Intro to R Programming Day 3

Pittard Consulting

Steve Pittard wsp@emory.edu

September 15, 2016

ggplot2 - Intro

R has a new graphics package called **ggplot2** that is based on a well developed *grammar of graphics*. The lattice package also attempts to adhere to a standard but ggplot2 takes it to a new level. In general plots have:

- aesthetics
  - coordinate positions (x,y)
  - element size, shape, and color
- geometry
  - lines, points
  - segments, bars
  - text
  - element size, shape, and color

ggplot2 - Terms

The terminology can be a little confusing:

- ggplot - The first version of the ggplot package
- ggplot2 - an updated and the most recent version of the ggplot package
- ggplot - is also an actual function in the ggplot2 package that allows you to build plots
- qplot - a simplified function to ease your transition into the ggplot2 package
- Many times I will just say or write "ggplot" as a synonym to ggplot2

ggplot2 - Terms

- Like Latticce graphics ggplot can support grouping and distinction of data within a single of plot
- ggplot can also support conditioning/panelling (though in ggplot it is called “faceting”)
- In practice and philosophy ggplot is closer to lattice graphics than it is to Base Graphics
- This does not mean that the Base Graphics System is bad - just that it lacks a unifying philosophy. It is very powerful for creating graphics programmatically.
- If you have the luxury of picking one then start with ggplot

ggplot2 - Learning

The idea is to first think of a plot in terms of these ideas after which you use specific ggplot2 commands to turn these ideas into an actual plot.

ggplot2 provides two points of entry into the package.

- **qplot** - a simplified version of more involved ggplot commands. Sort of like training wheels for becoming accustomed to ggplot. It is meant to mimic the Base graphics **plot** command though **qplot** offers more generality.

qplot is the basic plotting function in the ggplot2 package. It is a convenient wrapper for creating a number of different types of plots using a consistent calling scheme.

ggplot2 - Learning

The idea is to first think of a plot in terms of these ideas after which you use specific ggplot2 commands to turn these ideas into an actual plot.

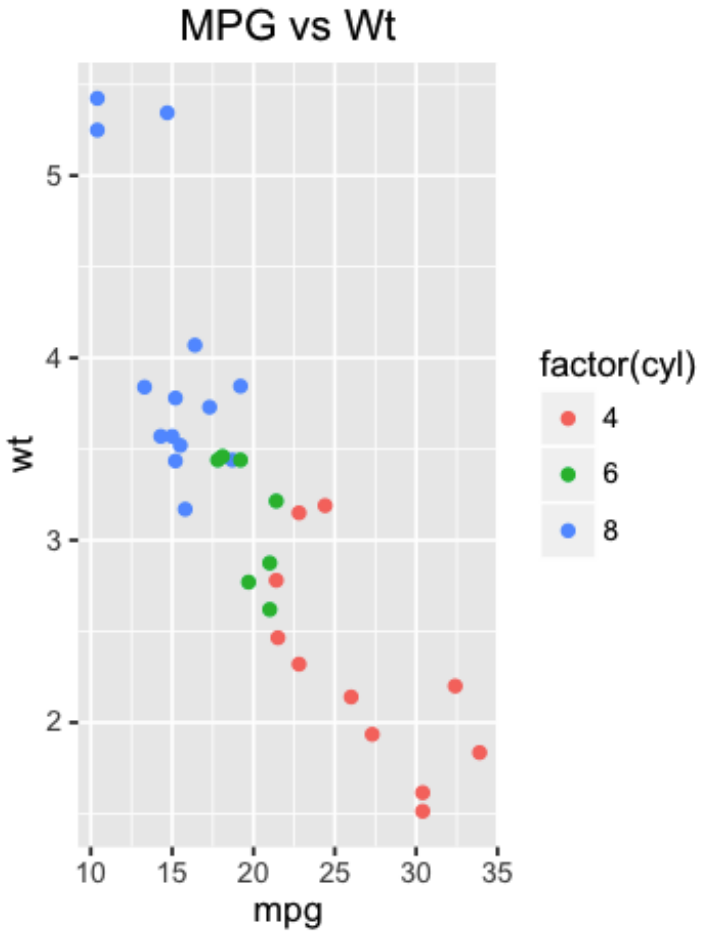
ggplot2 provides two points of entry into the package.

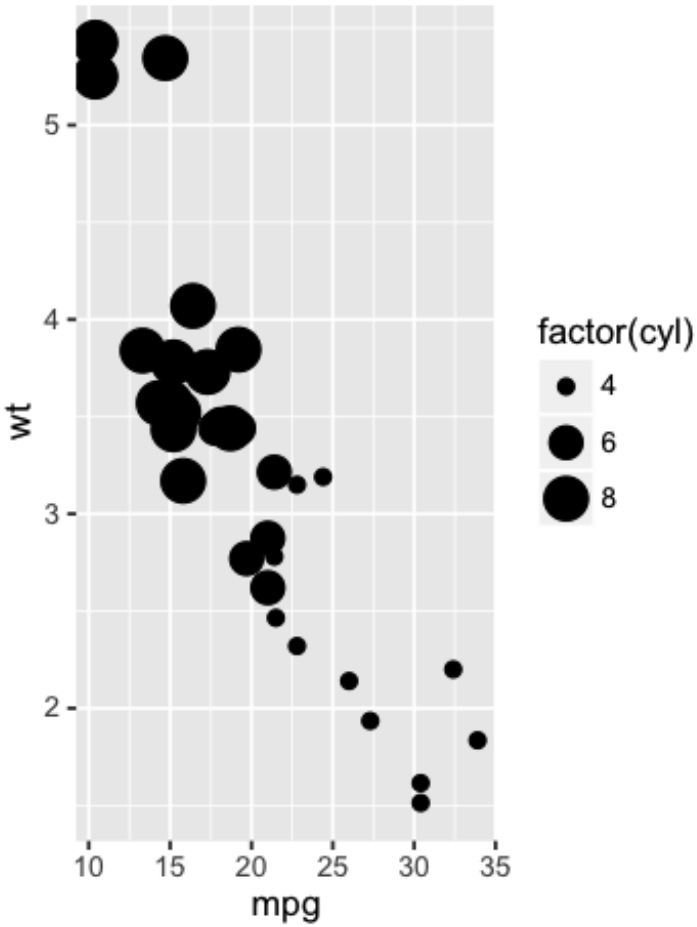
- **ggplot** - which is the more complex yet far more flexible command for creating plots. I favor this approach since it allows one to assemble plots in layers. Most literature describing analysis and visualization using ggplot will use the more general approach.

The **ggplot** command can be used to declare the input data frame for a graphic and to specify the set of plot aesthetics intended to be common throughout all subsequent layers unless specifically overridden

ggplot2 - qplot

```
qplot(mpg, wt, data=mtcars, main="MPG vs Wt",color=factor(cyl))
qplot(mpg, wt, data = mtcars, size = factor(cyl))
```

