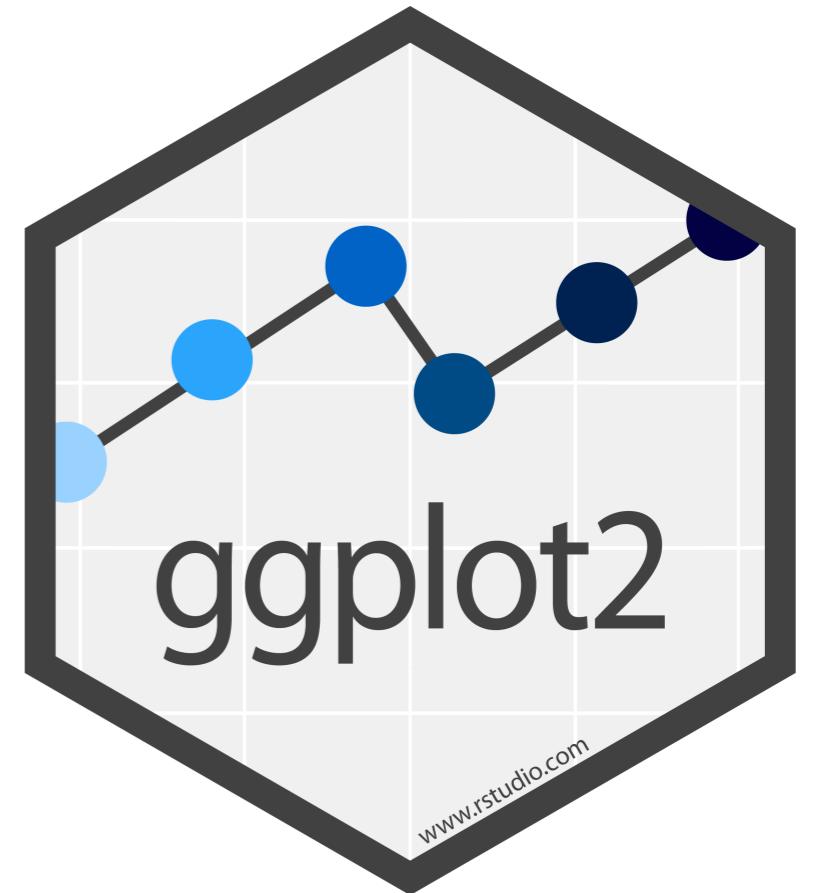
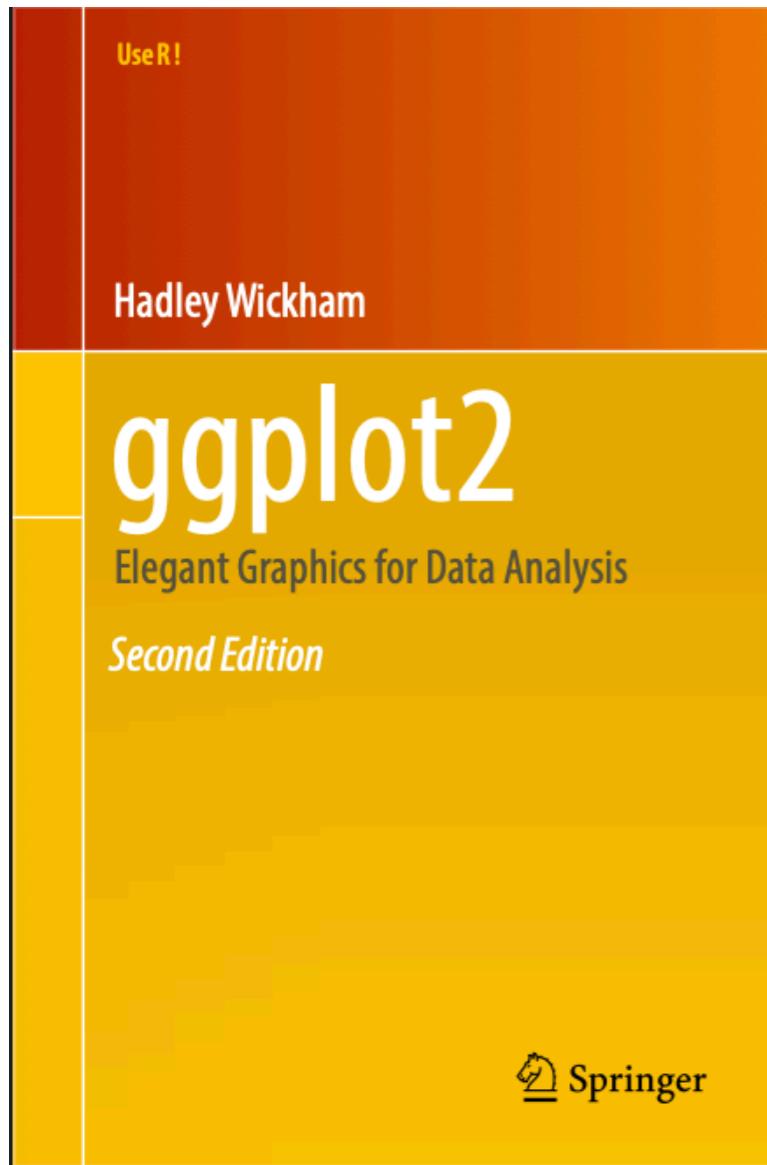


# Data Visualization Basics



# Outline

What is data visualization?

The Grammar of Graphics

Intro to ggplot2

Aesthetics

Scales

Geoms

Facets/Coordinate Systems/Theme

ggplot2 extensions!

# Data Visualization

“

The most important maxim for data analysis to heed, and one which many statisticians seem to have shunned, is this: “Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise.”

John Tukey

The Future of Data Analysis,  
*The Annals of Mathematical Statistics*, 1-67.

# Data Visualization

“

Modern data graphics can do much more than simply substitute for small statistical tables.

At their best, graphics are **instruments for reasoning** about quantitative information.

Often the most effective way to describe, explore, and summarize a set of numbers—even a very large set—is to look at pictures of those numbers.

Edward Tufte

*The Visual Display of Quantitative Information*, 2001.

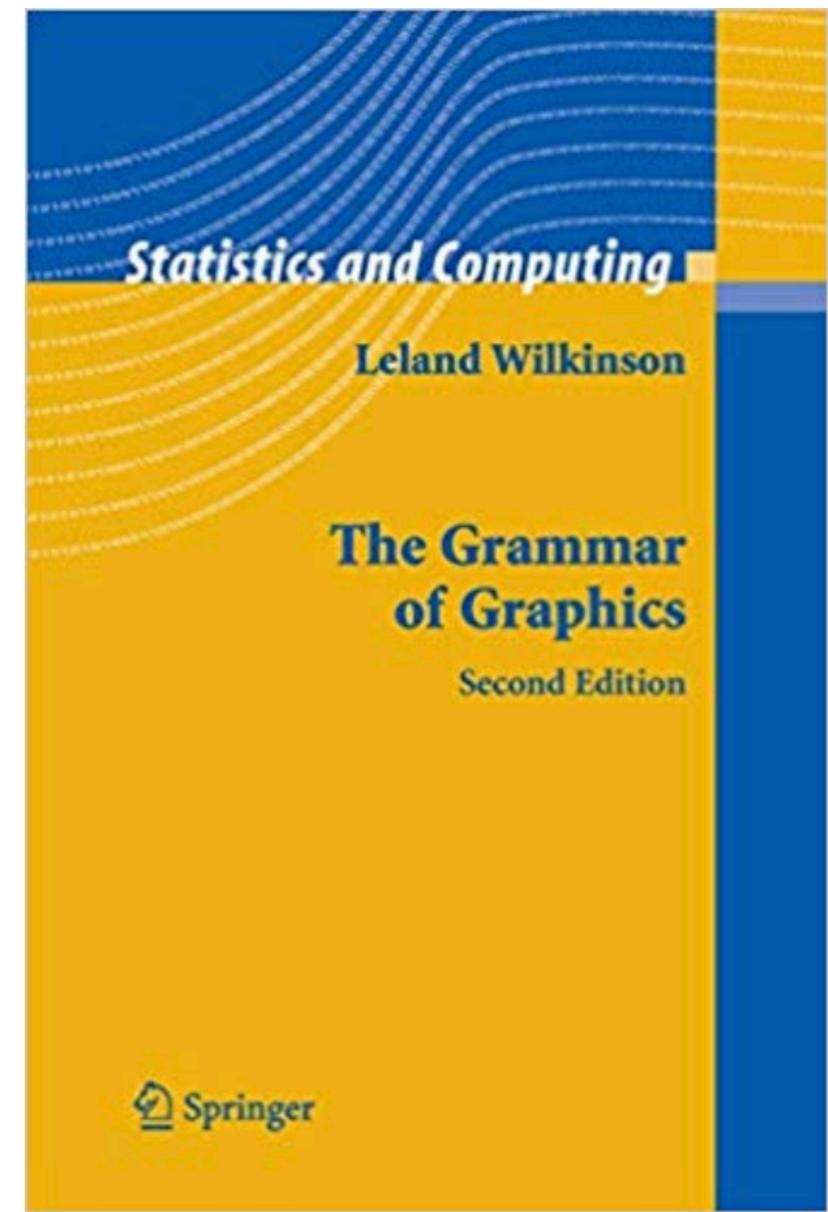
Emphasis added.

# The Grammar of Graphics

The grammar of graphics is a conceptual framework for statistical graphics

Originally by Wilkinson (2005)

Wickham (2009) expanded it and implemented it in the ggplot2 R package



# The Grammar of Graphics

“

The grammar of graphics is an answer to the question: what is a statistical graphic? ... In brief, the grammar tells us that a statistical graphic is a mapping from data to aesthetic attributes (colour, shape, size) of geometric objects (points, lines, bars).

Wickham, *ggplot2*, p.5

# The Grammar of Graphics

Every plot needs

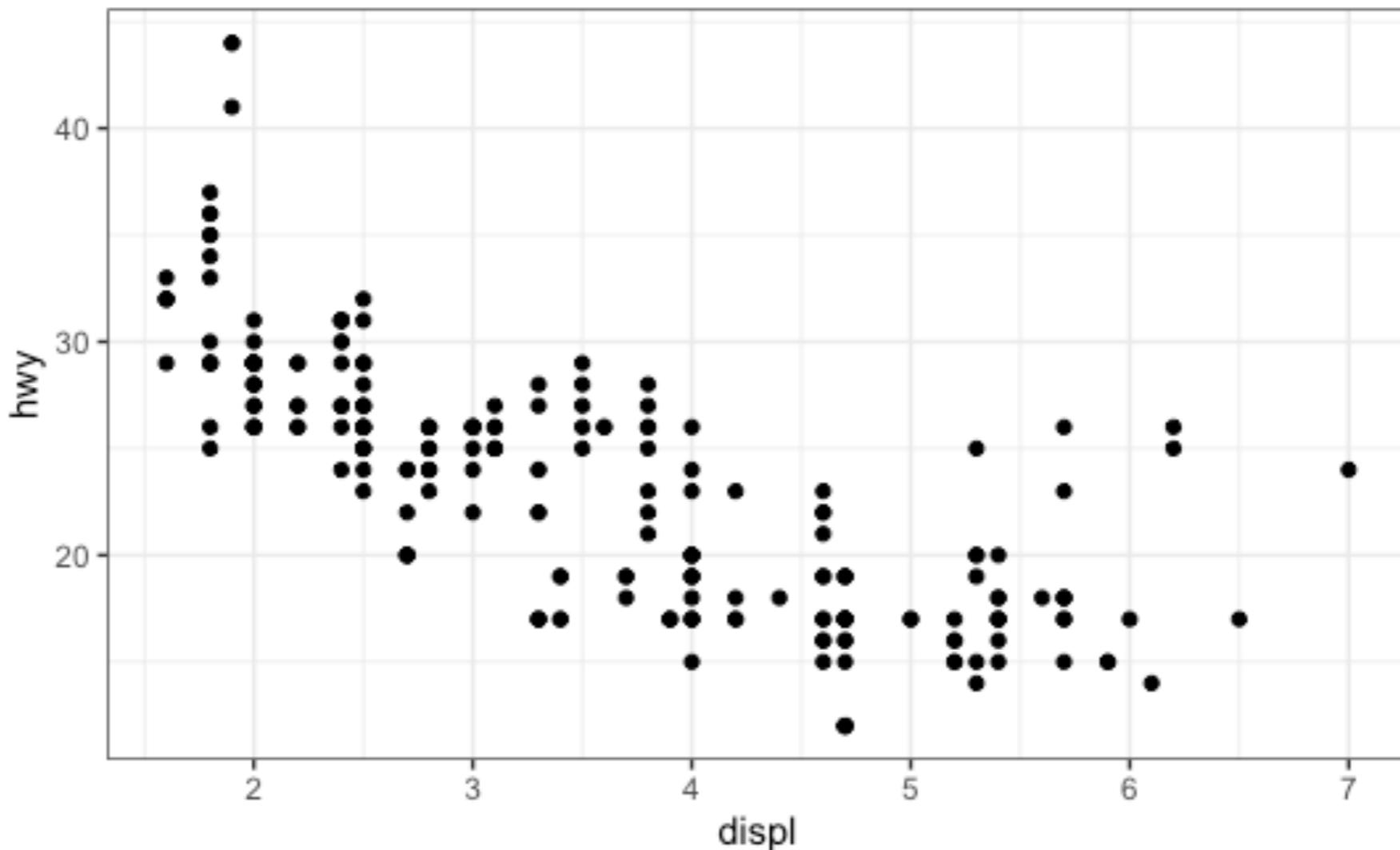
1. Data (with aesthetics)
2. Layers composed of
  - 2.1 Geometric objects (geoms)
  - 2.2 Statistical transformations
3. Scales
4. A coordinate system
5. A facet specification (possibly none)
6. A theme

