Data Analysis R - Problem set 3

Mario Bonilla

February 02, 2015

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

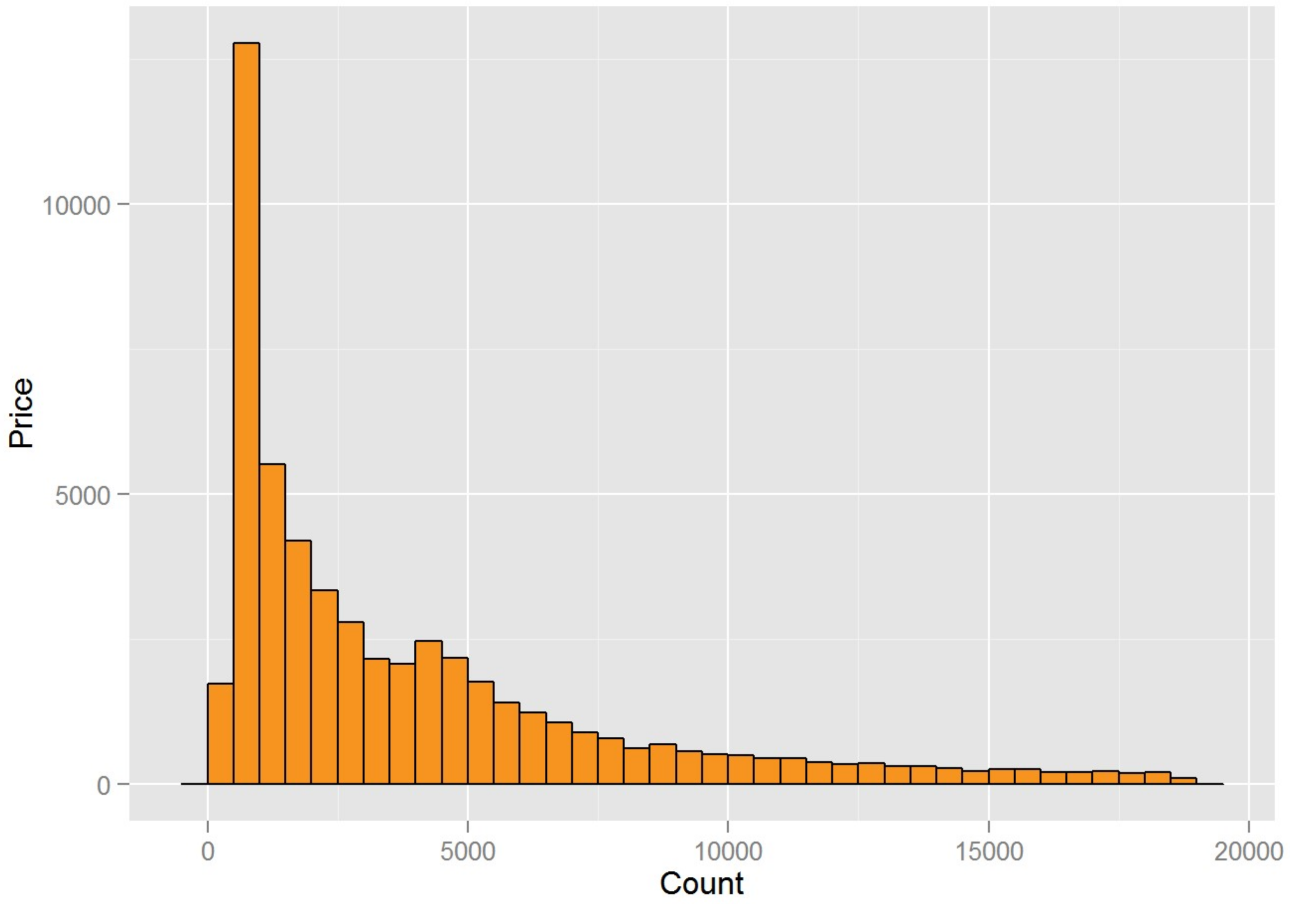
```
dim(diamonds)
```

```
## [1] 53940    10
```

```
?diamonds
```

```
## starting httpd help server ... done
```

```
#Price Histogram
qplot(x = price, data = diamonds,
      xlab = 'Count',
      ylab = 'Price',
      binwidth = 500,
      color = I('black'), fill = I('#F79420'))
```



```
#Price Histogram summary
# Shape: skewed
summary(diamonds$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     326     950    2401    3933    5324   18820
```

```
# Min 326, Median 2401, Mean 3933, Max 18820, 1st Q 950, 3rd Q 5324
```

```
#Diamonds counts: how many...?
