Making Statistical Graphics

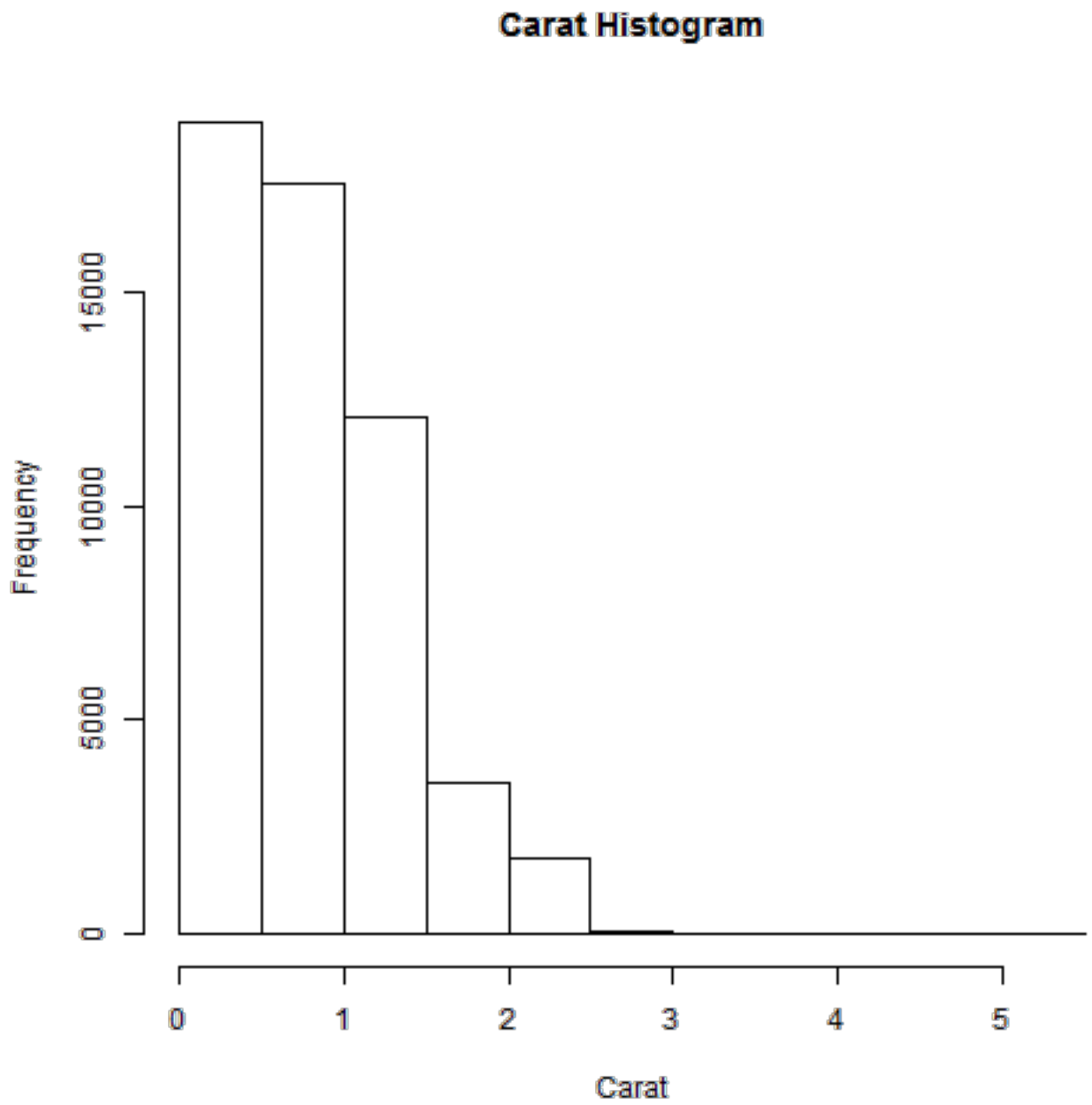
Find the diamonds data

```
require(ggplot2)
data(diamonds)
head(diamonds)
```

```
##   carat       cut color clarity depth table price    x    y    z
## 1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
## 2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
## 3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
## 4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
## 5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
## 6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48
```

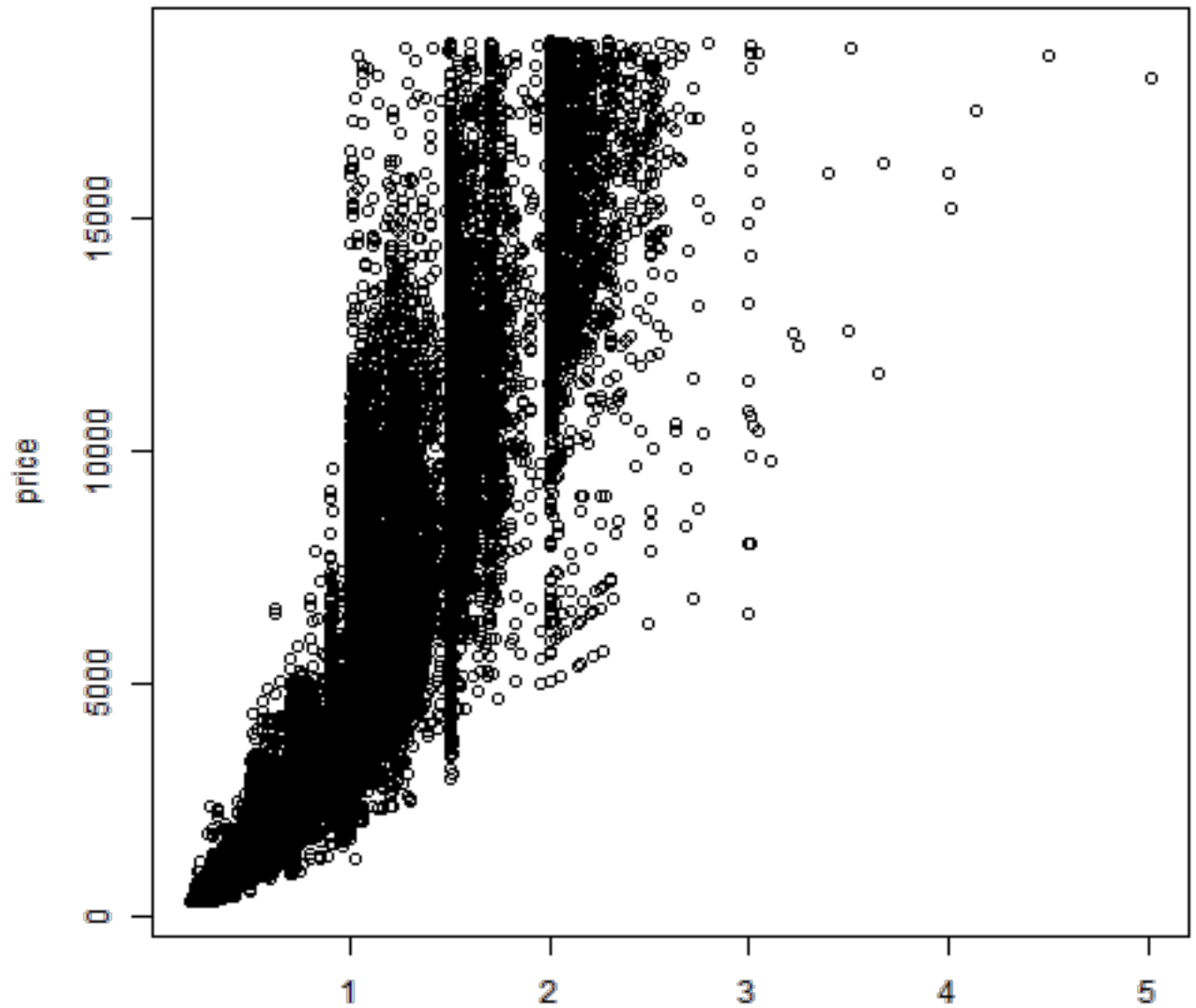
base Histogram

```
hist(diamonds$carat, main = "Carat Histogram", xlab = "Carat")
```



