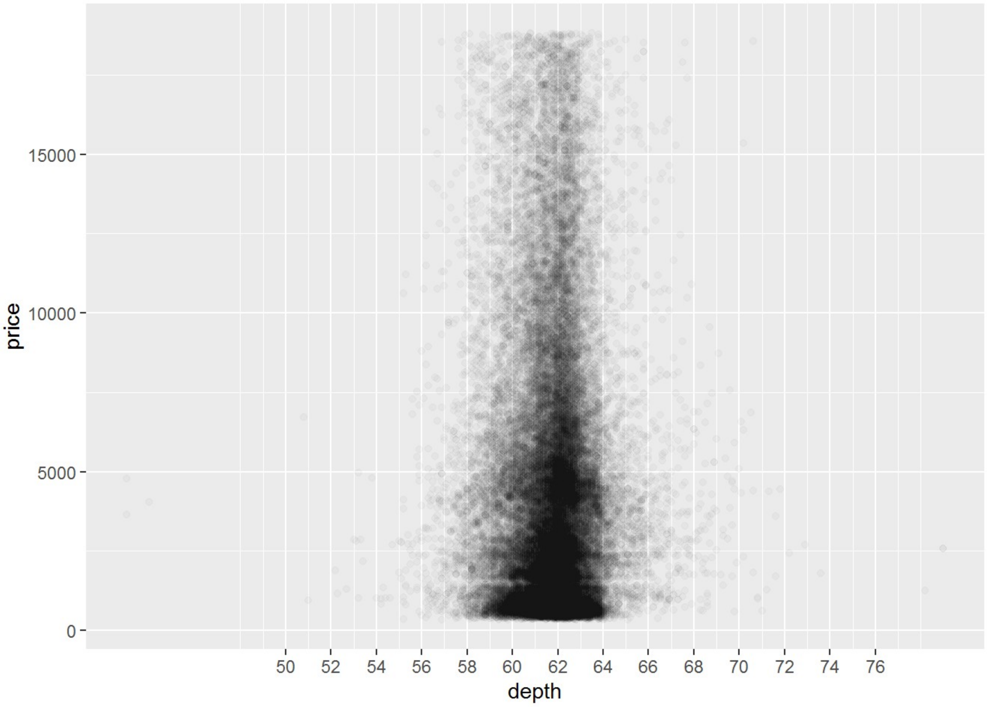
```
cor.test(diamonds$price, diamonds$y, method = 'pearson')
```

```
##
##  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
```

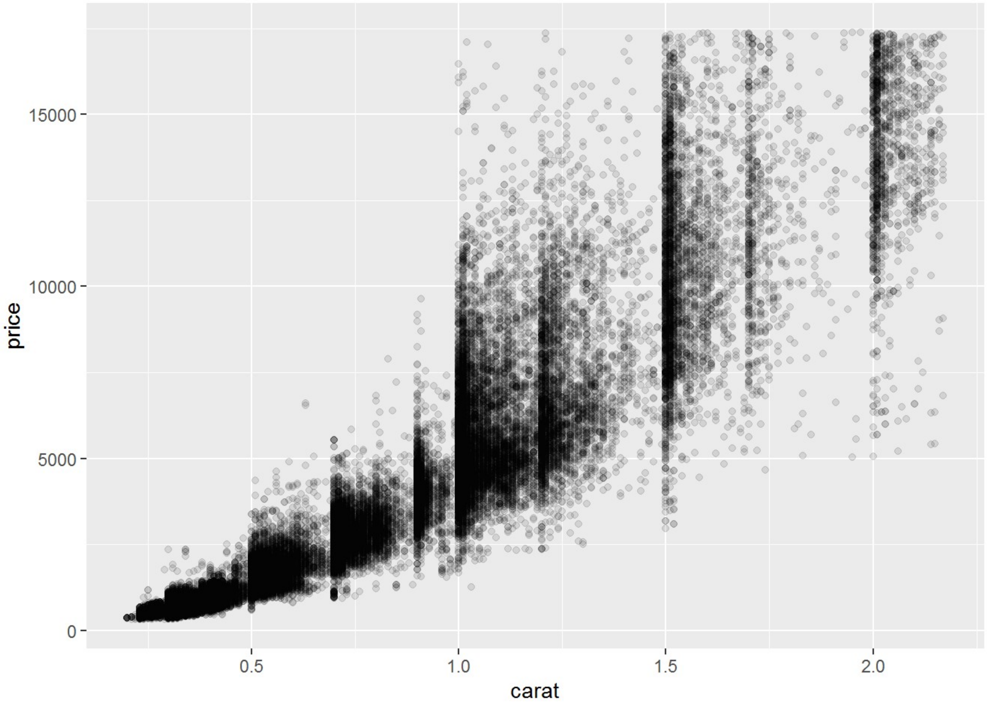
```
# ?scale_x_continuous

# simple scatter plot
# qplot(x = price, y = depth, data = diamonds)

# more complex scatter plot
ggplot(data = diamonds, aes(x = depth, y = price)) +
  geom_point(alpha=.025) +
  scale_x_continuous(breaks=seq(50,76,2))
```



