hw2

Weifeng She

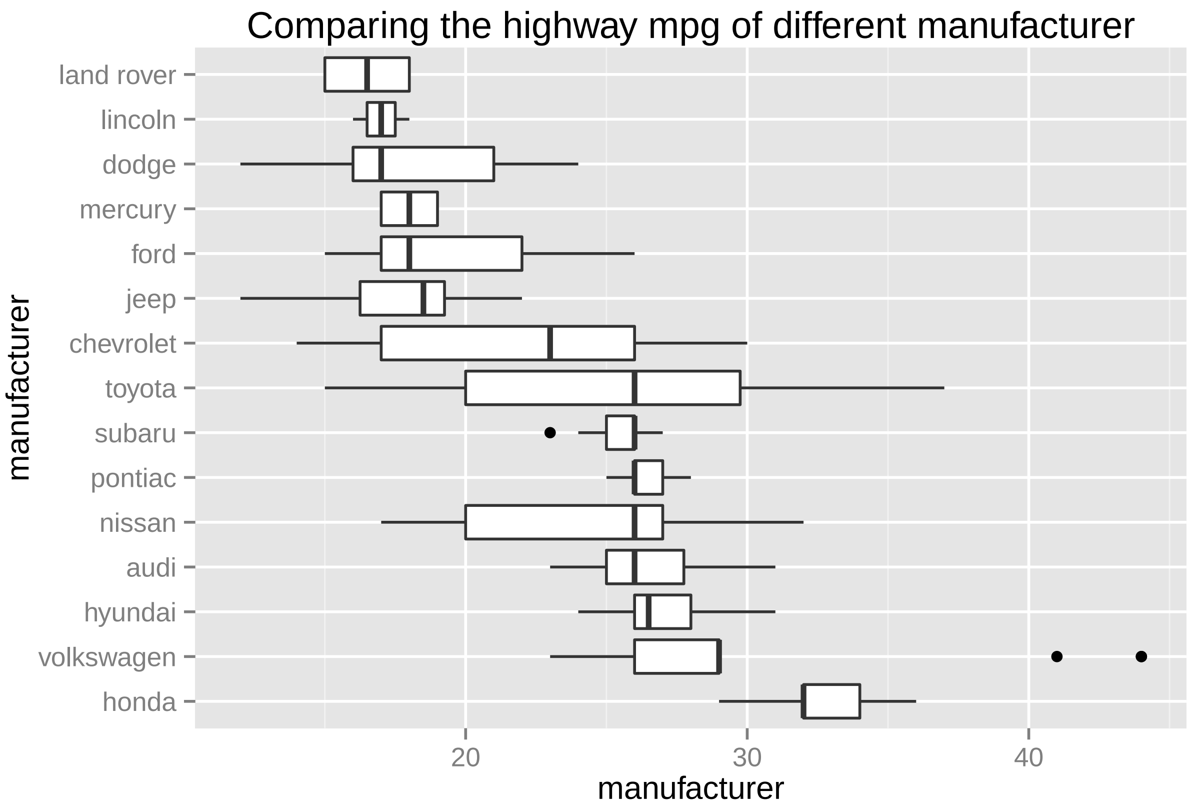
August 29, 2016

1.Using the mpg data, describe the relationship between highway mpg and car manufacturer. Describe which companies produce the most and least fuel e!cient cars, and display a graph supporting your conclusion.

```
library(ggplot2)
library(reshape)
data(mpg)
head(mpg)
```

```
##   manufacturer model displ year cyl      trans drv cty hwy fl   class
## 1         audi    a4   1.8 1999   4   auto(l5)   f  18  29  p compact
## 2         audi    a4   1.8 1999   4 manual(m5)   f  21  29  p compact
## 3         audi    a4   2.0 2008   4 manual(m6)   f  20  31  p compact
## 4         audi    a4   2.0 2008   4   auto(av)   f  21  30  p compact
## 5         audi    a4   2.8 1999   6   auto(l5)   f  16  26  p compact
## 6         audi    a4   2.8 1999   6 manual(m5)   f  18  26  p compact
```

```
ggplot(mpg, aes(reorder(manufacturer, -hwy, median), hwy)) +
        geom_boxplot() +
        coord_flip() +
        scale_x_discrete("manufacturer") +
        ggtitle("Comparing the highway mpg of different manufacturer") +
        ylab("manufacturer") +
        xlab("high mpg")
```



When we graph the boxplot of highway mpg versus each manufacturer and then order the graph from the lowest to highest of mpg median. We can see Land rover has the lowest mpg and Honda has the highest mpg.

2. Using the mpg data, explore the three-way relationship between highway mpg, city mpg, and model class. What are your observations? Display a graph supporting these observations.

```
qplot(x = hwy,
      y = cty,
      facets = .~class,
      data = mpg,
      main = "High way mpg vs. City mpg by class") +
        stat_smooth(se=FALSE)
```

```
## Warning in loop_apply(n, do.ply): span too small. fewer data values than
## degrees of freedom.
```

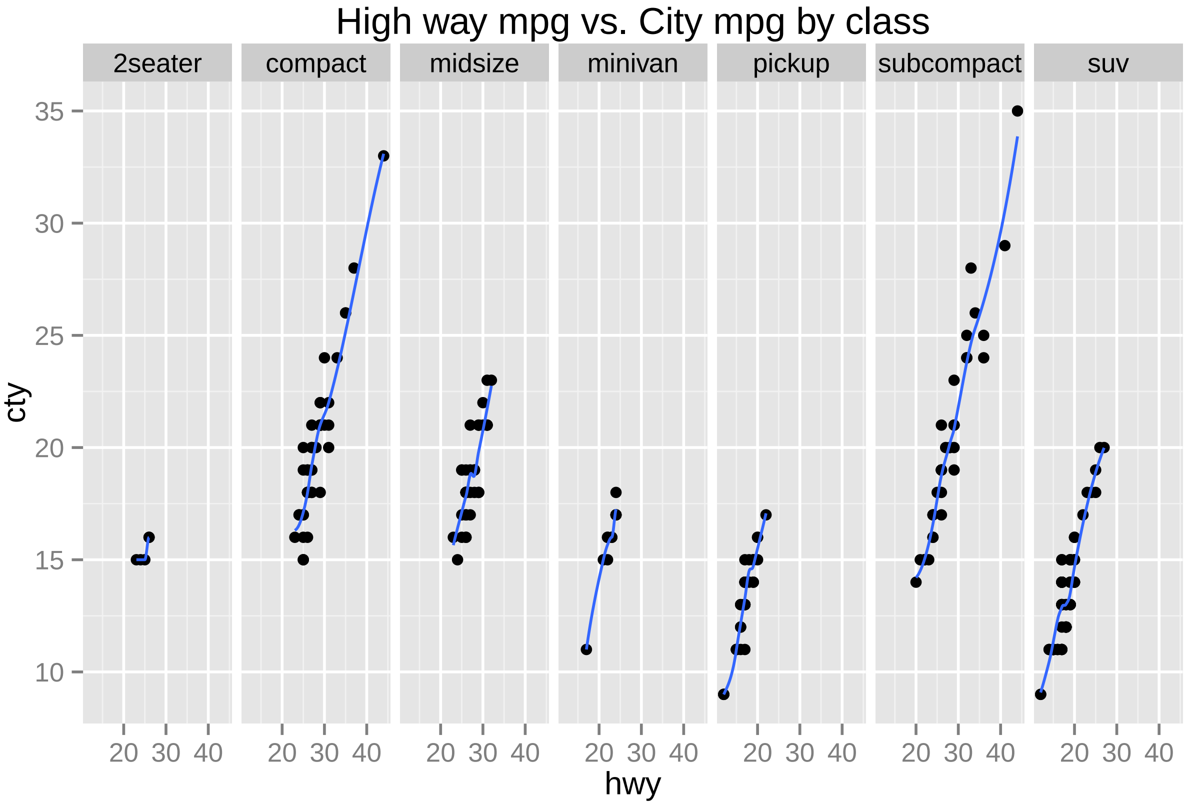
```
## Warning in loop_apply(n, do.ply): pseudoinverse used at 22.985
```

```
## Warning in loop_apply(n, do.ply): neighborhood radius 2.015
```

```
## Warning in loop_apply(n, do.ply): reciprocal condition number 0
```

```
## Warning in loop_apply(n, do.ply): There are other near singularities as
## well. 1.0302
```

```
## Warning in loop_apply(n, do.ply): pseudoinverse used at 24.035
## Warning in loop_apply(n, do.ply): neighborhood radius 2.035

## Warning in loop_apply(n, do.ply): reciprocal condition number 7.8765e-17

## Warning in loop_apply(n, do.ply): There are other near singularities as
## well. 1
```