Base Scatterplot

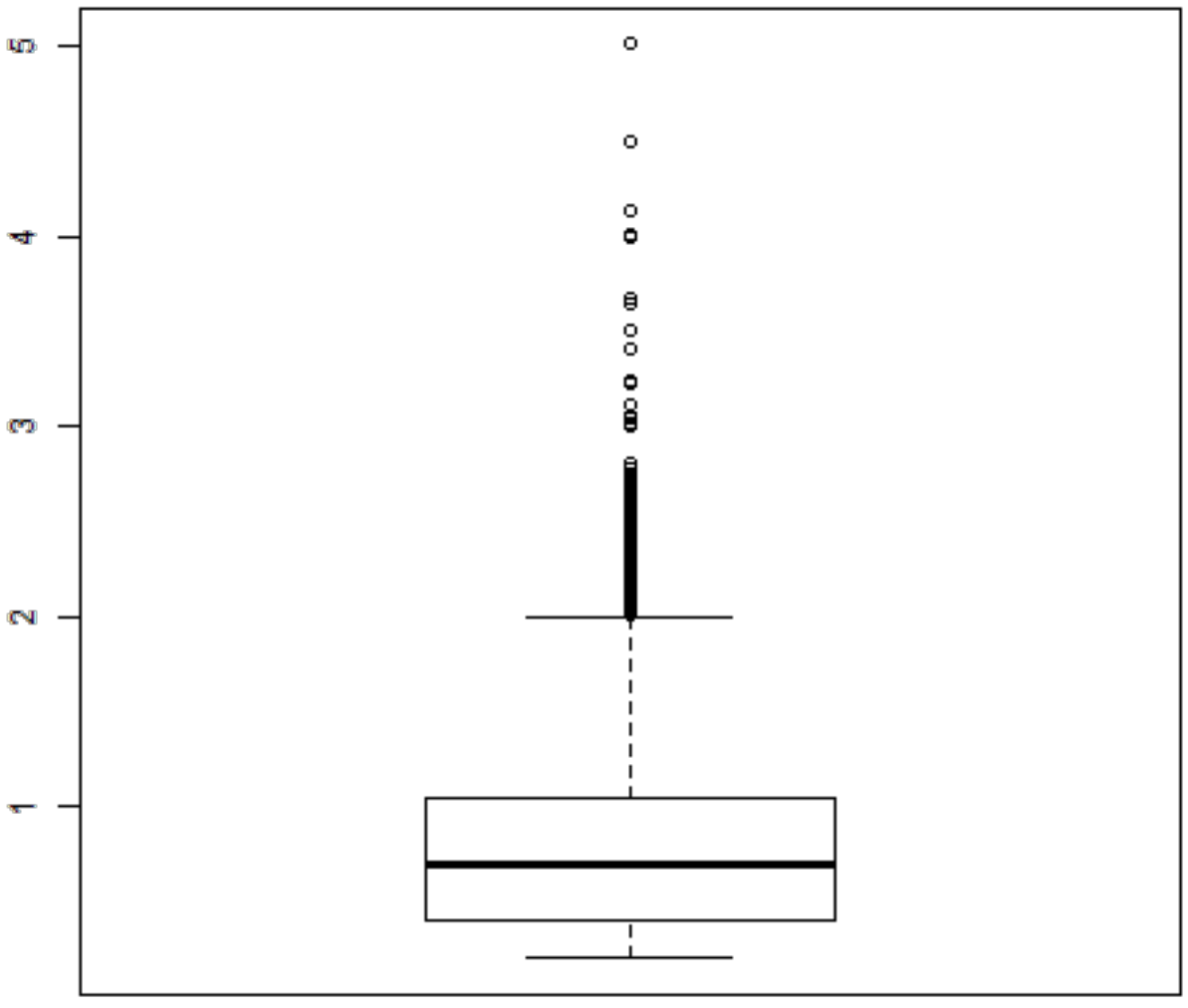
```
plot(price ~ carat, data = diamonds)
```




```
plot(diamonds$carat, diamonds$price)
```

Base Boxplot

```
boxplot(diamonds$carat)
```



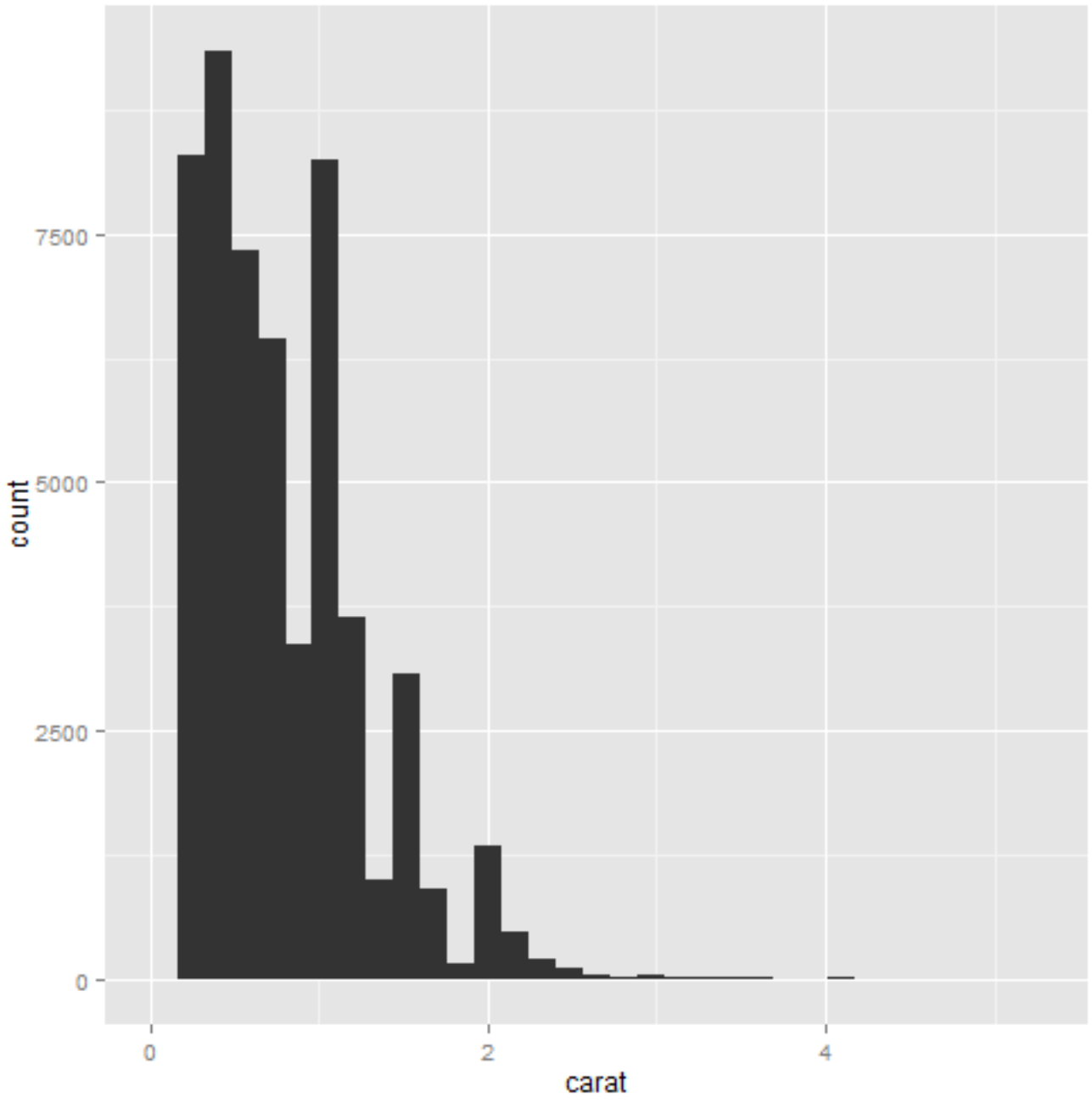
Get Familiar with ggplot2

the whole + paradigm grammer of graphics

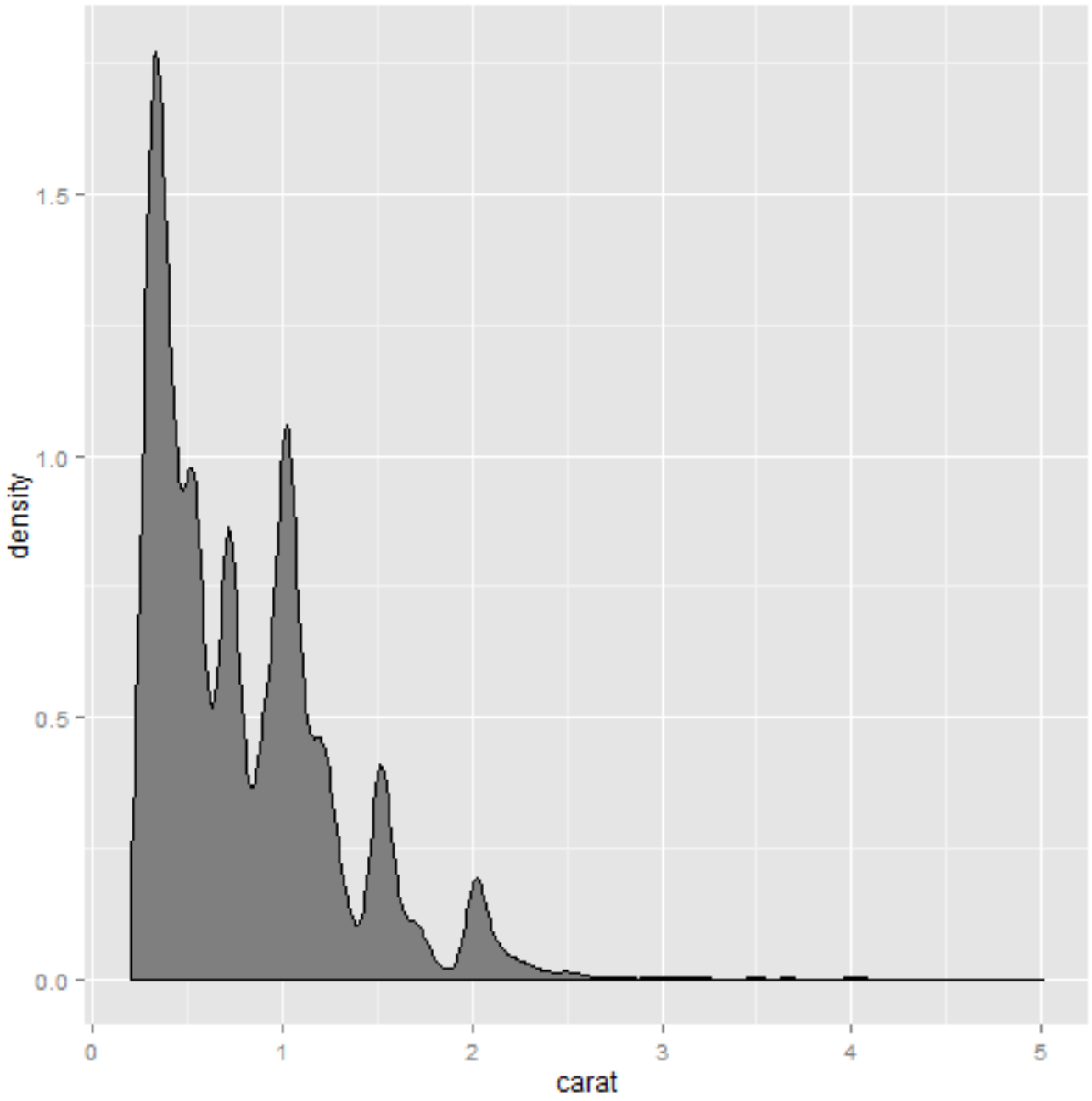
ggplot2 Histograms and Density Plots

```
ggplot(data = diamonds) + geom_histogram(aes(x = carat))
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to
adjust this.
```