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

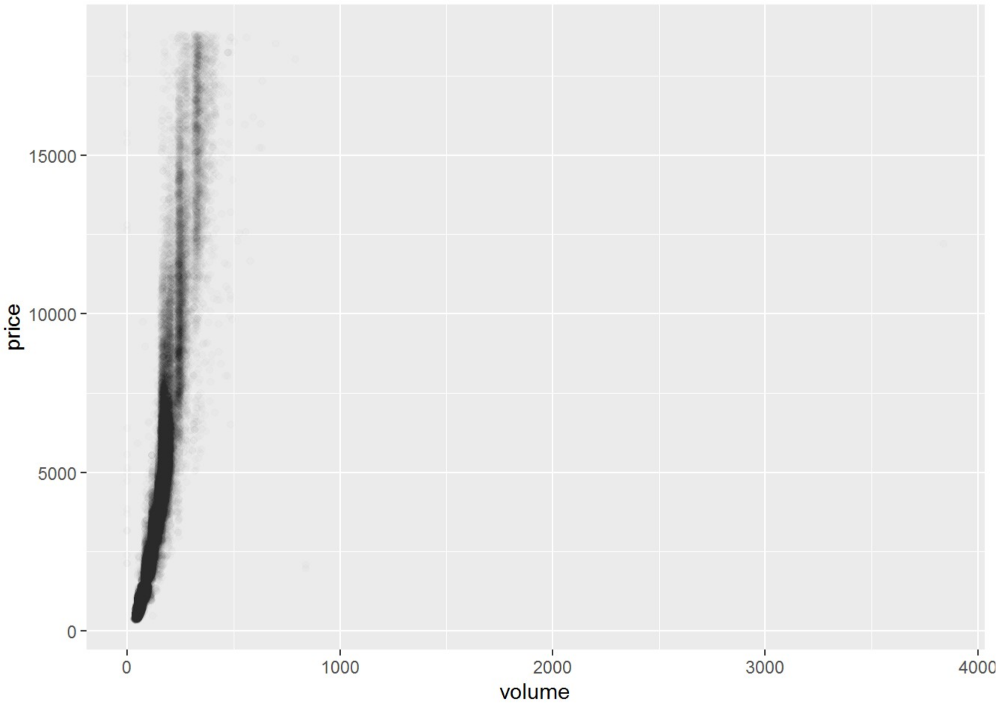
```
##
##  Pearson's product-moment correlation
##
## data:  diamonds$depth and diamonds$price
## t = -2.473, df = 53938, p-value = 0.0134
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.019084756 -0.002208537
## sample estimates:
##        cor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

```
#?geom_point()

# more complex scatter plot and I took out the top 1% of all data (outliers perhaps)
ggplot(aes(x = carat, y = price),
       data = subset(diamonds, diamonds$price < quantile(diamonds$price, 0.99) &
                        diamonds$carat < quantile(diamonds$carat, 0.99))) +
  geom_point(alpha=0.10)
```



```
  #scale_x_continuous(breaks=seq(50,76,2))
```

```
# add new feature to data set: volume
diamonds$volume <- diamonds$x * diamonds$y * diamonds$z

