R_ggplot2_Package

Norman Lo

1/10/2022

Data Visualization in R

1 - Base R Graphic Tools

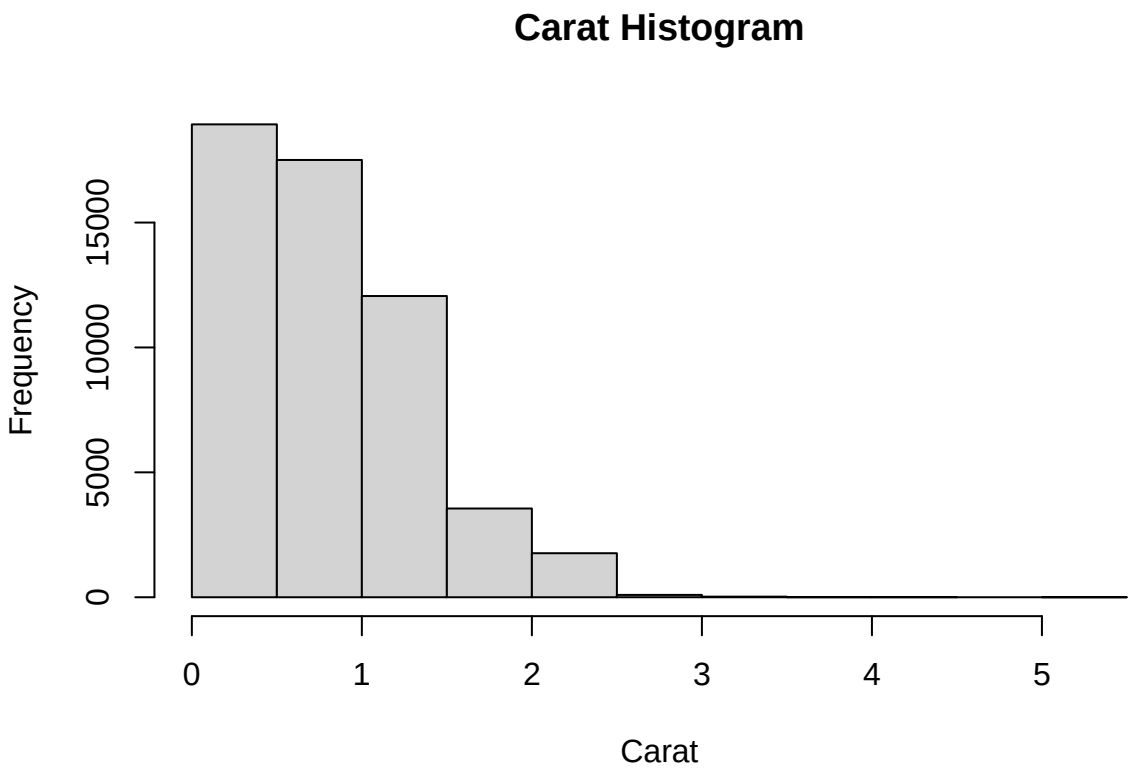
Most people use the **base R** built-in graphic tools as the starting point for data visualization, which base R alreayd provides a great set of graphic tools. Here we are going to demonstrate the use of these graphic tools and later compare it to the more sophisticated **ggplot2** package. In this demonstration, we are using the 'diamonds' data set from ggplot2 package.

Histogram is the most fundamental graphic tools to show the frequency distribution of the numerical data. Base R provides this very easy way to plot the histogram with hist() function. Suppose we are plotting the histrogram for the numerical column "carat" from the diamonds tbl.

```
library(ggplot2)
data(diamonds)
head(diamonds)
```

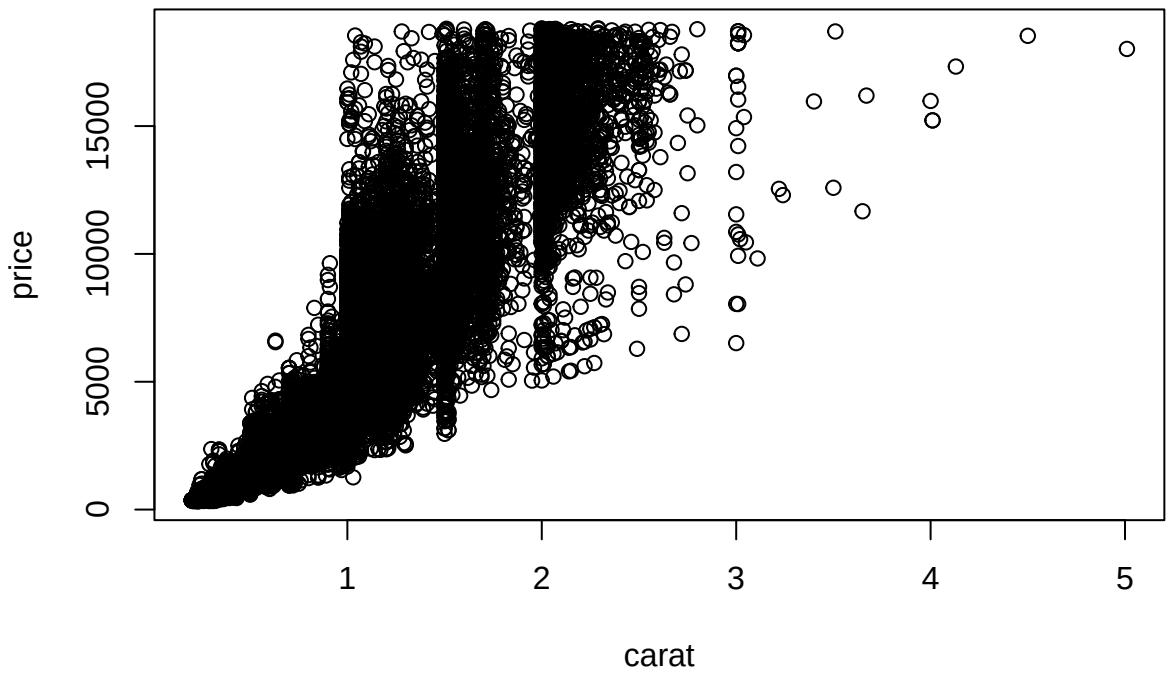
```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  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  4.2   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
```

```
# Create histogram with base R
hist(diamonds$carat, main='Carat Histogram', xlab='Carat')
```



Scatter plot is a very useful tool to compare two sets of numerical data. We can use "formula" to plot the diamond price against carat.

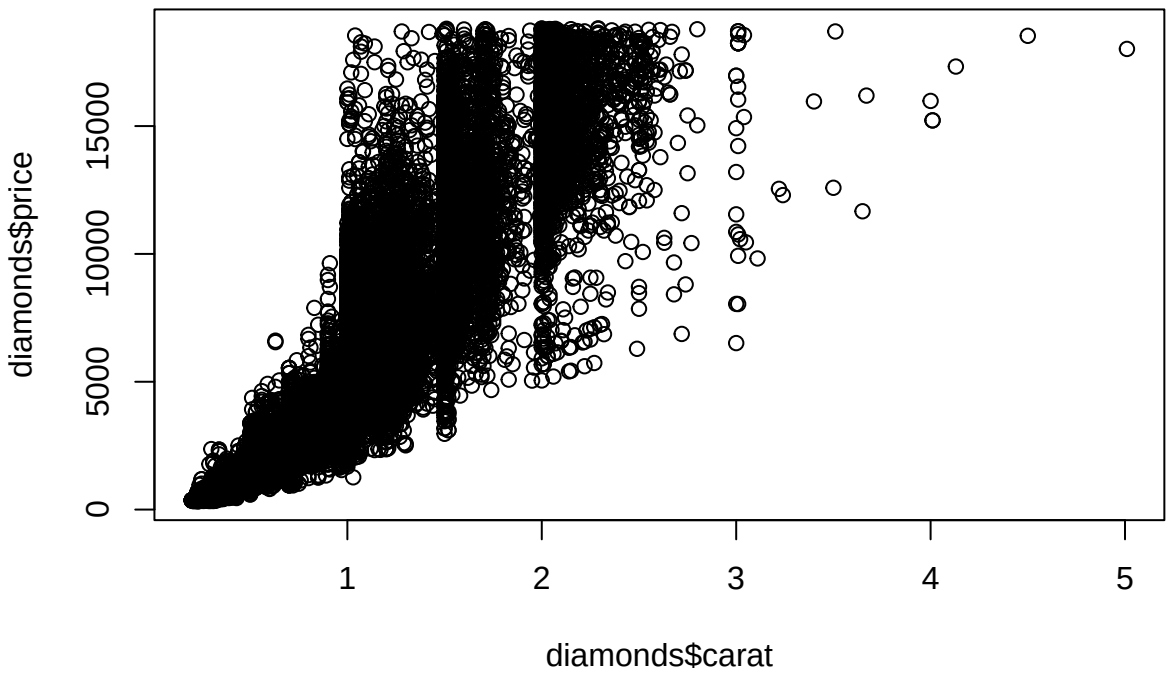
```
plot(price ~ carat, data=diamonds)
```



Note: In the formula, price is the y value and carat is the x value.