# The ggplot2 package

Code:

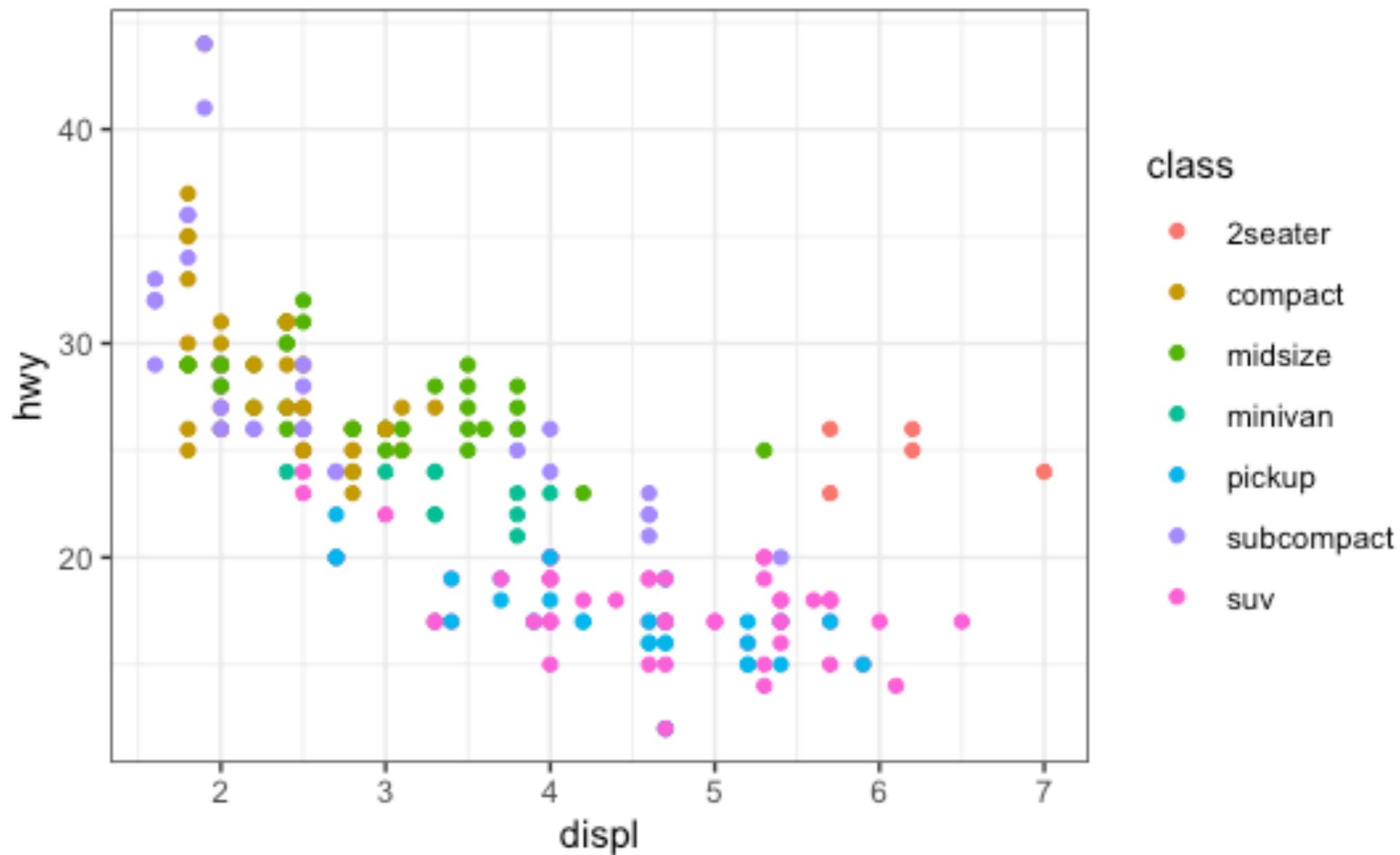
```
ggplot(data = mpg, mapping = aes(x = displ,  
y = hwy)) + geom_point()
```

Output:



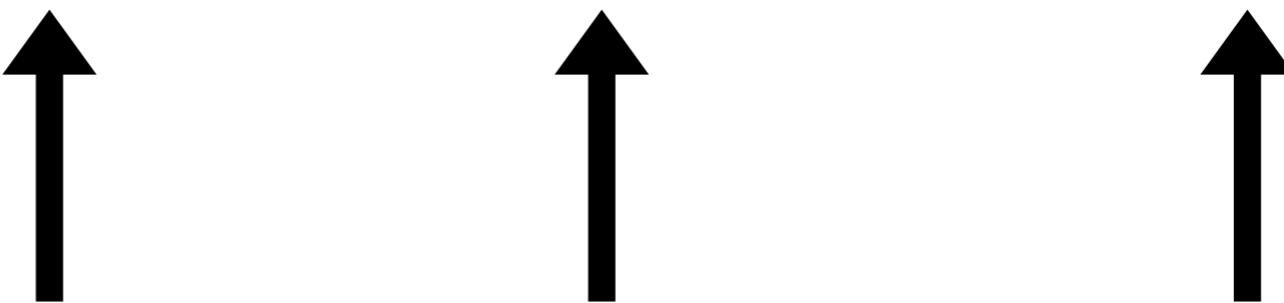
# The ggplot2 package

```
ggplot(data = mpg, mapping = aes(x = displ,  
y = hwy, color = class)) + geom_point()
```



# The ggplot2 package

```
mapping = aes(x = displ, y = hwy, color = class))
```



x, y, and color are all **aesthetics**

Other aesthetics include shape and size

# The Diamonds Dataset

# Your Turn!

The `ggplot2` package has a dataset called *diamonds*

Look at the variables in the dataset and determine the type of data you are working with (categorical, quantitative, etc.)

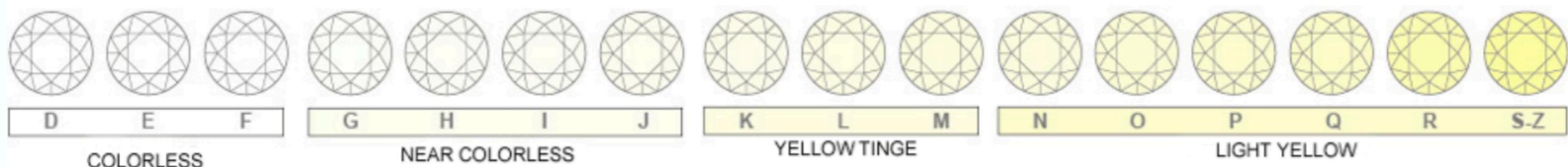
Choose one categorical and one quantitative variable to plot!  
(think about what plots are appropriate for the types of data)

Follow my general template in the previous slides to make the plots (changing names and the geom type)

# Diamonds

Cut      Ideal > Premium > Very Good > Good > Fair

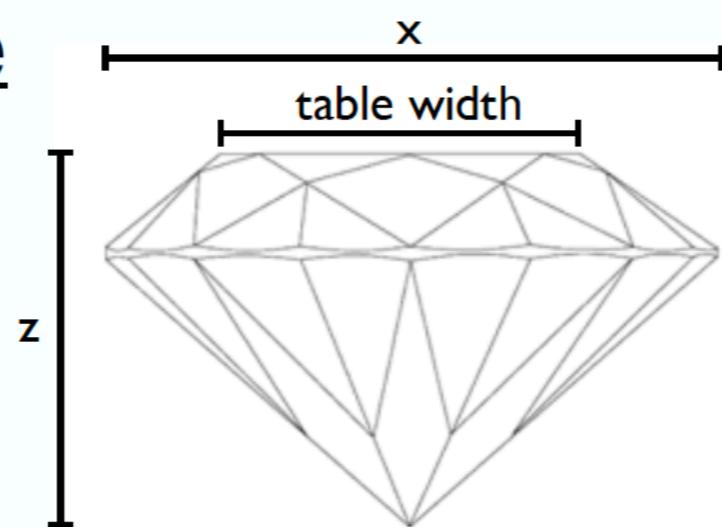
Color



Clarity

Clarity	Flawless	Internally Flawless	Very Very Slightly Included	Very Slightly Included	Slightly Included	Included					
Grade	FL	IF	VVS1	VVS2	VS1	VS2	SI1	SI2	I1	I2	I3

Everything else



also

y  
carat  
price  
depth = z / diameter  
table = table width / x

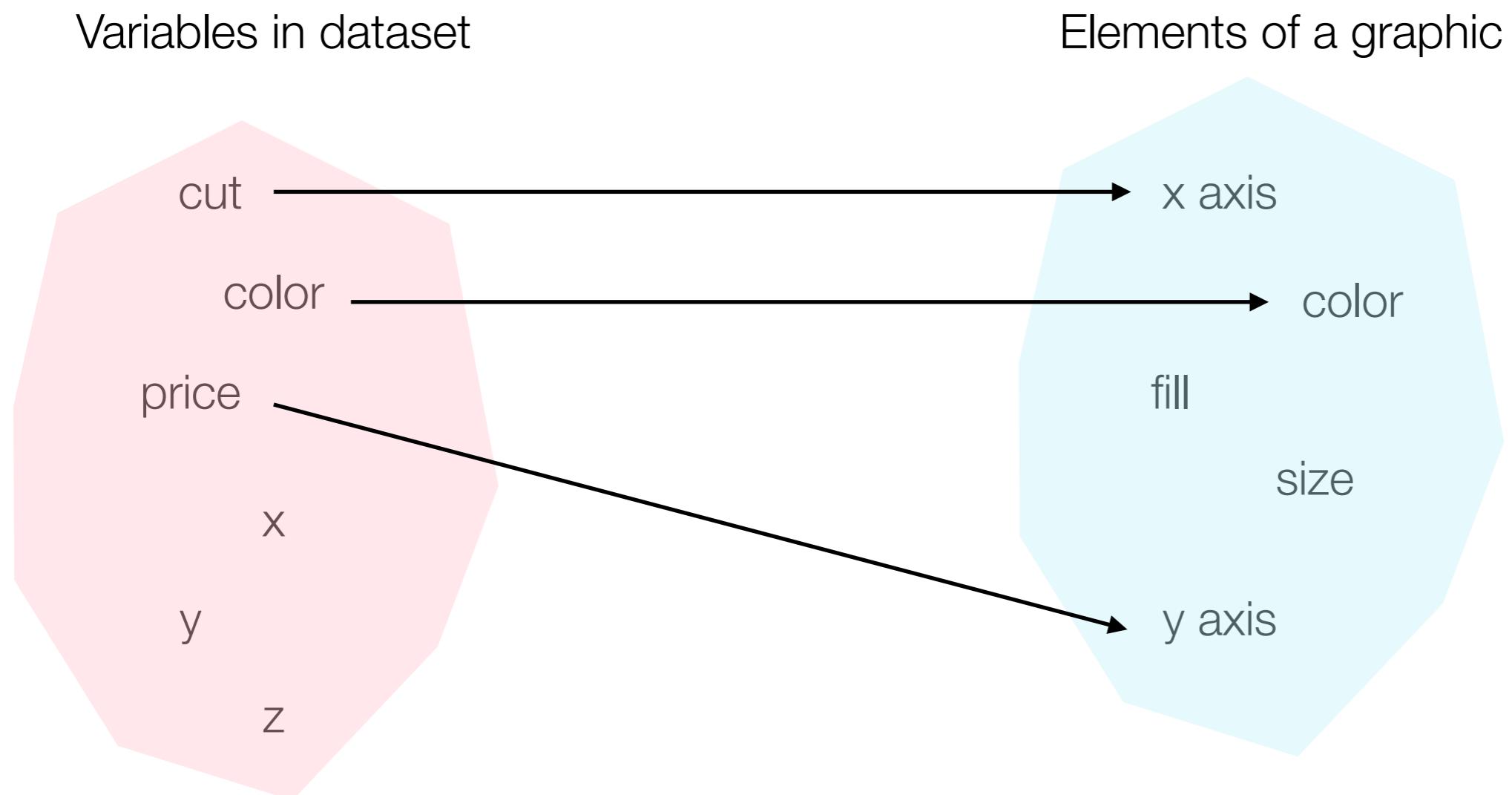
Source : <http://www.affordablediamondsonline.com/diamondshapes.jpg>, [http://beaufortsjeweler.com/images/diamond\\_color\\_chart.gif](http://beaufortsjeweler.com/images/diamond_color_chart.gif), [http://www.am-diamonds.com/images/anatomy\\_diamond.gif](http://www.am-diamonds.com/images/anatomy_diamond.gif)