# scatterplot
ggplot(data = diamonds, aes(x = volume, y = price)) +
  geom_point(alpha=.010)
```

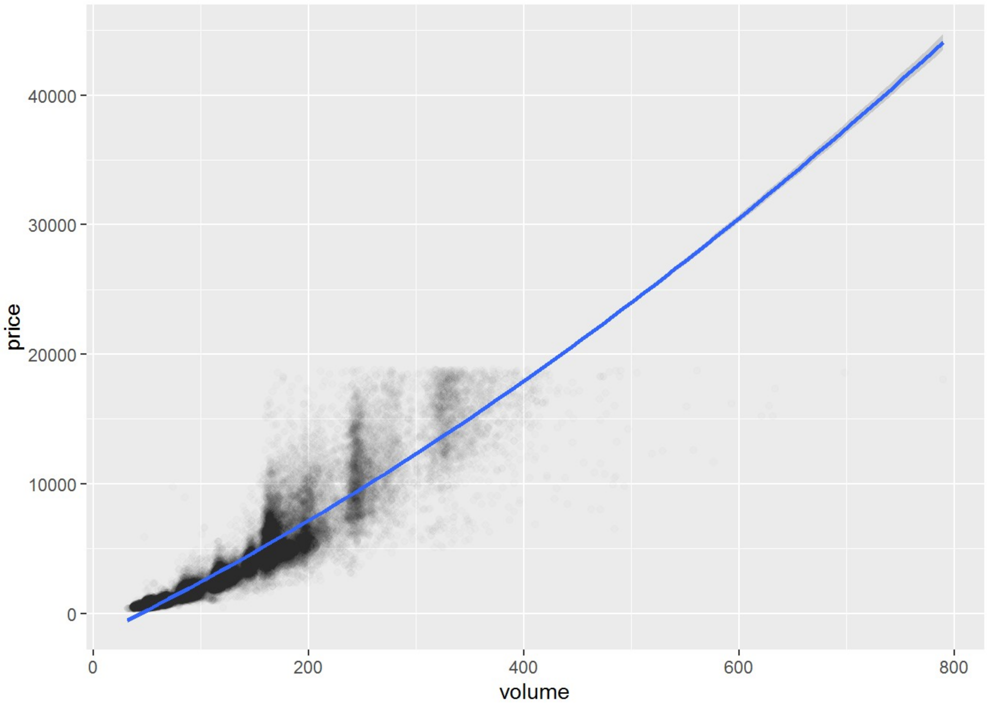


```
# depends on plyr library
# count(diamonds$volume == 0)
# count(diamonds$volume > 1000)
# count(diamonds$volume > 750)
```

```
pf_800 <- subset(diamonds, !(volume == 0 | volume >= 800)) #or pf_800 <-subset(pf, volume > 0 & volume < 80
0)

cor.test(pf_800$price, pf_800$volume , method = 'pearson')
```

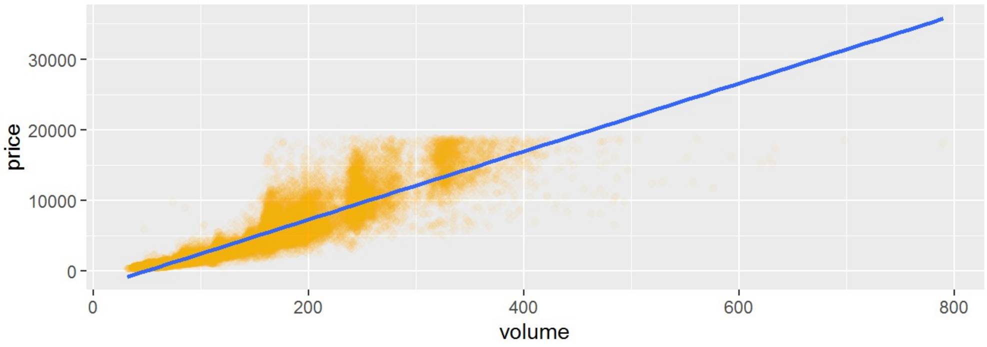
```
##
##  Pearson's product-moment correlation
##
## data:  pf_800$price and pf_800$volume
## t = 559.19, df = 53915, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9222944 0.9247772
## sample estimates:
##       cor
## 0.9235455
```

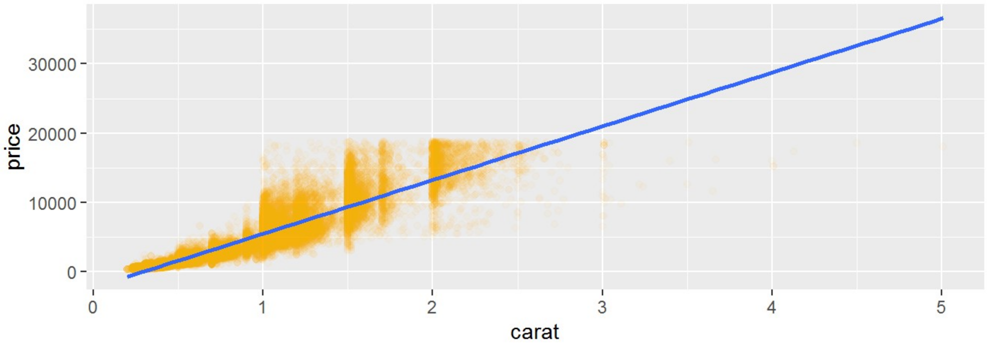
```
ggplot(data = pf_800, aes(x = volume, y = price)) +
  geom_point(alpha=.010) +
#  geom_smooth(method = 'lm', color = 'red')
  geom_smooth(method = "lm", formula = y ~ poly(x,2), size = 1)
```