It is not necessary to use "formula" for a scatter plot,

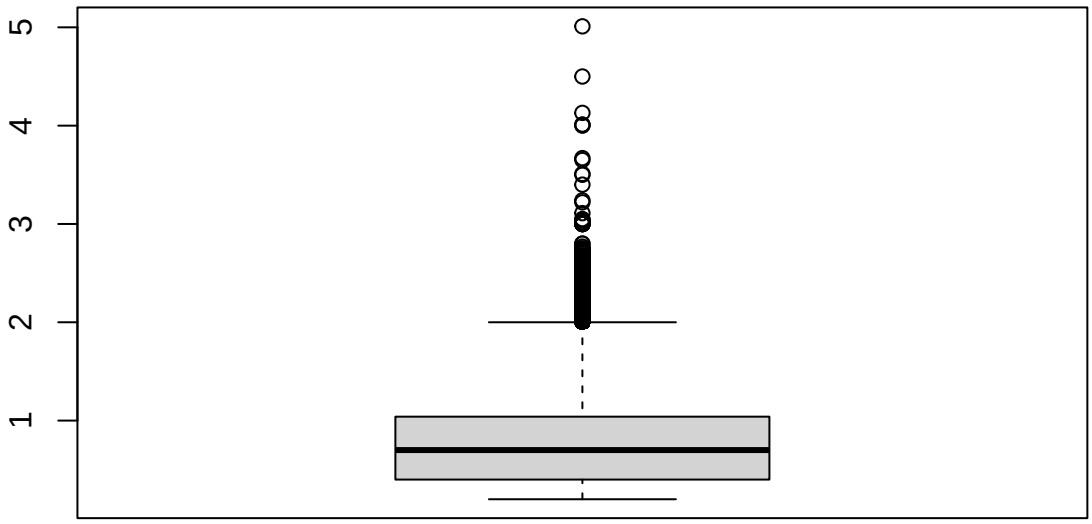
```
plot(diamonds$carat, diamonds$price)
```



Note: the first argument in the plot() function is the x value and the second argument is the y value.

Box plot is probably the first graphical tool learned in fundermental statistic class. It is used to depict the range, mean, median, and quartiles of the numerical data. We can use the boxplot() function to depict the numerical data 'carat'

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



2 - ggplot2 Package

Base R already provides a powerful graphic tools for the average R users. However, when we want to modify some specific features in the graph, the base R graphic tools requires more lines of code. **ggplot2** package offers a structure of coding for achieving the complex modification of graphic features. In some cases, adding some additional features to a graph may take 30 additional lines of code to run in base R, but may only require 1 line of code using ggplot2 package.

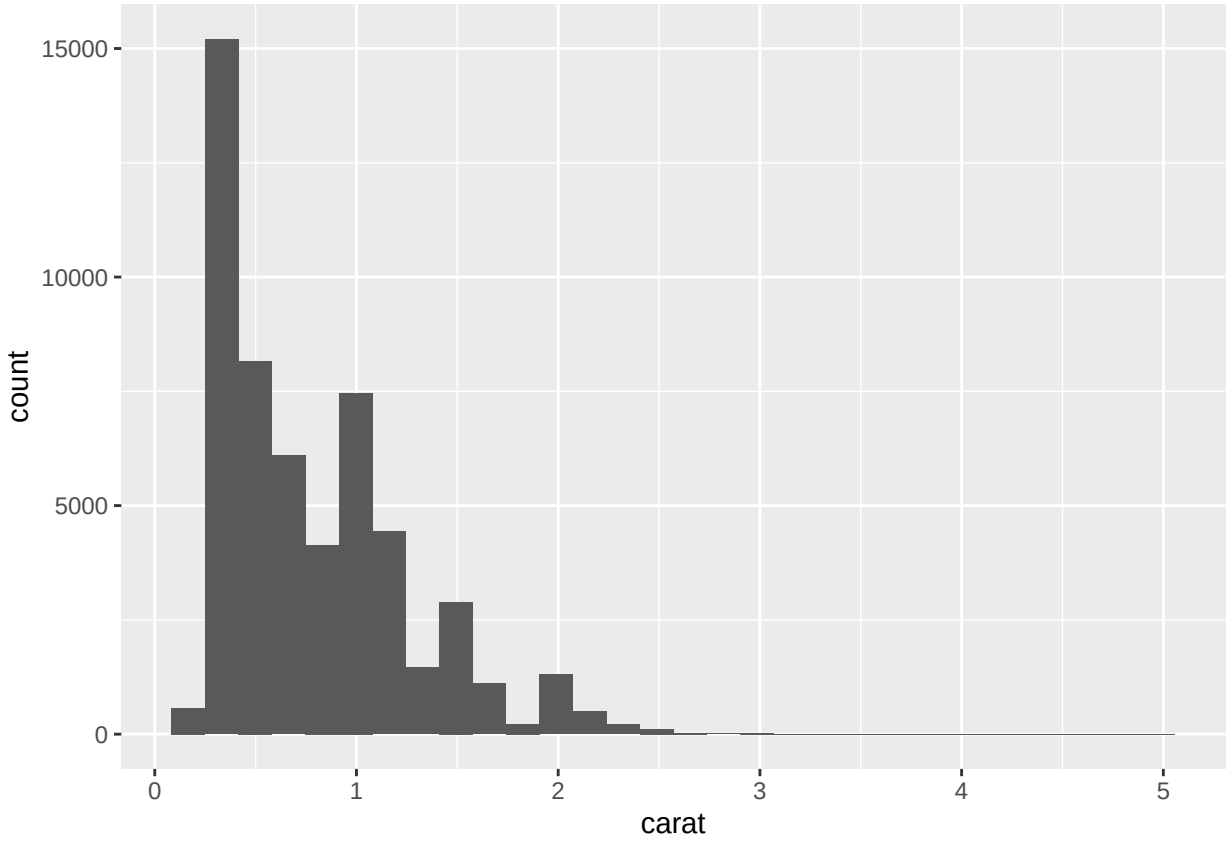
We start with taking the data as the first argument in ggplot() function and build on top of it with "+". Think of "+" is like adding an additional layer to the plot for different feature added. The most important thing to add into the new layer is aes(), represents "aesthetic" adjustment.

Histogram with ggplot2

Using ggplot2 for plotting the histogram for numerical data 'Carat' in the diamonds data set.

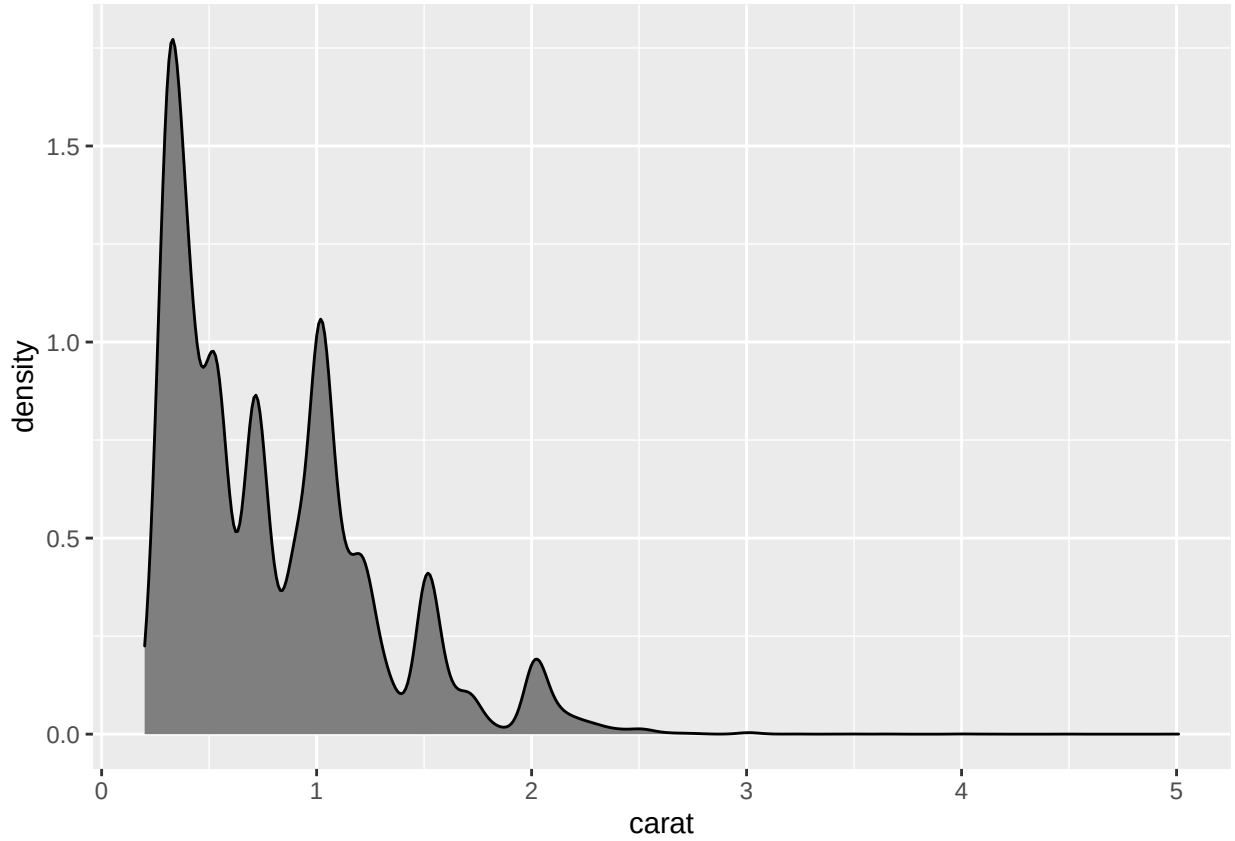
```
ggplot(data=diamonds) + geom_histogram(aes(x=carat))
```

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



Similar to histrogram, we can easily plot the density distribution with geom_density().