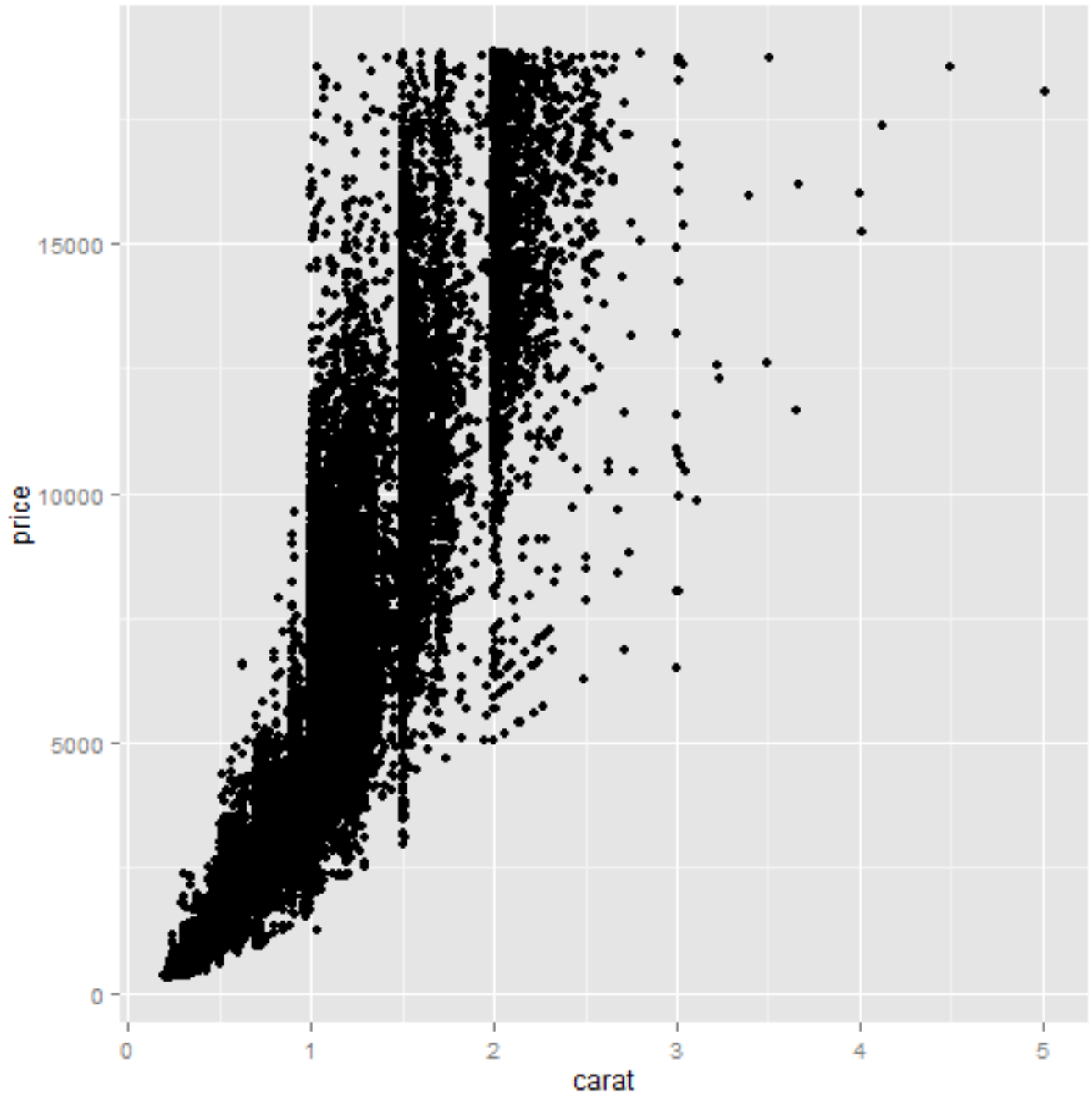



```
ggplot(data = diamonds) + geom_density(aes(x = carat), fill =
"grey50")
```



ggplot2 Scatterplots

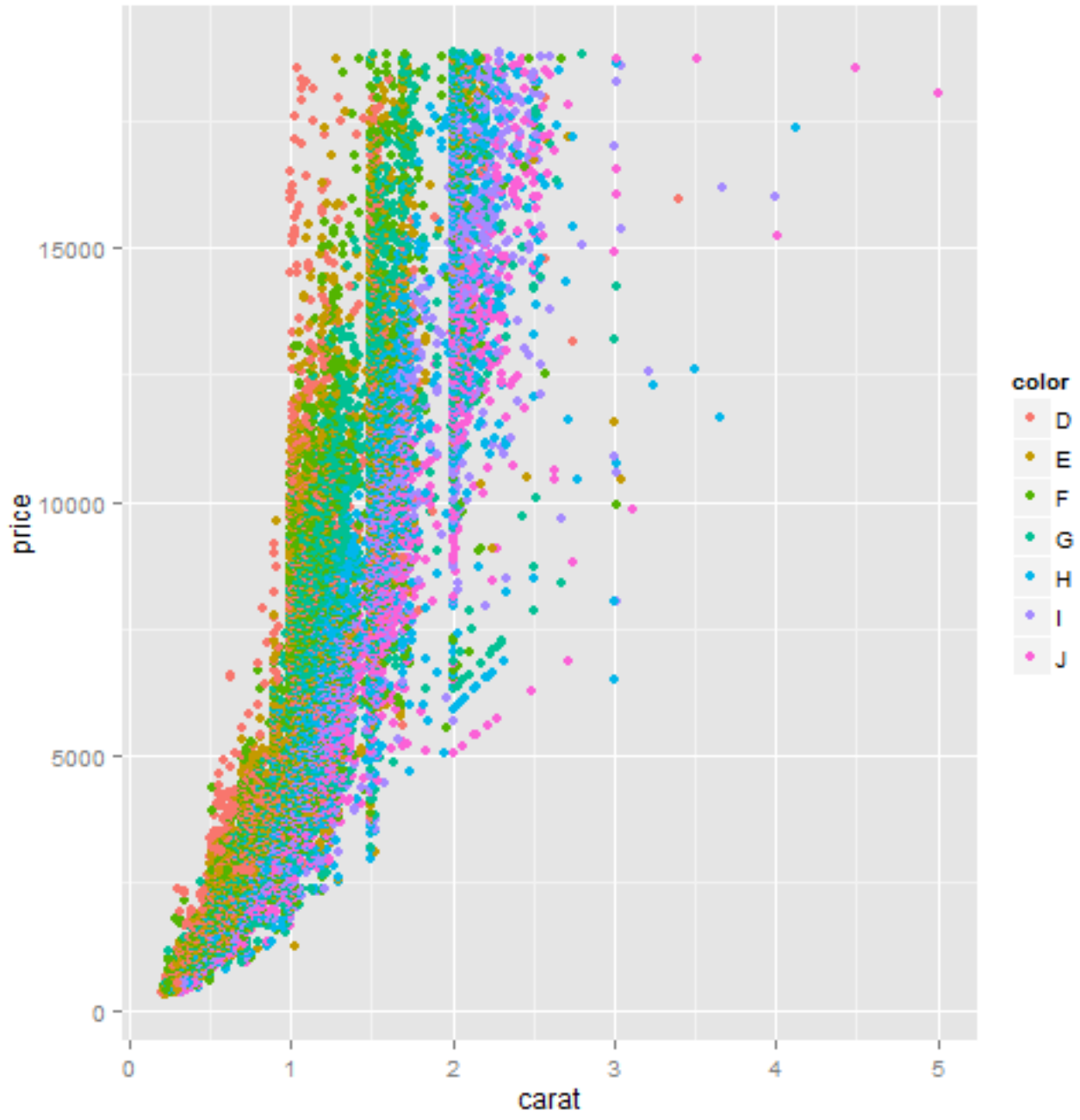
```
ggplot(diamonds, aes(x = carat, y = price)) + geom_point()
```




```
# save basics of ggplot object to a variable
g <- ggplot(diamonds, aes(x = carat, y = price))
```