From the scatter plot of city mpg vs. highway mpg by class, except for 2seater only has very few points, we can see for all other classes of cars, city mpg is positively related with highway mpg.

3. What are the pros and cons of using a histogram vs a box plot? Which one will you prefer for what purpose?

Histograms are useful, easy to graph one dimensional data. It could be used for discrete, continuous and even unordered data. With proper bin width, we can know the distribution of the data. But it could not be used for multiple categorical data and also little information is displayed.

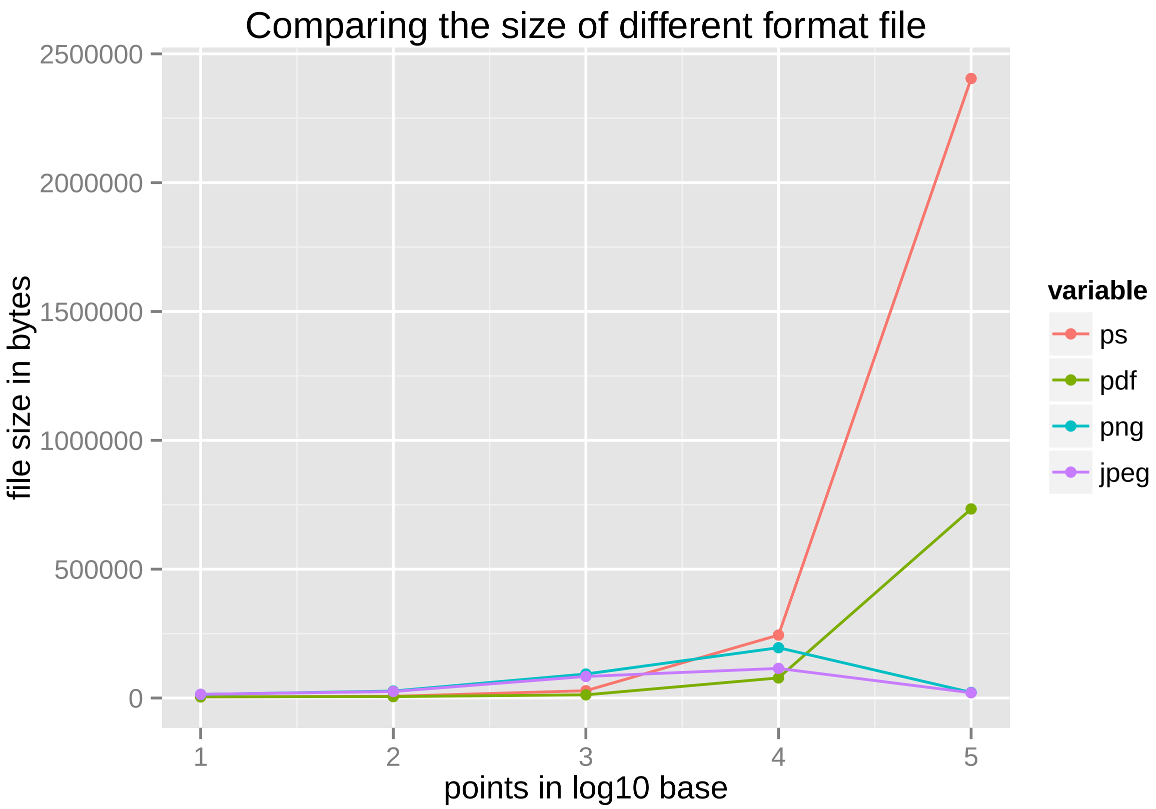
Boxplot is an alternative of histogram. The centerline of the box is the median and edges correspond to the first and third quantiles. Whiskers extend out to 1.5 times the IQR(inner quantile region) and points outside of the whiskers denote the outliers. Box plot does not contain the information about the data distribution. But it is easier to read the median and IQR.

4. Generate two sets of N random points using the function runif and display a corresponding scatter plot. If you save the file to disk, what is the resulting file size for the following file formats: ps, pdf, jpeg, png? How do these values scale with increasing N?

```
points <- c(10, 100, 1000, 10000, 100000)

df <- data.frame(ps = numeric(0),
                 pdf = numeric(0),
                 png = numeric(0),
                 jpeg = numeric(0))
postfix <- c('.ps', '.pdf', '.png', '.jpeg')
functions <- c(postscript, pdf, png, jpeg)

for(i in seq_along(points)) {
  x = runif(points[[i]], min = 0, max = 100)
  y = runif(points[[i]], min = 0, max = 100)
  for(j in seq_along(functions)){
      filename <- paste("plot", toString(i), postfix[[j]], sep = '')

      functions[[j]](filename)
      plot(x, y)
      dev.off()
      df[i, j] <-  file.info(filename)$size
}
}
df$points <- points
df
```

```
##        ps    pdf    png   jpeg points
## 1    4450   4587  13810  14305  1e+01
## 2    6663   5380  27197  25288  1e+02
## 3   28266  12309  93151  83663  1e+03
## 4  244266  77942 195178 114902  1e+04
## 5 2404266 733774  21701  20391  1e+05
```

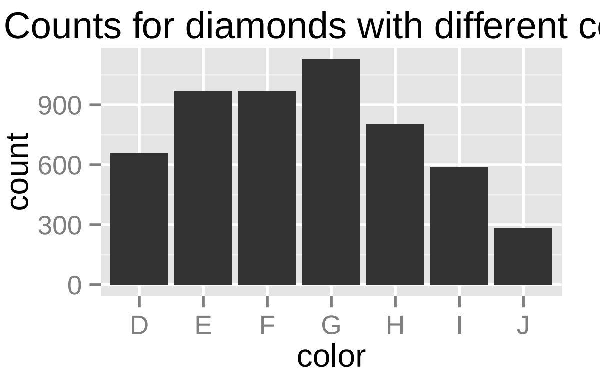
```
meltdf <- melt(df, id = c("points"))

ggplot(meltdf, aes(x=log10(points), y=value, color = variable)) +
  geom_line() +
  geom_point() +
  ggtitle("Comparing the size of different format file") +
  xlab("points in log10 base") +
  ylab("file size in bytes")
```