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

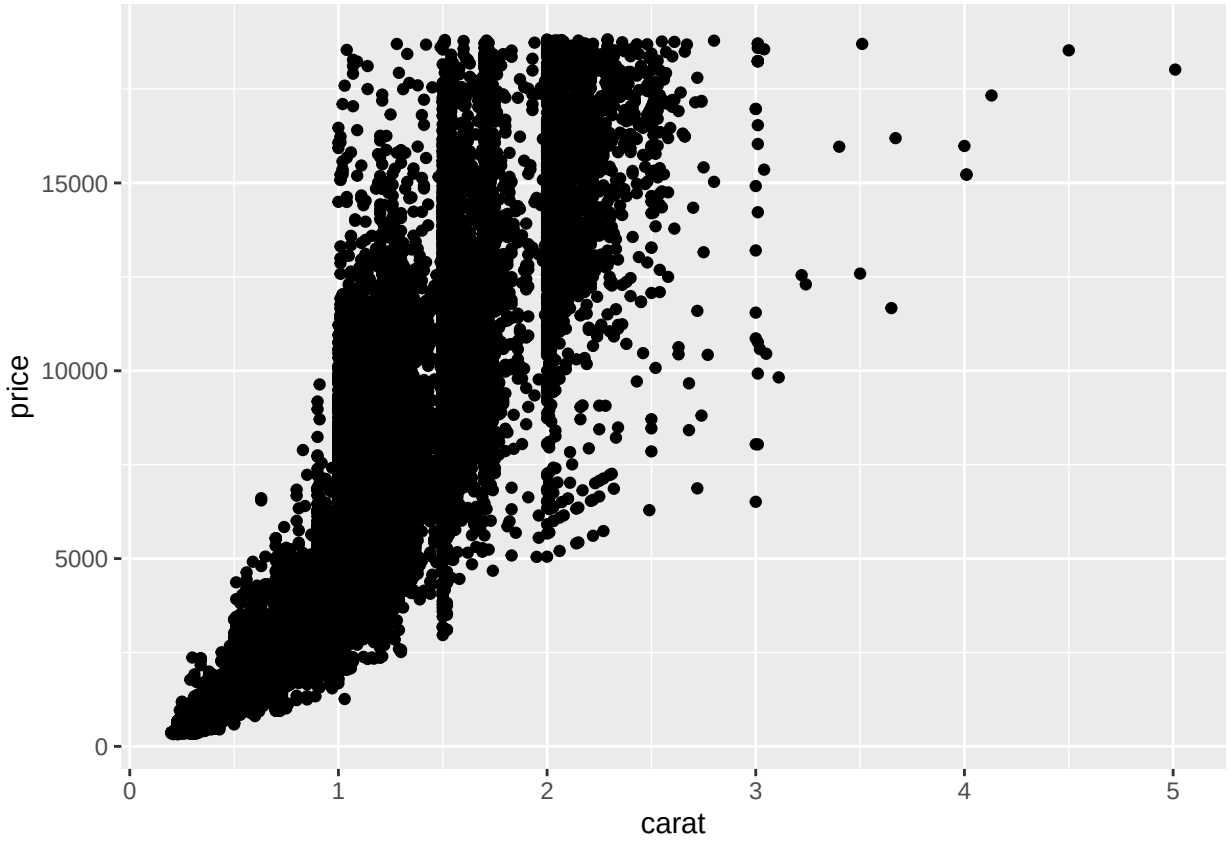


Note: Histogram divide the continuous variable into groups (x-axis) and gives the frequency (y-axis) in each group. Density distribution is evaluated at any given point on the continuous variable (x-axis)

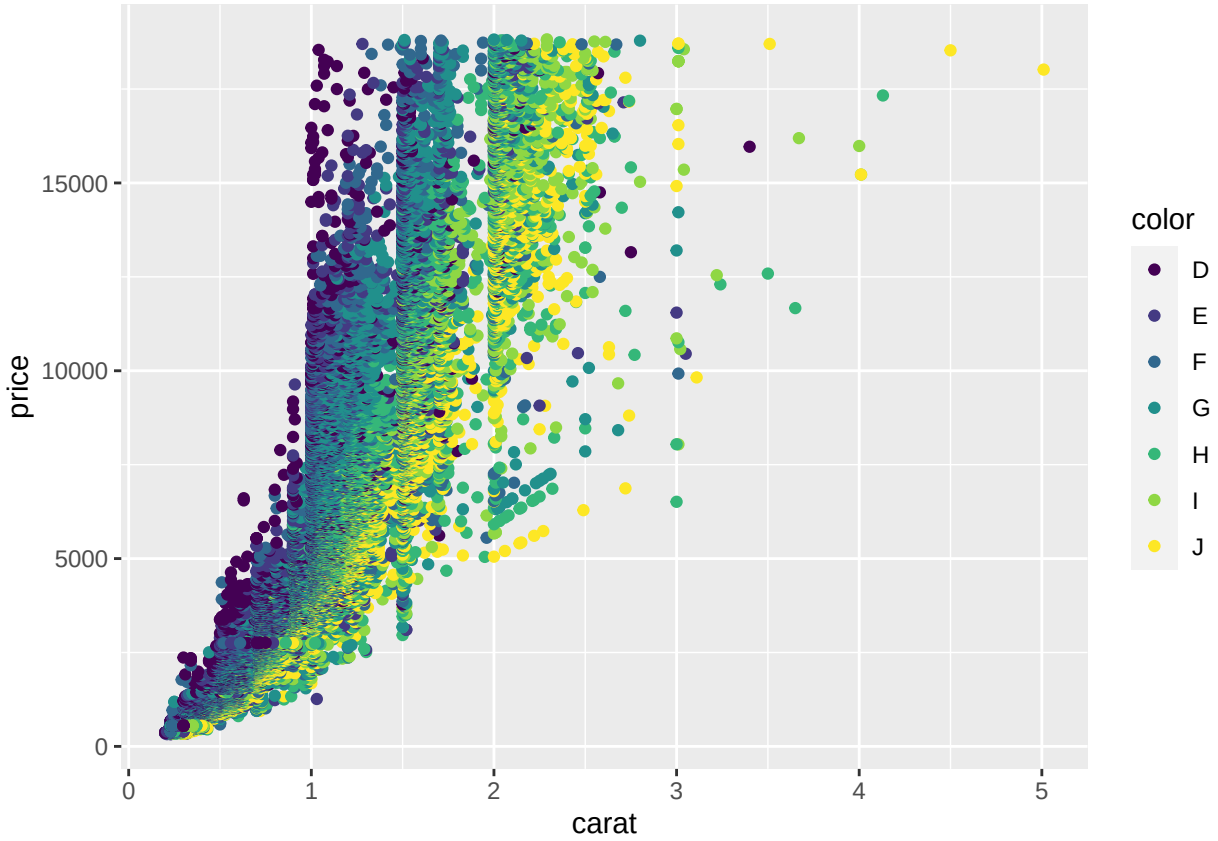
Scatter Plot with ggplot2

In this section, we are not only demonstrate how to use scatter plot with ggplot2 package, we also try to demonstrate the simplicity for adding some useful features to our graph. Plot the diamond price and carat within a scatter plot. This time we are including the aes() in the ggplot() function, but not in the geom()

```
# Create scatter plot with ggplot2
ggplot(diamonds, aes(x=carat, y=price)) + geom_point()
```