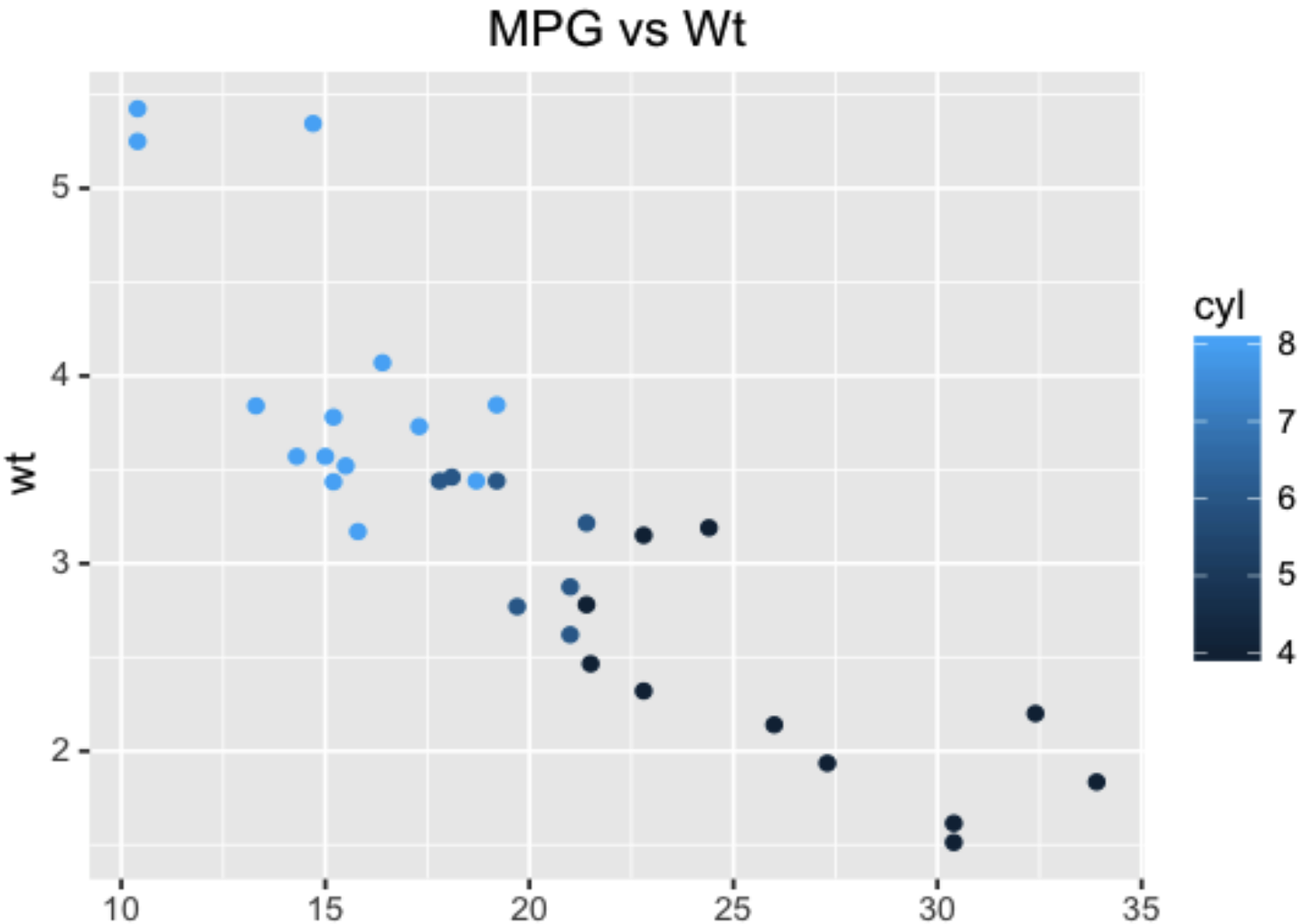




ggplot2 - qplot

You should explicitly make factors out of variables that you want to use as factors. Base and Lattice graphics are more forgiving about this than ggplot.

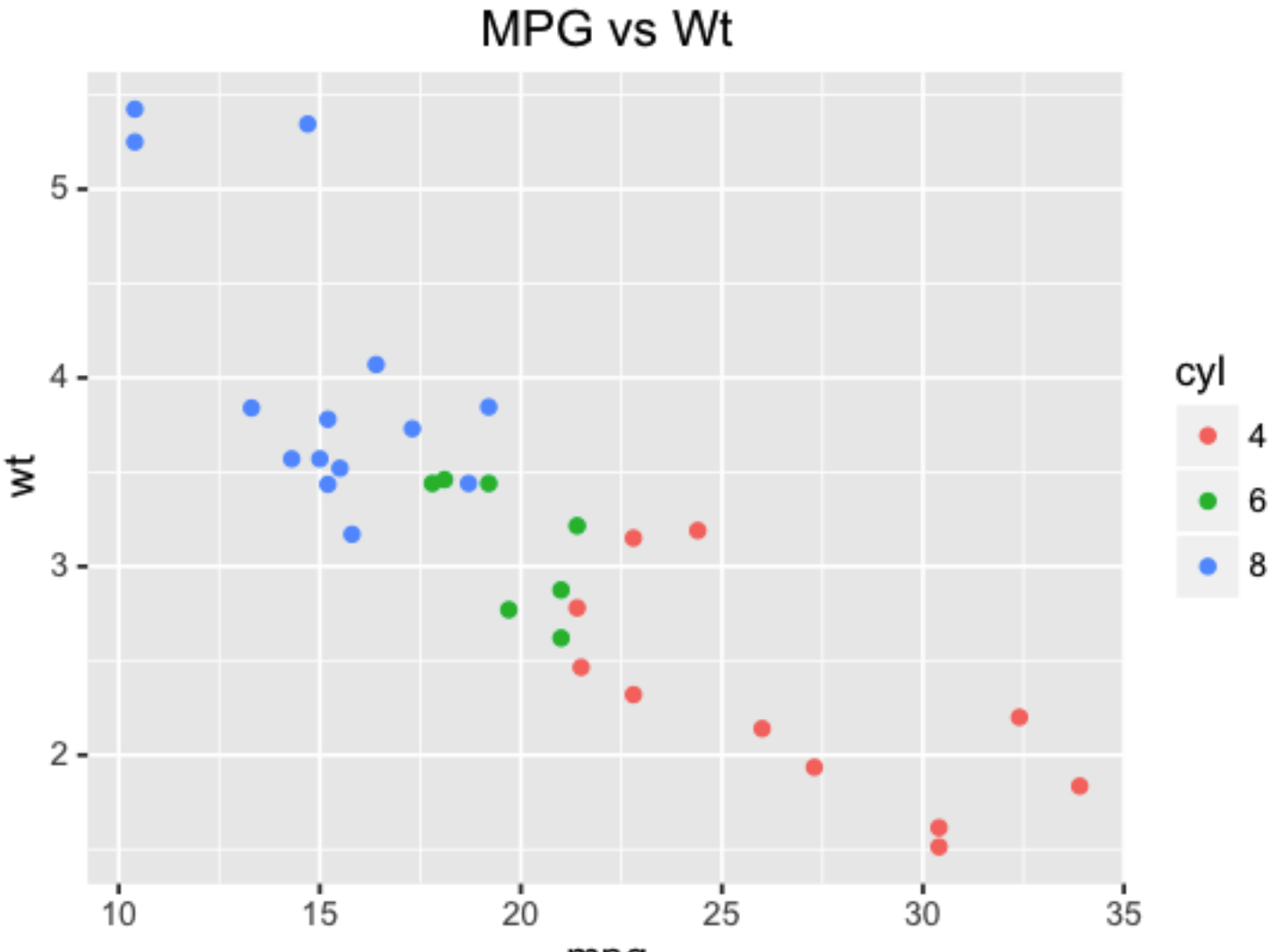
```
qplot(mpg, wt, data=mtcars, main="MPG vs Wt",color=cyl)
```