sum(diamonds$price < 500)
```

```
## [1] 1729
```

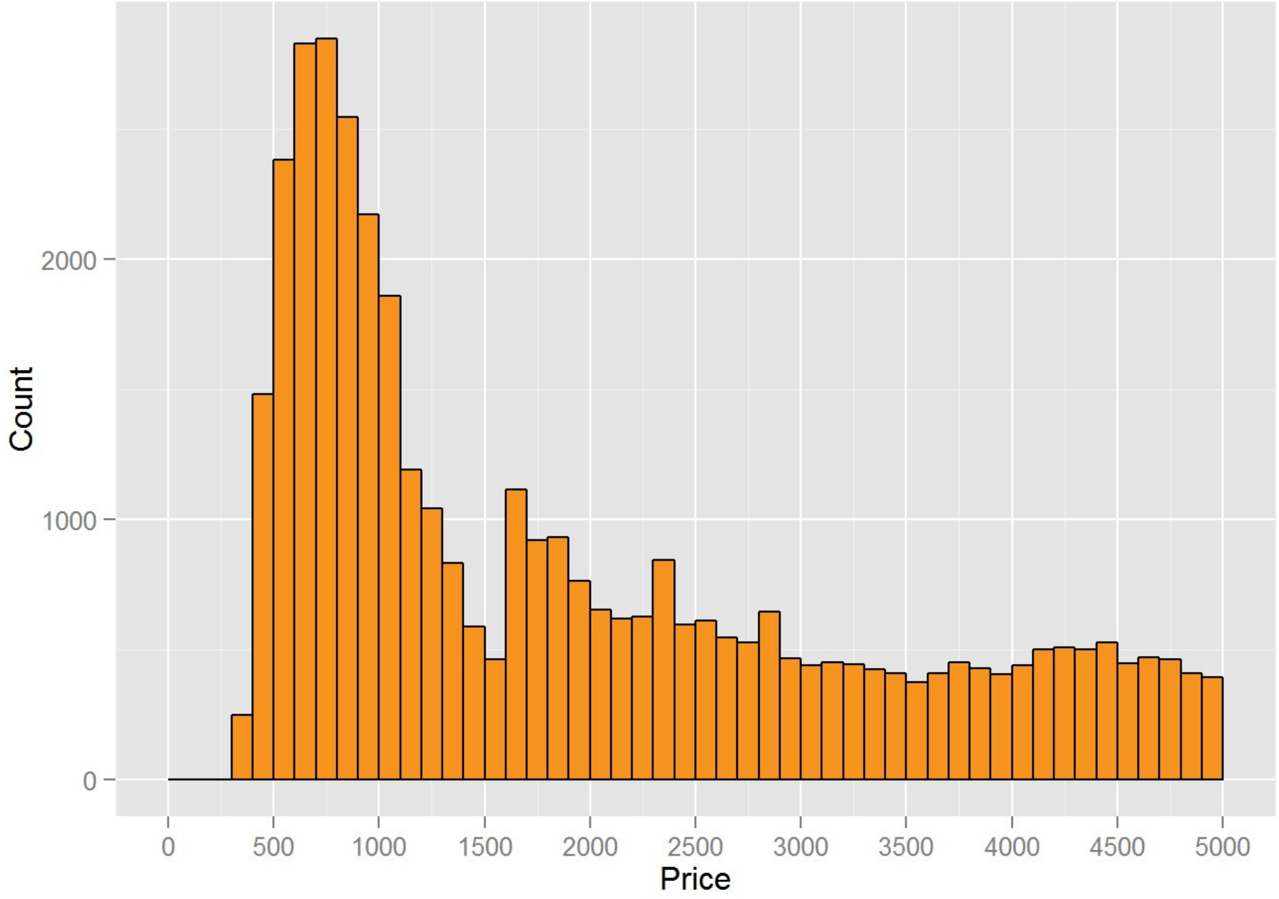
```
sum(diamonds$price < 250)
```

```
## [1] 0
```

```
sum(diamonds$price >= 15000)
```

```
## [1] 1656
```

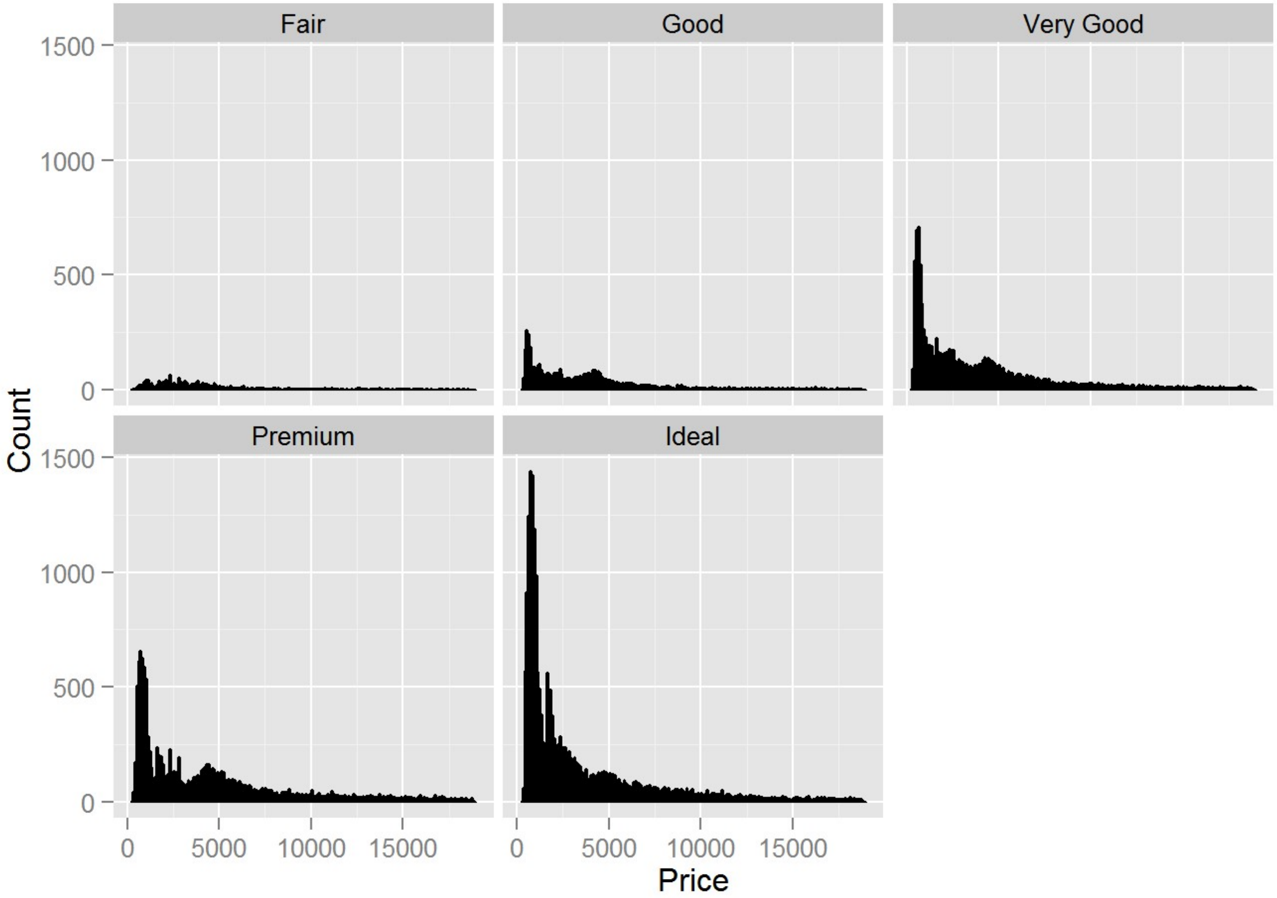
```
# Explore the largest peak in the
# price histogram
qplot(x = price, data = diamonds,
      xlab = 'Price',
      ylab = 'Count',
      binwidth = 100,
      color = I('black'), fill = I('#F79420')) +
  scale_x_continuous(limits = c(0, 5000),
                     breaks = seq(0, 5000, 500))
```



```
ggsave('priceHistogram.jpeg')
```

```
## Saving 7 x 5 in image
```

```
# Break out the histogram of diamond prices by cut.
qplot(x = price, data = diamonds,
      xlab = 'Price',
      ylab = 'Count',
      binwidth = 100,
      color = I('black'), fill = I('#F79420')) +
  facet_wrap(~cut)
```



```
#Price by cut
by(diamonds$price, diamonds$cut, summary)
```

```
## diamonds$cut: Fair
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     337    2050    3282    4359    5206   18570