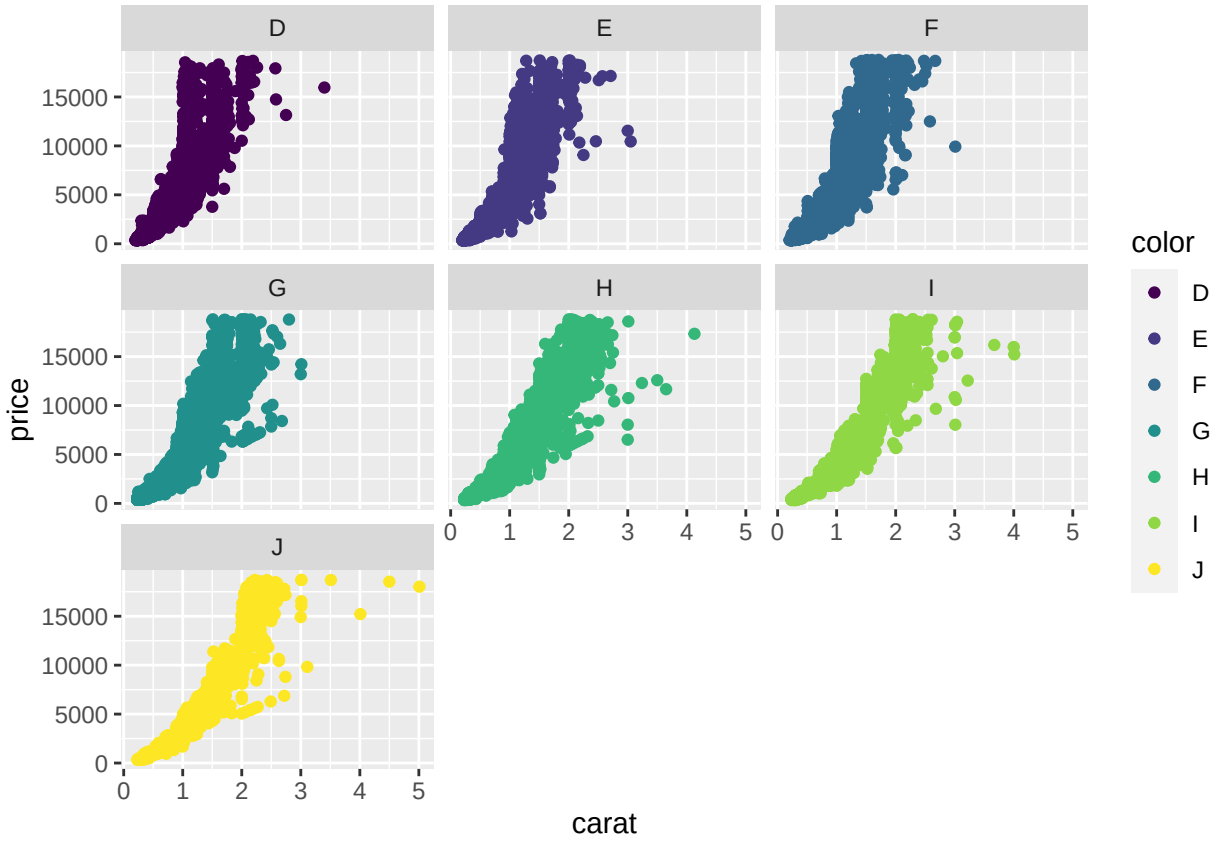

```
# We are going to repeat using "ggplot(diamonds, aes(x=carat, y=price))", so we save it into a variable
g <- ggplot(diamonds, aes(x=carat, y=price))

# Suppose we would like to identify the colors of the diamond in the graph with different color codes.
g + geom_point(aes(color=color))
```



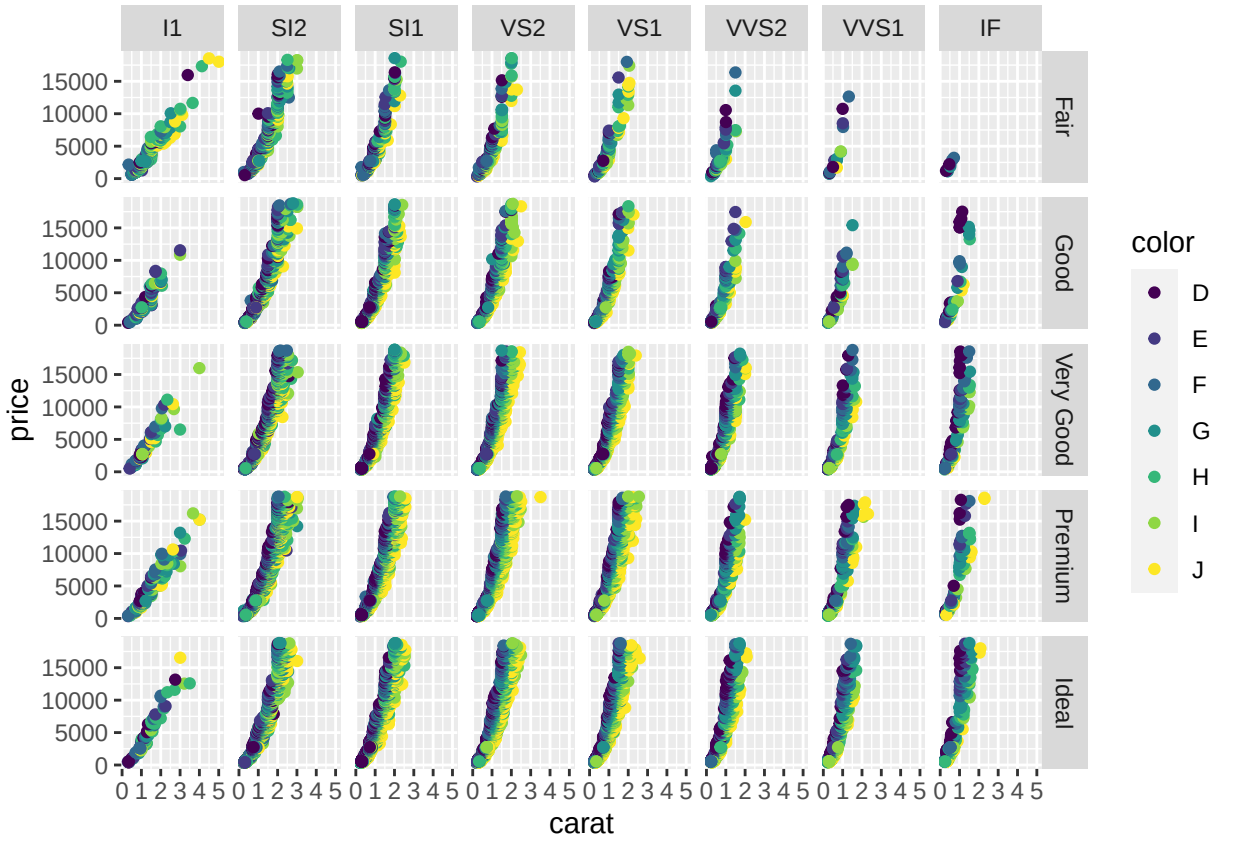
Note: The argument color= is to define what color to use for the data points in the graph and we define the color will be different according to the value under column "color" in the data set.

ggplot2 can also create layers of plot with functions like facet_wrap() or facet_grid(). For instance, if we are interested to split each color grade into its own price vs. carat scatter plot.