```
# Do you think this would be a useful model to estimate
# the price of diamonds? Why or why not?
```

```
diamonds$volume<- diamonds$x * diamonds$y * diamonds$z
set_volume <- subset(diamonds, volume > 0 & volume < 800)

p1 <- ggplot(aes(x = volume, y = price), data = set_volume ) +
  geom_point(alpha = 1/25, color = 'orange')
p_v <- p1 + stat_smooth(method = "lm", formula = y ~ x, size = 1)

p2 <- ggplot(aes(x = carat, y = price), data = set_volume ) +
  geom_point(alpha = 1/25, color = 'orange')
p_c <- p2 + stat_smooth(method = "lm", formula = y ~ x, size = 1)

grid.arrange(p_v, p_c)
```



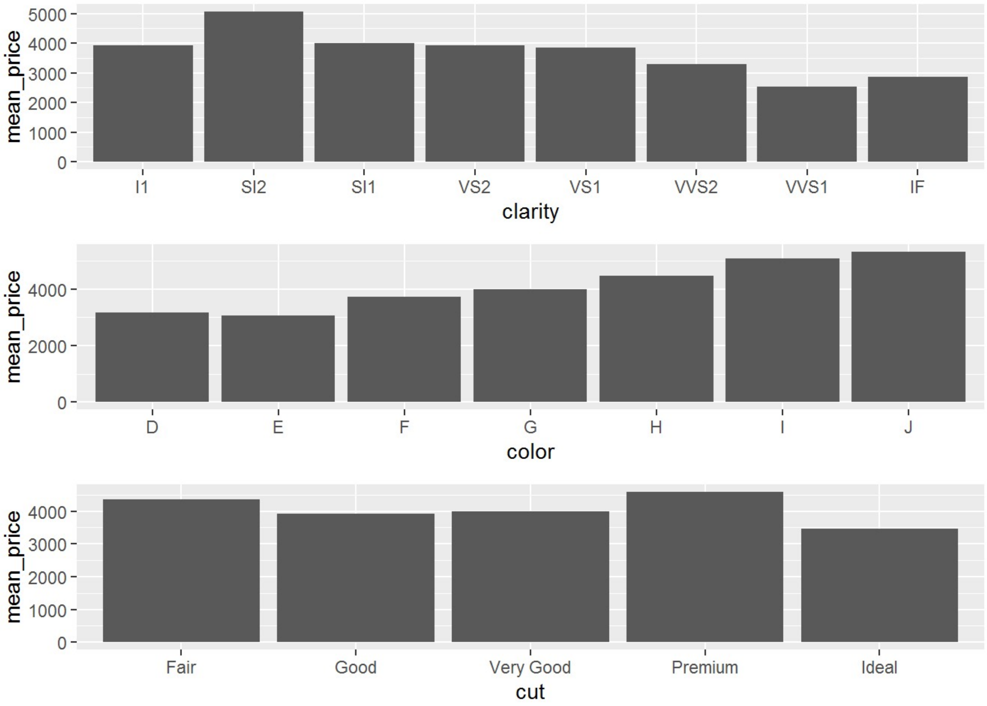


```
cor.test(set_volume$carat, set_volume$volume, method = "pearson")
```

```
## 
##  Pearson's product-moment correlation
## 
## data:  set_volume$carat and set_volume$volume
## t = 5041.9, df = 53915, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9989232 0.9989589
## sample estimates:
##       cor 
## 0.9989412
```

```
# detach("package:plyr", unload=TRUE)

# Use the function dplyr package
# to create a new data frame containing
# info on diamonds by clarity.
#       (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.

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

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

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))

diamonds_by_cut <- group_by(diamonds, cut)
diamonds_mp_by_cut <- summarise(diamonds_by_cut, mean_price = mean(price))


p1 <- ggplot(aes(clarity, mean_price), data = diamonds_mp_by_clarity) +
geom_bar(stat = 'identity')

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

p3 <- ggplot(aes(cut, mean_price), data = diamonds_mp_by_cut) +
geom_bar(stat = 'identity')


grid.arrange(p1, p2, p3)
```



```r
#
# We think something odd is going here. These trends seem to go against our intuition.
#
# Mean price tends to decrease as clarity improves. The same can be said for color.
# cut is fairly standartd though the highest grade cut does decrease...
```