ggplot2 - qplot

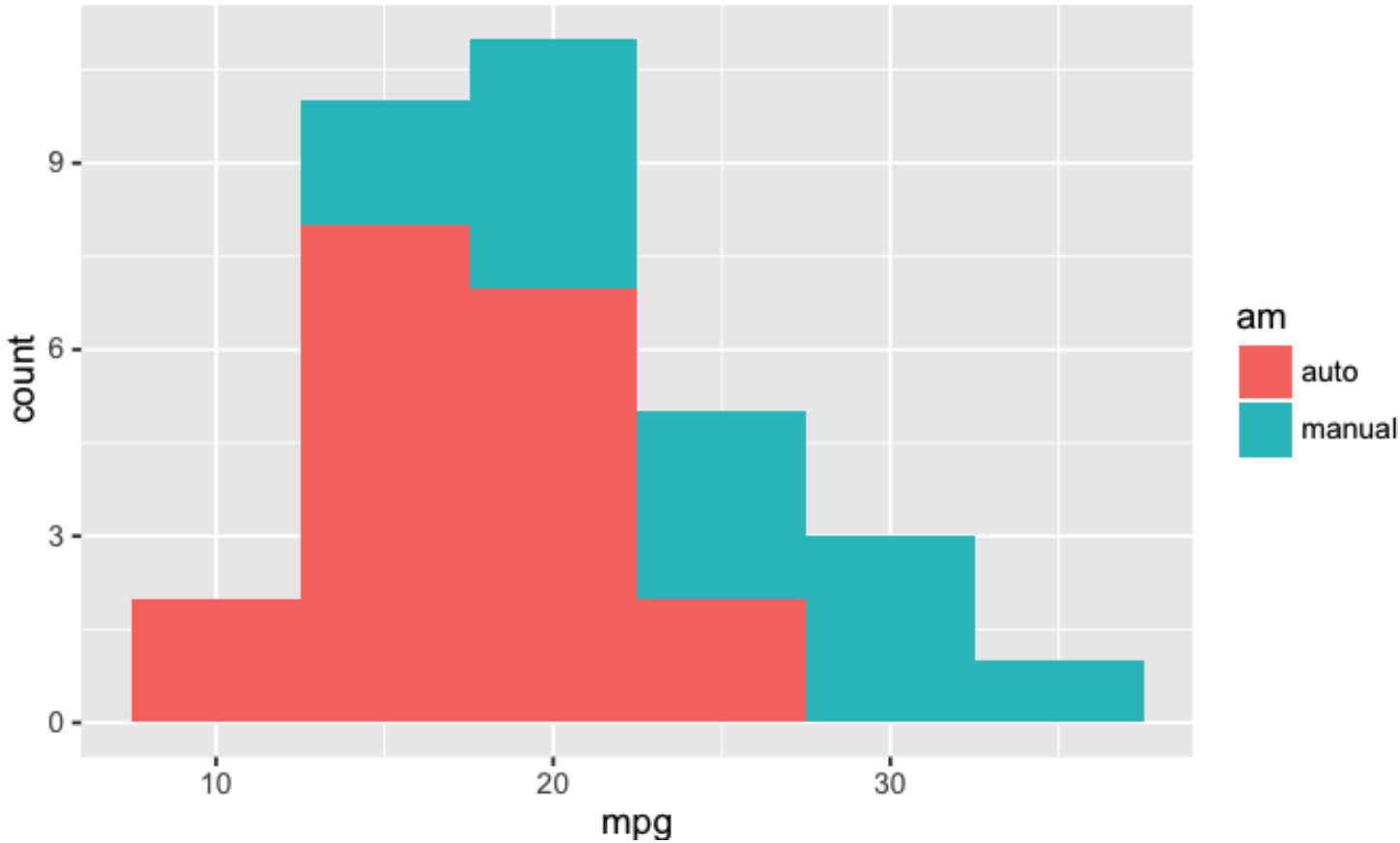
Here we make the cylinder variable into a factor

```
mtcars$cyl <- factor(mtcars$cyl)
qplot(mpg, wt, data=mtcars, main="MPG vs Wt",color=cyl)
```



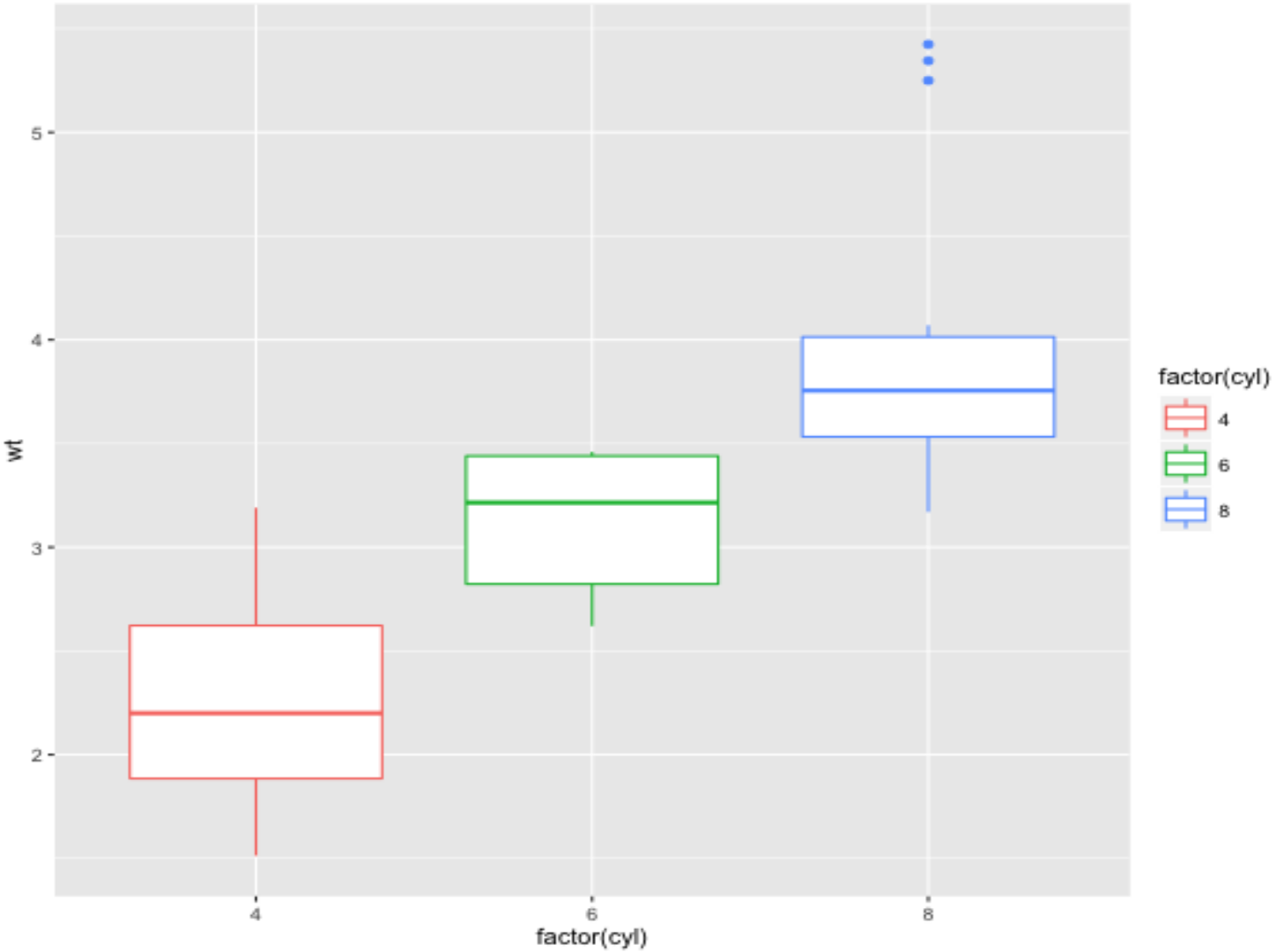
ggplot2 - qplot

```
mtcars$am <- factor(mtcars$am,labels=c("auto","manual"))
qplot(mpg, data=mtcars, geom="histogram",binwidth=5,fill=am)
```