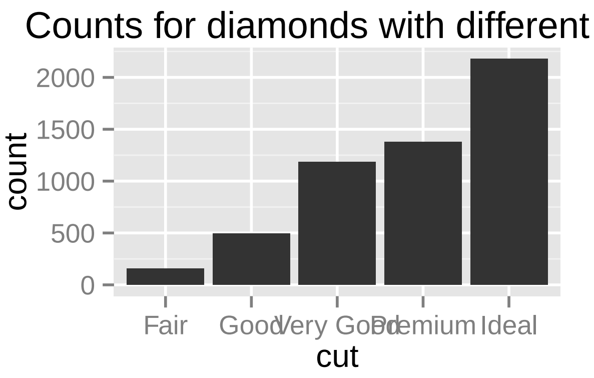


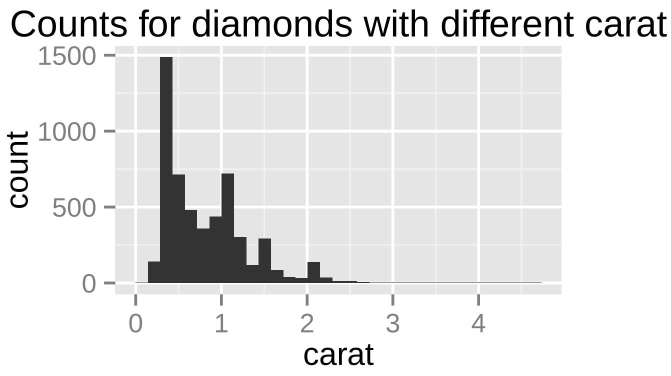
Here I chose different amount of uniform distributed points from 10 to 100000. With the expotientially increasing of the points before 10000, generally the size of file is also inceasing linearly. But from 10000 to 100000, the size of png and jpeg file did not crease, only ps and pdf file increased exponentially.

5. The diamonds dataset within ggplot2 contains 10 columns (price, carat, cut, color, etc.) for 53940 diferent diamonds. Type help(diamonds) for more information. Plot histograms for color, carat, and price, and comment on their shapes. Investigate the three-way relationship between price, carat, and cut. What are your conclusions? Provide graphs that support your conclusions. If you encounter computational difculties, consider using a smaller dataframe whose rows are sampled from the original diamonds dataframe. Use the function sample to create a subset of indices that may be used to create the smaller dataframe.

```
data(diamonds)
set.seed(1)
diamondsSubset = diamonds[sample(dim(diamonds)[1], dim(diamonds)[1]/10),]
mfrow = c(1, 3)
p1 <- qplot(x = color, data = diamondsSubset, main = "Counts for diamonds with different color")
p2 <- qplot(x = carat, data = diamondsSubset, main = "Counts for diamonds with different carat")
p3 <- qplot(x = cut, data = diamondsSubset, main = "Counts for diamonds with different cut")

multiplot(p1, p2, p3, cols=2)
```