With Color

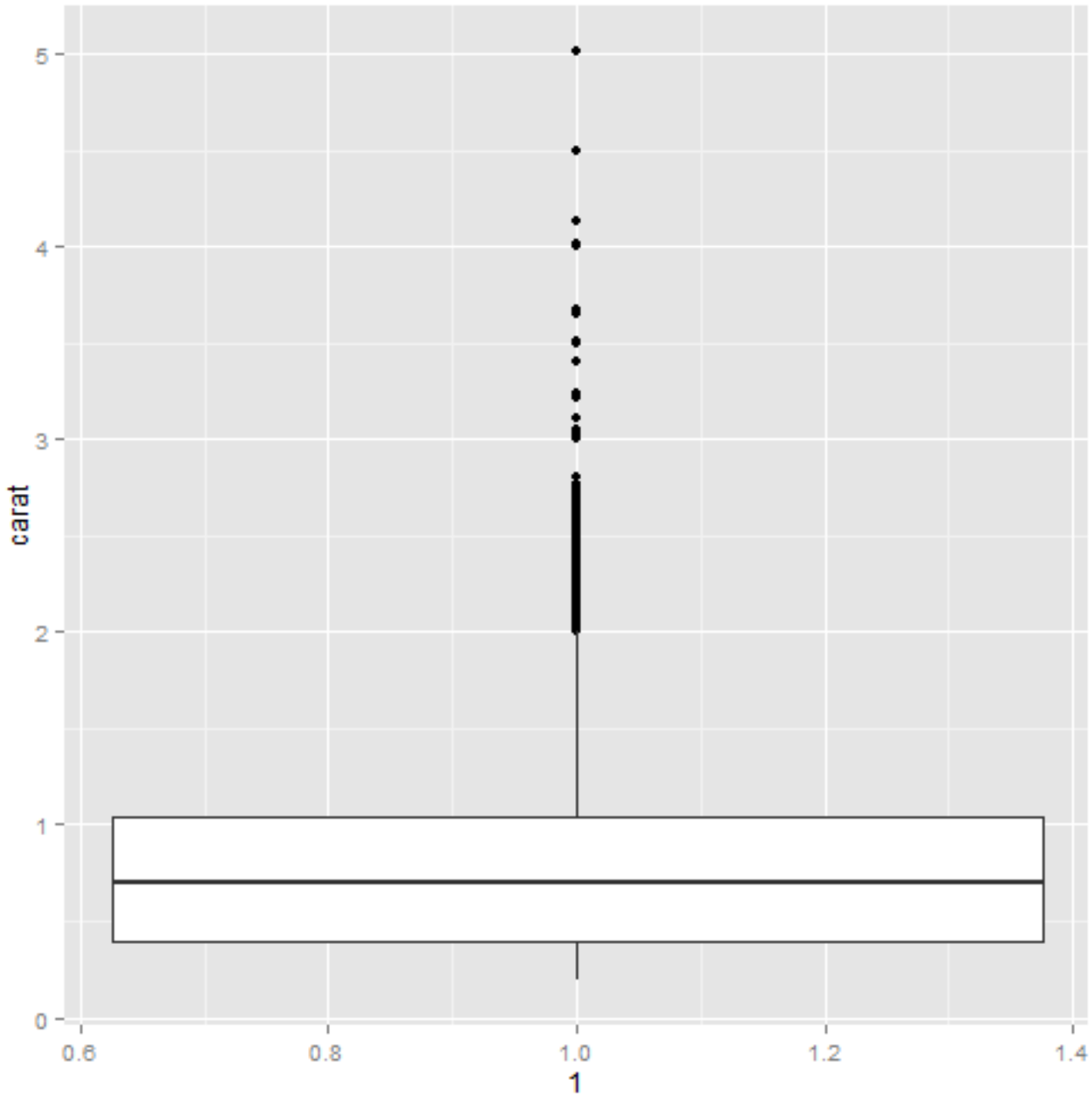
```
g + geom_point(aes(color = color))
```



ggplot2 Boxplots and Violin Plots

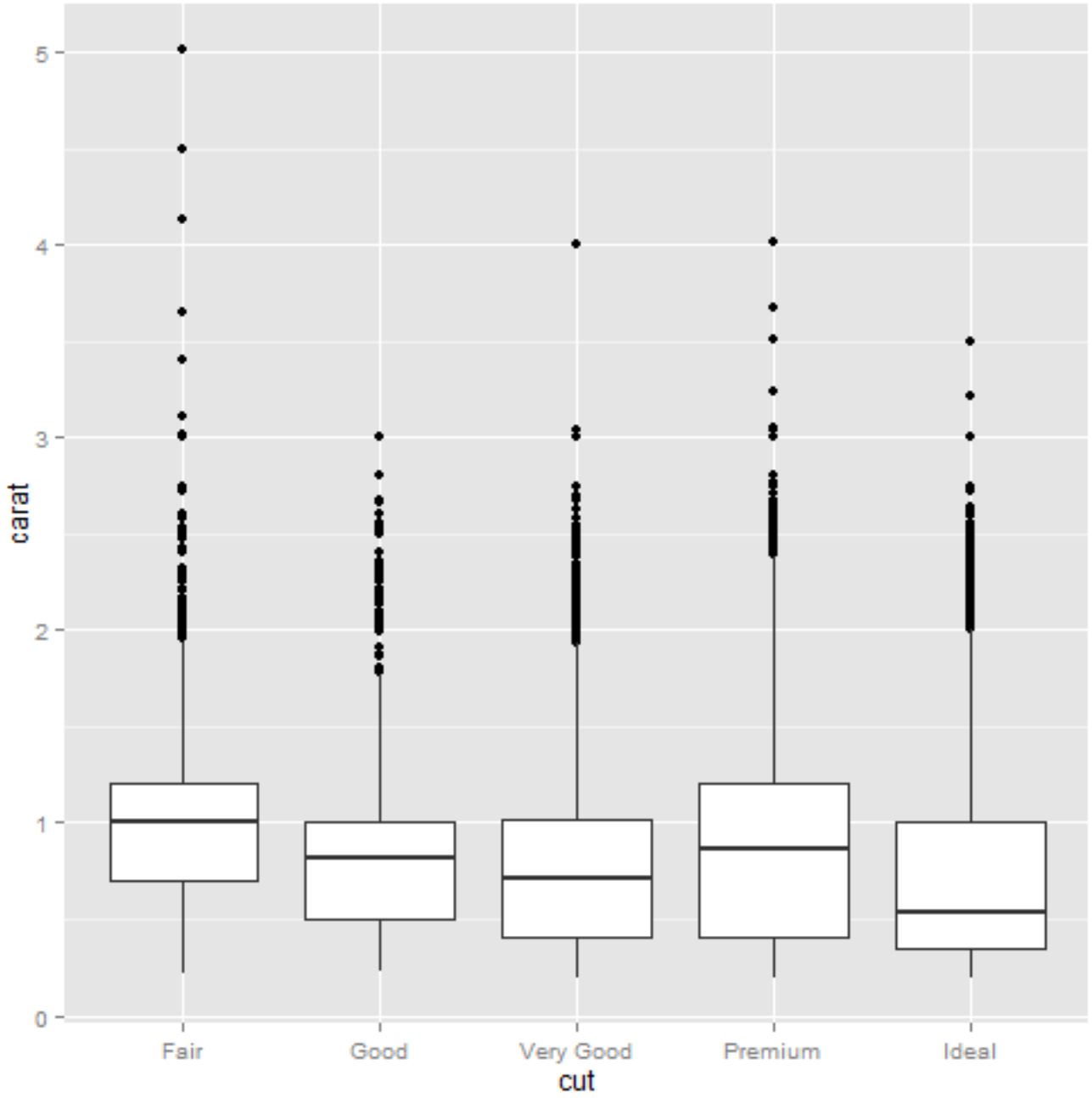
One variable

```
ggplot(diamonds, aes(y = carat, x = 1)) + geom_boxplot()
```