ggplot2 - qplot

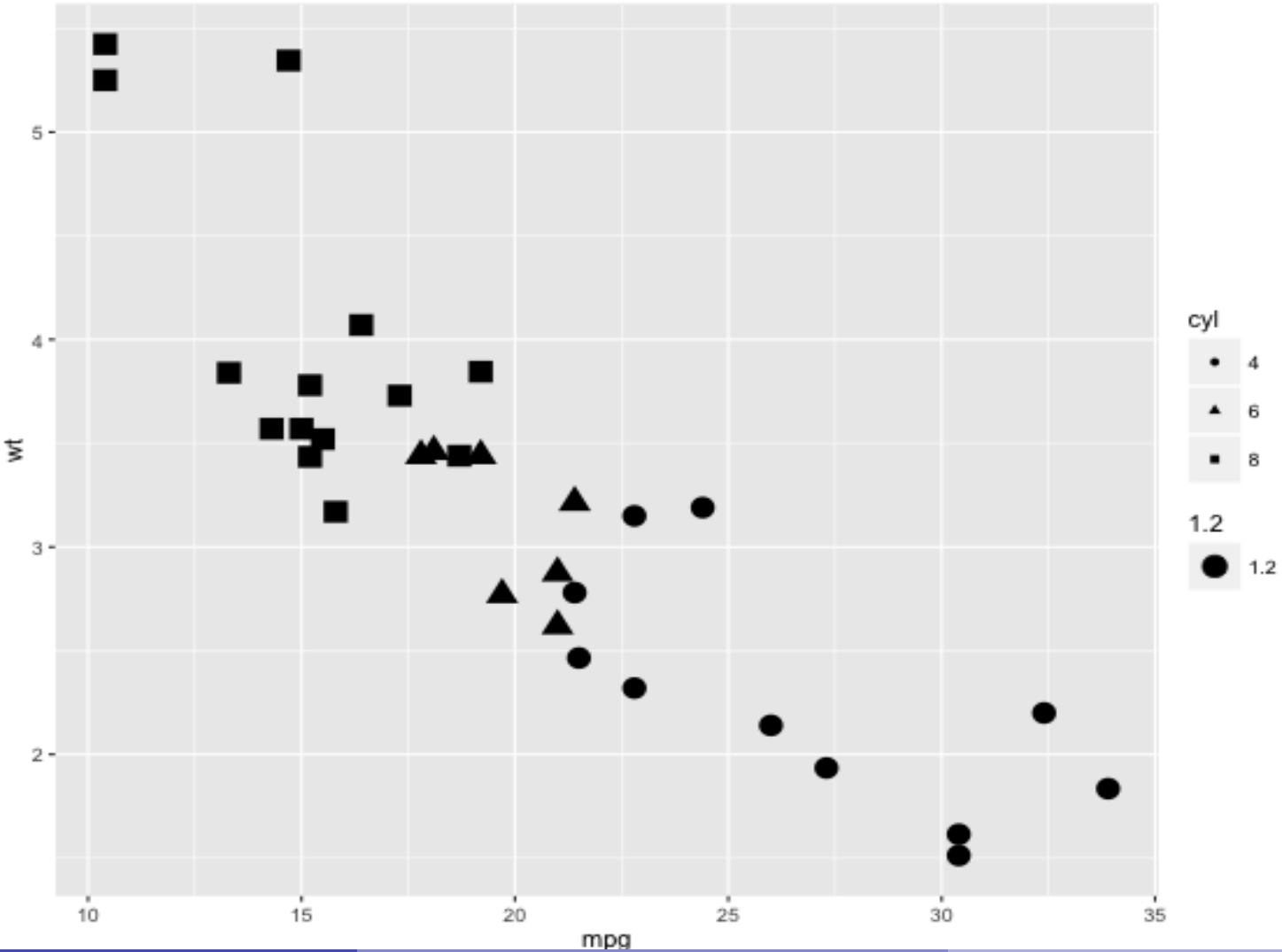
```
mtcars$cyl <- factor(mtcars$cyl)
qplot(factor(cyl),wt,data=mtcars,geom="boxplot",color=factor(cyl))
```



ggplot2 - qplot

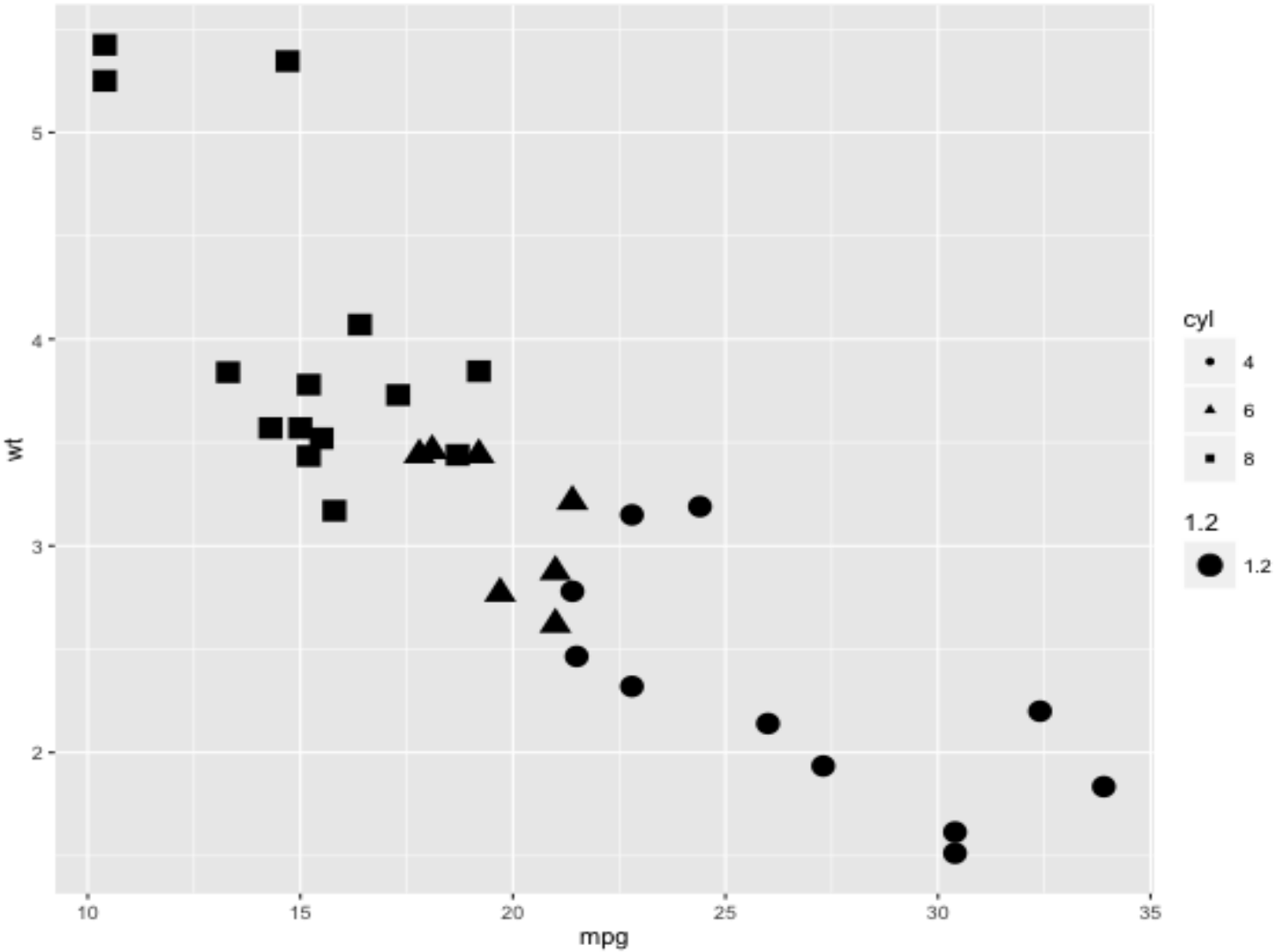
So each geometry itself can have aesthetics.

```
mtcars$cyl <- factor(mtcars$cyl)
qplot(mpg,wt, data=mtcars, geom="point", shape=cyl,size=1.2)
```