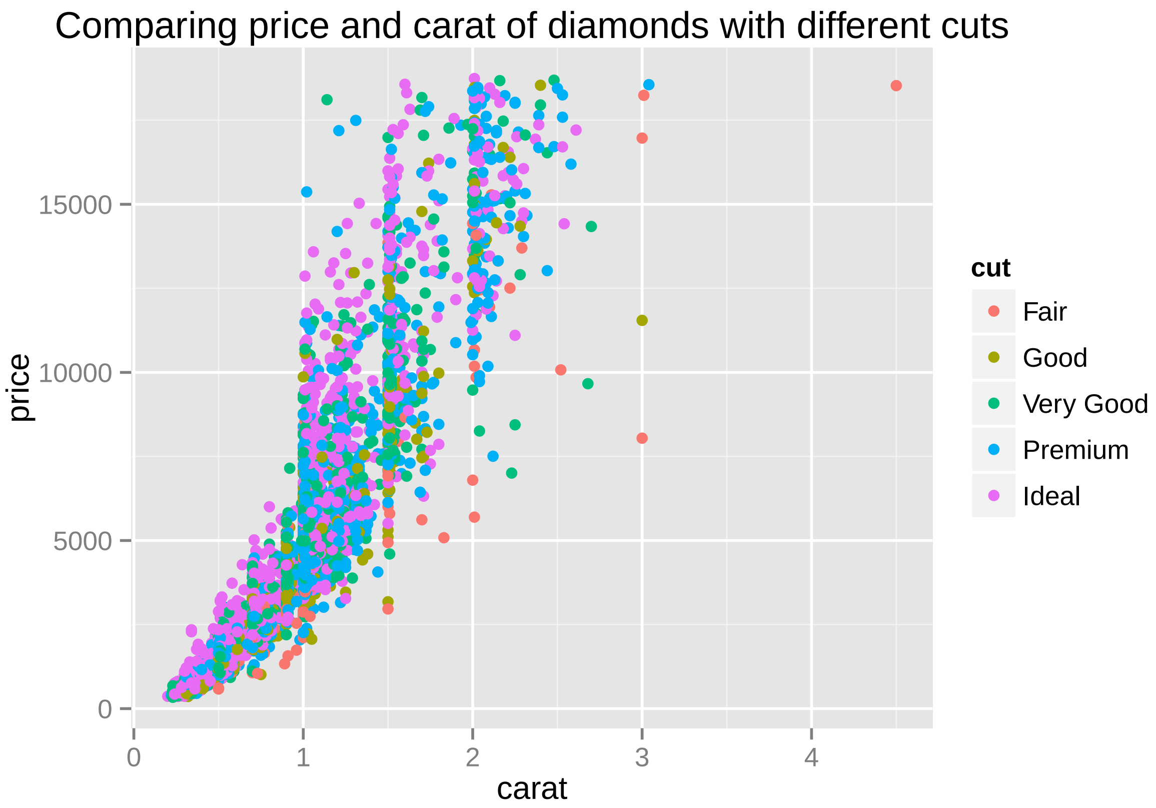




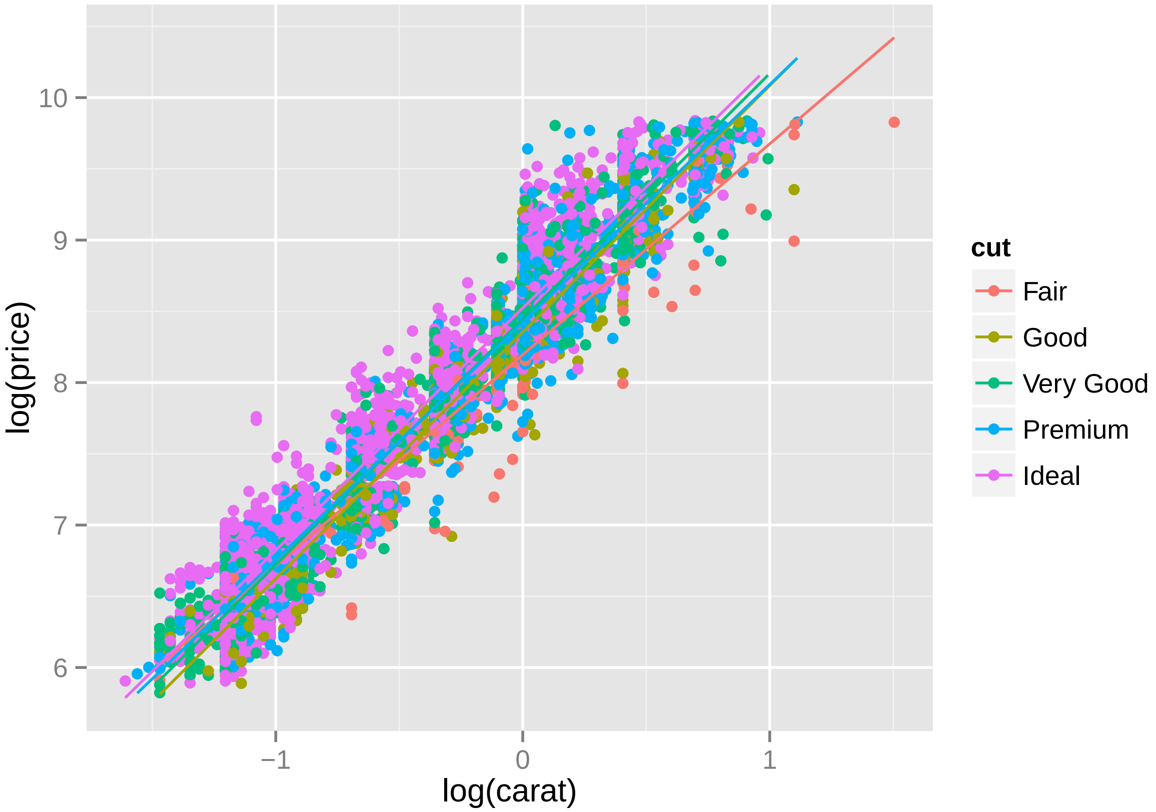
```
#ggplot(diamonds, aes(carat, price, color= cut)) + geom_point()
```

We can get following conclusions from the above three histgrams for color, carat and cut of the diamonds: (a) there are the least amount of diamond with worst color(J), the most amount for G and the best color(D) is in the middle. (b) The lower the carat, the higher amount the diamonds. (c) The increasing the quality of the cut, the higher amount of the diamonds.

```
ggplot(diamondsSubset, aes(carat, price, color= cut)) +
  geom_point()  +
  ggtitle('Comparing price and carat of diamonds with different cuts')
```



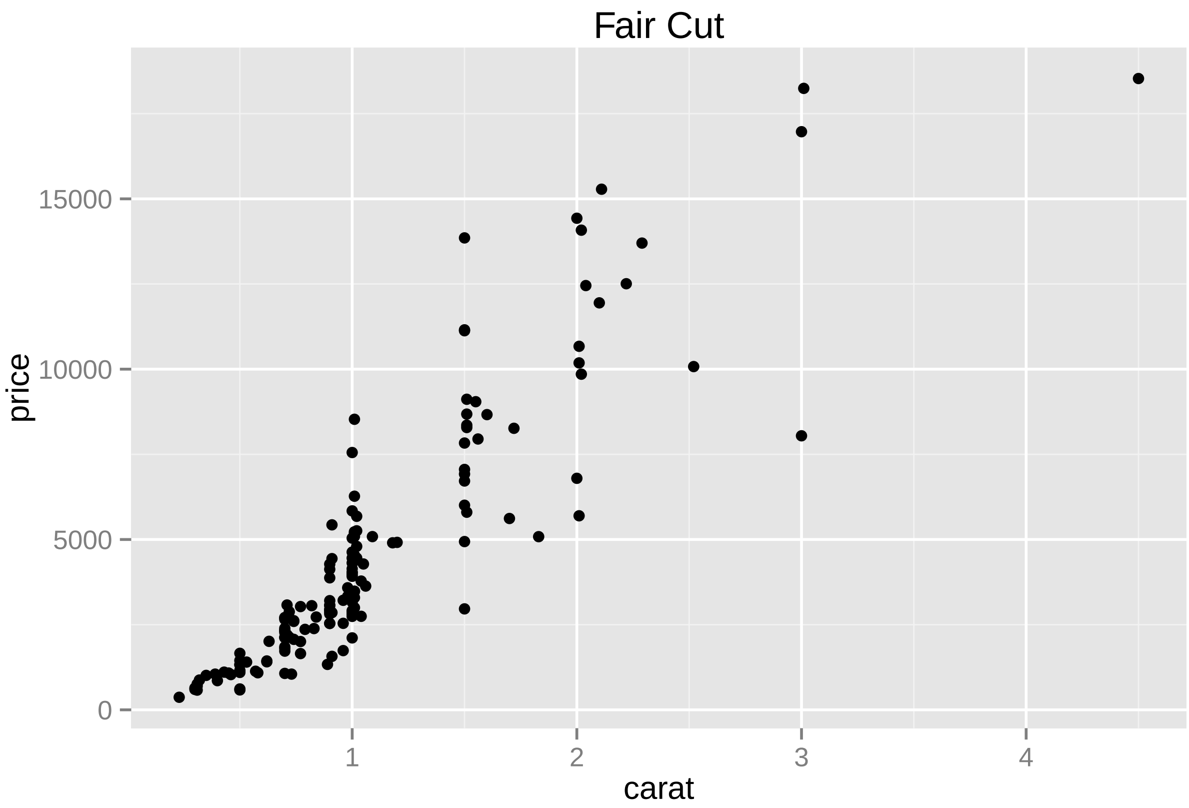
```
ggplot(diamondsSubset, aes(log(carat), log(price), color= cut)) +
  geom_point()  +
  stat_smooth(method = "lm",se = F)
```