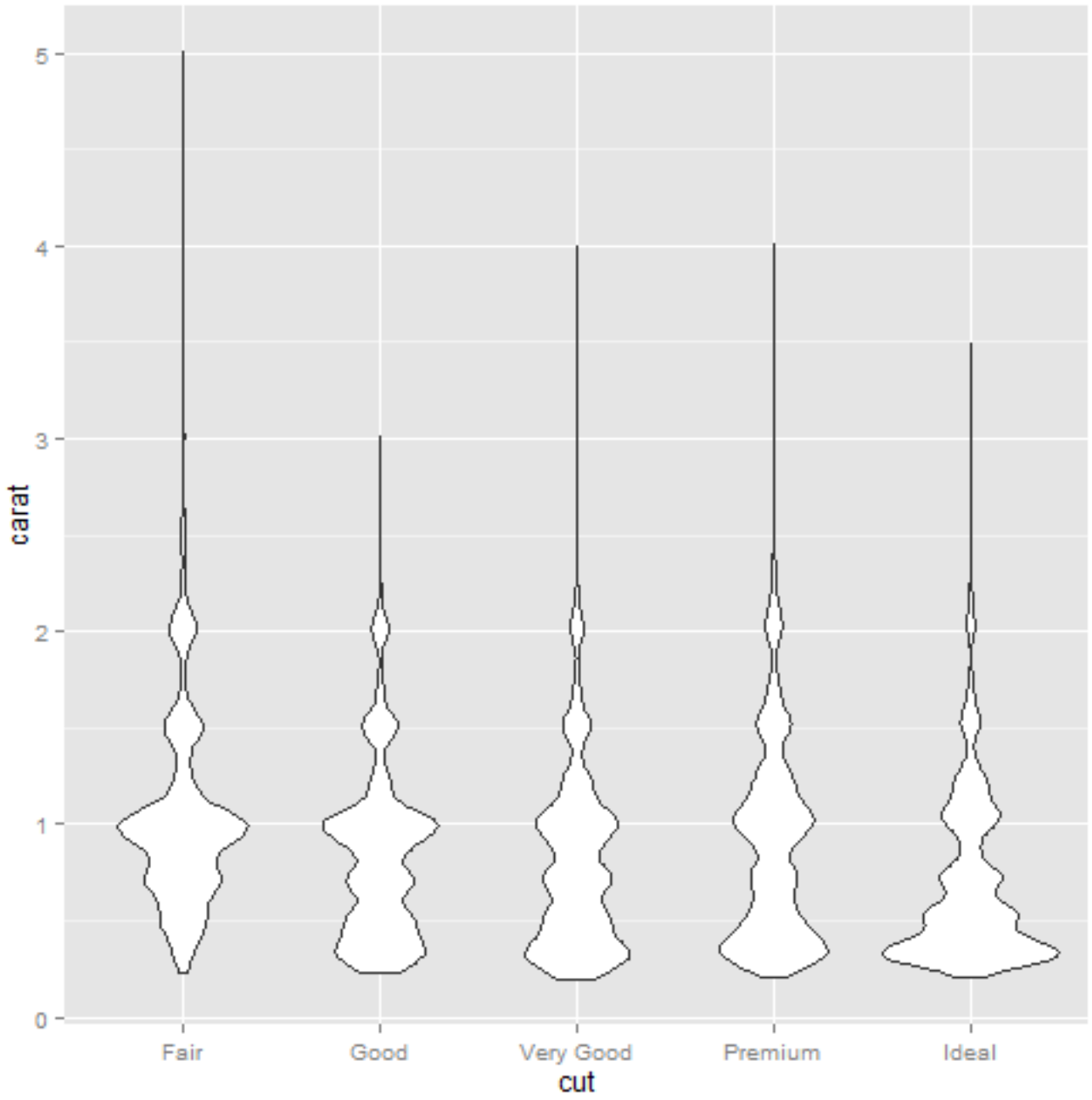
Multiple variables

```
ggplot(diamonds, aes(y = carat, x = cut)) + geom_boxplot()
```



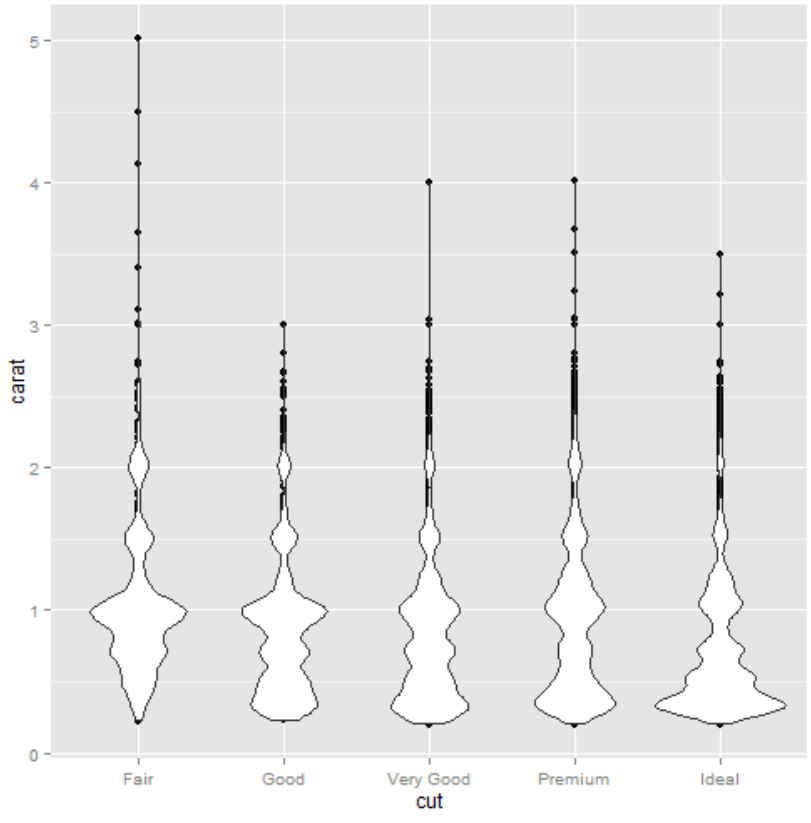
Violin

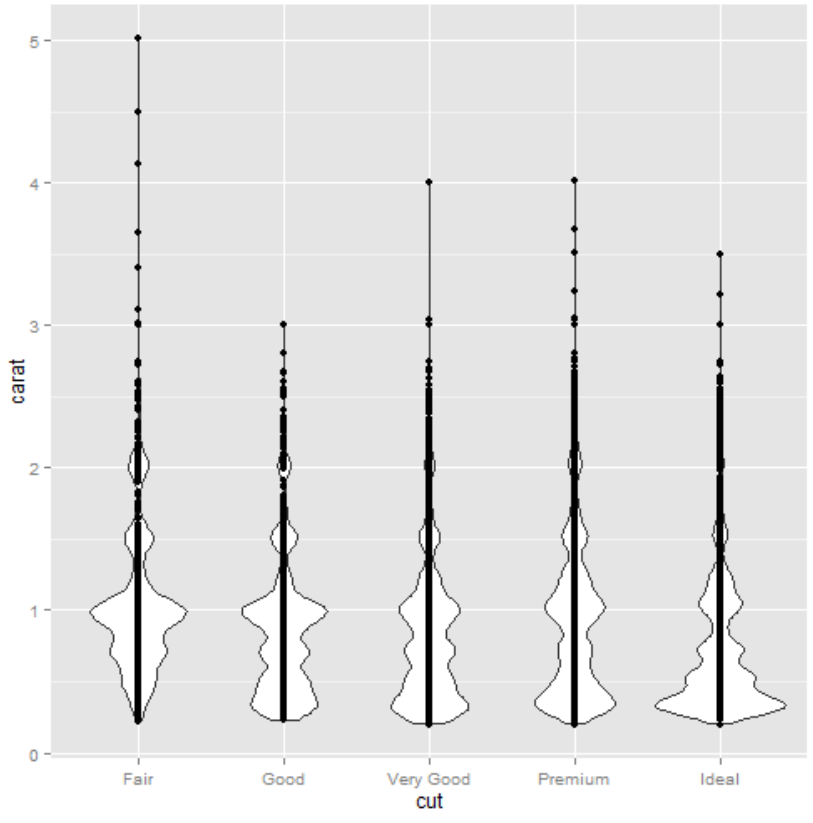
```
ggplot(diamonds, aes(y = carat, x = cut)) + geom_violin()
```



Layer order

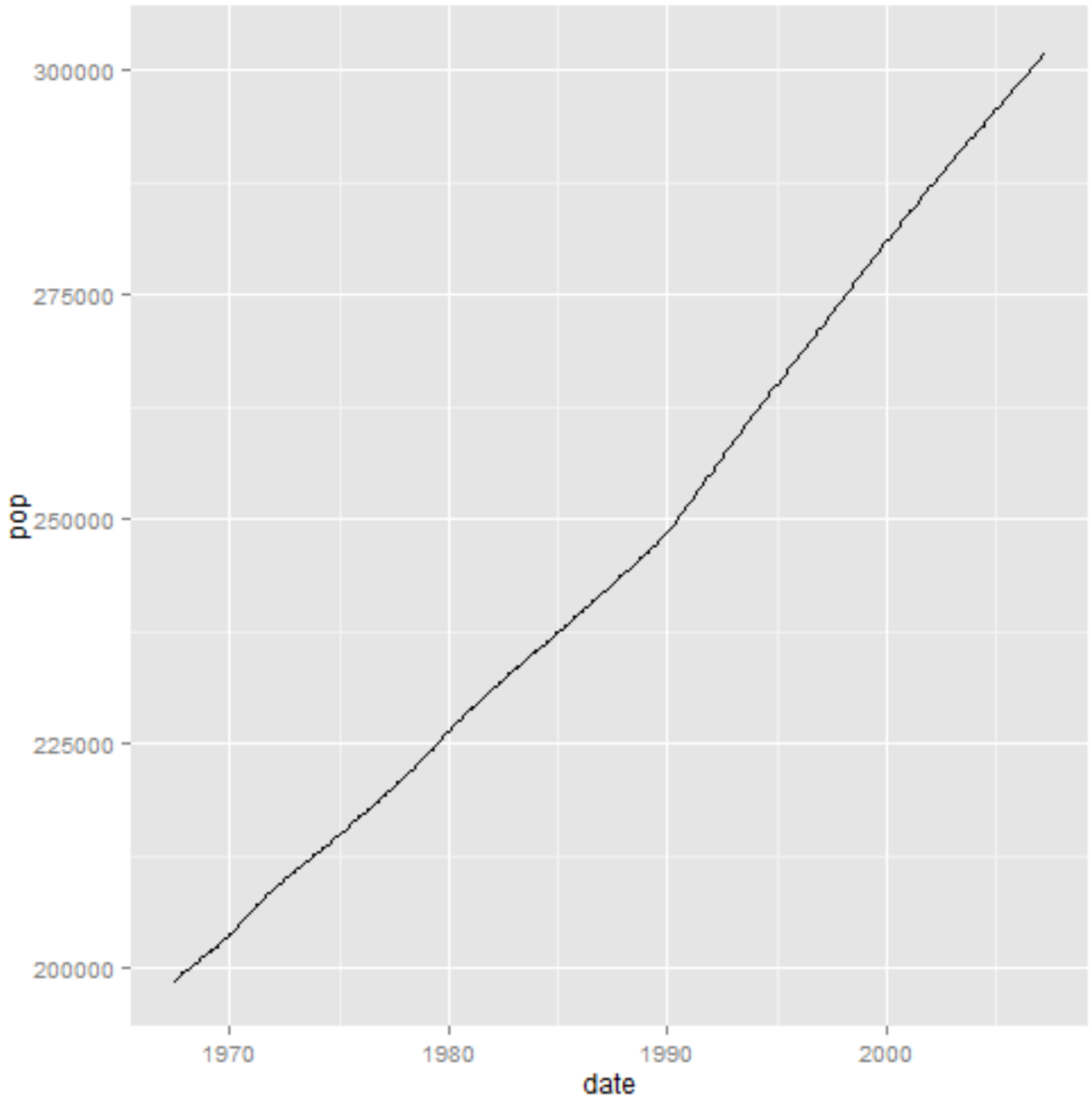
```
ggplot(diamonds, aes(y = carat, x = cut)) + geom_point() +
geom_violin()
ggplot(diamonds, aes(y = carat, x = cut)) + geom_violin() +
geom_point()
```