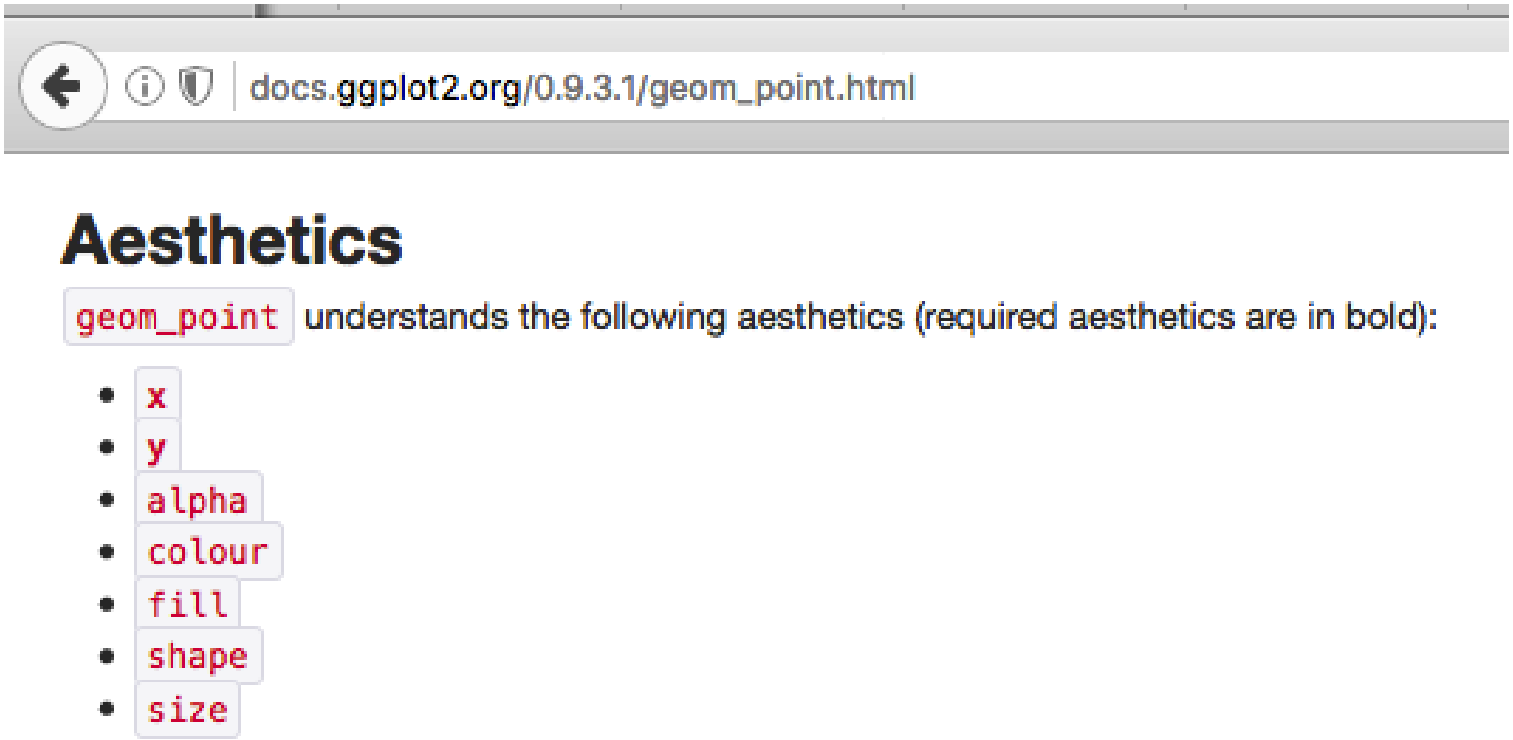
ggplot2 - qplot

```
mtcars$cyl <- factor(mtcars$cyl)
qplot(mpg,wt, data=mtcars, geom="point", shape=cyl,size=1.2)
```



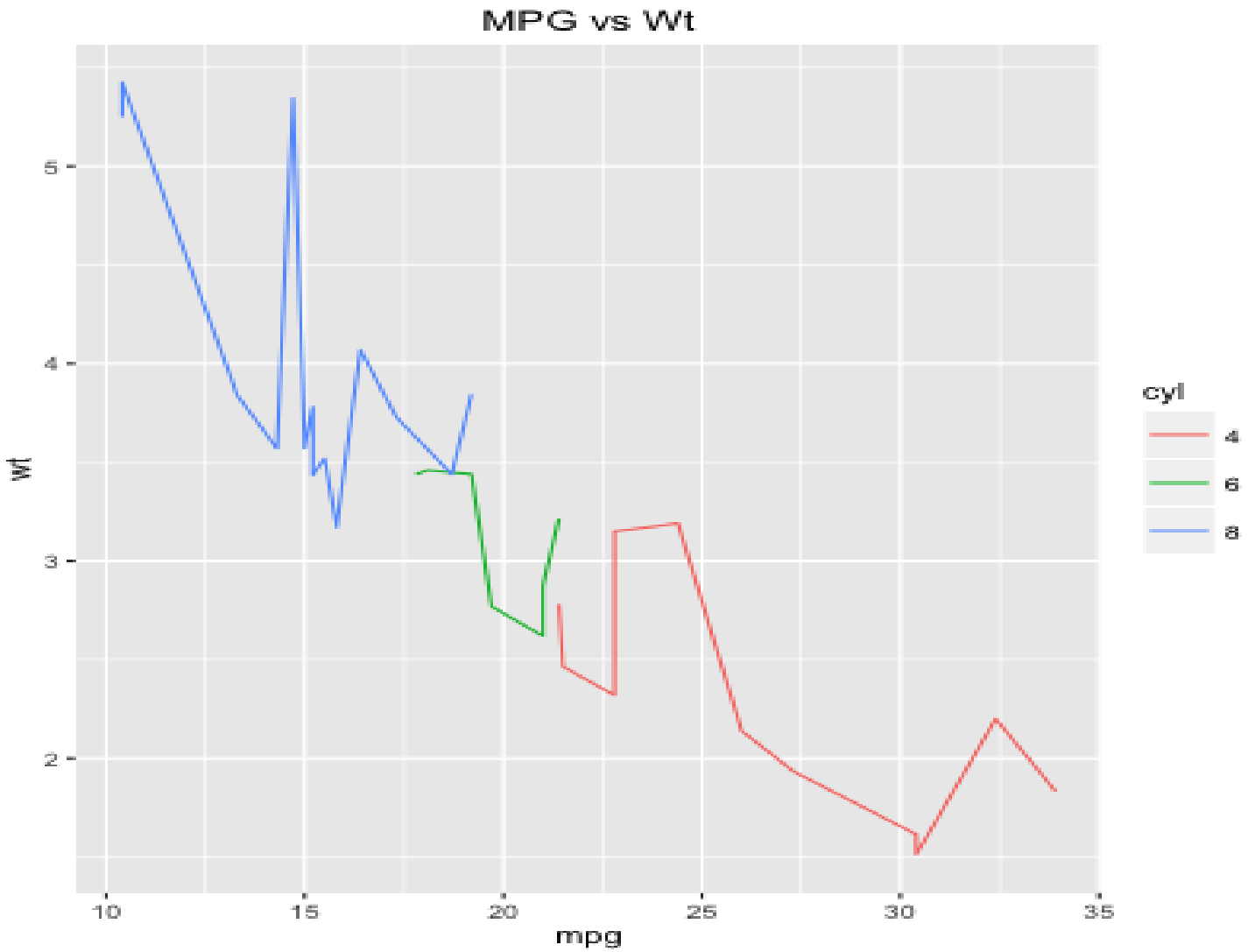
ggplot2 - qplot

Each geometry can have aesthetics. Check out the `docs.ggplot2.org/` for information on what aesthetics are associated with a given geometry.