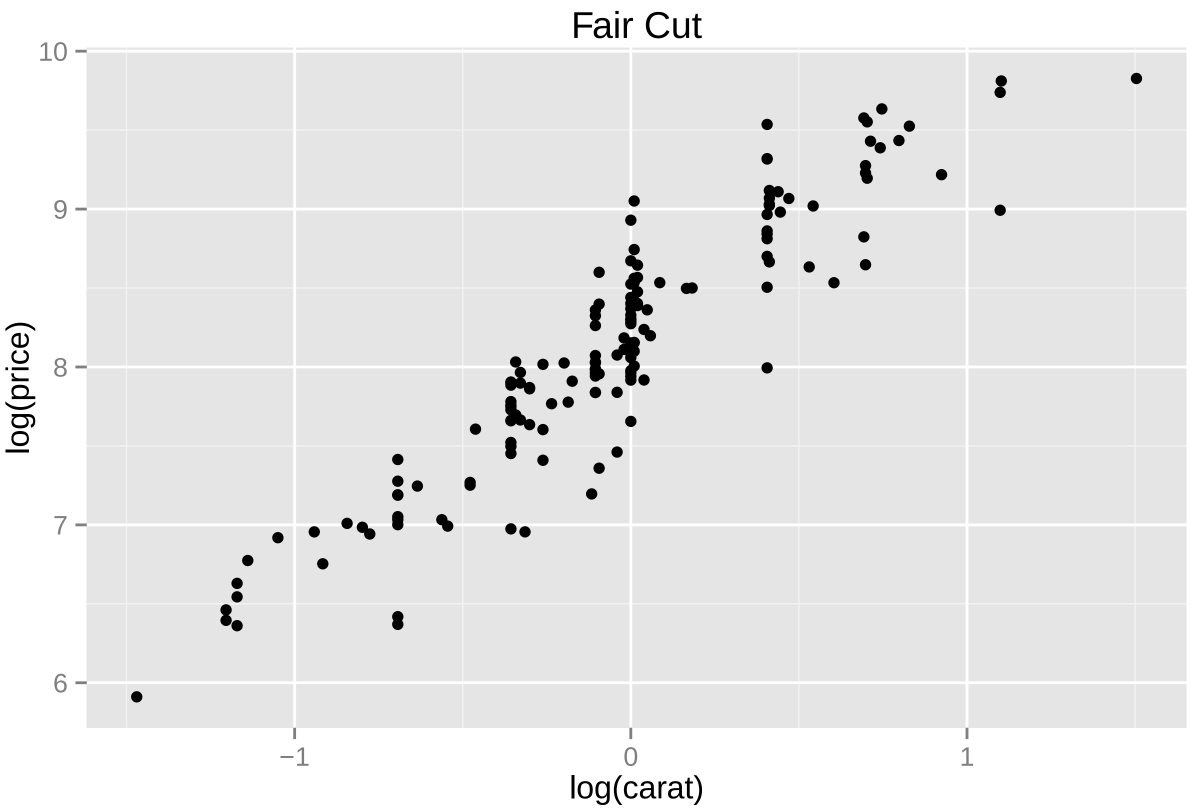
```
  ggtitle('Comparing price and carat of diamonds with different cuts')
```

```
## $title
## [1] "Comparing price and carat of diamonds with different cuts"
##
## attr(,"class")
## [1] "labels"
```

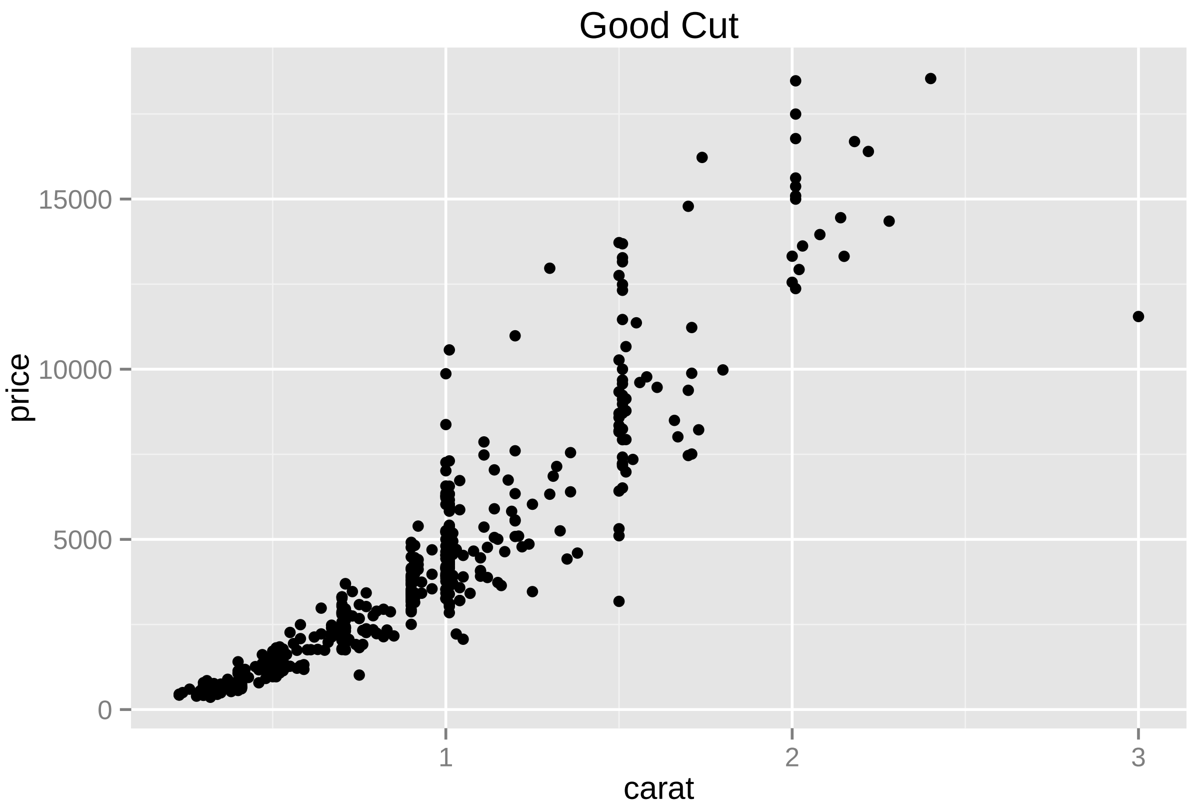
```
fair <- subset(diamondsSubset, cut == "Fair")

ggplot(fair, aes(carat, price)) +
  geom_point() +
  ggtitle("Fair Cut")
```



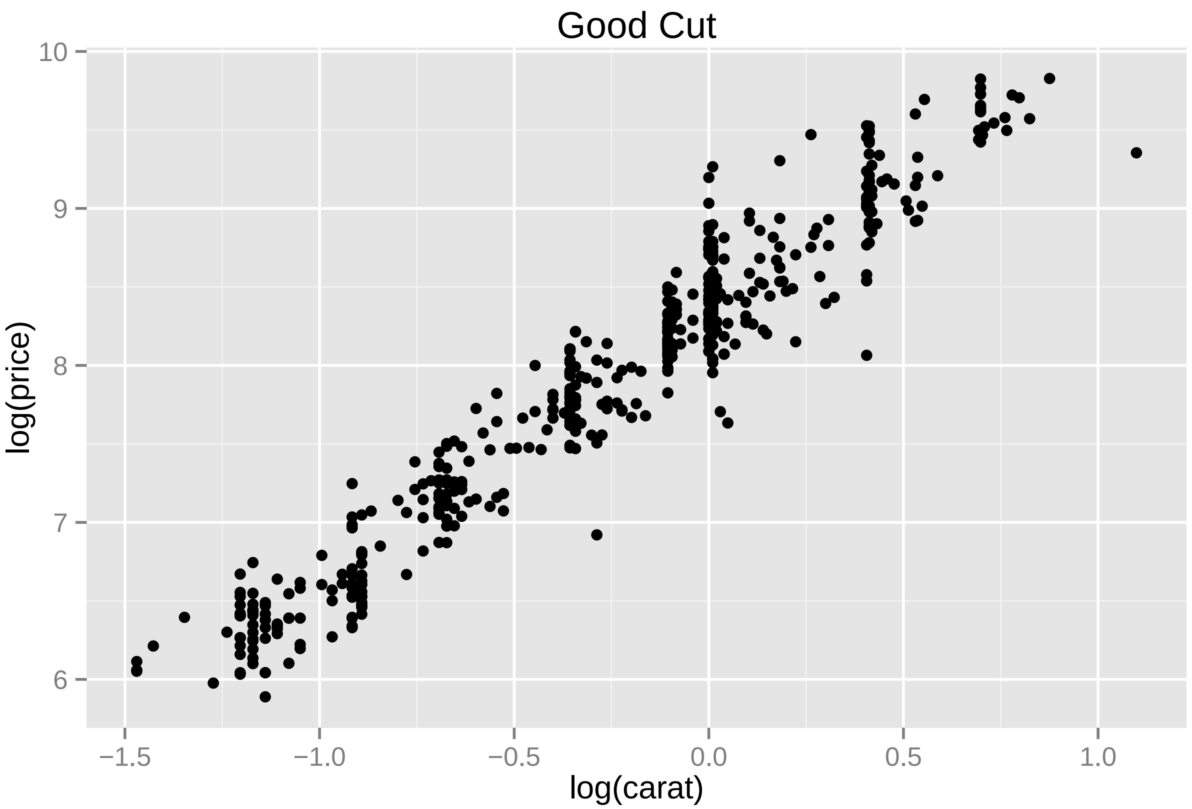
```
ggplot(fair, aes(log(carat), log(price))) +
  geom_point() +
  ggtitle("Fair Cut")
```