Credit: David J Kahle

# Aesthetics

# Aesthetics

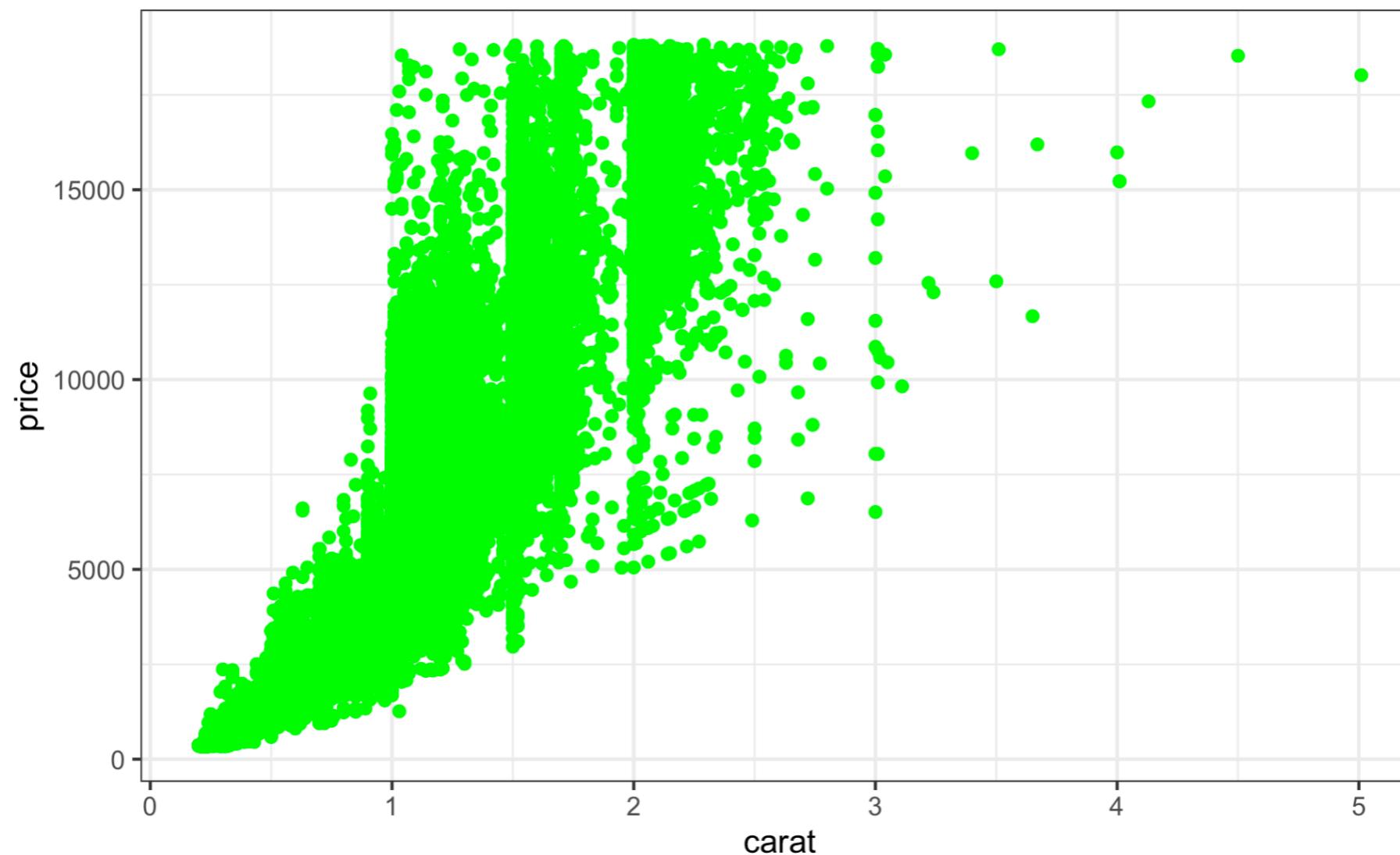
Aesthetics are a mapping from the variables to a part of the graph



There are a lot of aesthetics (like 40!) so you can make very complex graphics! **But don't abuse the power!!**

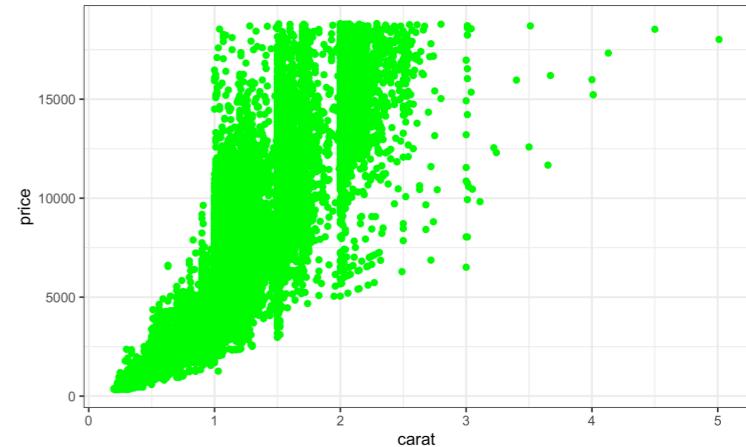
# Set vs. Map

How would I make the following plot?



# Set vs. Map

How would I make the following plot?

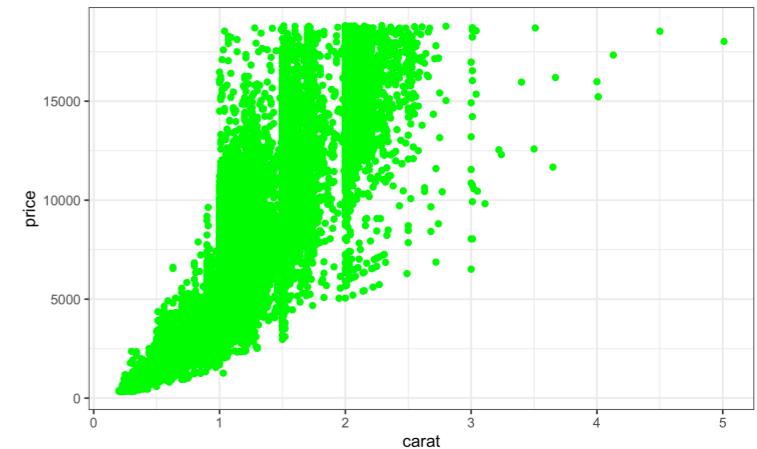


```
ggplot(data = diamonds, aes(carat, price, color = "green"))  
+ geom_point()
```

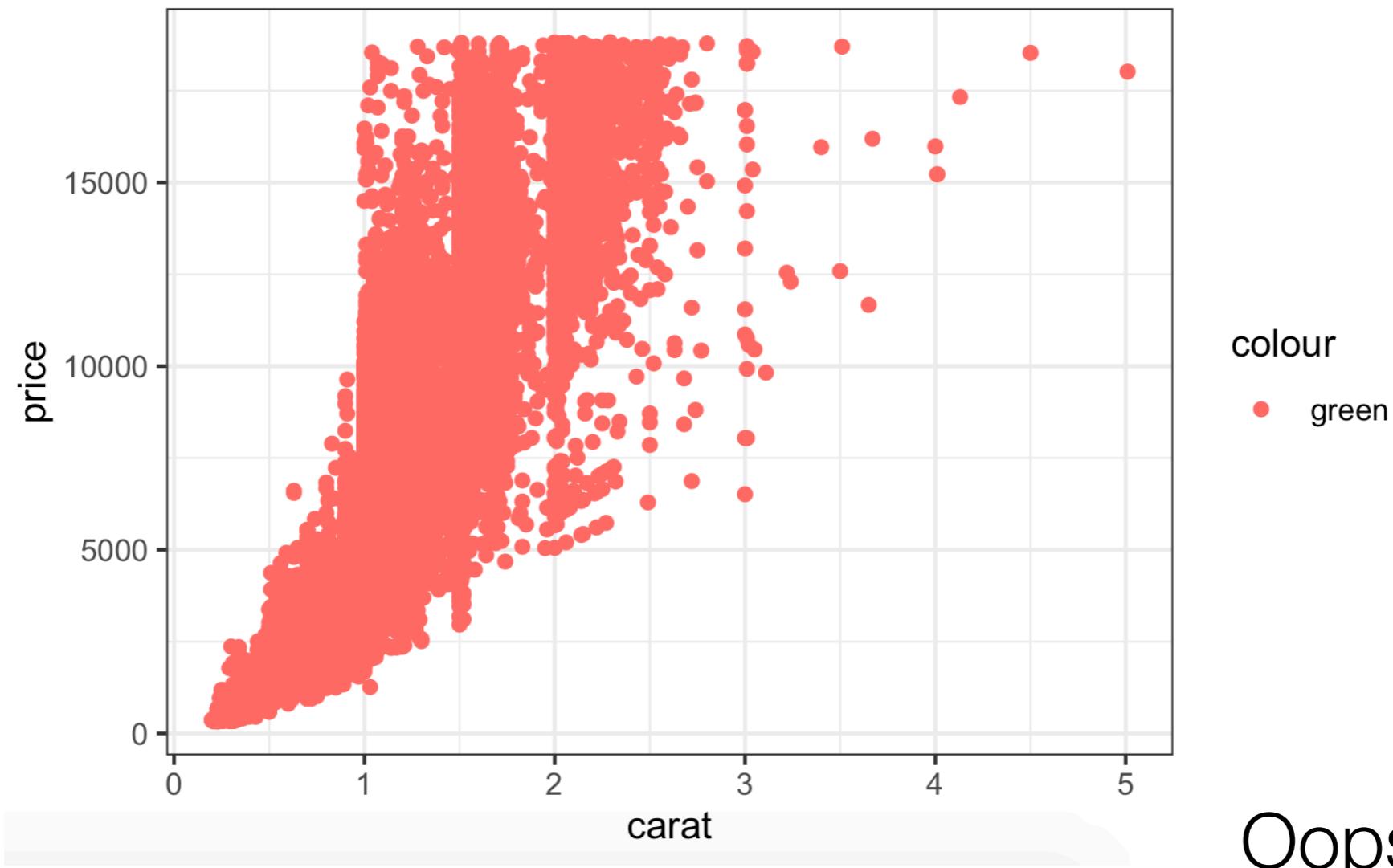
```
ggplot(data = diamonds, aes(carat, price)) +  
  geom_point(color = "green")
```

# Set vs. Map

How would I make the following plot?



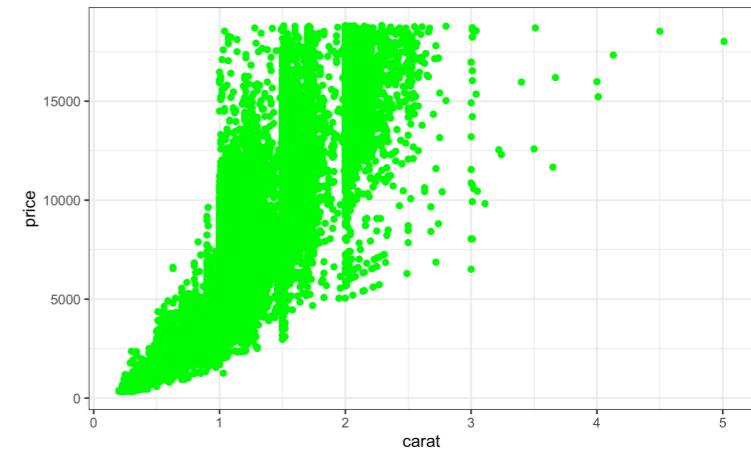
```
ggplot(data = diamonds, aes(carat, price, color = "green"))  
+ geom_point()
```