ggplot2 - qplot

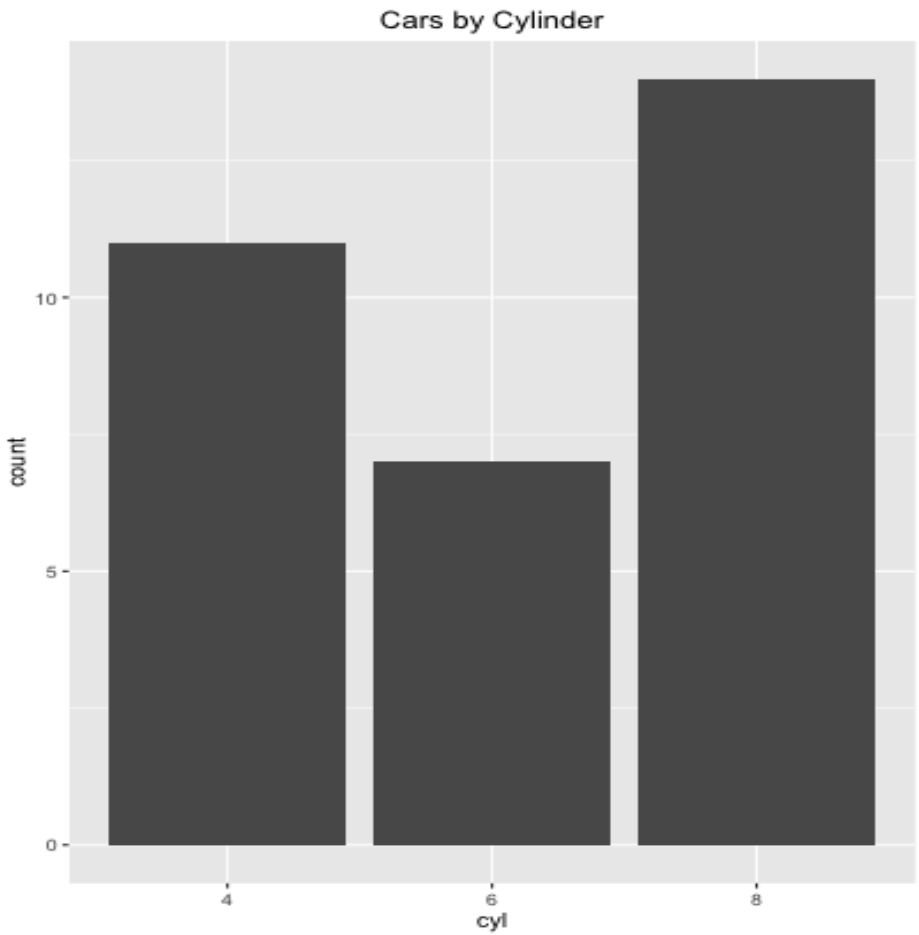
```
qplot(mpg,wt, data=mtcars,geom="line",main="MPG vs Wt",color=cyl)
```



ggplot2 - qplot

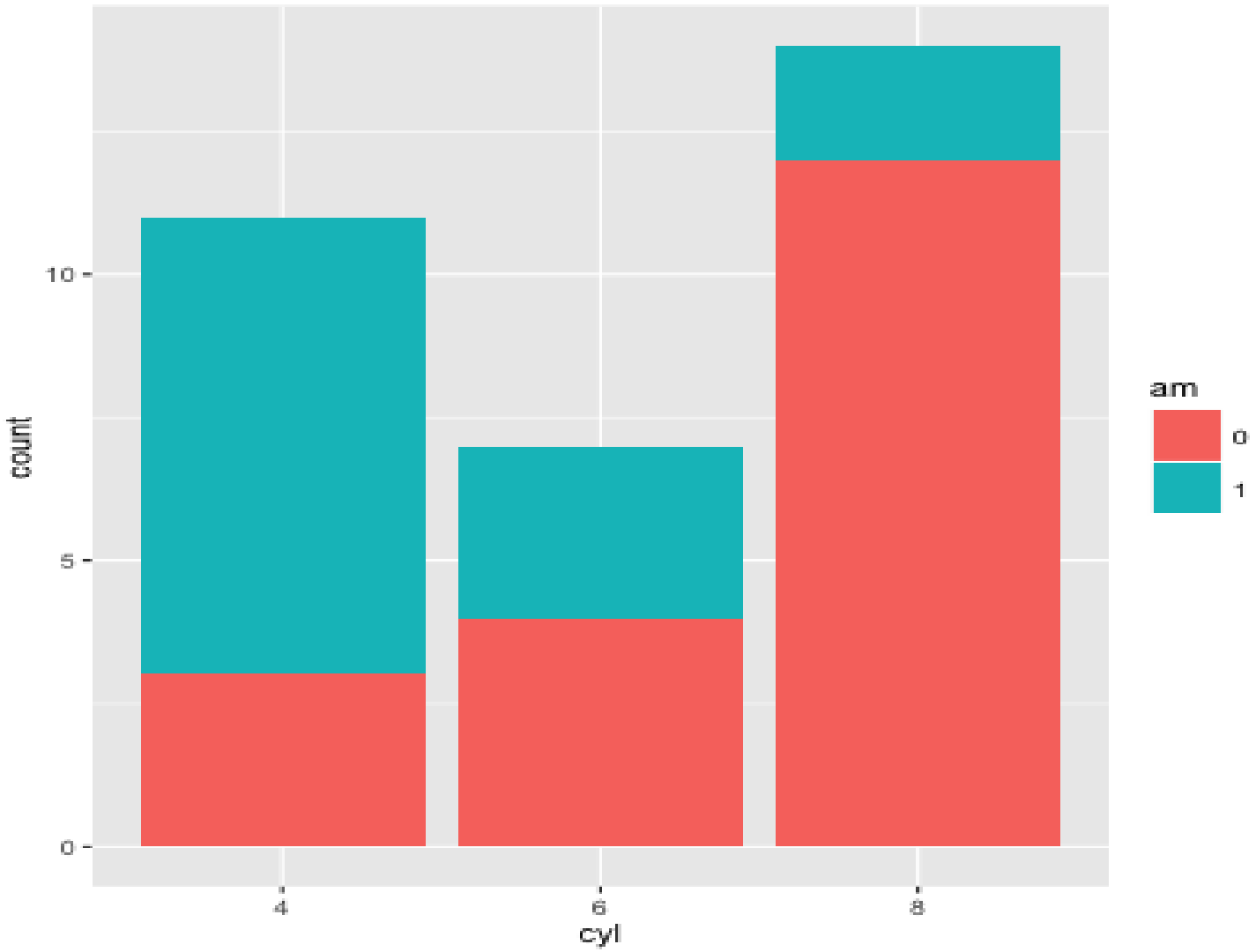
The cool thing about the **bar** geom is that it will tabulate the number of observations in each category for you - unlike Base Graphics

```
mtcars$cyl <- factor(mtcars$cyl)
qplot(cyl, data=mtcars, geom="bar",main="Cars by Cylinder")
```