ggplot2 Line Plots

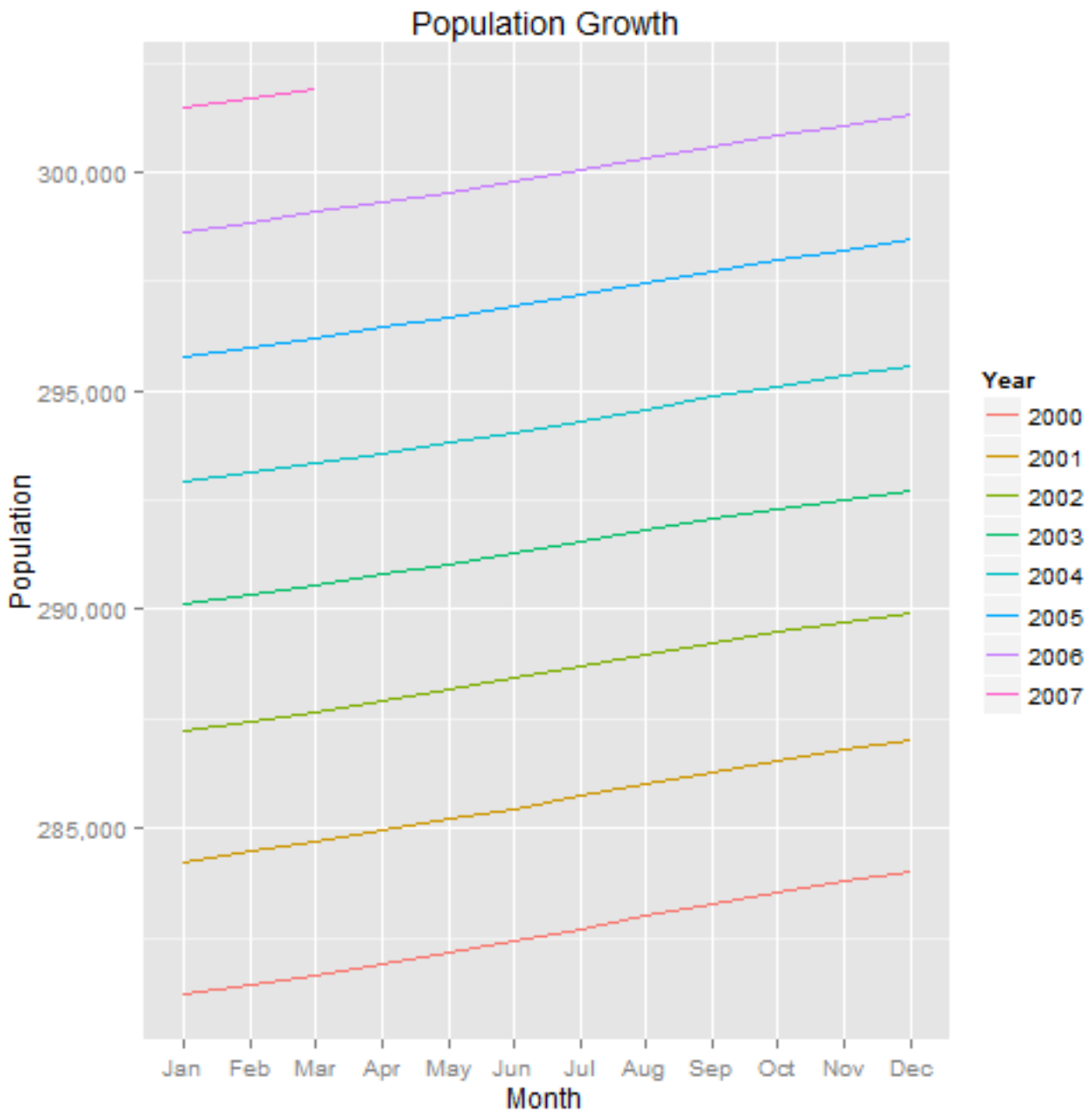
```
ggplot(economics, aes(x = date, y = pop)) + geom_line()
```




```
# load the lubridate package
require(lubridate)
```

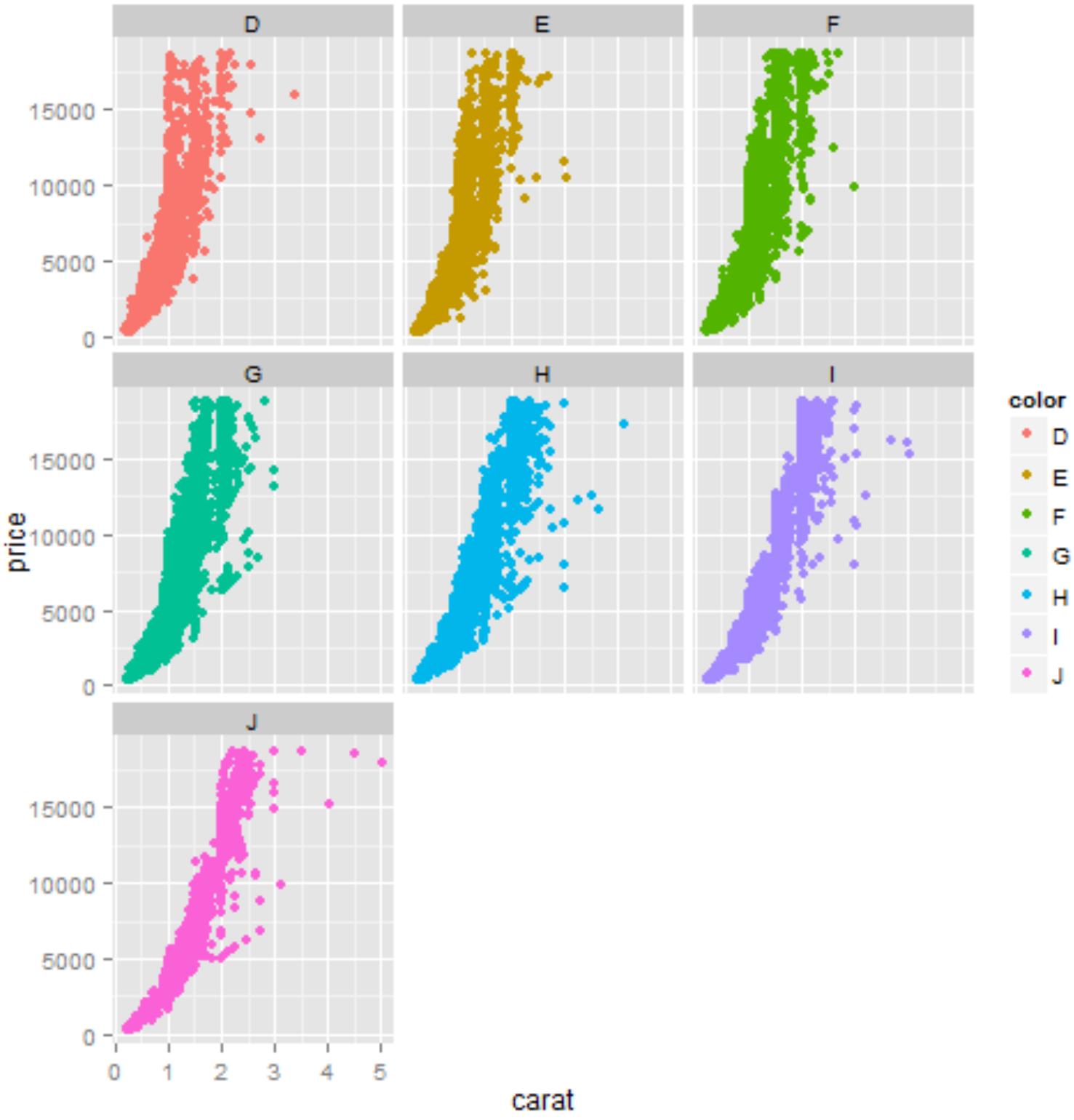
```
## Loading required package: lubridate
```

```
## create year and month variables
economics$year <- year(economics$date)
# the label argument to month means that the result should be the
# names of the month instead of the number
economics$month <- month(economics$date, label=TRUE)

# subset the data
# the which function returns the indices of observations where the
# tested condition was TRUE
econ2000 <- economics[which(economics$year >= 2000), ]

# load the scales package for better axis formatting
require(scales)

# build the foundation of the plot
g <- ggplot(econ2000, aes(x=month, y=pop))
# add lines color coded and grouped by year
# the group aesthetic breaks the data into separate groups
g <- g + geom_line(aes(color=factor(year), group=year))
#  name the legend "Year"
g <- g + scale_color_discrete(name="Year")
# format the y axis
g <- g + scale_y_continuous(labels=comma)
# add a title and axis labels
g <- g + labs(title="Population Growth", x="Month", y="Population")
# plot the graph
g
```