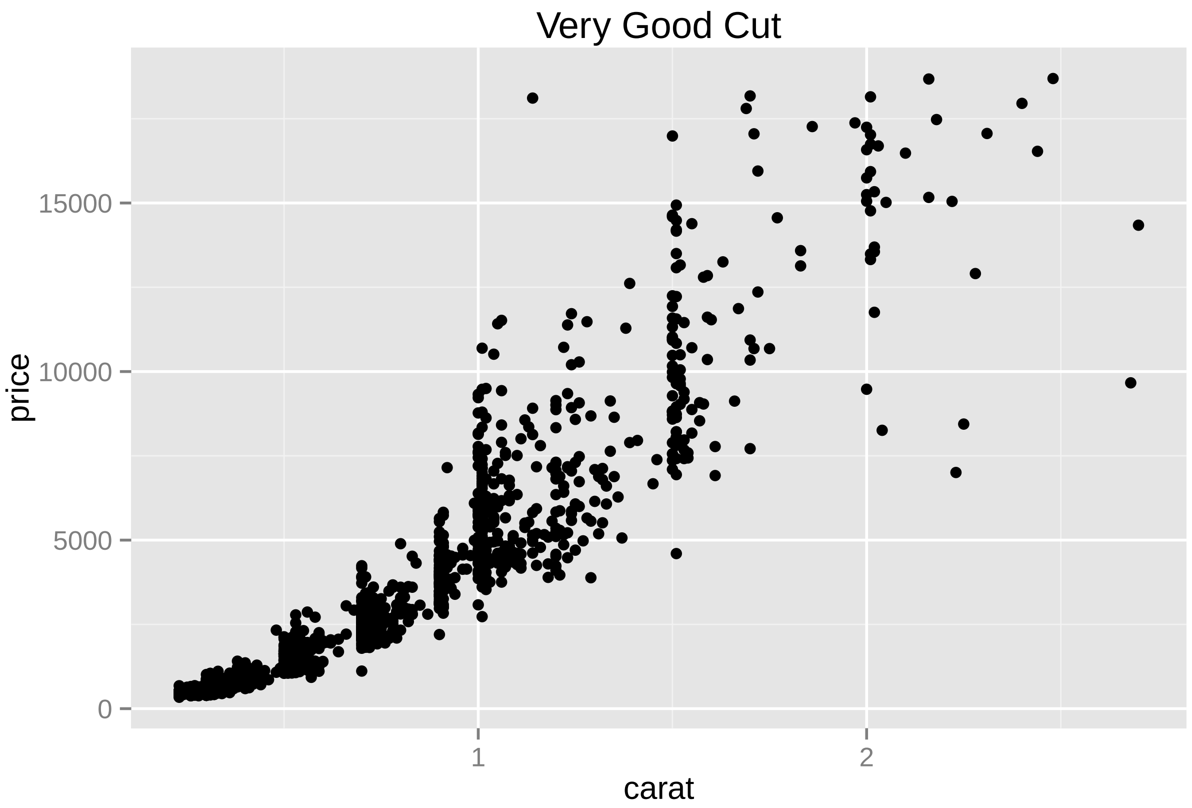
```
good <- subset(diamondsSubset, cut == "Good")

ggplot(good, aes(carat, price)) +
  geom_point() +
  ggtitle("Good Cut")
```



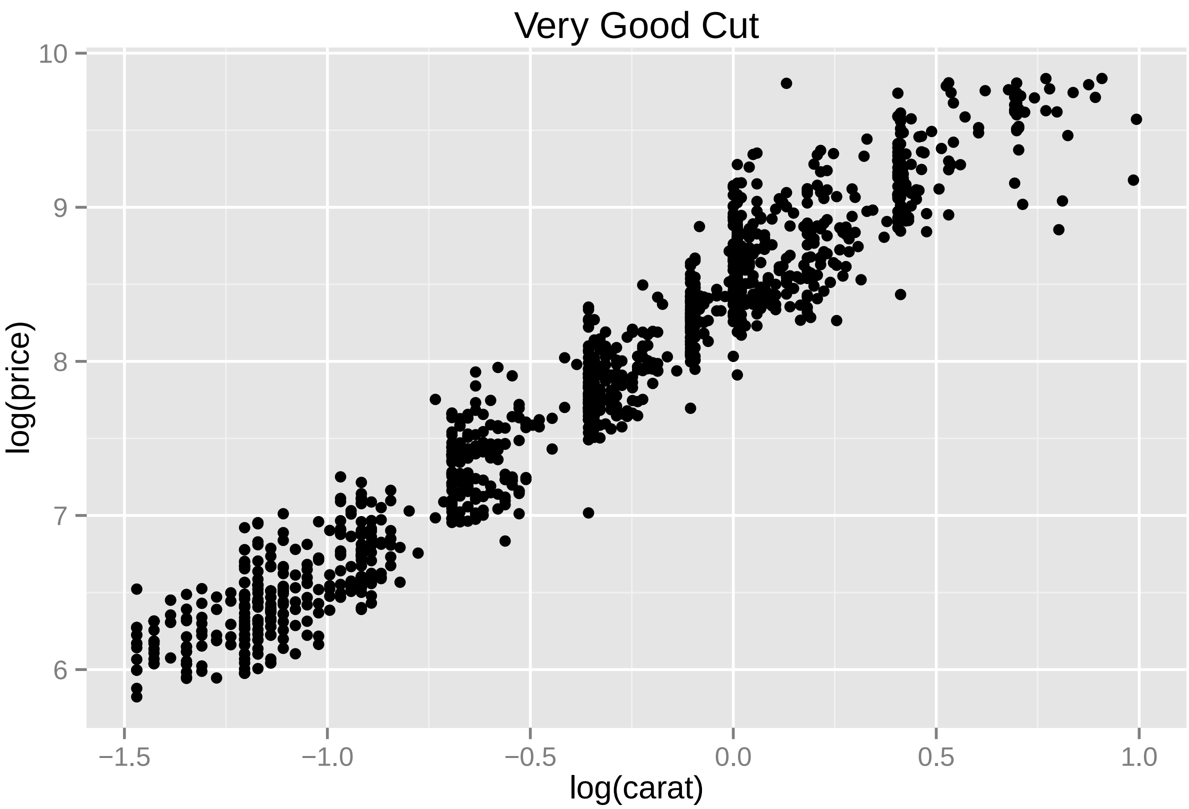
```
ggplot(good, aes(log(carat), log(price))) +
  geom_point() +
  ggtitle("Good Cut")
```