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

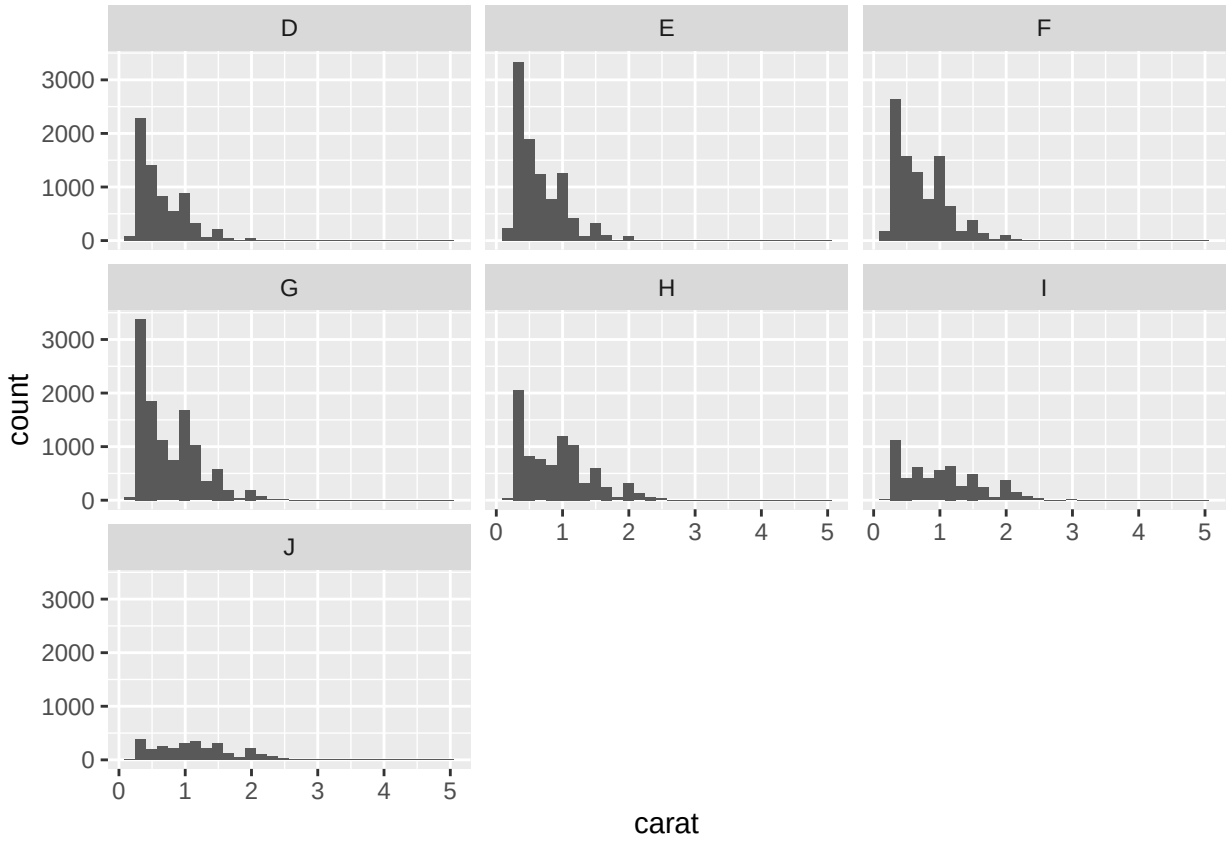



```
# Or, if we want to splite the scatter plots into different combinations of "cut" and "clarity" grade.
g + geom_point(aes(color=color)) + facet_grid(cut~clarity)
```




```
# Note: Both facet_wrap() and facet_grid() can be applied to any geom() functions.
# e.g. We can use facet_wrap() to plot the histrogram for carat in different color grade.
ggplot(diamonds, aes(x=carat)) + geom_histogram() + facet_wrap(~color)
```

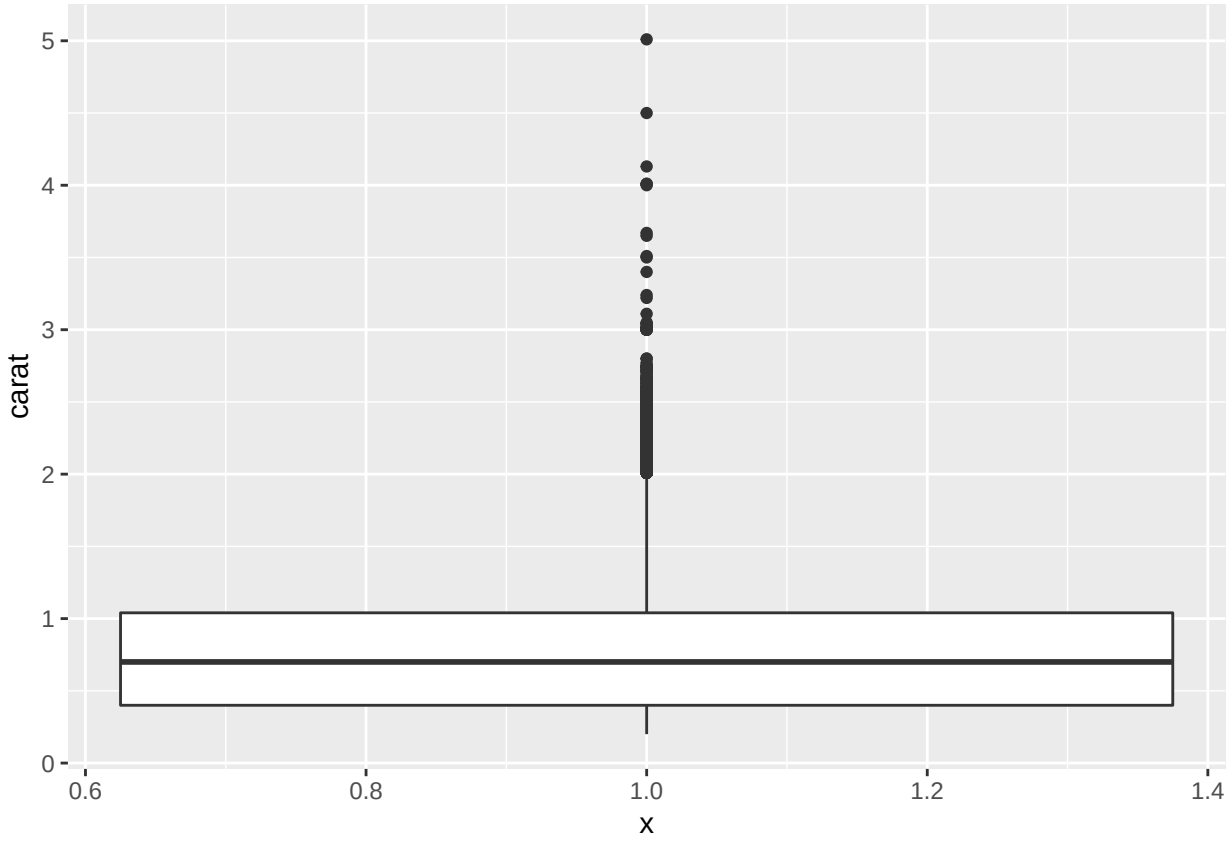
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Box Plot with ggplot2

ggplot2 provides a more complete and intuitive features to the box plot graphic. Plotting the box plot for carat

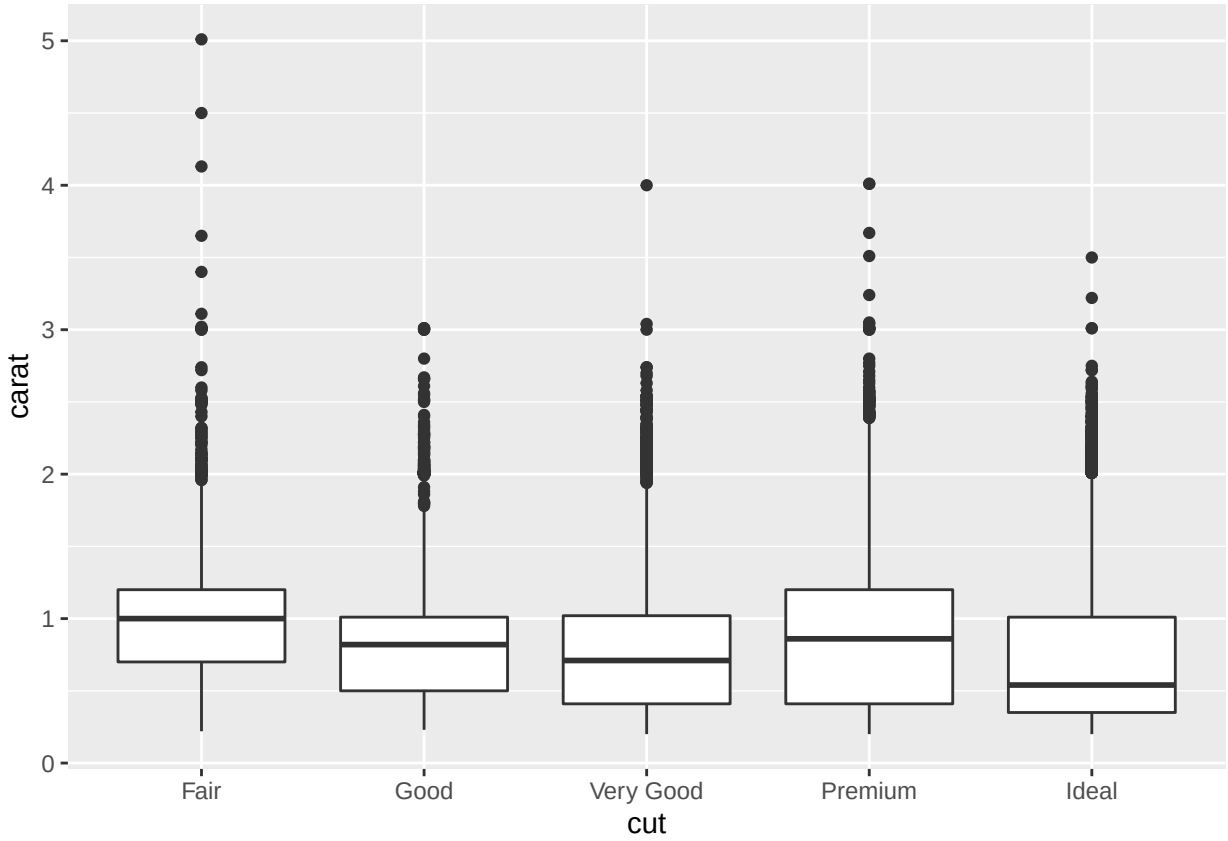
```
# Create box plot with ggplot2
ggplot(diamonds, aes(y=carat, x=1)) + geom_boxplot()
```