## --------------------------------------------------------
## diamonds$cut: Good
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     327    1145    3050    3929    5028   18790
## --------------------------------------------------------
## diamonds$cut: Very Good
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     336     912    2648    3982    5373   18820
## --------------------------------------------------------
## diamonds$cut: Premium
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     326    1046    3185    4584    6296   18820
## --------------------------------------------------------
## diamonds$cut: Ideal
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     326     878    1810    3458    4678   18810
```

```
by(diamonds$price, diamonds$cut, max)
```

```
## diamonds$cut: Fair
## [1] 18574
## --------------------------------------------------------
## diamonds$cut: Good
## [1] 18788
## --------------------------------------------------------
## diamonds$cut: Very Good
## [1] 18818
## --------------------------------------------------------
## diamonds$cut: Premium
## [1] 18823
## --------------------------------------------------------
## diamonds$cut: Ideal
## [1] 18806
```

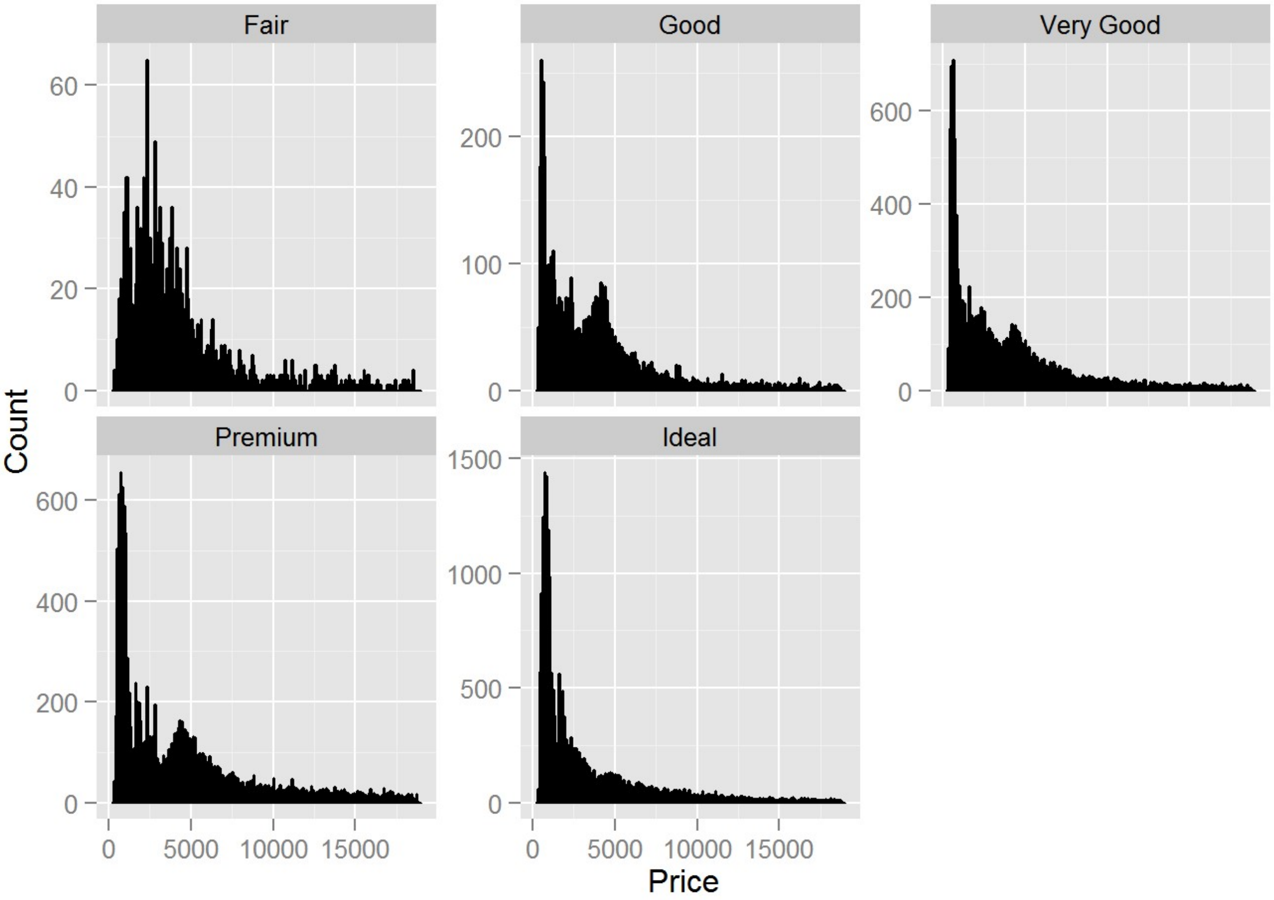
```
by(diamonds$price, diamonds$cut, min)
```

```
## diamonds$cut: Fair
## [1] 337
## --------------------------------------------------------
## diamonds$cut: Good
## [1] 327
## --------------------------------------------------------
## diamonds$cut: Very Good
## [1] 336
## --------------------------------------------------------
## diamonds$cut: Premium
## [1] 326
## --------------------------------------------------------
## diamonds$cut: Ideal
## [1] 326
```

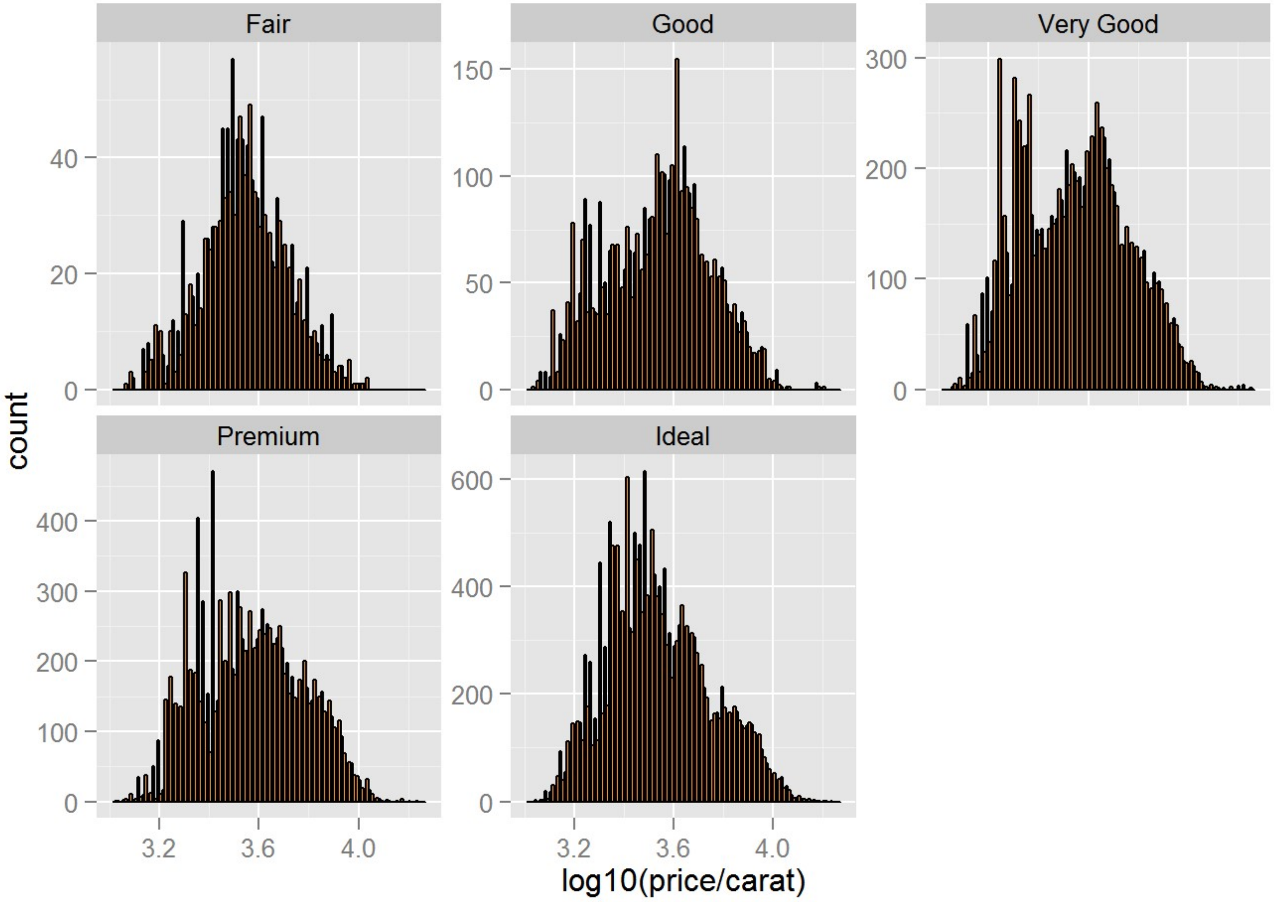
```
by(diamonds$price, diamonds$cut, median)
```

```
## diamonds$cut: Fair
## [1] 3282
## --------------------------------------------------------
## diamonds$cut: Good
## [1] 3050.5
## --------------------------------------------------------
## diamonds$cut: Very Good
## [1] 2648
## --------------------------------------------------------
## diamonds$cut: Premium
## [1] 3185
## --------------------------------------------------------
## diamonds$cut: Ideal
## [1] 1810
```