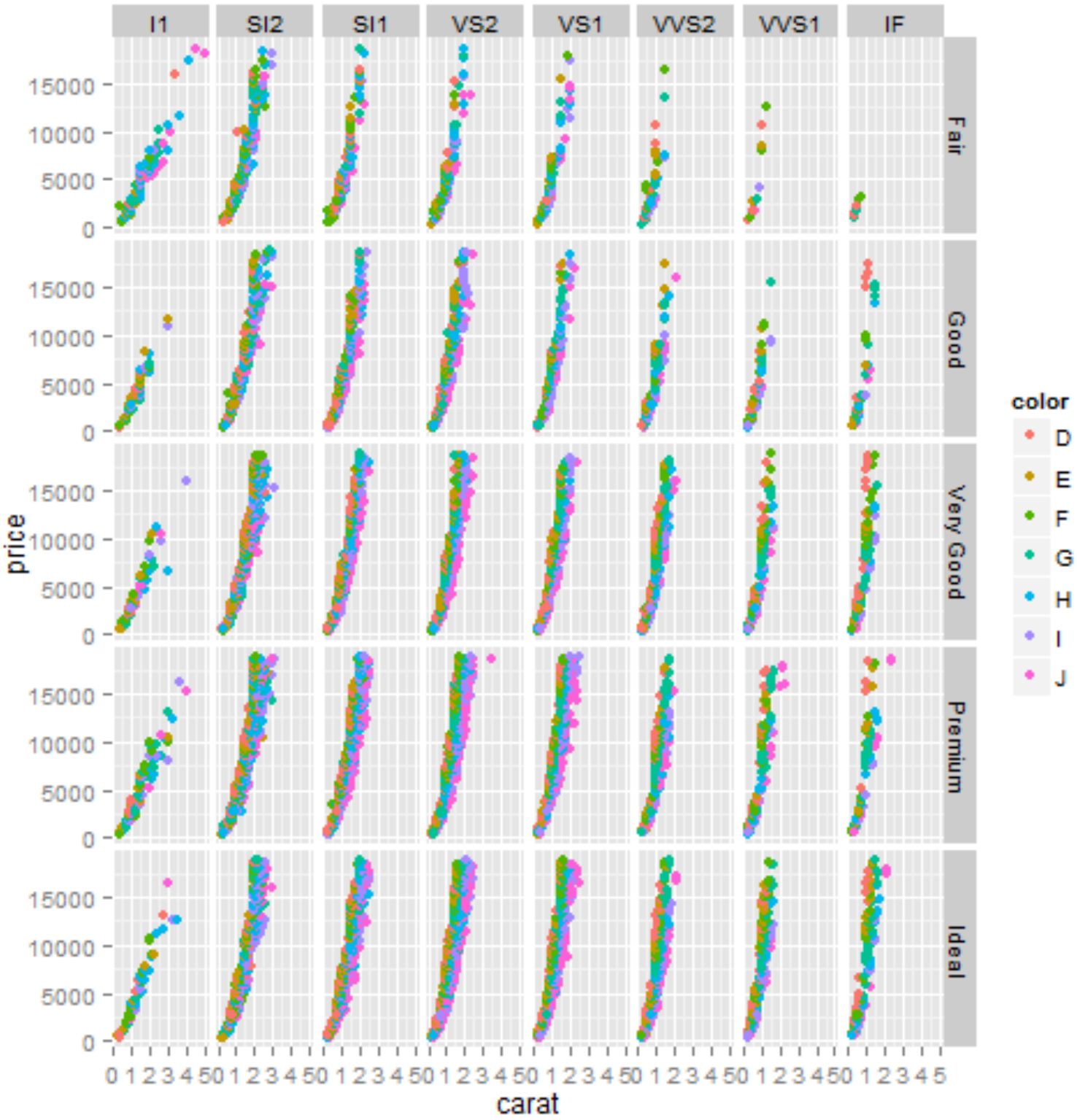


Faceting

```
g <- ggplot(diamonds, aes(x = carat, y = price))
g + geom_point(aes(color = color)) + facet_wrap(~color)
```

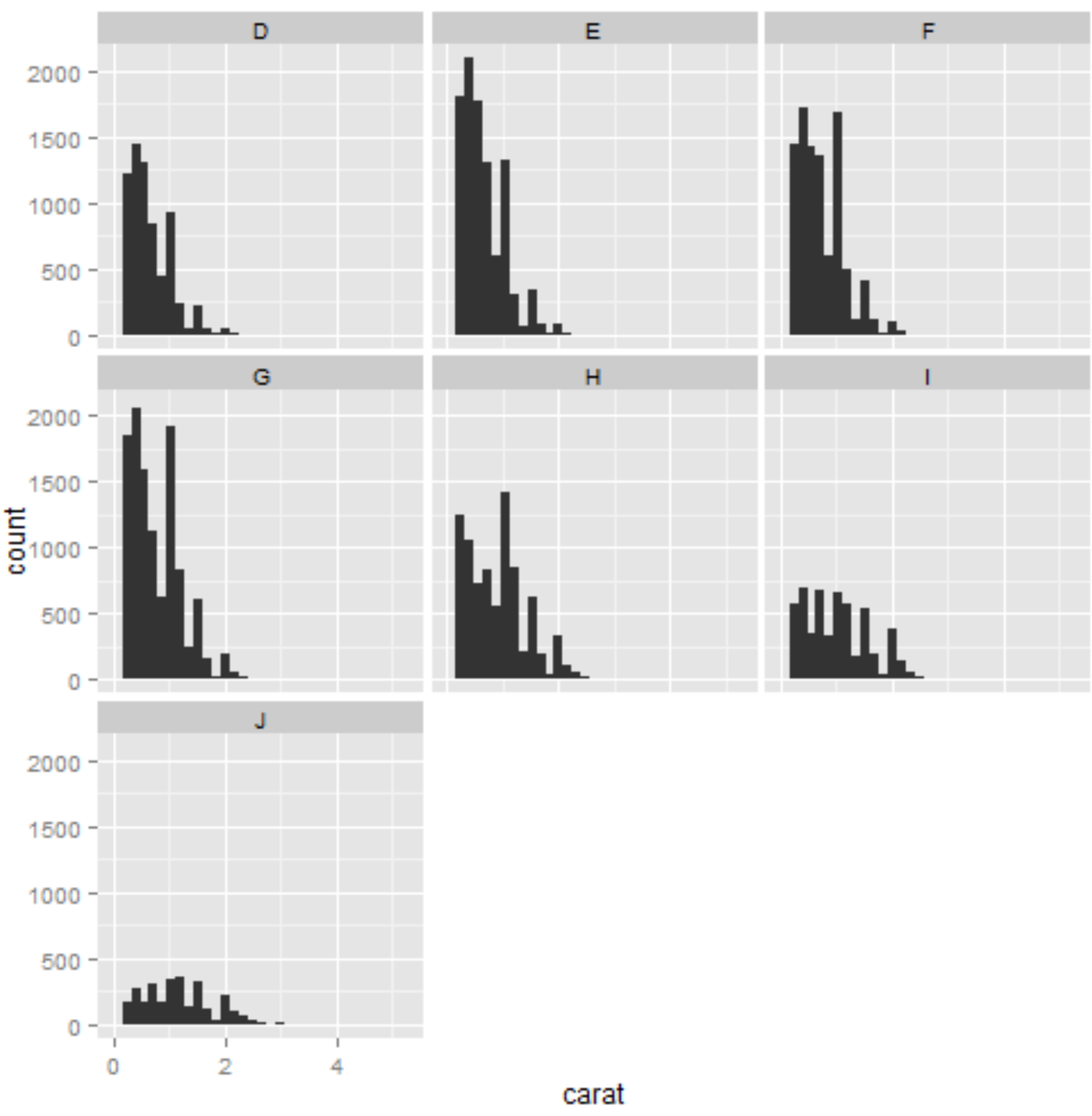



```
g + geom_point(aes(color = color)) + facet_grid(cut ~ clarity)
```




```
ggplot(diamonds, aes(x = carat)) + geom_histogram() +
facet_wrap(~color)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to
adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to
adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to
adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to
adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to
adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to
adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to
adjust this.
```