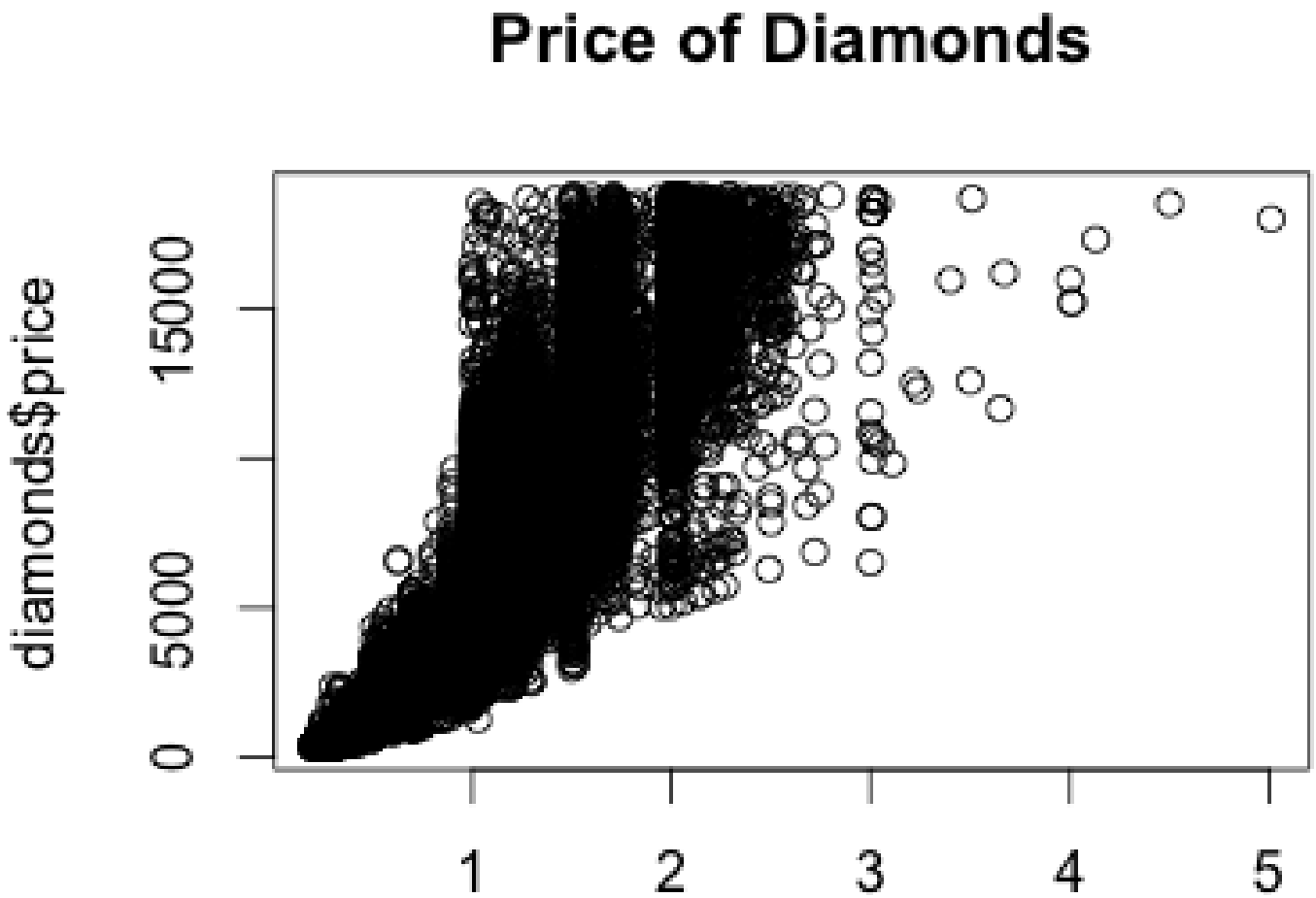
ggplot2 - qplot

```
mtcars$am <- factor(mtcars$am)
qplot(cyl, data=mtcars, fill=am, geom="bar")
```



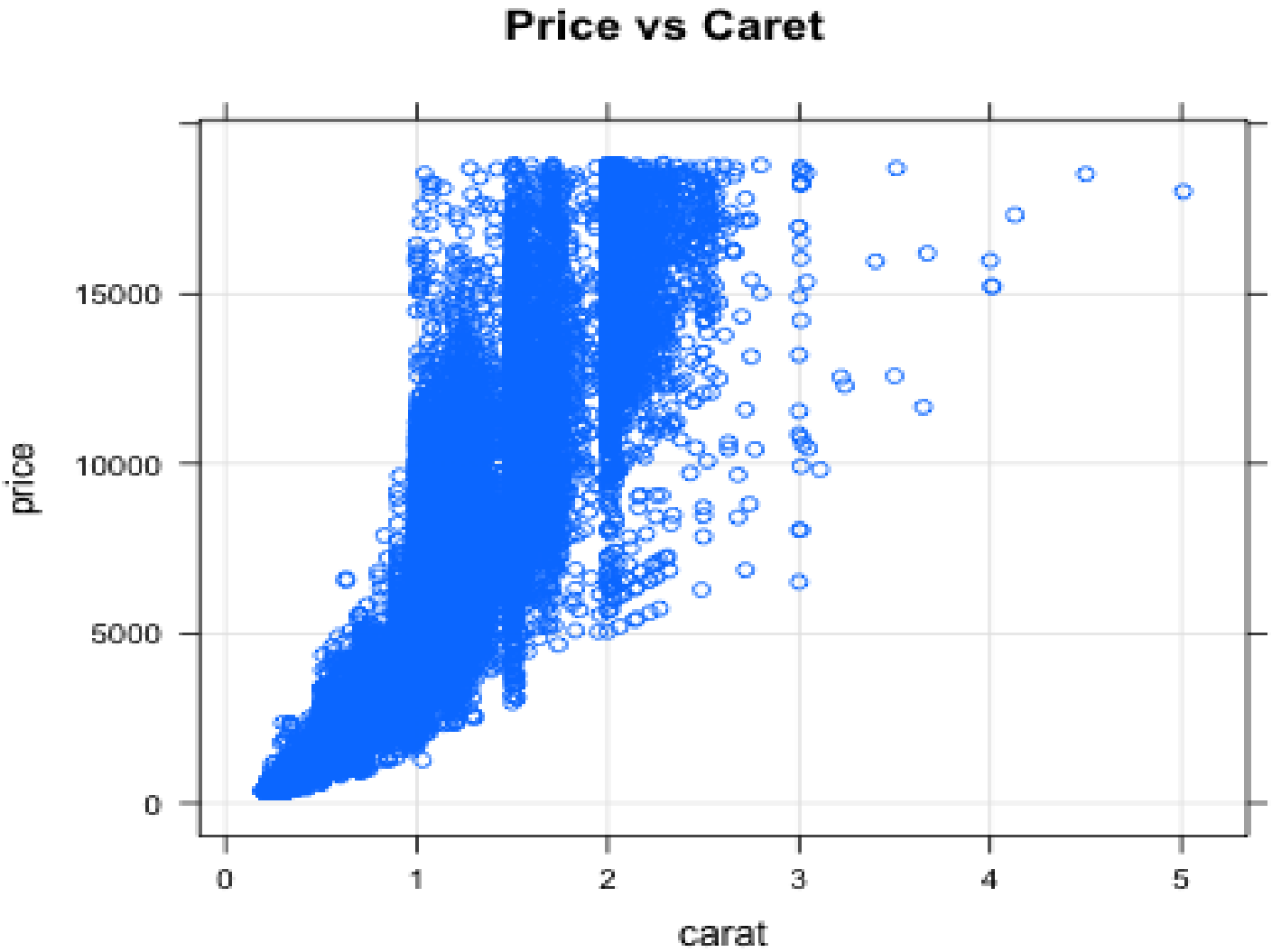
ggplot2 - alpha

```
# Let's plot this with Base graphics
data(diamonds)  # A dataset on diamonds
plot(diamonds$carat,diamonds$price,main="Price of Diamonds")
grid()
```