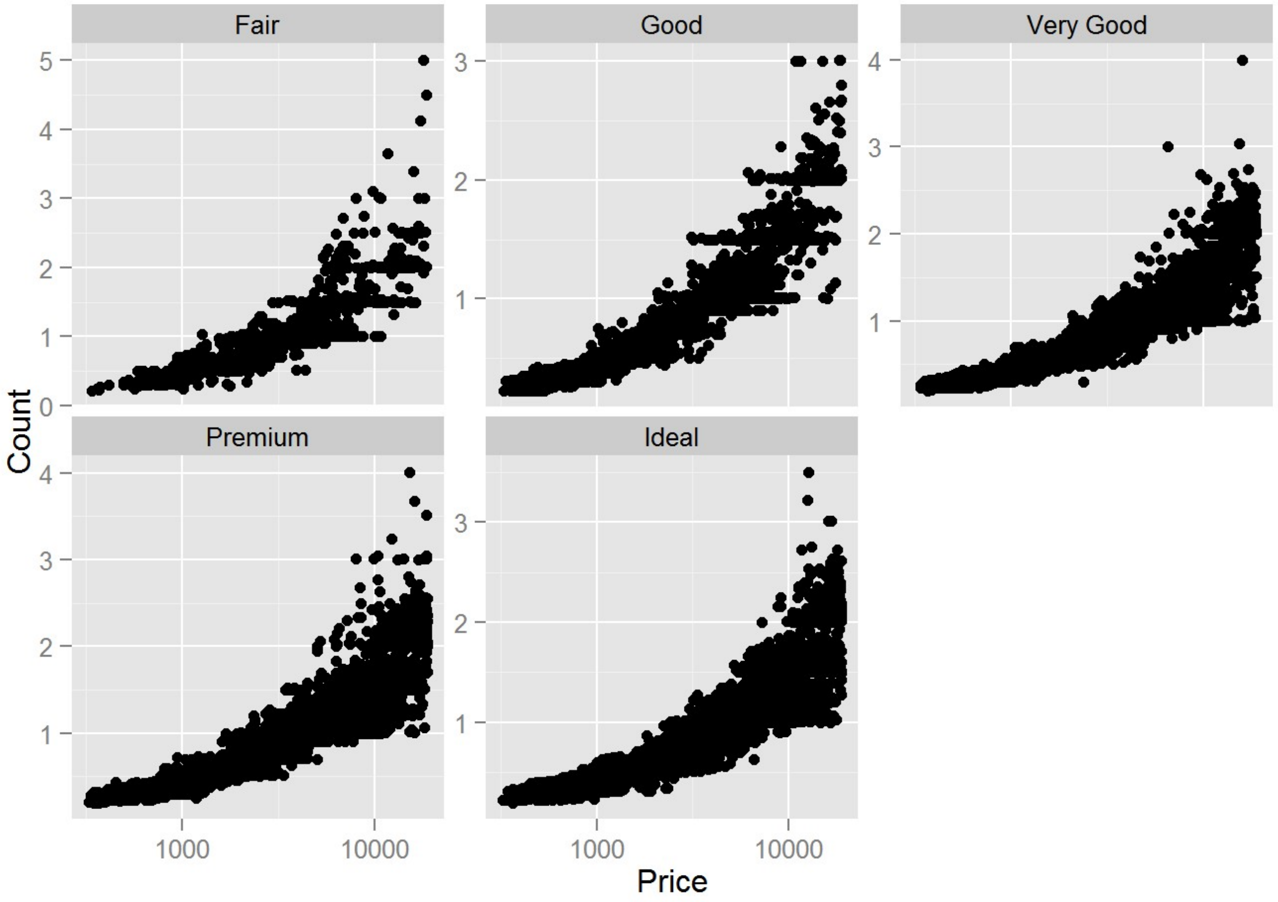
```
#facet_wrap so that
# the y-axis in the histograms is not fixed
qplot(x = price, data = diamonds,
      xlab = 'Price',
      ylab = 'Count',
      binwidth = 100,
      color = I('black'), fill = I('#F79420')) +
  facet_wrap(~cut, scales = 'free_y')
```



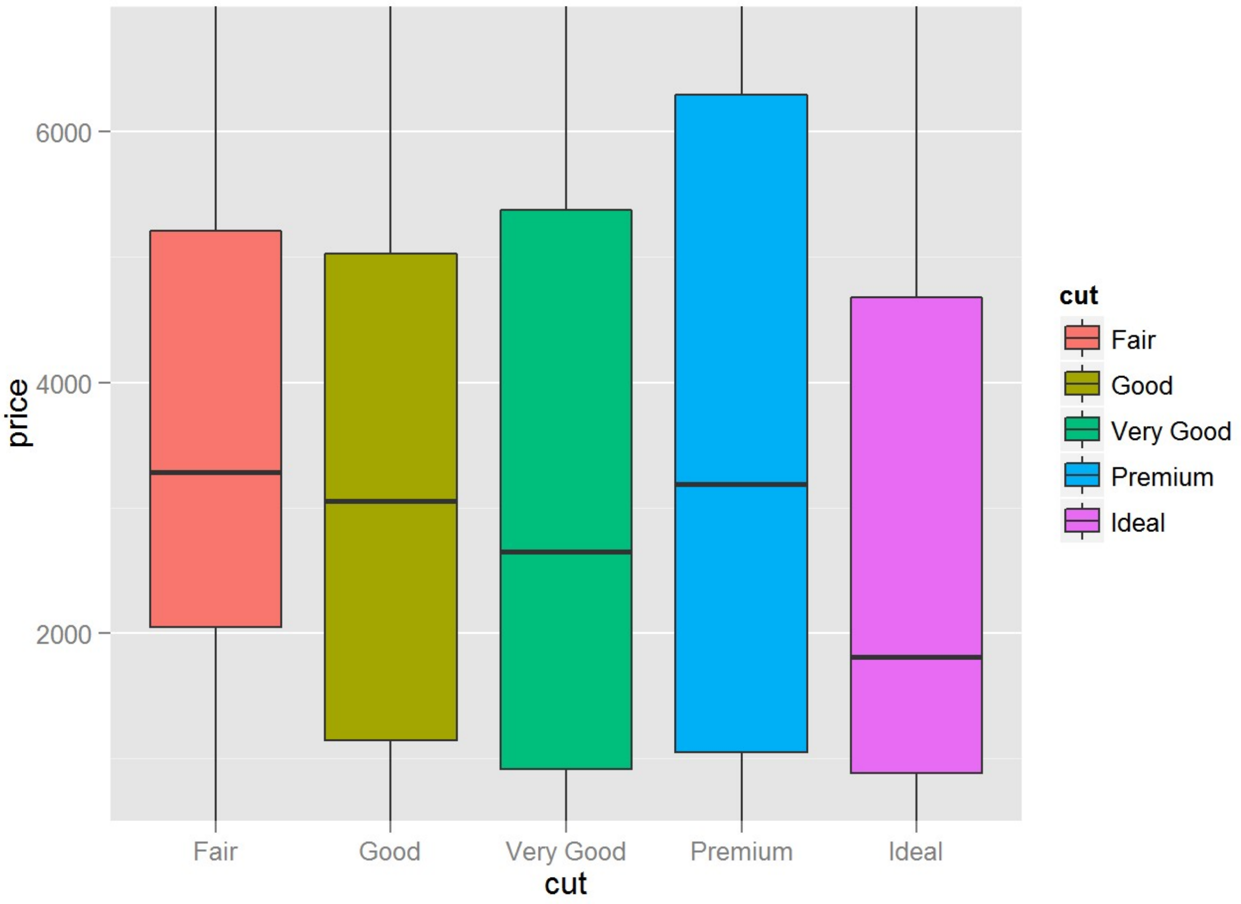
```
#price per carat, facet by cut.
qplot(x = log10(price/carat), data = diamonds,
      binwidth = 0.01,
      color = I('black'), fill = I('#F79420')) +
  facet_wrap(~cut, scales="free_y")
```



```
#other
qplot(x = price, y = carat, data = diamonds,
      xlab = 'Price',
      ylab = 'Count',
      binwidth = 100,
      color = I('black'), fill = I('#F79420')) +
  facet_wrap(~cut, scales = 'free_y') +
  scale_x_log10()
```



```
#Price Box Plots
qplot(x = cut, y = price,
      data = subset(diamonds, !is.na(cut)),
      geom = 'boxplot', fill=cut) +
  coord_cartesian(ylim = c(500, 7000))
```



```
ggsave('priceBoxPlots.jpeg')
```