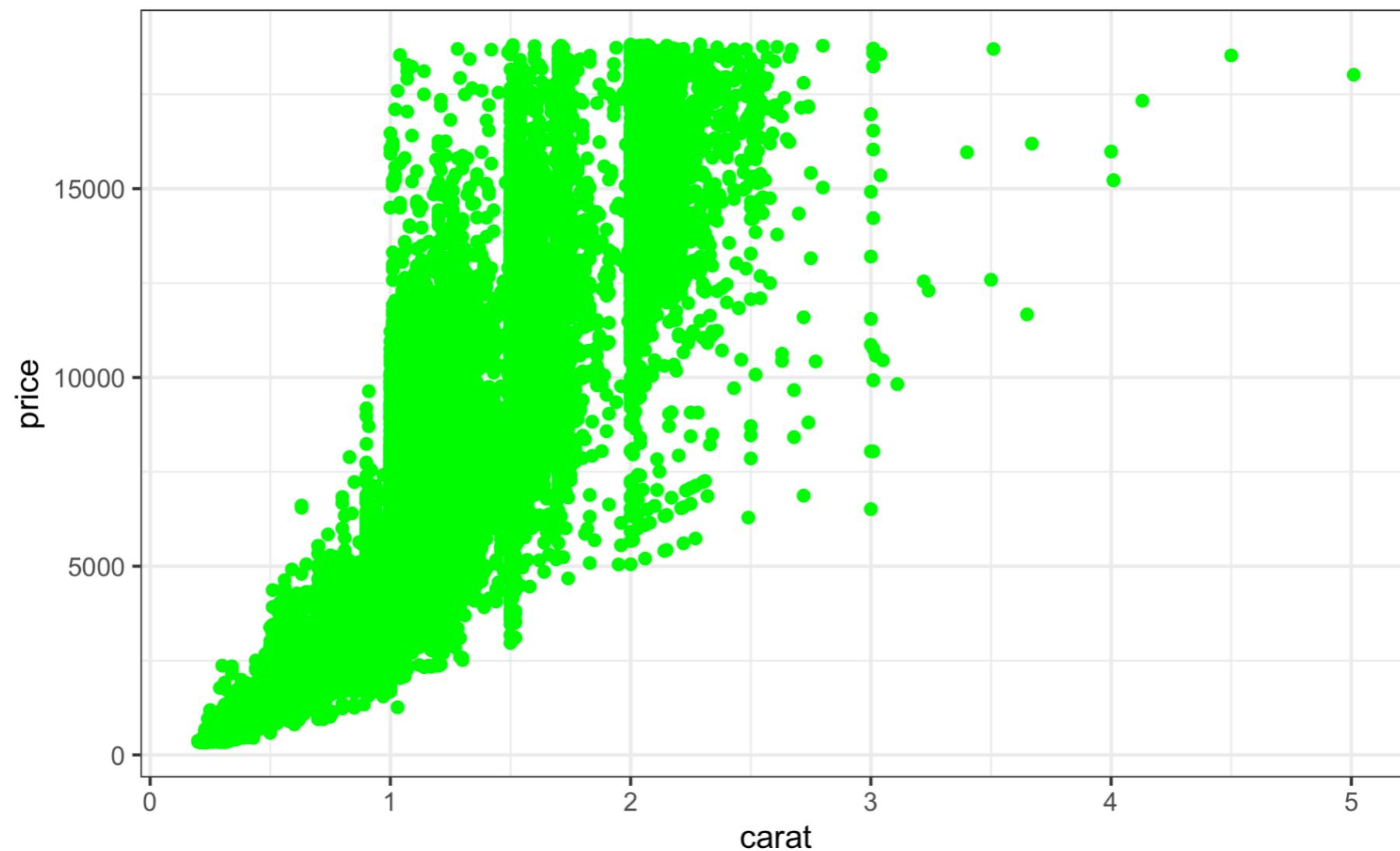
Oops!!!

# Set vs. Map

How would I make the following plot?



```
ggplot(data = diamonds, aes(carat, price)) +  
  geom_point(color = "green")
```

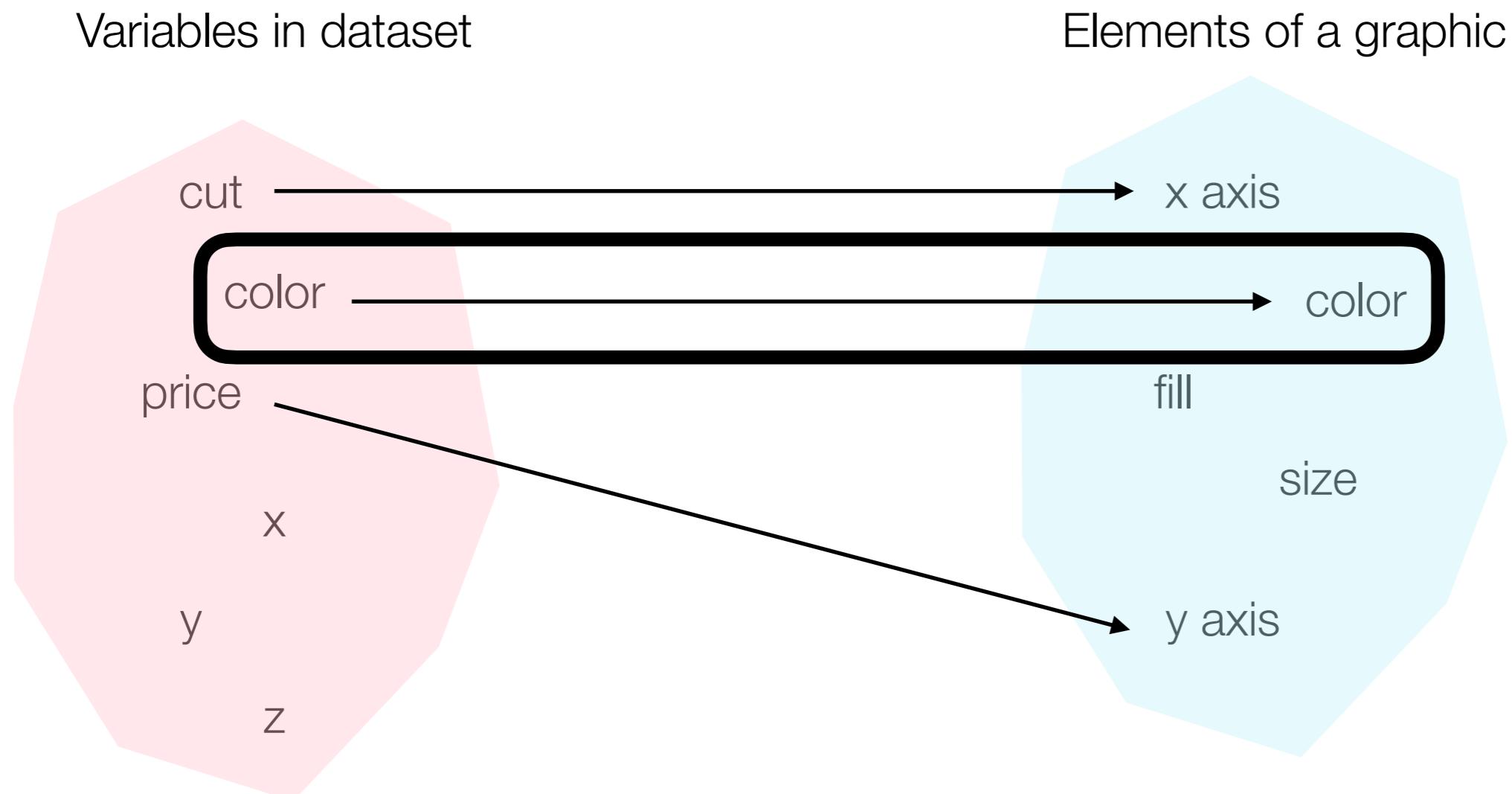


# Scales

# Scales

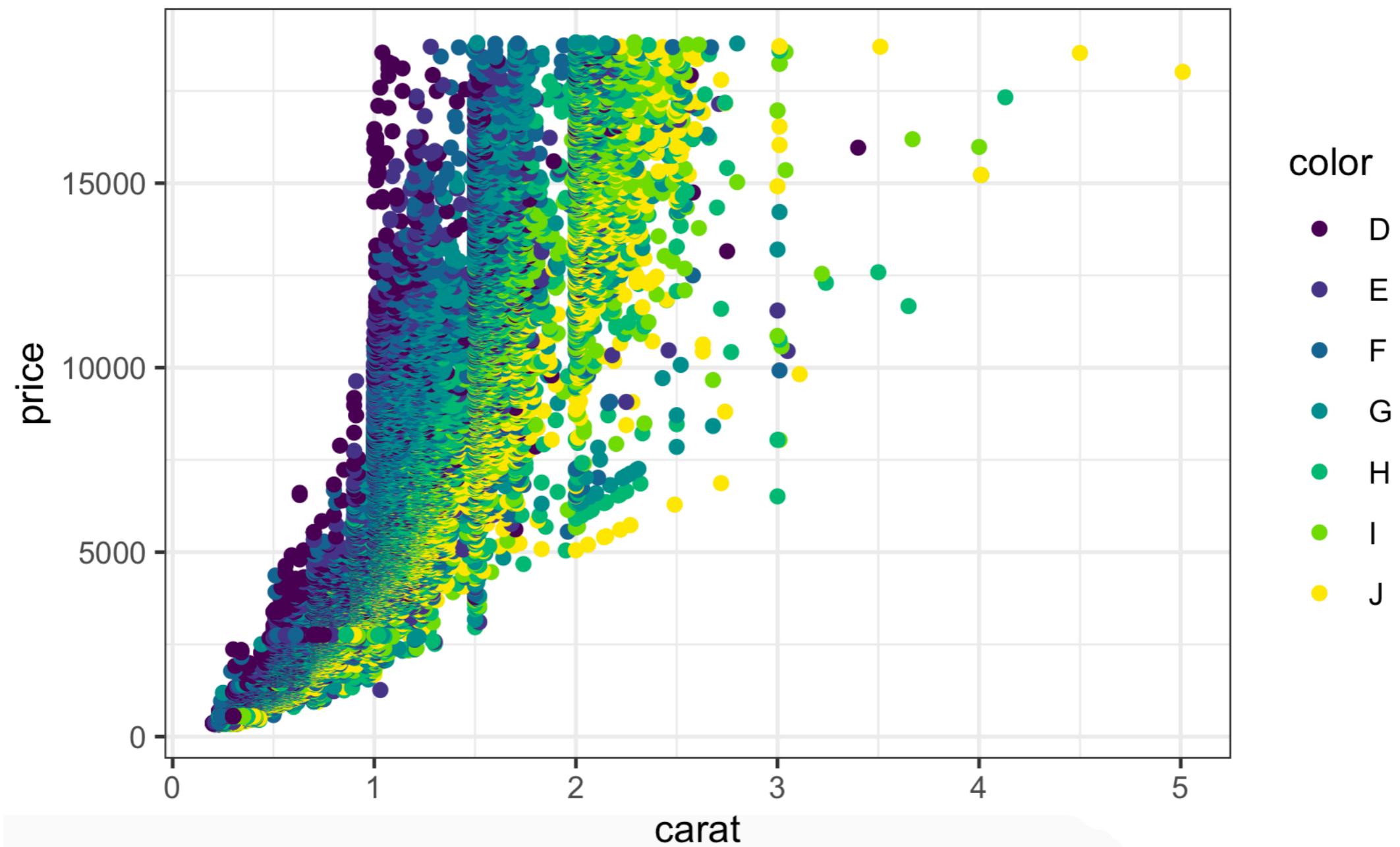
Scales are functions from the data space to the range of visual properties

A scale is a function that maps levels of a variable to aesthetic values



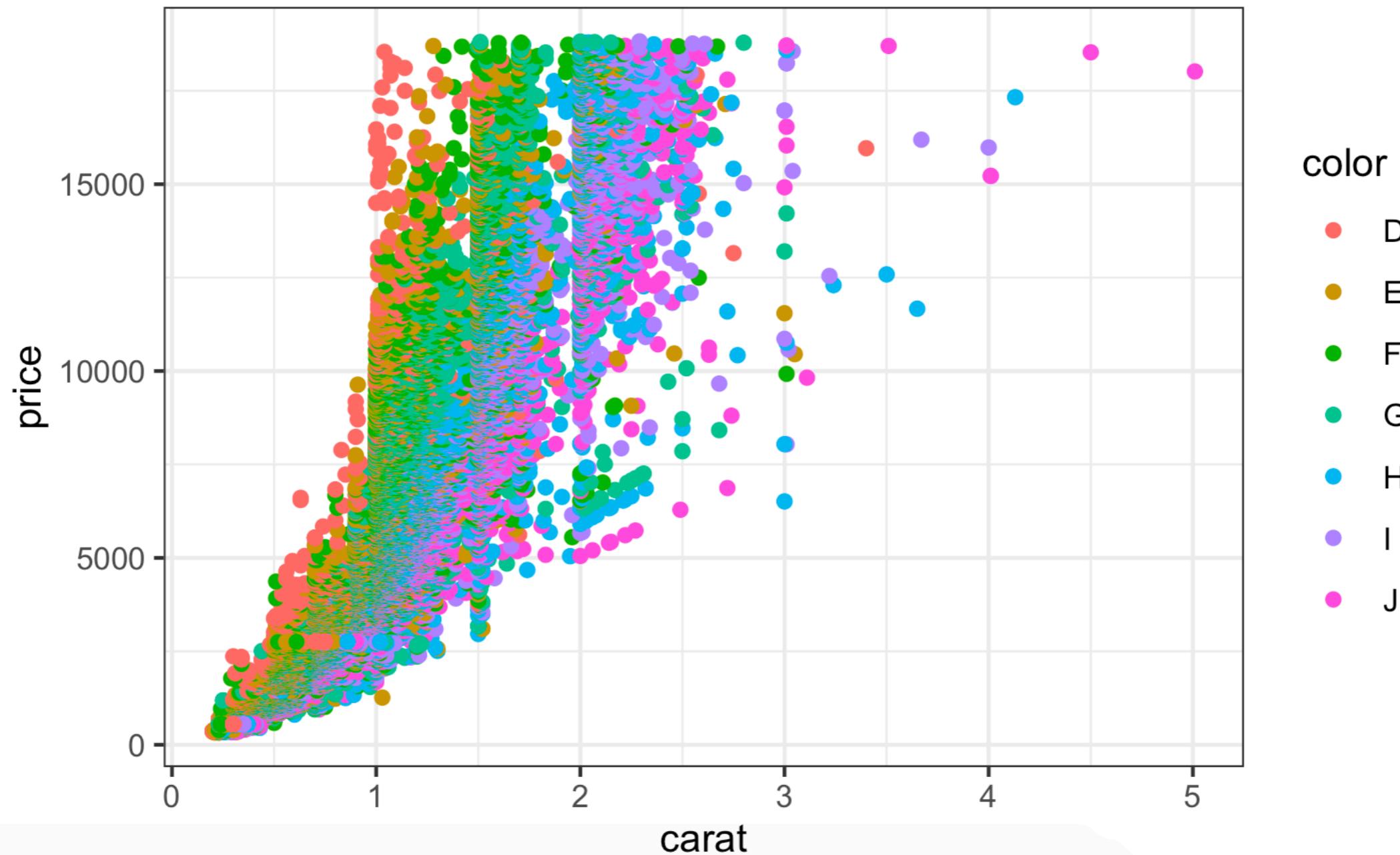
# Scales

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

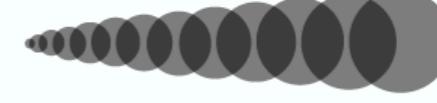
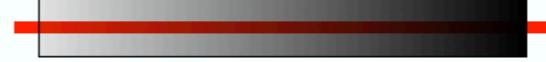