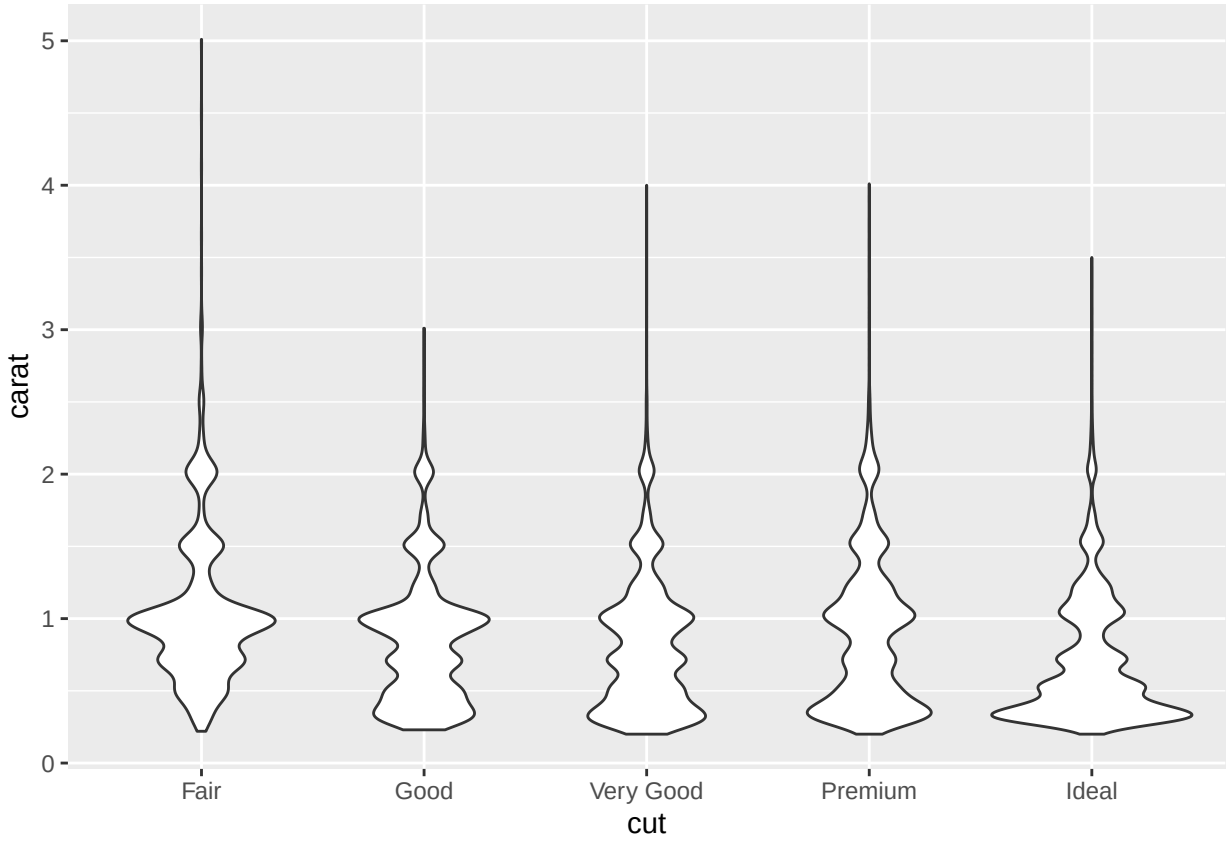
Note: Unlike the base R boxplot(), ggplot2 require to define the x-axis even for a one dimensional plot. Therefore, we put in the value 1 for x.

In ggplot2, we can easily extend the boxplot into different classes by defining a categorical data. We can divide the box plot for carat by the cut grade.

```
# Create boxplot by different cut grade
ggplot(diamonds, aes(y=carat, x=cut)) + geom_boxplot()
```



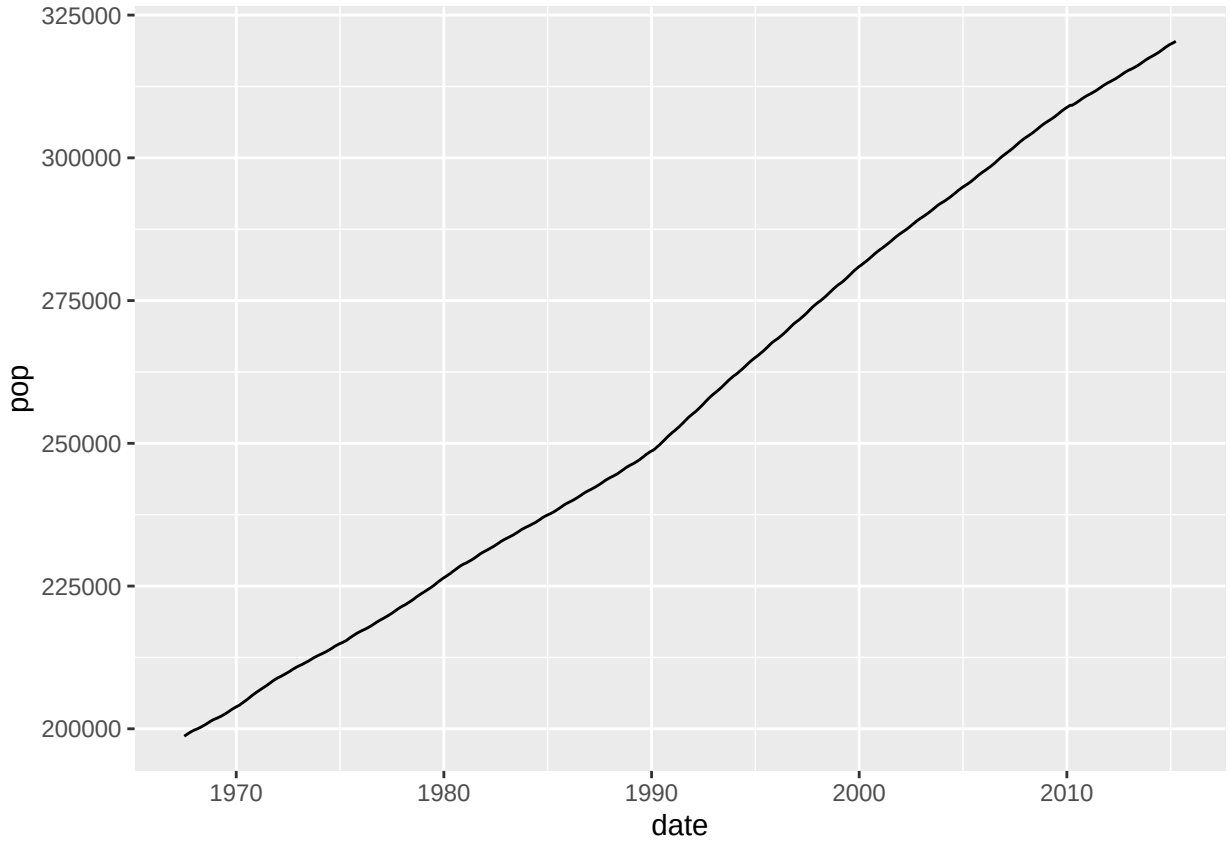

```
# In a more specific case, we can change the box plot into a violin plot
ggplot(diamonds, aes(y=carat, x=cut)) + geom_violin()
```



Line Graph with ggplot2

Line graph is usually applied to express the time series data. We are using the economics data set from ggplot2 package to demonstrate the use of line graph with date. Plot the population data with the given date.

```
# Create line graph with ggplot2
ggplot(economics, aes(x=date, y=pop)) + geom_line()
```



Suppose we are trying to break the data into year and plot into the same graph. We can use the lubridate package to help us on this task.

```
library(lubridate)
```