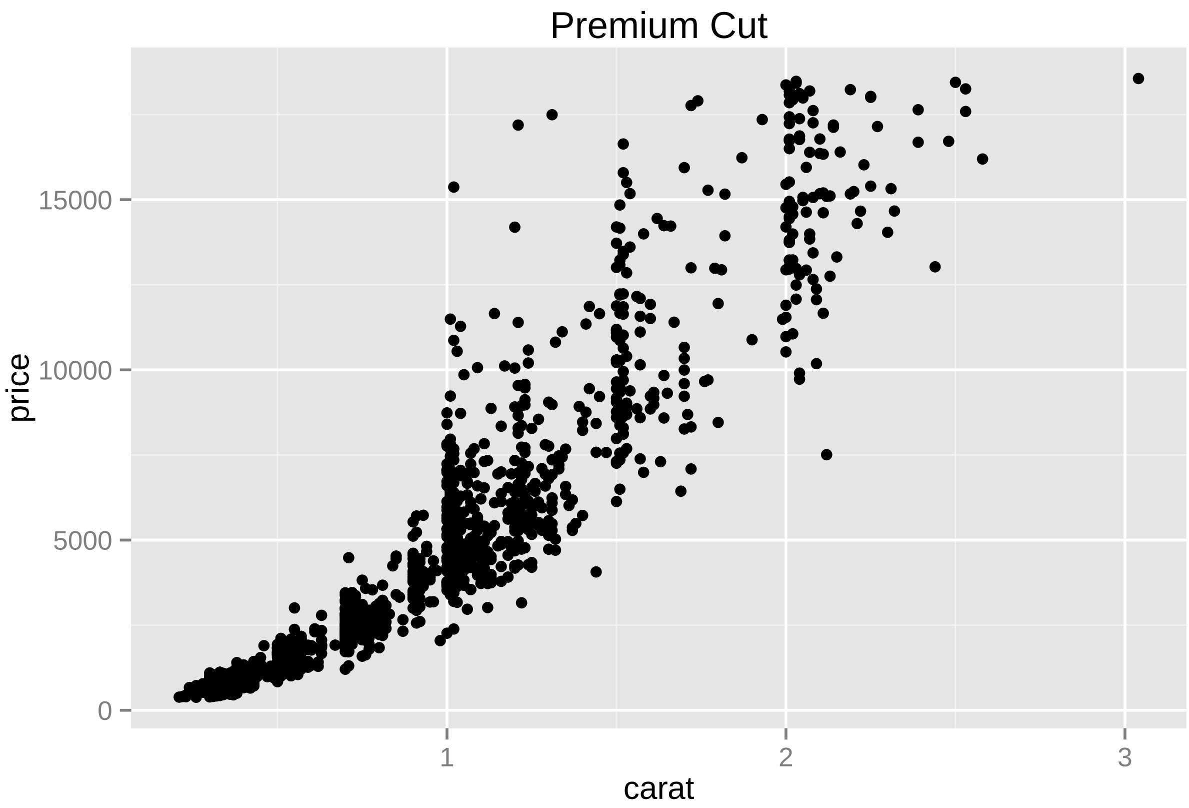
```
verygood <- subset(diamondsSubset, cut == "Very Good")

ggplot(verygood, aes(carat, price)) +
  geom_point() +
  ggtitle("Very Good Cut")
```



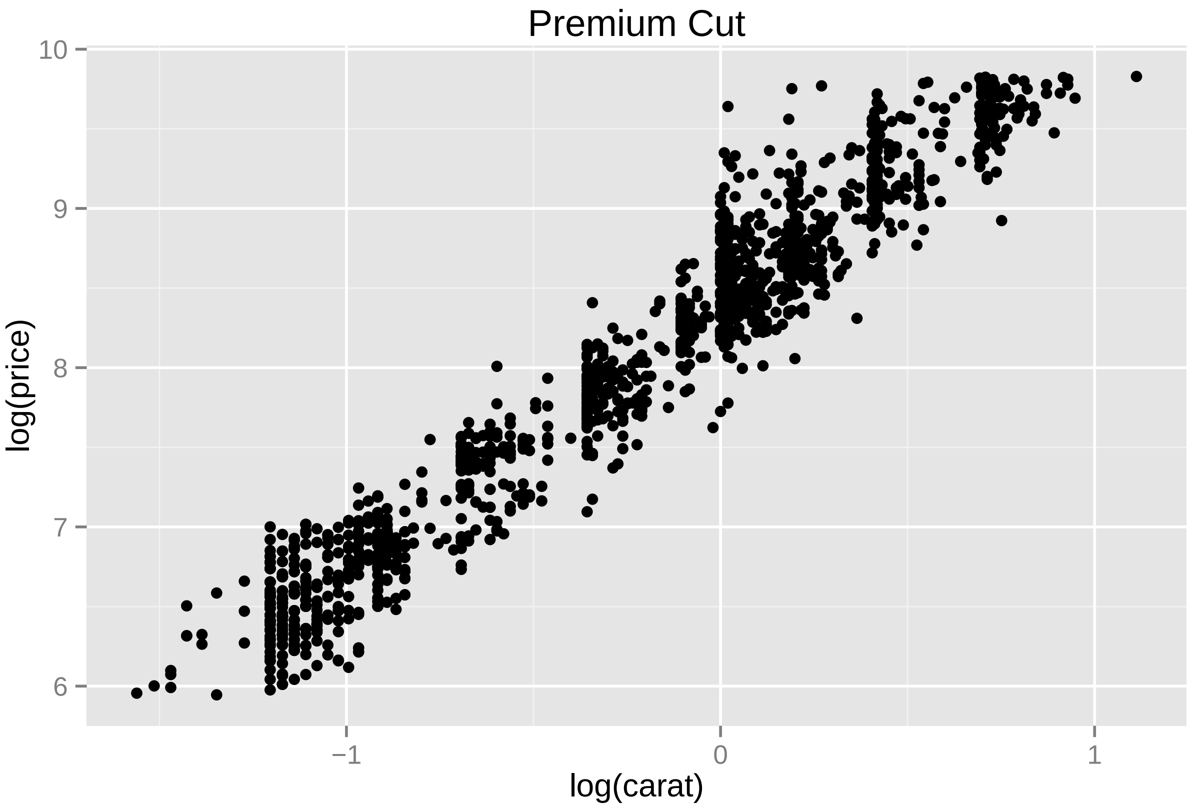
```
ggplot(verygood, aes(log(carat), log(price))) +
  geom_point() +
  ggtitle("Very Good Cut")
```