# Scales

```
ggplot(data = diamonds, aes(carat, price, color = color)) +  
  geom_point() + scale_color_hue()
```



# Scales

Aesthetic effect by variable type	Discrete	Continuous
Color	Rainbow of colors 	Color gradient 
Size Bubble Charts	Discrete size steps 	Continuous size gradient 
Shape	Different shape for points 	
Alpha	Discrete transparency steps 	Continuous transparency gradient 

# Geoms

# Geoms

A Geom is shorthand for geometric object

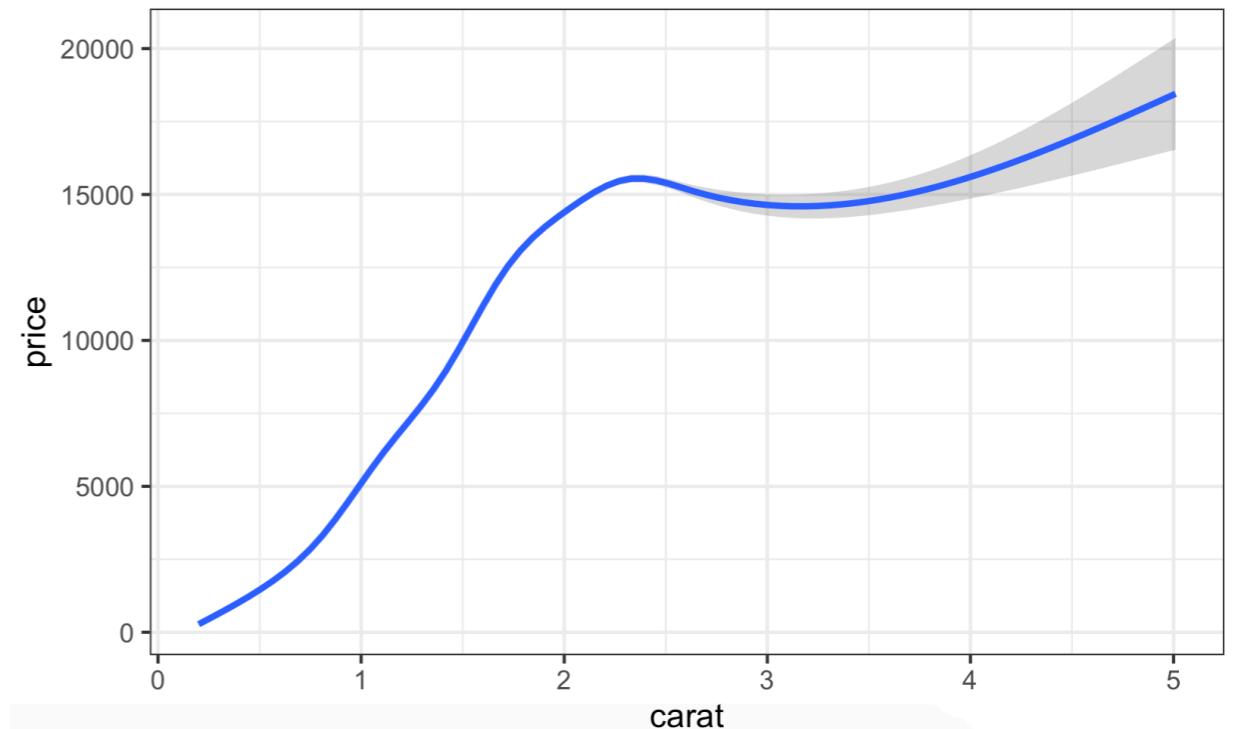
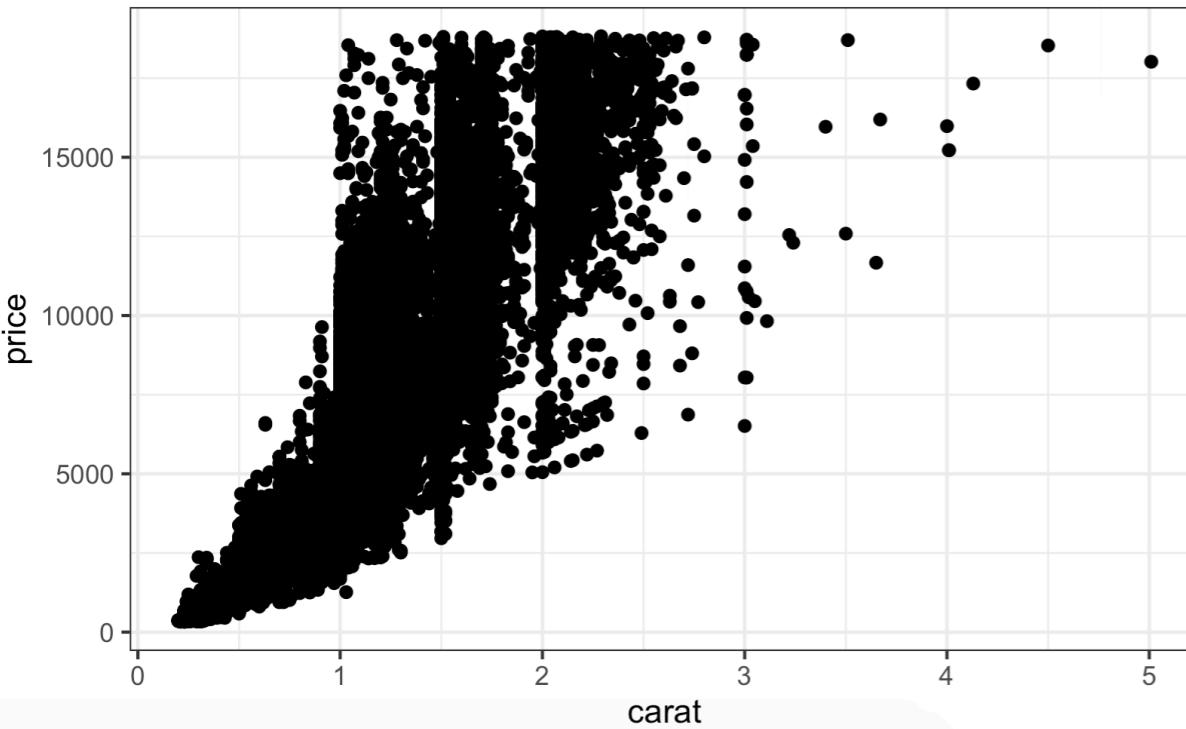
Geoms are made up of basic graphical elements

There are a lot of geoms as well!!!

For details on geoms, scales, aesthetics, etc. go to the  
[ggplot2 reference guide](#)

# Geoms

What is the difference between these two plots?

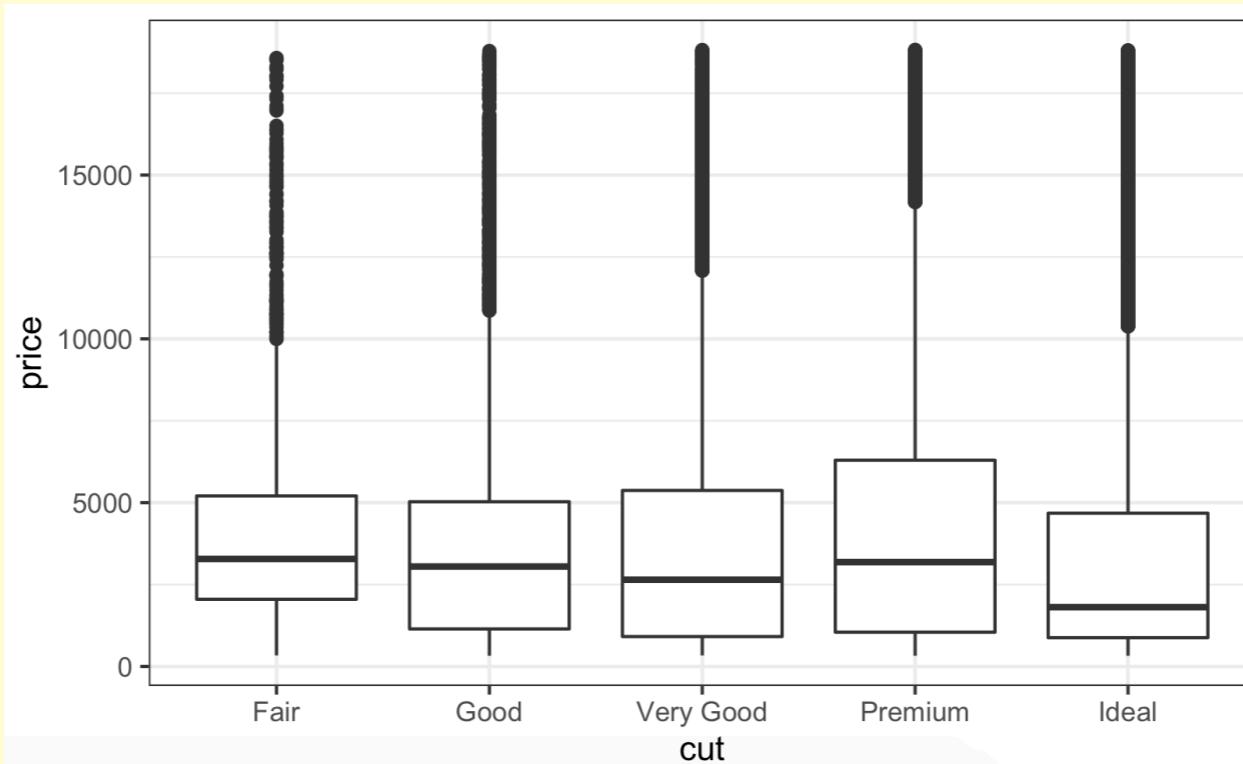
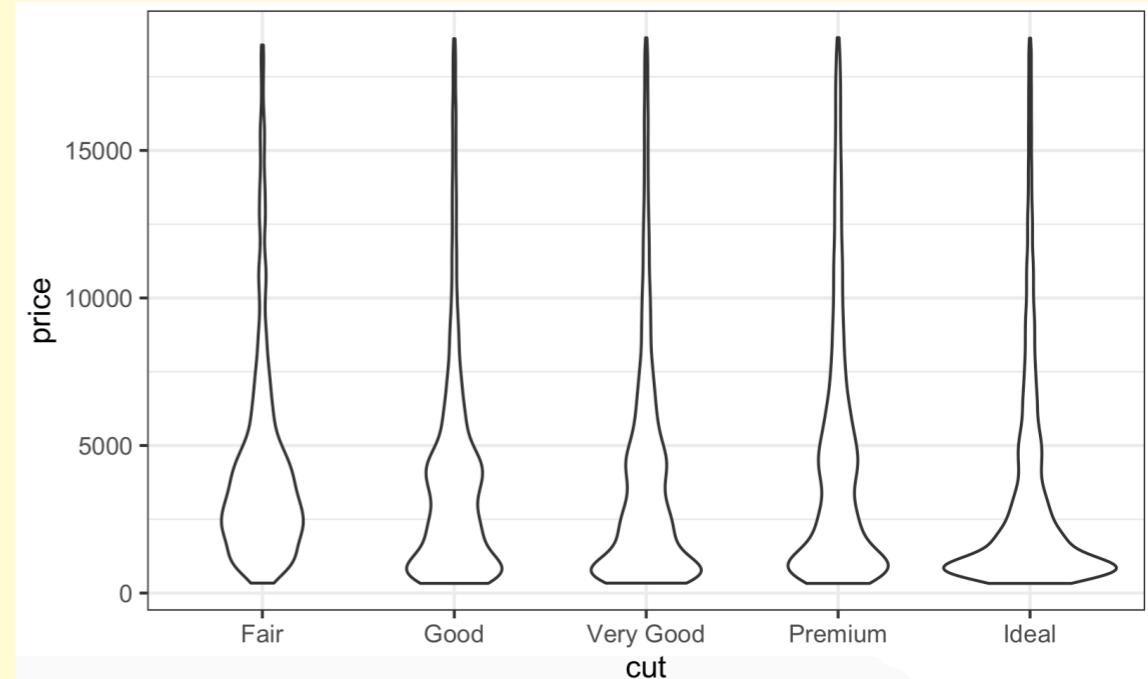
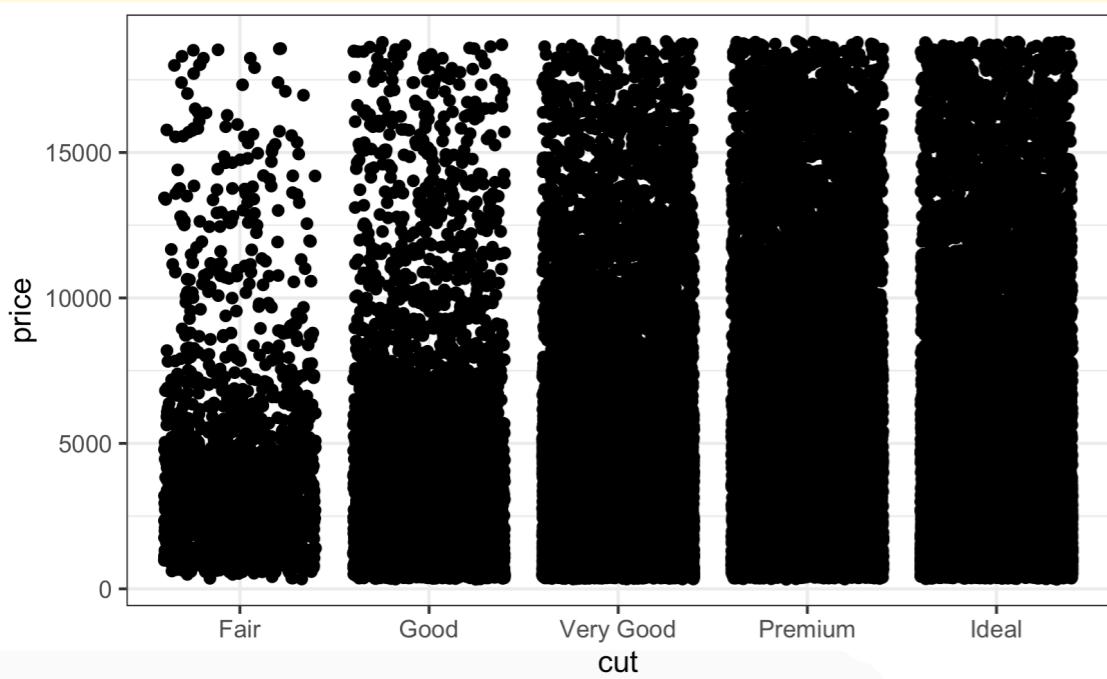