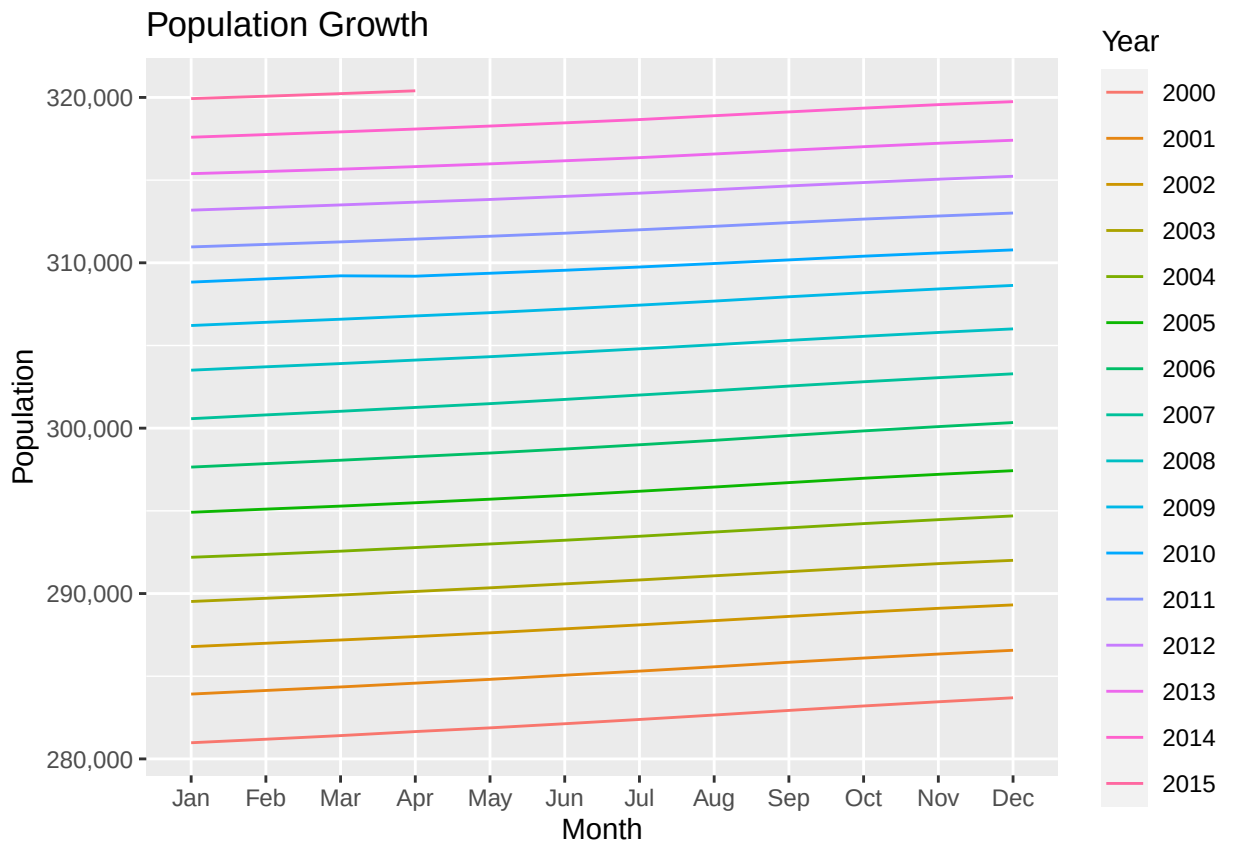
```
## Warning: package 'lubridate' was built under R version 4.1.2
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
# Build year and month variables
economics$year <- year(economics$date)
# The argument label=TRUE means to return month by its name, not a numeric value
economics$month <- month(economics$date, label=TRUE)
# Let's save the data in or beyond 2000 to an object "econ2000"
econ2000 <- economics[which(economics$year >= 2000),]

# We can use scales package to give better format to the graph
library(scales)

# Build the base of the line graph
```

```
g <- ggplot(econ2000, aes(x=month, y=pop))
# Define different colors for different years.
# Use argument group to group the data into different year.
g <- g +geom_line(aes(color=factor(year), group=year))
# Define the color by the given "Year"
g <- g + scale_color_discrete(name="Year")
# Format the y-axis
g <- g + scale_y_continuous(labels=comma)
# Adding the title, x-axis, and y-axis to the graph
g <- g + labs(title="Population Growth", x="Month", y="Population")
# Plot the graph with all the added features
g
```



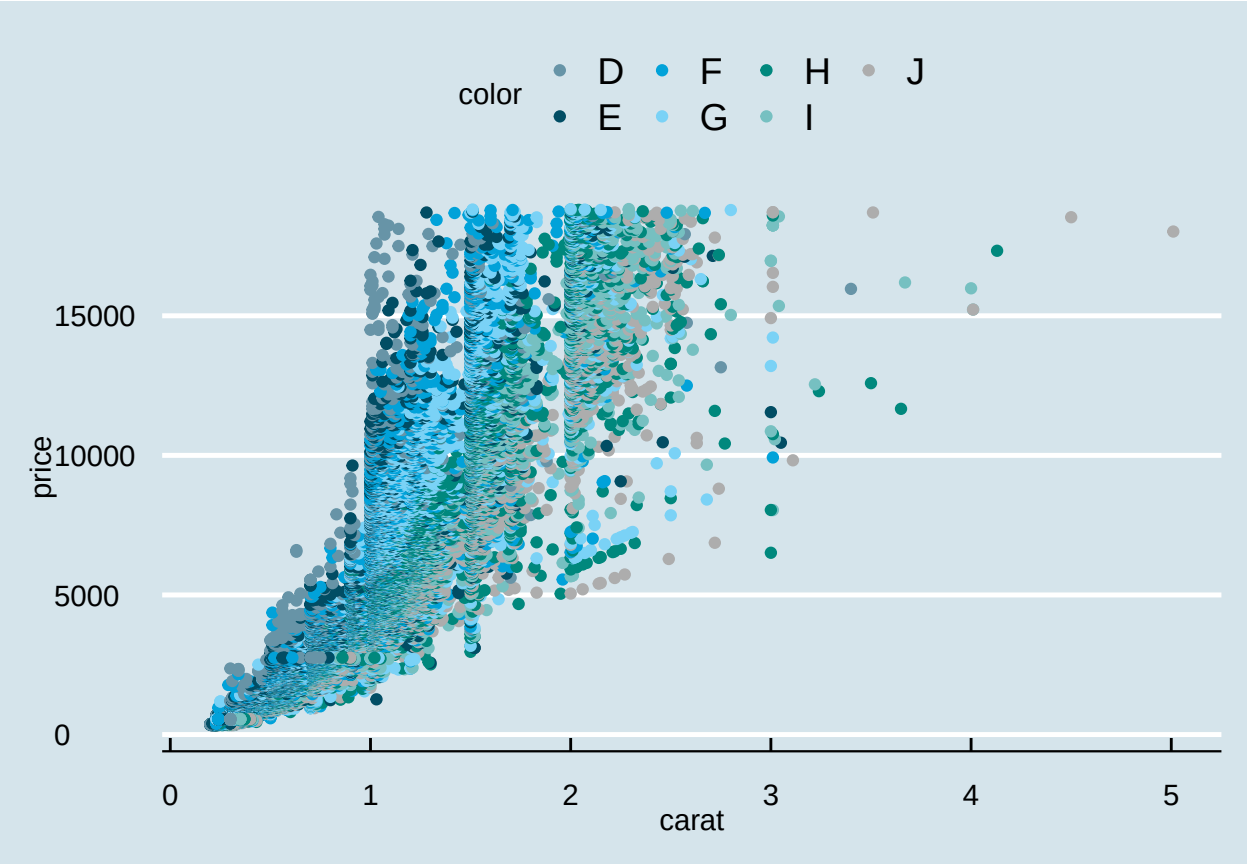
Themes in ggplot2

Another big advantage using ggplot2 is that the package provides users variety of theme options, such as "The Economist", "Excel", "Edward Tufte", and "The Wall Street Journal". To access the themes, we need to load the **ggthemes** library package.