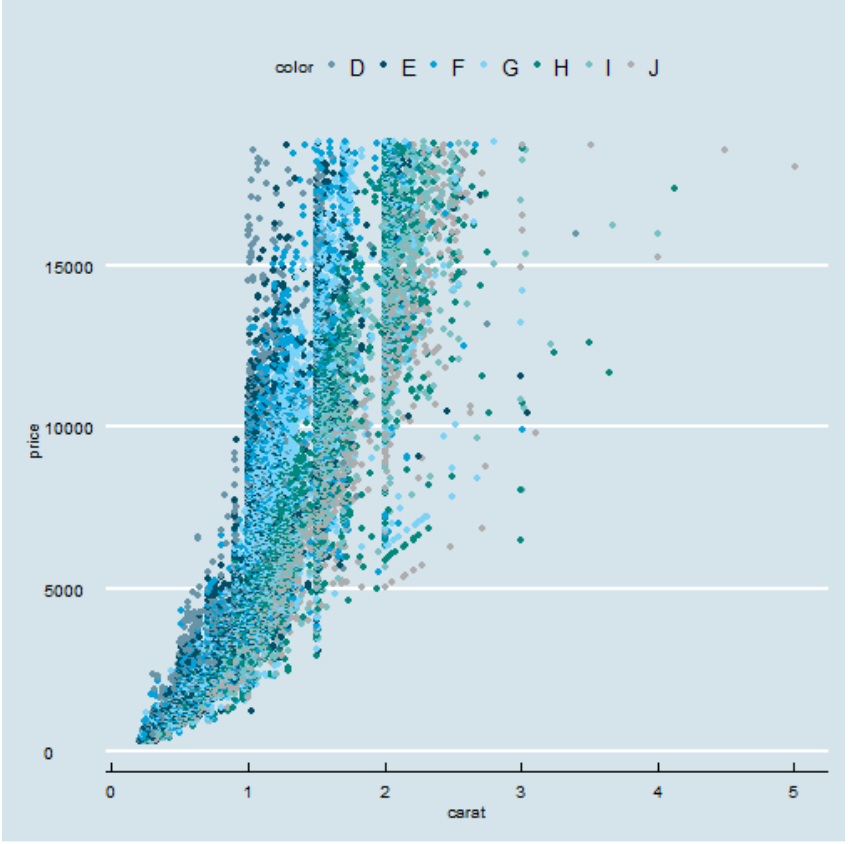


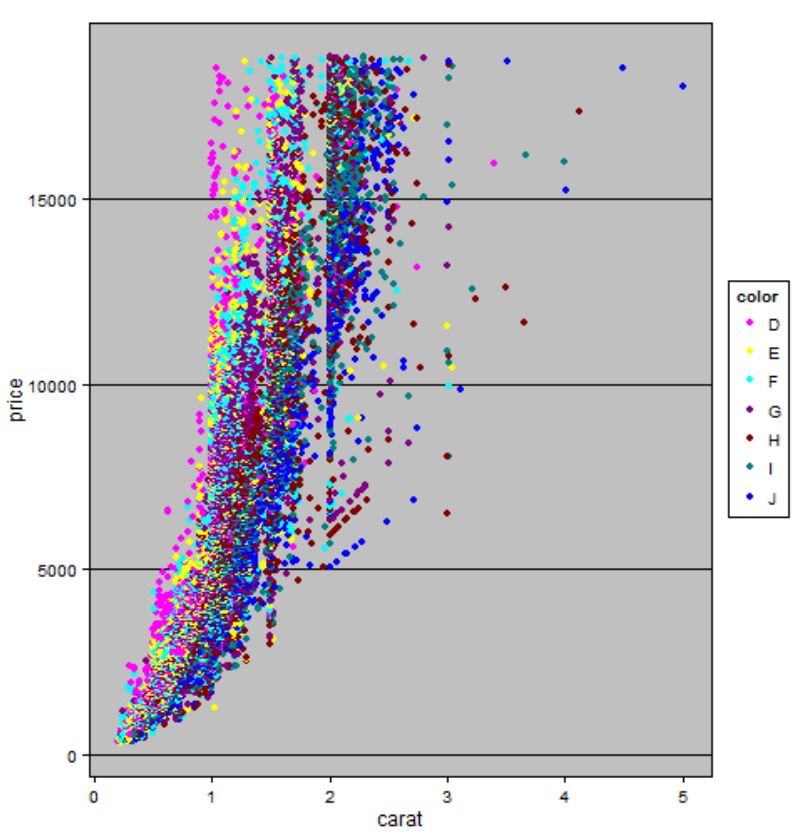
Themes

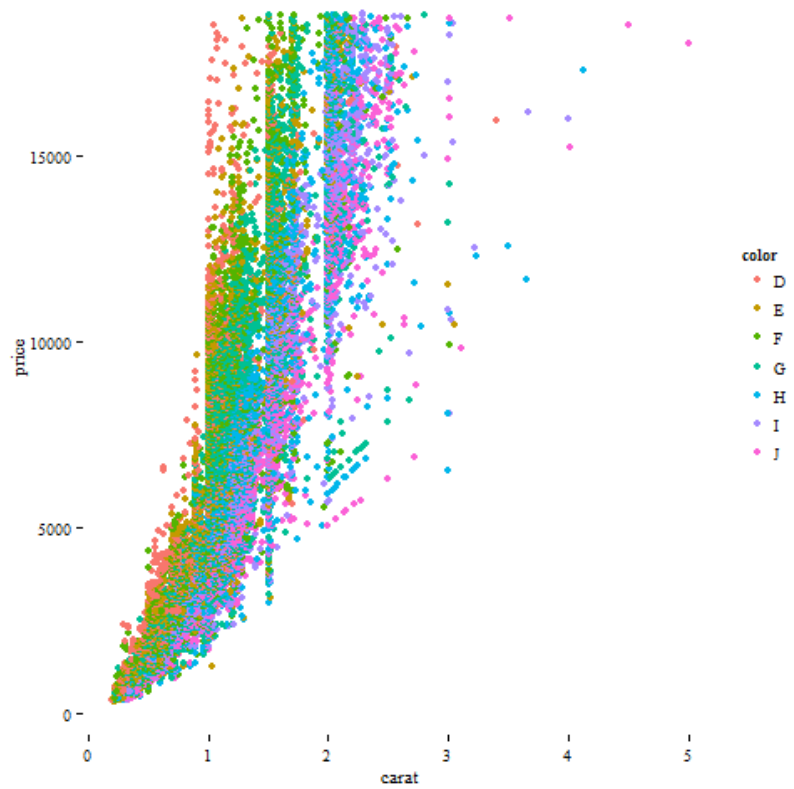
```
require(ggthemes)
```

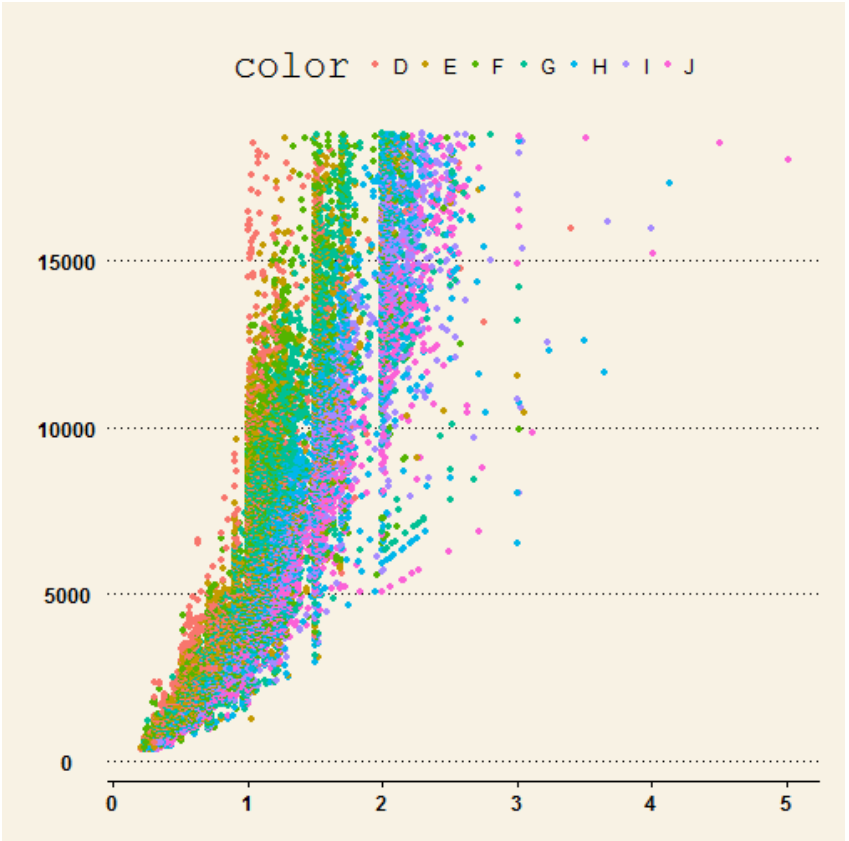
```
## Loading required package: ggthemes
```

```
# build a plot and store it in g2
g2 <- ggplot(diamonds, aes(x=carat, y=price)) +
    geom_point(aes(color=color))

# apply a few themes
g2 + theme_economist() + scale_colour_economist()
g2 + theme_excel() + scale_colour_excel()
g2 + theme_tufte()
g2 + theme_wsj()
```


color D E F G H I J
15000
10000
5000
0
price
0 1 2 3 4 5
carat


15000
10000
5000
0
price
0 1 2 3 4 5
carat
color
D
E
F
G
H
I
J


15000
10000
5000
0
price
0 1 2 3 4 5
carat
color
D
E
F
G
H
I
J


color D E F G H I J
15000
10000
5000
0
0 1 2 3 4 5