They both have the same axes and the same data!

The difference is the geometric object that represents the data!

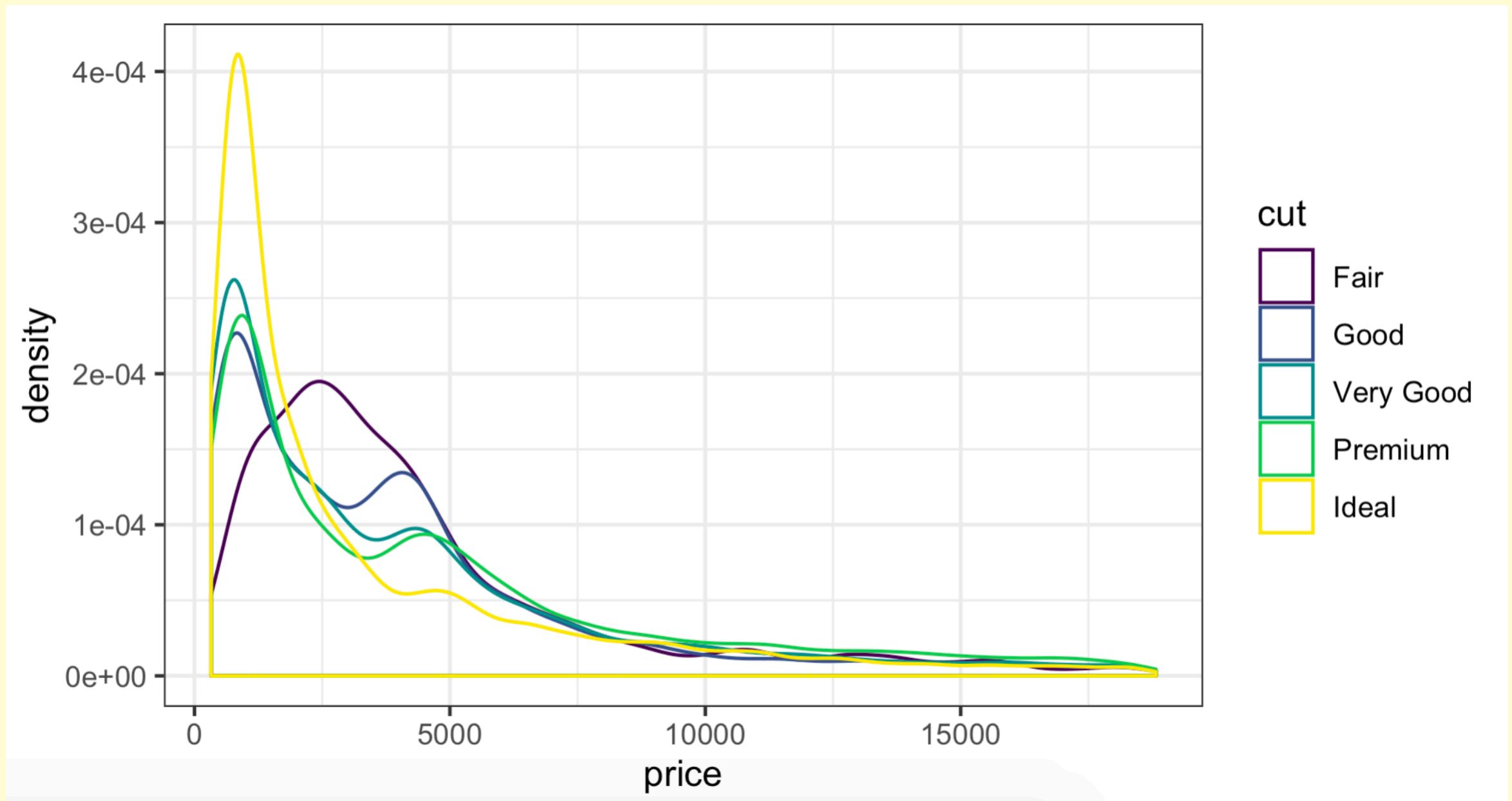
# Your Turn!

With those around you, discuss how you would make the following plots:



# Your Turn!

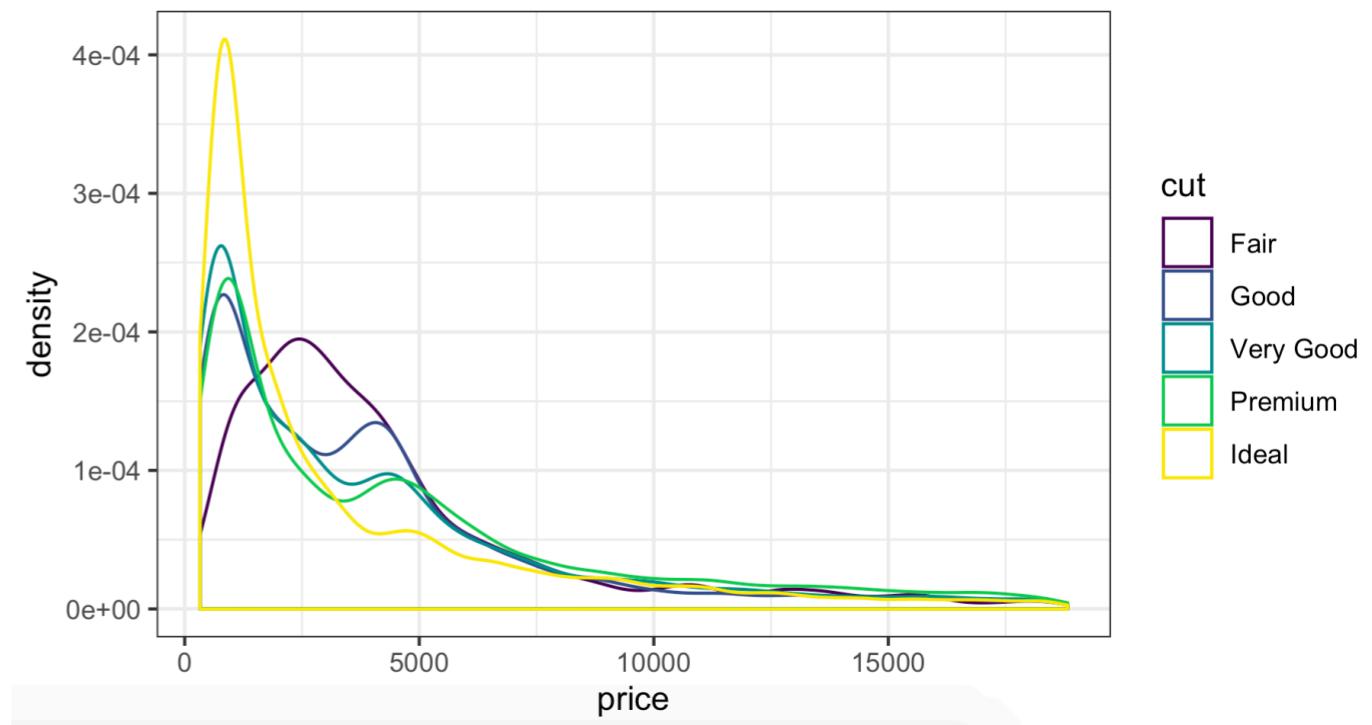
With those around you, discuss how you would make this plot:



# Facets

This plot looks a little crowded!

Maybe we could split up the plot so that each category gets its own plot!

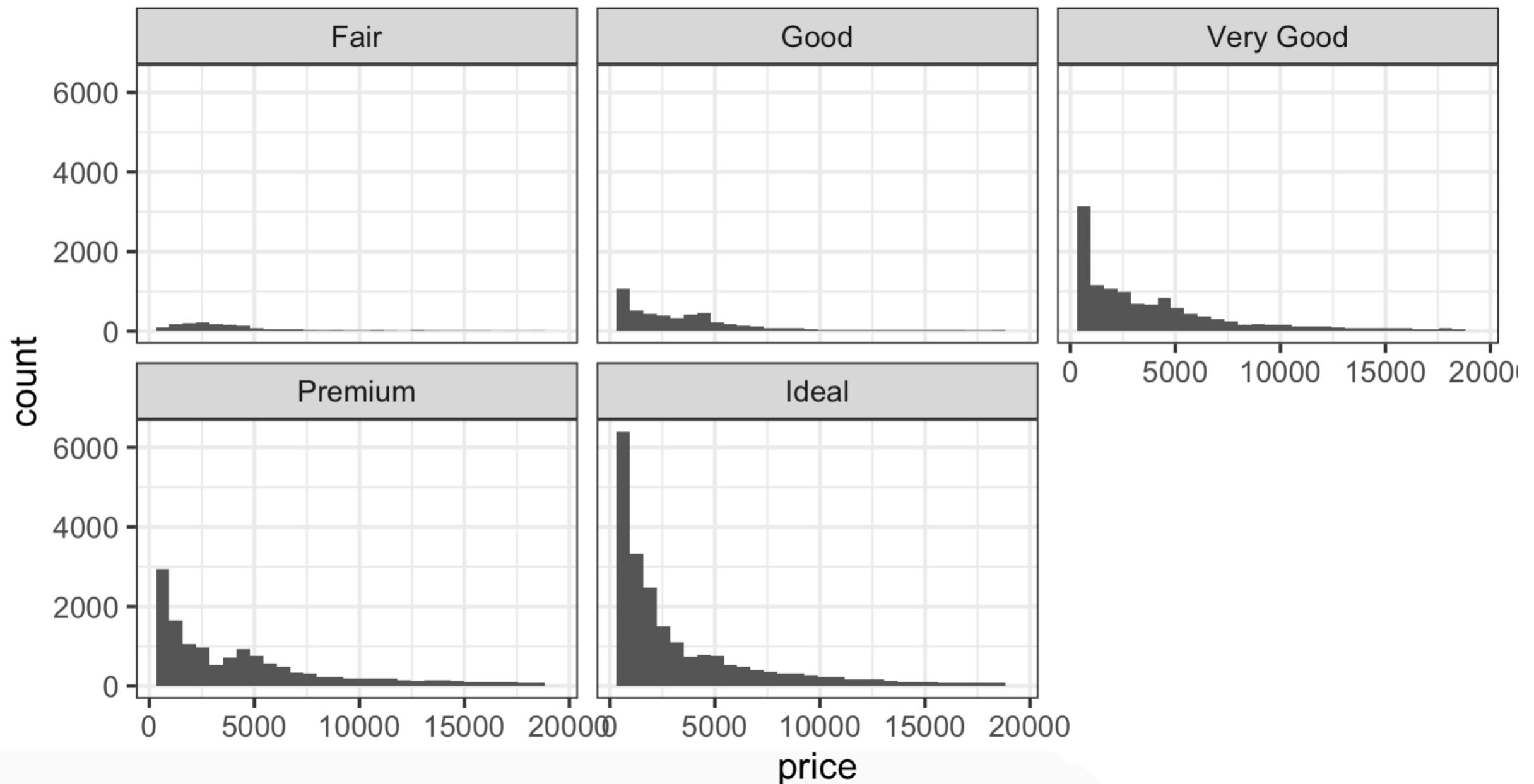


Splitting up the plot into conditional plots based on a discrete variable is called faceting!!