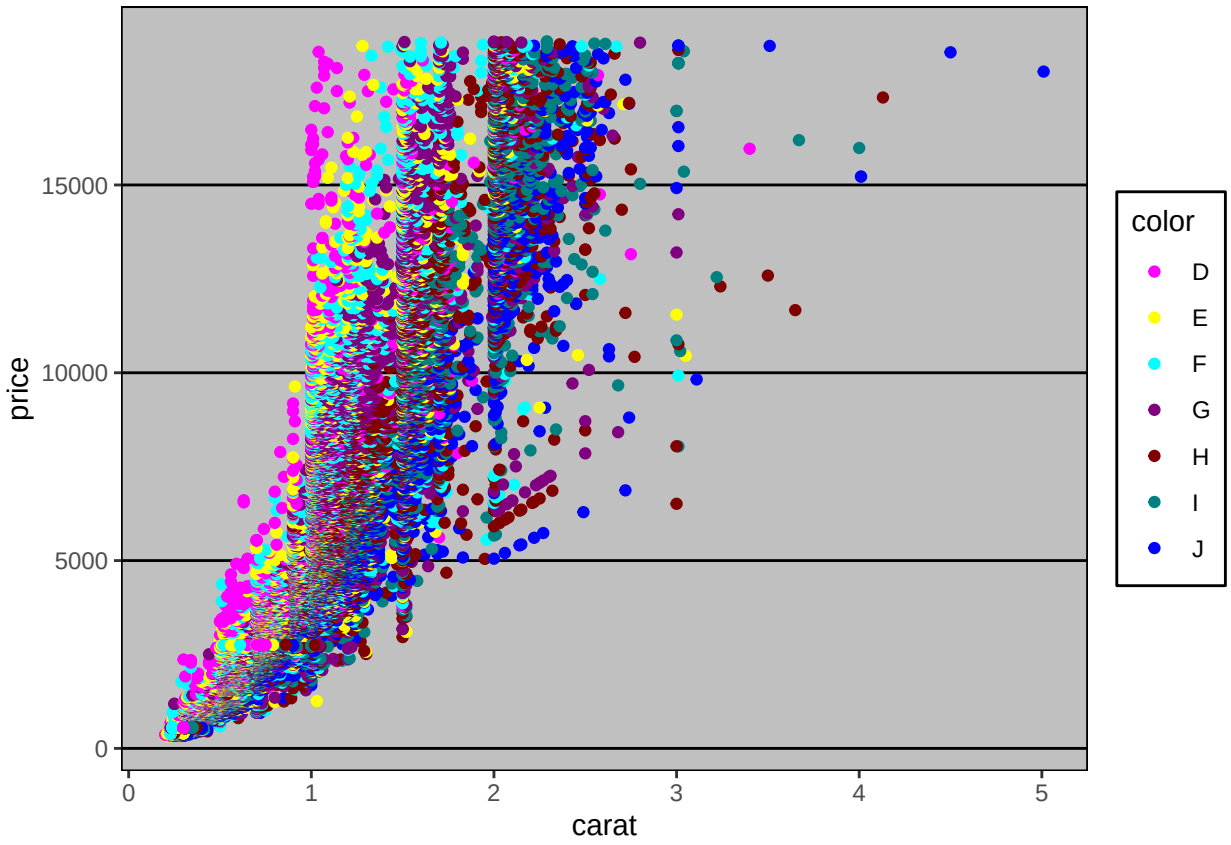
```
# install.packages("ggthemes")
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 4.1.2
```

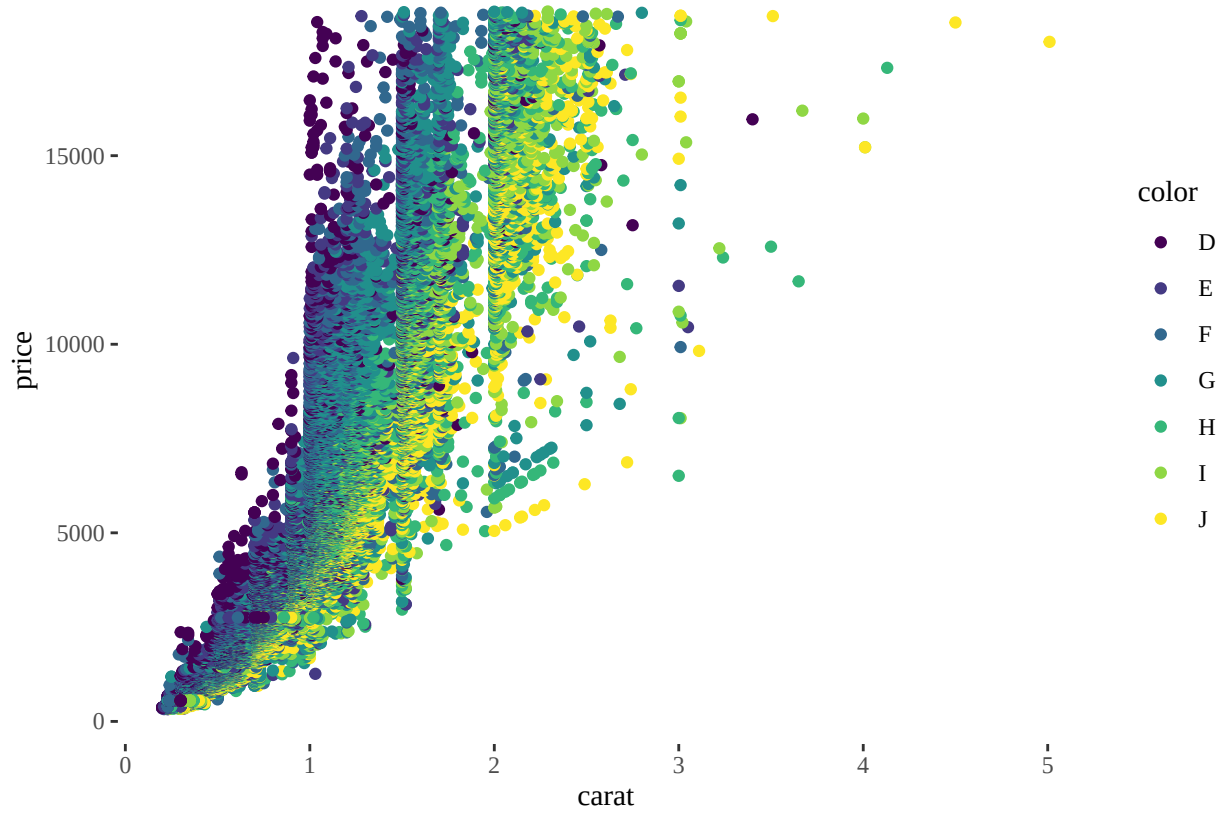
```
# e.g. Applying differen themes on a scatter plot.
g2 <- ggplot(diamonds, aes(x=carat, y=price)) +
  geom_point(aes(color=color))

g2 + theme_economist() + scale_color_economist()
```



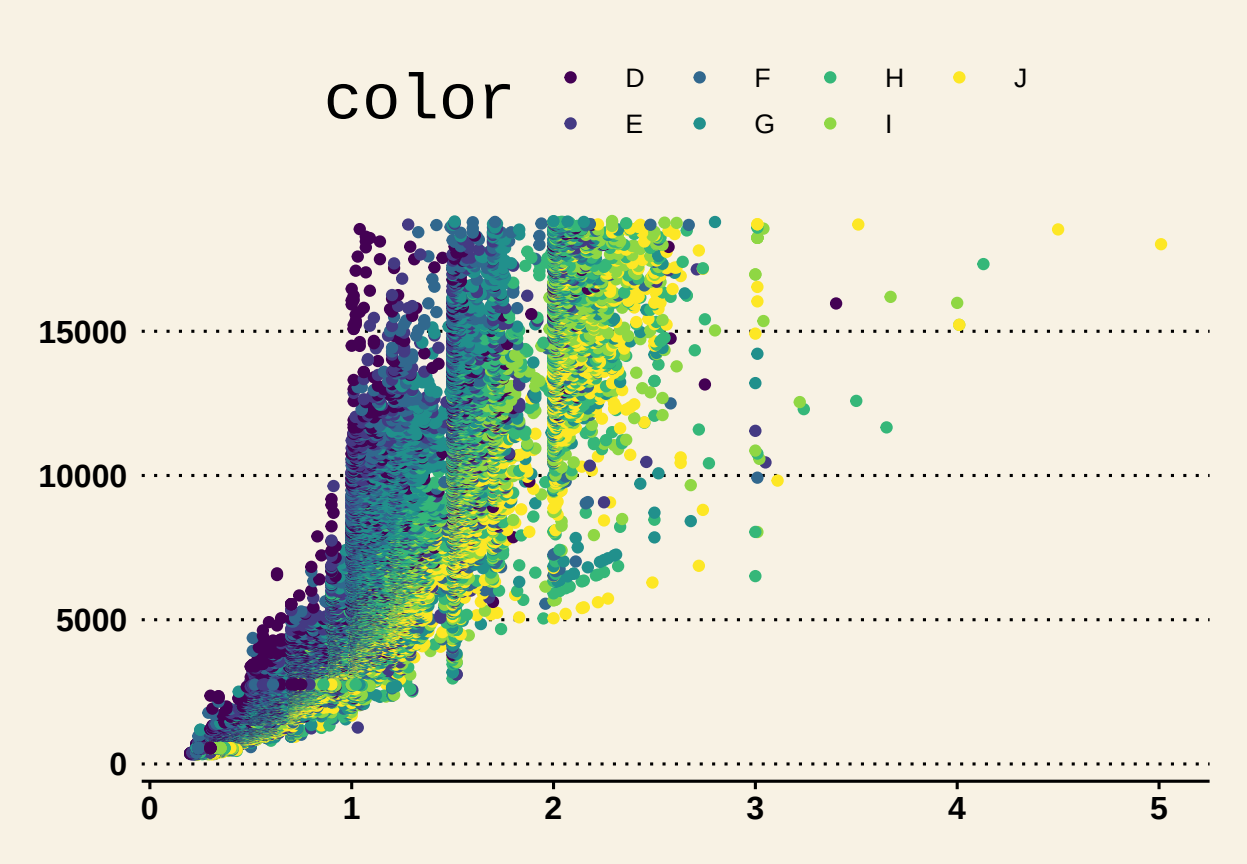

```
g2 + theme_excel() + scale_color_excel()
```




```
g2 + theme_tufte()
```




```
g2 + theme_wsj()
```


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