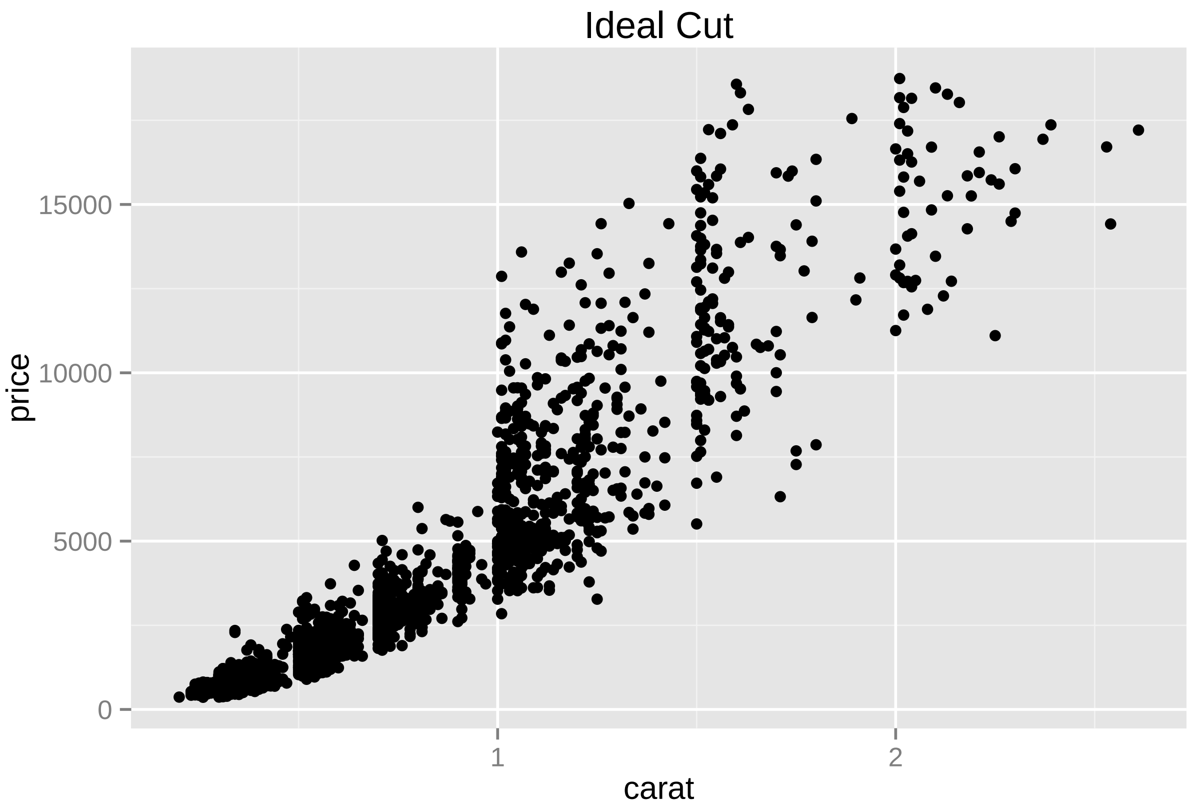
```
premium <- subset(diamondsSubset, cut == "Premium")

ggplot(premium, aes(carat, price)) +
  geom_point() +
  ggtitle("Premium Cut")
```



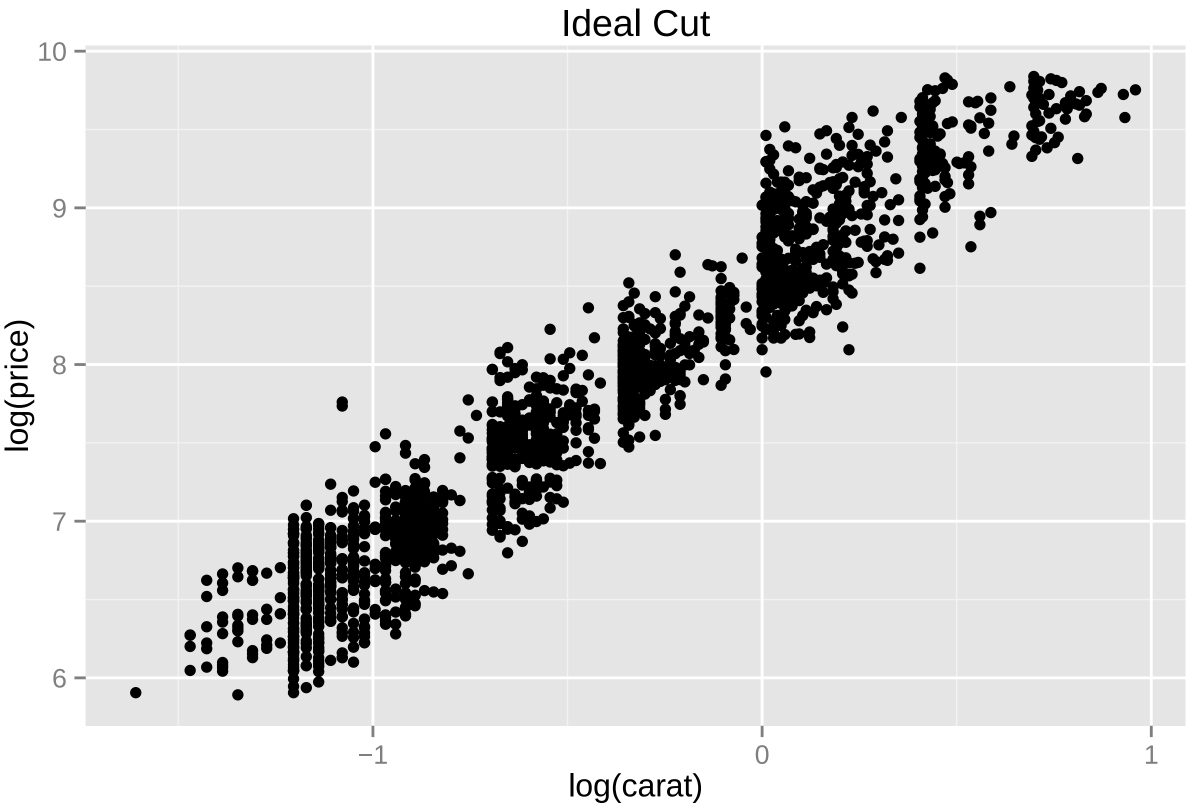
```
ggplot(premium, aes(log(carat), log(price))) +
  geom_point() +
  ggtitle("Premium Cut")
```



```
ideal <- subset(diamondsSubset, cut == "Ideal")
ggplot(ideal, aes(carat, price)) +
  geom_point() +
  ggtitle("Ideal Cut")
```



```
ggplot(ideal, aes(log(carat), log(price))) +
  geom_point() +
  ggtitle("Ideal Cut")
```



Here we subset 10% of the data and graph the price vesus the carat for all cuts. When we look at the relationship between carat an price for all cuts, we can see price increases exponentially with the increase of carats. Then when we graph the log of carat with the log of price, we can see there is a clear linear relationship between them. This fact also holds true when we graph each cut individually. Also from the linear regression line, we can see the best quality of the cut(Ideal) has the highest price and the worst quality of the cut (Fair) has the lowest price.