# Facets

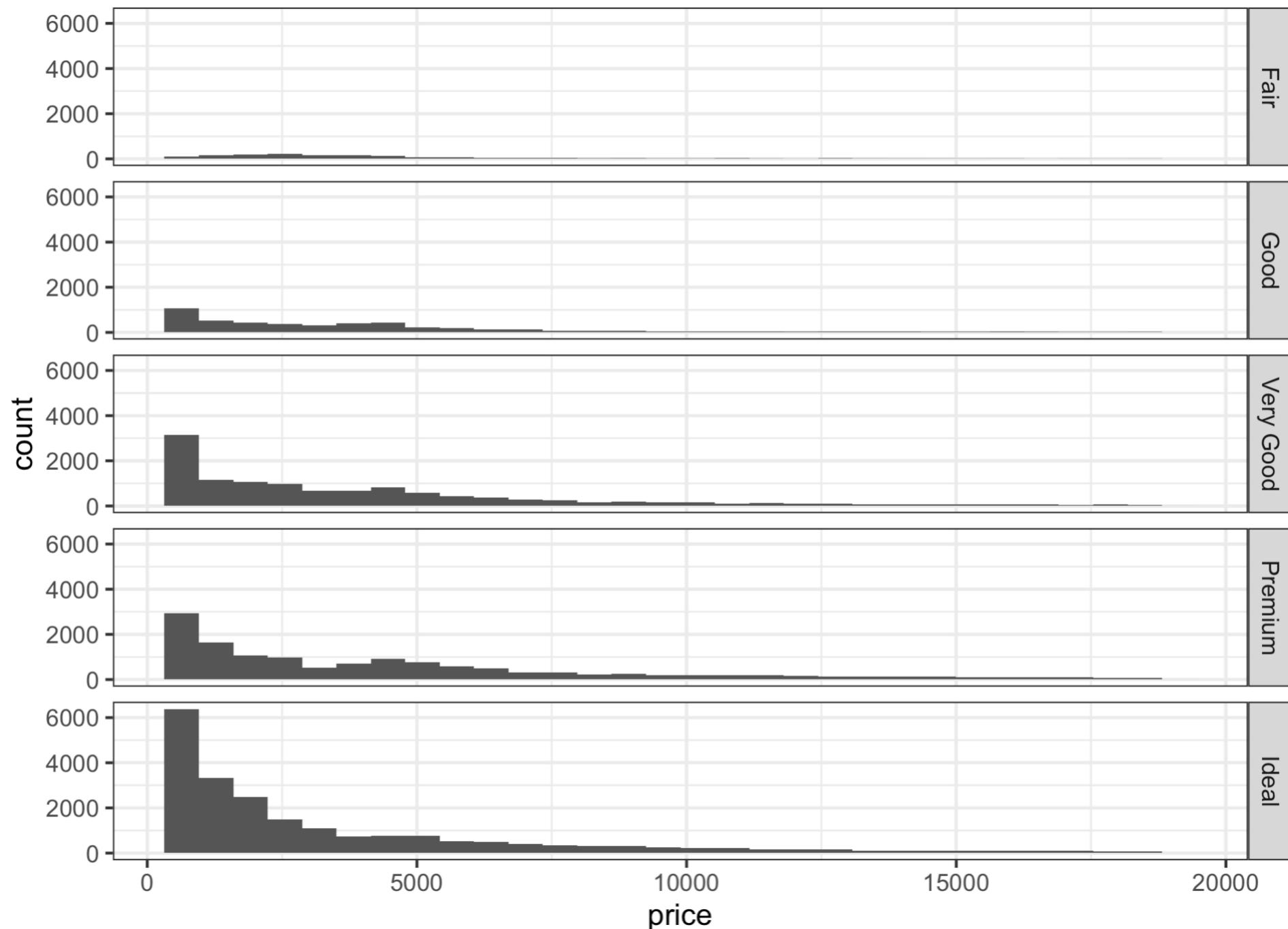
```
ggplot(diamonds, aes(price)) + geom_histogram() + facet_wrap(vars(cut))
```

Note vars() not aes()



# Facets

```
ggplot(diamonds, aes(price)) + geom_histogram() + facet_grid(vars(cut))
```

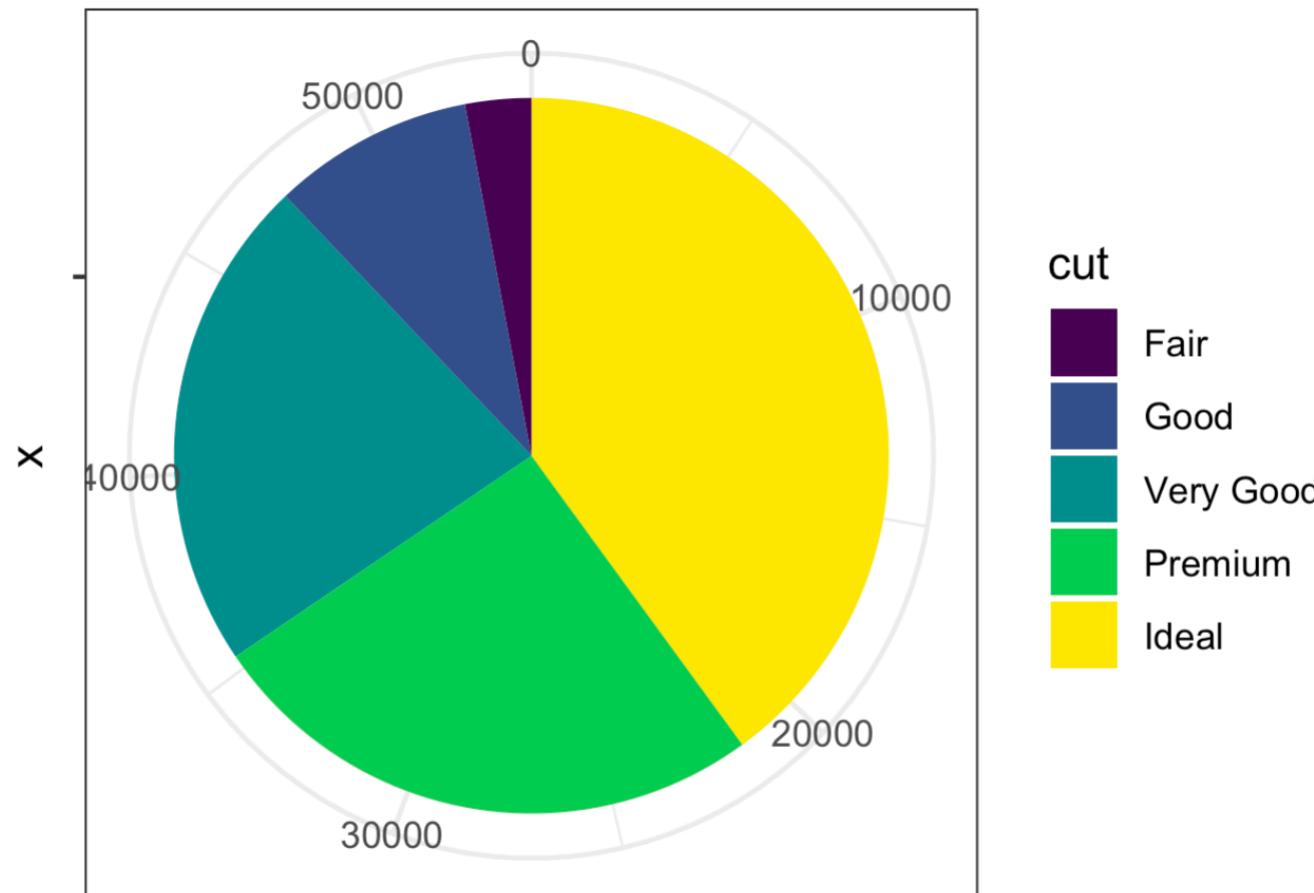


# Coordinate System

Sometimes you may need to change the coordinate system you are working with

Ex: To make a pie chart (Eww!), you need to use polar coordinates

```
diamonds %>%  
  group_by(cut) %>%  
  summarise(n = n()) %>%  
  ggplot(aes(x = "", y = n, fill = cut)) + geom_bar(width = 1, stat = "identity") +  
  coord_polar("y")
```

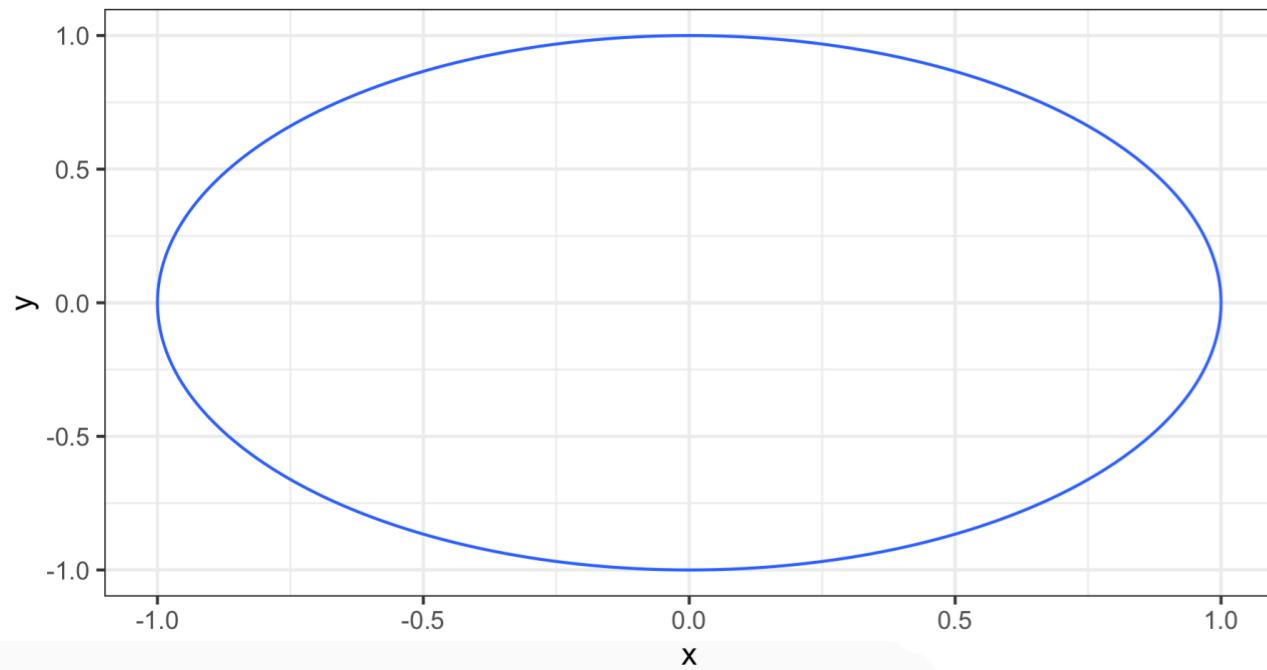


# Coordinate System

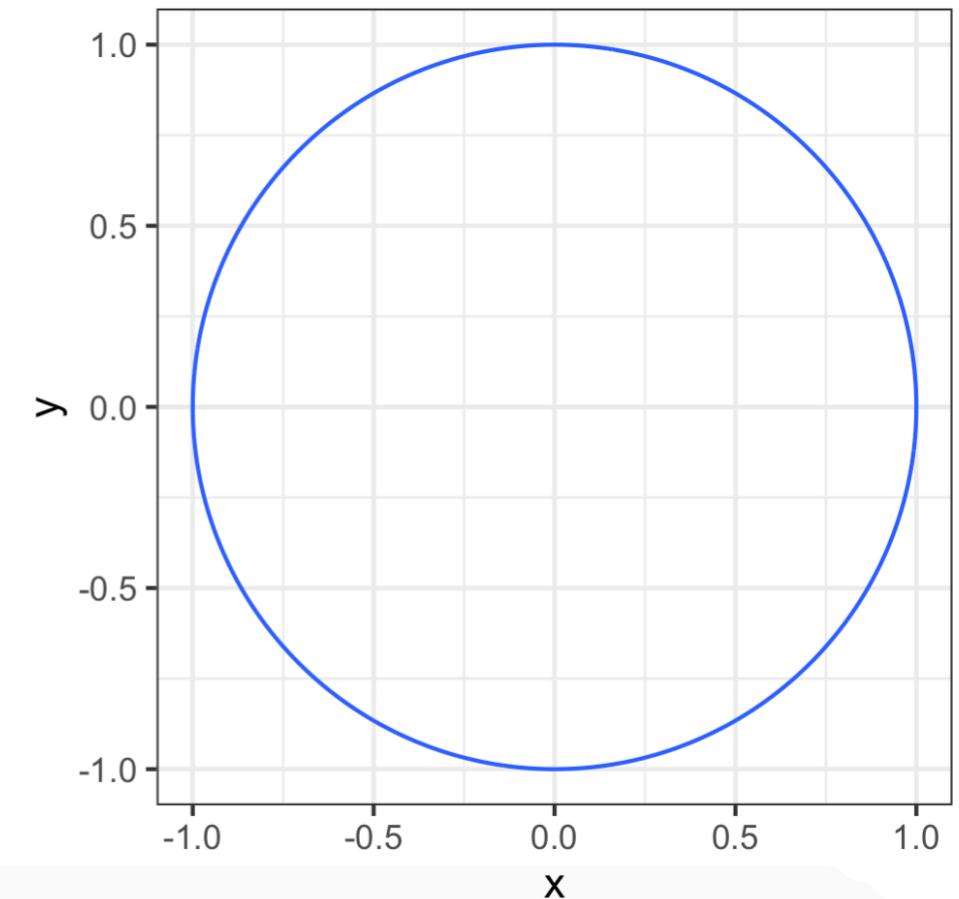
You may also want to change/fix the plot ratio:

Ex: I want to plot a circle!

```
ggvariety("x^2 + y^2 - 1")
```



```
ggvariety("x^2 + y^2 - 1")  
+ coord_equal()
```

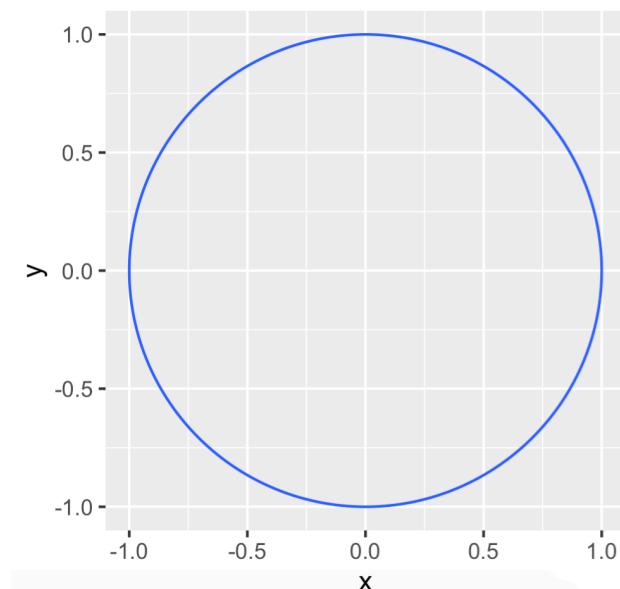


`ggvariety()` is from the `algstat` R package

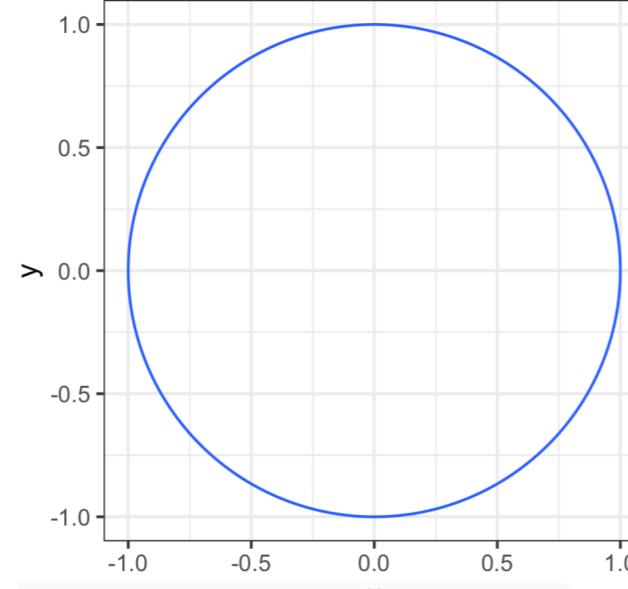
# Plot Theme

Depending on where your plot is being presented, the background theme may matter. Here's a bunch of them:

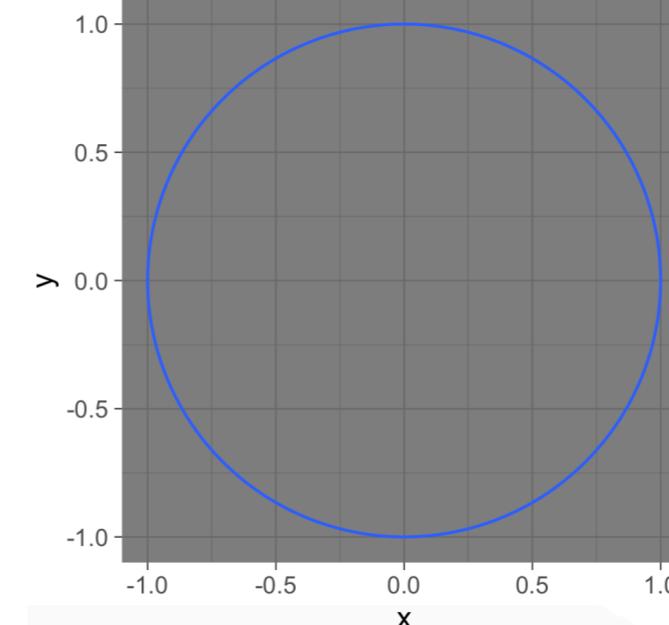
`theme_gray()`



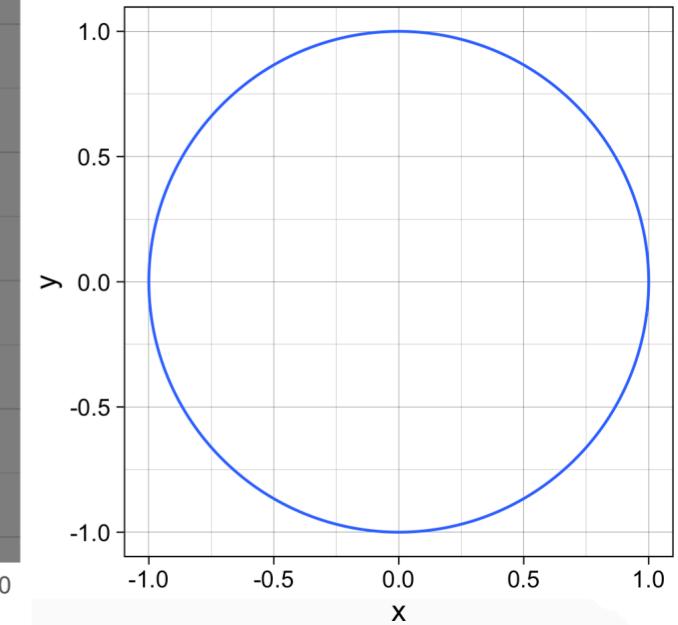
`theme_bw()`



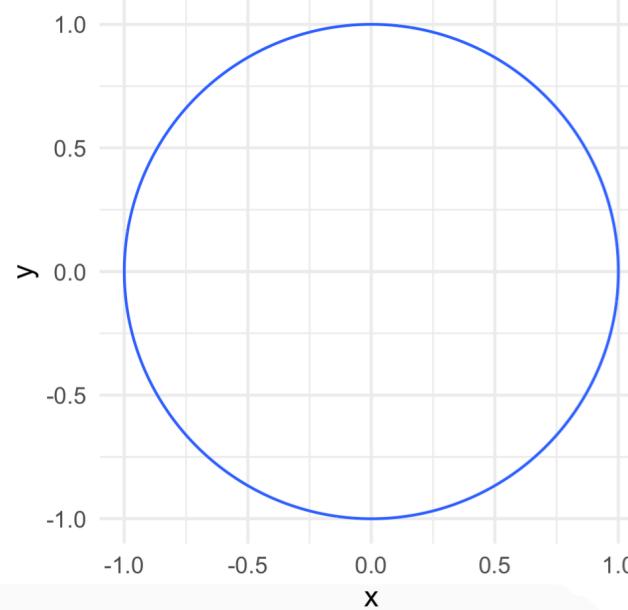
`theme_dark()`



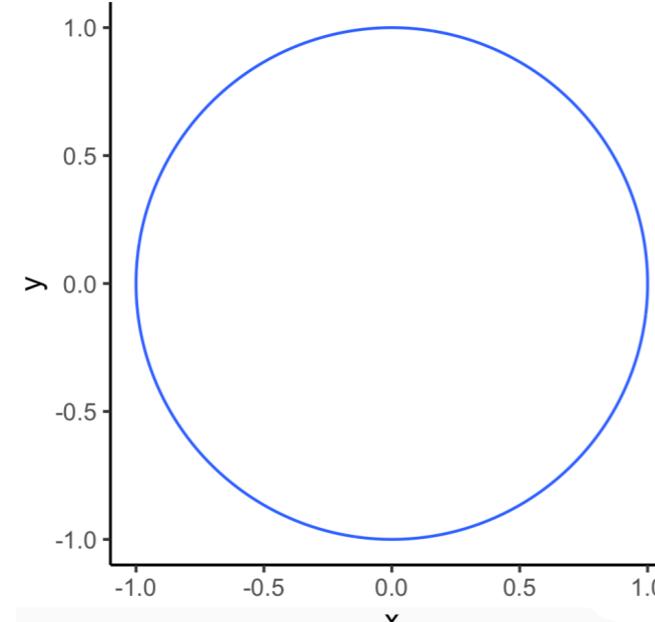
`theme_linedraw()`



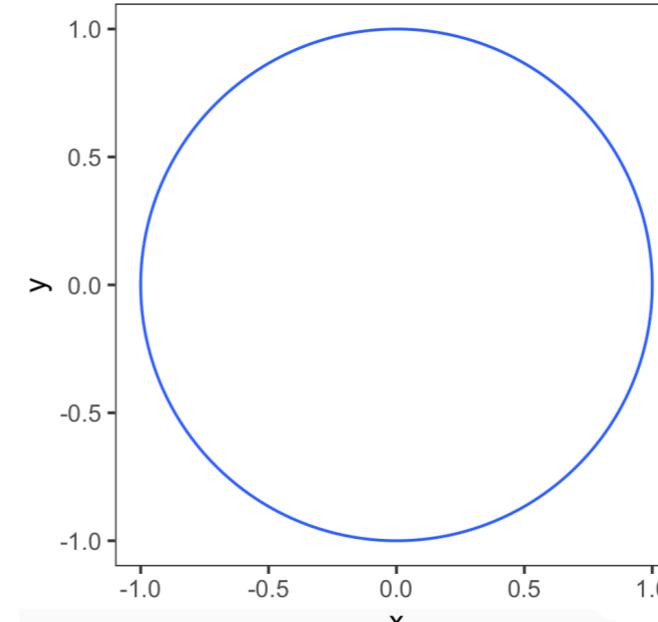
`theme_minimal()`



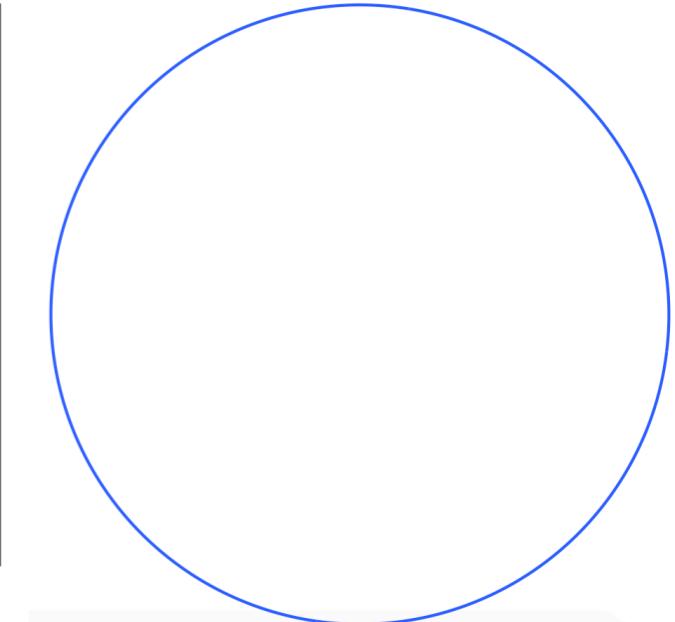
`theme_classic()`



`theme_test()`



`theme_void()`



# Extension Packages

ggplot2 is a very popular package and many packages have been built up on it (63+)!

Here is a website of a bunch of different extension packages: [ggplot2 extensions](#)

One cool one is ganimate!