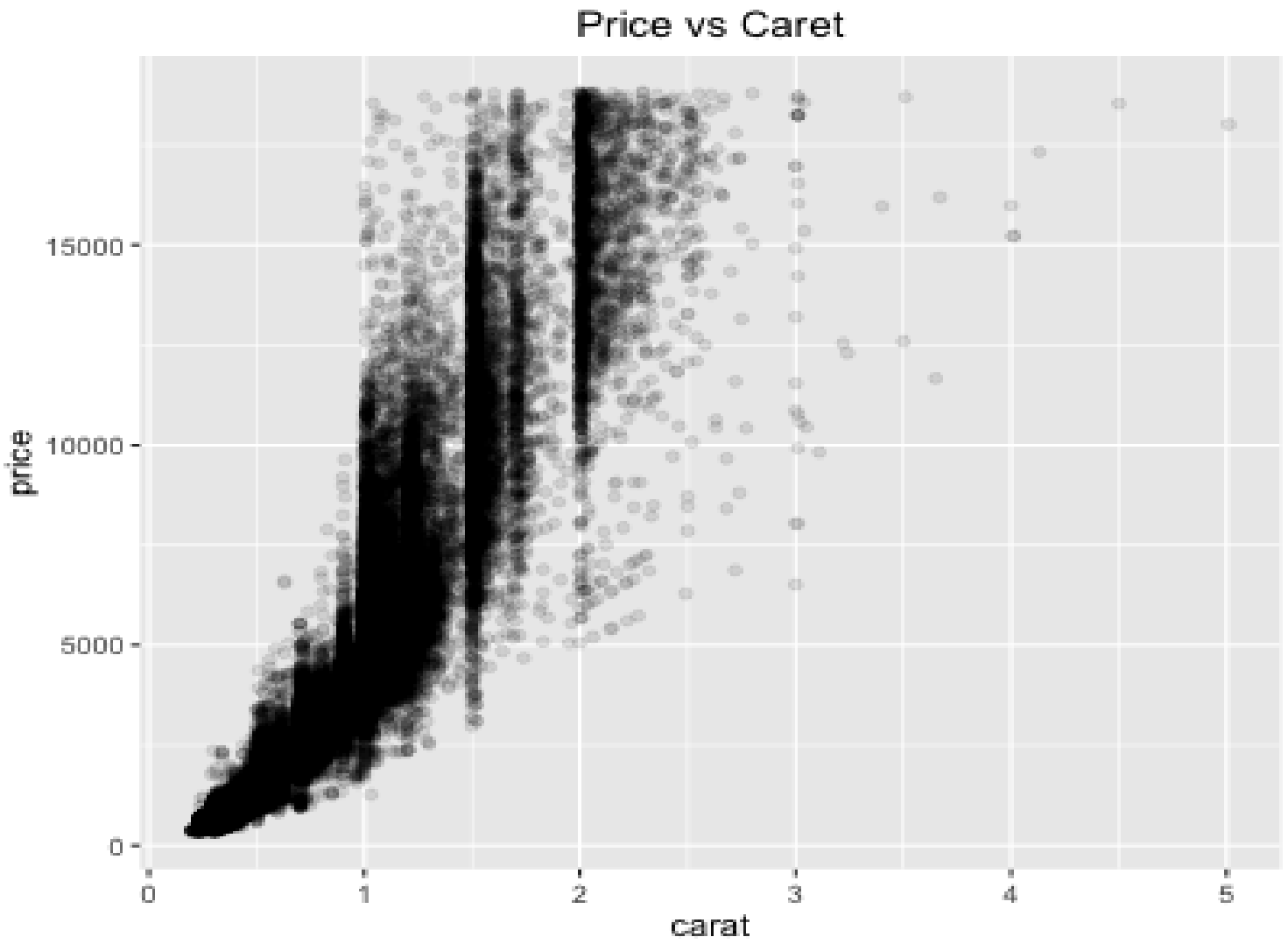
ggplot2 - alpha

```
# Let's plot this with lattice graphics
data(diamonds)  # A dataset on diamonds
xyplot(price~carat,data=diamonds,main="Price vs Caret",type=c("p","g"))
```



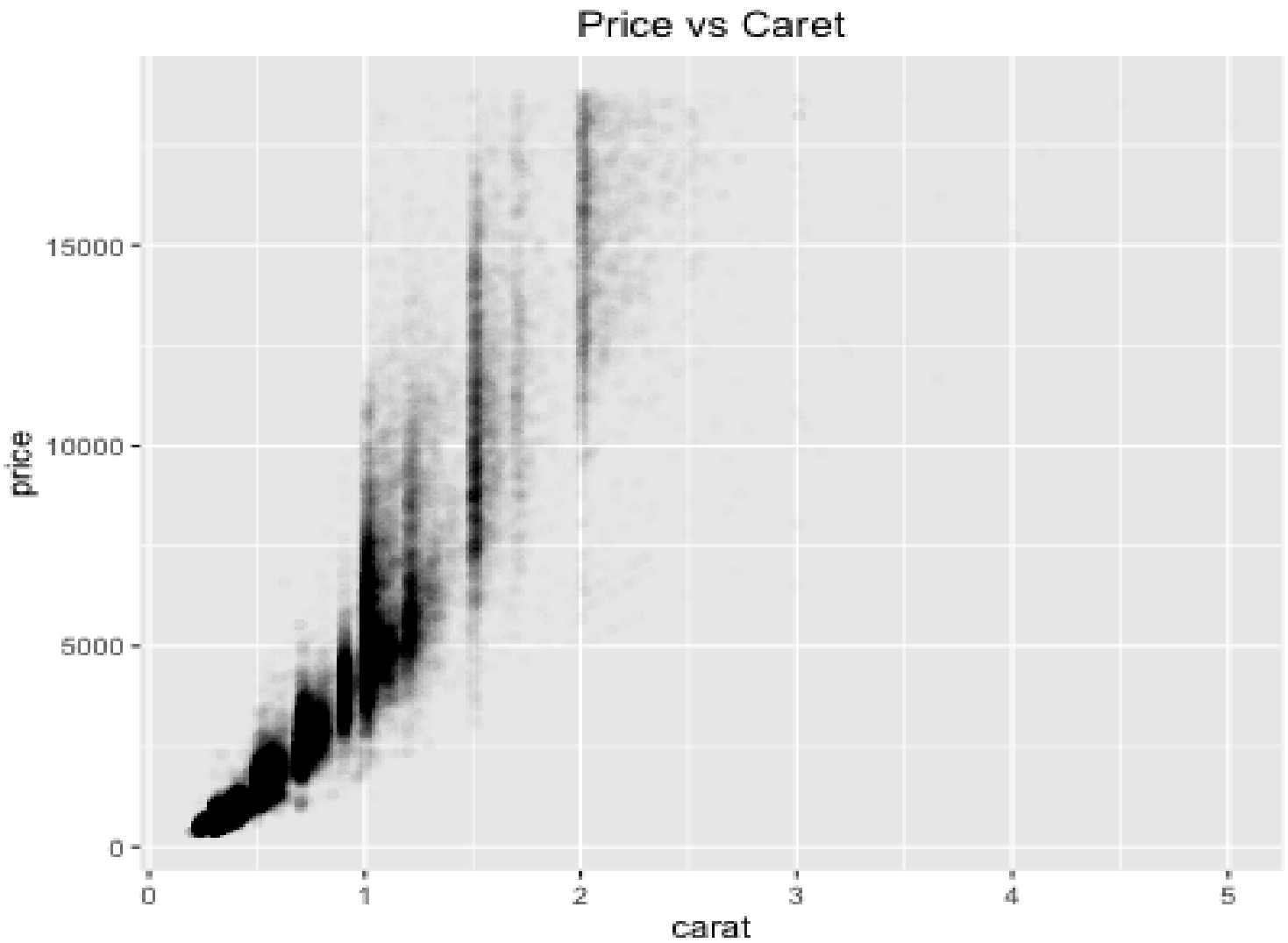
ggplot2 - alpha

```
data(diamonds)  # A dataset on diamonds
title <- "Price vs Caret"
qplot(carat, price, data = diamonds, alpha = I(1/10),main=title)
```



ggplot2 - alpha

```
data(diamonds)  # A dataset on diamonds
title <- "Price vs Caret"
qplot(carat, price, data = diamonds, alpha = I(1/100),main=title)
```



ggplot2 - qplot

```
qplot(displ, hwy, data=mpg, facets = . ~ year) + geom_smooth()
```