```
## Saving 7 x 5 in image
```

```
#Some detailed information. IQR: interquartile range
priceD <- subset(diamonds, color == 'D')
summary(priceD)
```

```
##      carat               cut        color       clarity         depth
##  Min.   :0.2000   Fair     : 163   D:6775   SI1    :2083   Min.   :52.2
##  1st Qu.:0.3600   Good     : 662   E:   0   VS2    :1697   1st Qu.:61.0
##  Median :0.5300   Very Good:1513   F:   0   SI2    :1370   Median :61.8
##  Mean   :0.6578   Premium  :1603   G:   0   VS1    : 705   Mean   :61.7
##  3rd Qu.:0.9050   Ideal    :2834   H:   0   VVS2   : 553   3rd Qu.:62.5
##  Max.   :3.4000                    I:   0   VVS1   : 252   Max.   :71.6
##                                    J:   0   (Other): 115
##      table          price               x               y
##  Min.   :52.0   Min.   :  357   Min.   :0.000   Min.   :0.000
##  1st Qu.:56.0   1st Qu.:  911   1st Qu.:4.590   1st Qu.:4.600
##  Median :57.0   Median : 1838   Median :5.230   Median :5.240
##  Mean   :57.4   Mean   : 3170   Mean   :5.417   Mean   :5.421
##  3rd Qu.:59.0   3rd Qu.: 4214   3rd Qu.:6.180   3rd Qu.:6.180
##  Max.   :73.0   Max.   :18693   Max.   :9.420   Max.   :9.340
##
##        z
##  Min.   :0.000
##  1st Qu.:2.820
##  Median :3.220
##  Mean   :3.343
##  3rd Qu.:3.840
##  Max.   :6.270
##
```

```
priceJ <- subset(diamonds, color == 'J')
summary(priceJ)
```

```
##      carat              cut      color       clarity        depth
##  Min.   :0.230   Fair     :119   D:   0   SI1    :750   Min.   :43.00
##  1st Qu.:0.710   Good     :307   E:   0   VS2    :731   1st Qu.:61.20
##  Median :1.110   Very Good:678   F:   0   VS1    :542   Median :62.00
##  Mean   :1.162   Premium  :808   G:   0   SI2    :479   Mean   :61.89
##  3rd Qu.:1.520   Ideal    :896   H:   0   VVS2   :131   3rd Qu.:62.70
##  Max.   :5.010                   I:   0   VVS1   : 74   Max.   :73.60
##                                  J:2808   (Other):101
##      table           price               x                y
##  Min.   :51.60   Min.   :  335   Min.   : 3.930   Min.   : 3.900
##  1st Qu.:56.00   1st Qu.: 1860   1st Qu.: 5.700   1st Qu.: 5.718
##  Median :58.00   Median : 4234   Median : 6.640   Median : 6.630
##  Mean   :57.81   Mean   : 5324   Mean   : 6.519   Mean   : 6.518
##  3rd Qu.:59.00   3rd Qu.: 7695   3rd Qu.: 7.380   3rd Qu.: 7.380
##  Max.   :68.00   Max.   :18710   Max.   :10.740   Max.   :10.540
##
##        z
##  Min.   :2.460
##  1st Qu.:3.530
##  Median :4.110
##  Mean   :4.033
##  3rd Qu.:4.580
##  Max.   :6.980
##
```

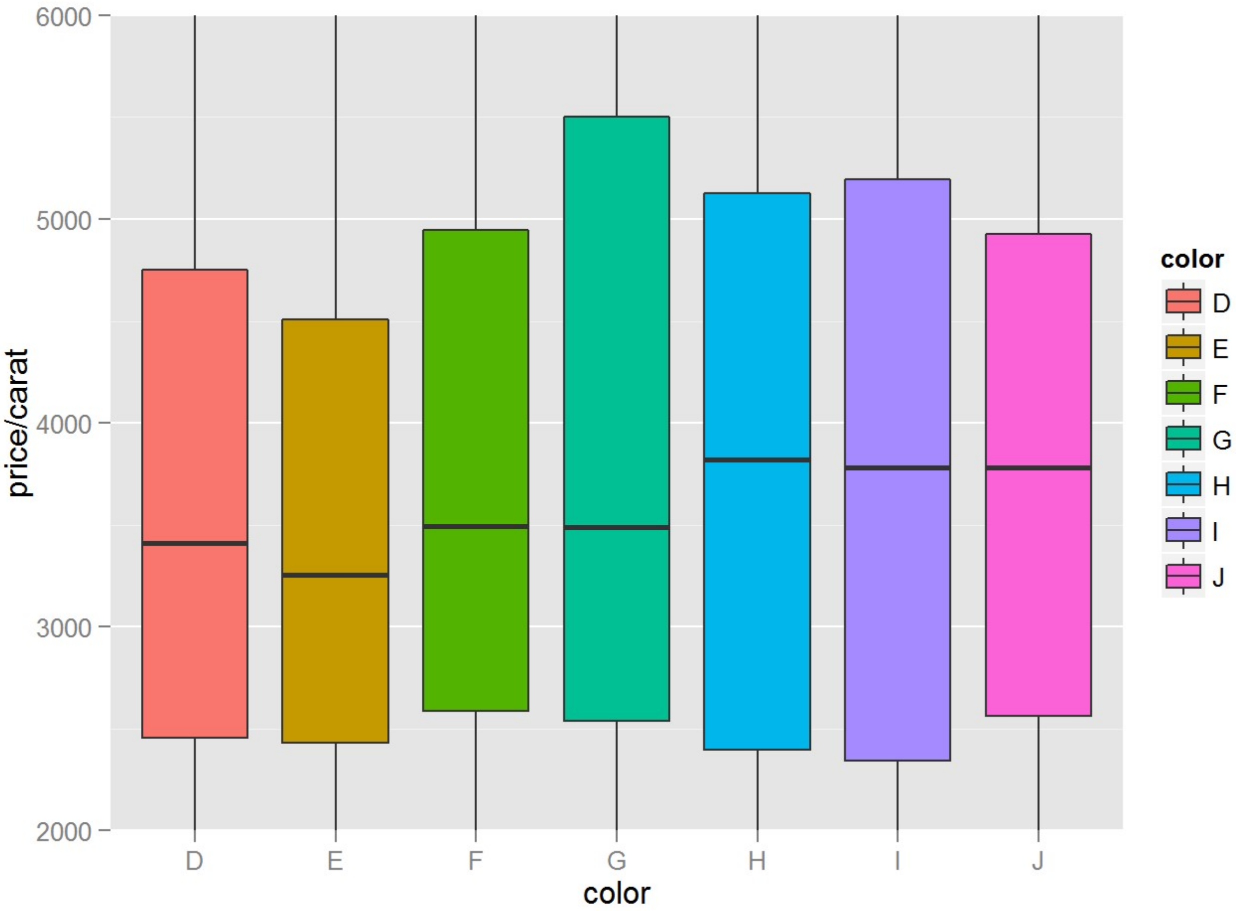
```
IQR(subset(diamonds, color == 'D')$price)
```

```
## [1] 3302.5
```

```
IQR(subset(diamonds, color == 'J')$price)
```

```
## [1] 5834.5
```

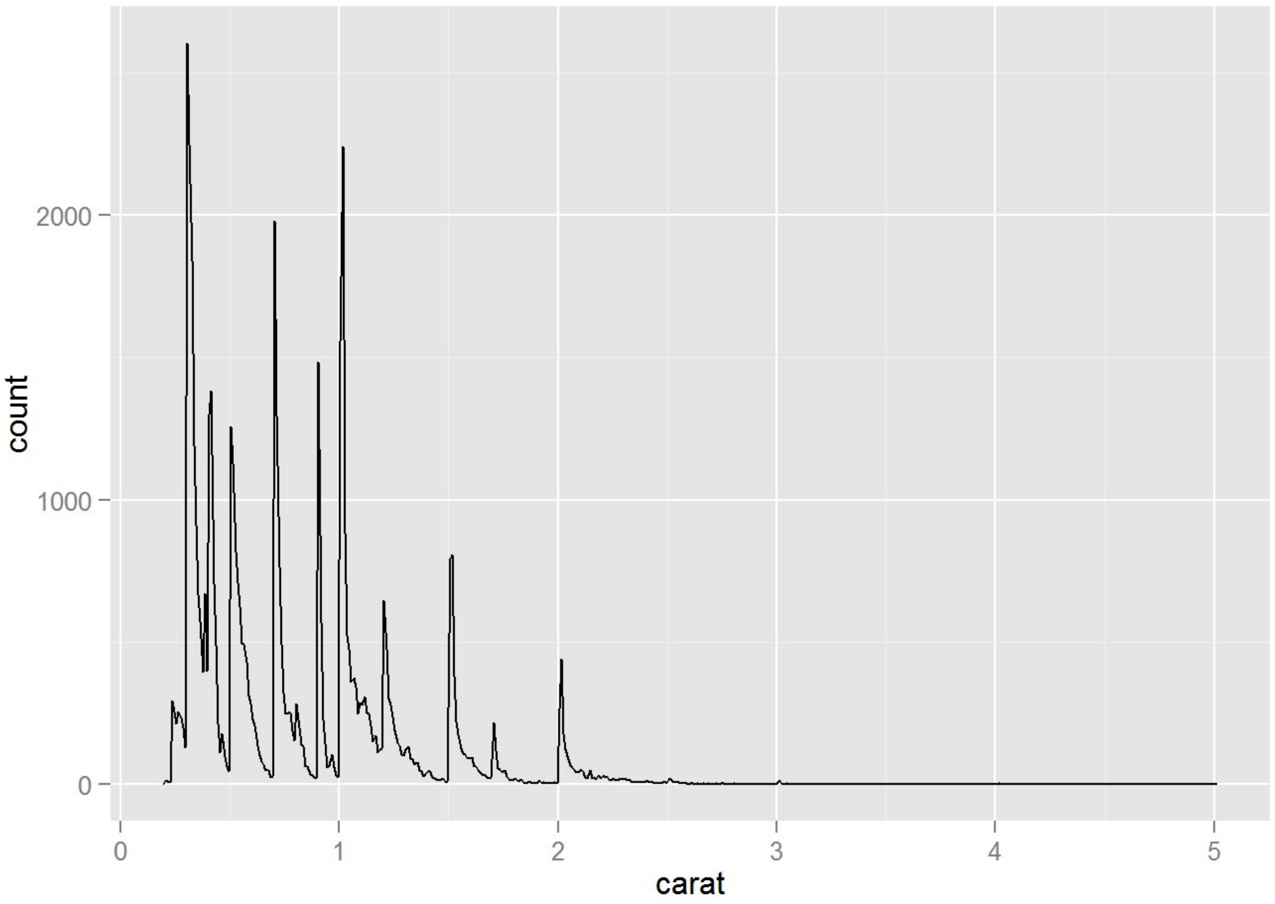
```
#Price per Carat Box Plots by Color
qplot(x = color, y = price/carat,
      data = subset(diamonds, !is.na(color)),
      geom = 'boxplot', fill=color) +
  coord_cartesian(ylim = c(2000, 6000))
```



```
ggsave('pricePerCaratBoxPlots.jpeg')
```

```
## Saving 7 x 5 in image
```

```
#Carat Frequency Polygon
qplot(x = carat, data = diamonds,
      binwidth = 0.01, geom = 'freqpoly')
```


count
2000
1000
0
0
1
2
3
4
5
carat