

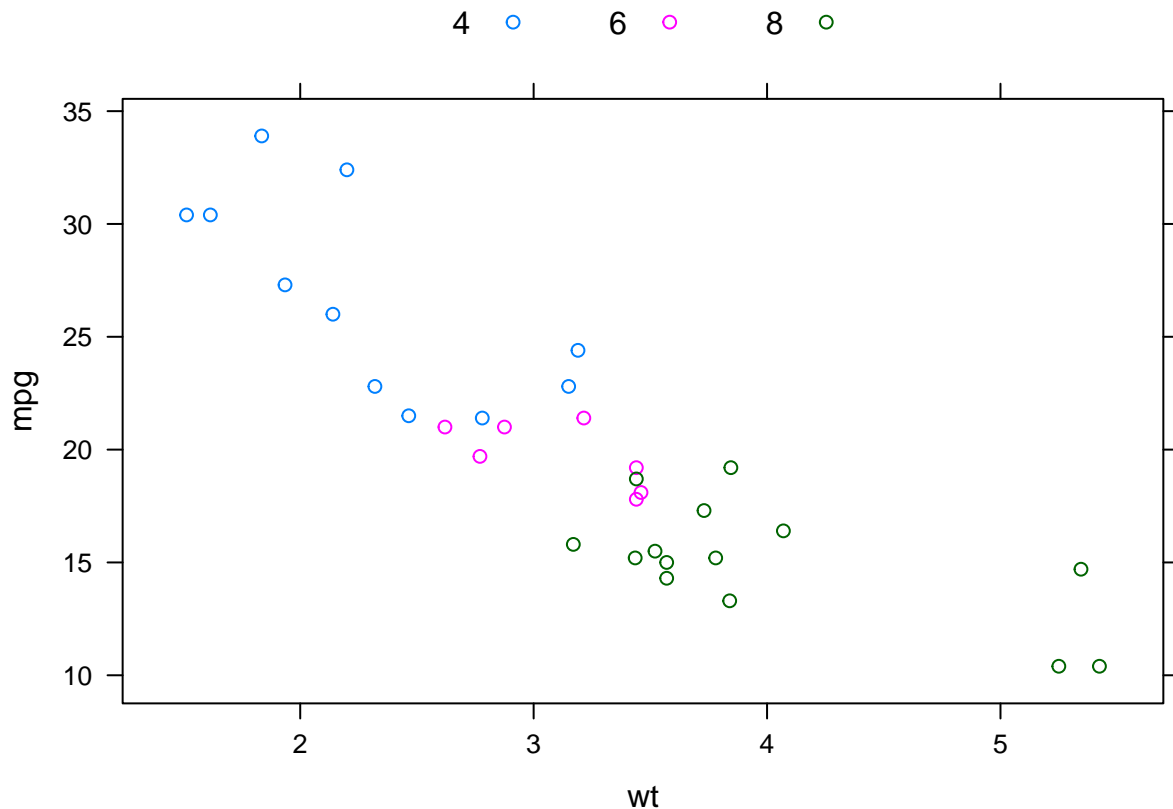
lattice legends

Steve Pittard

February 11, 2016

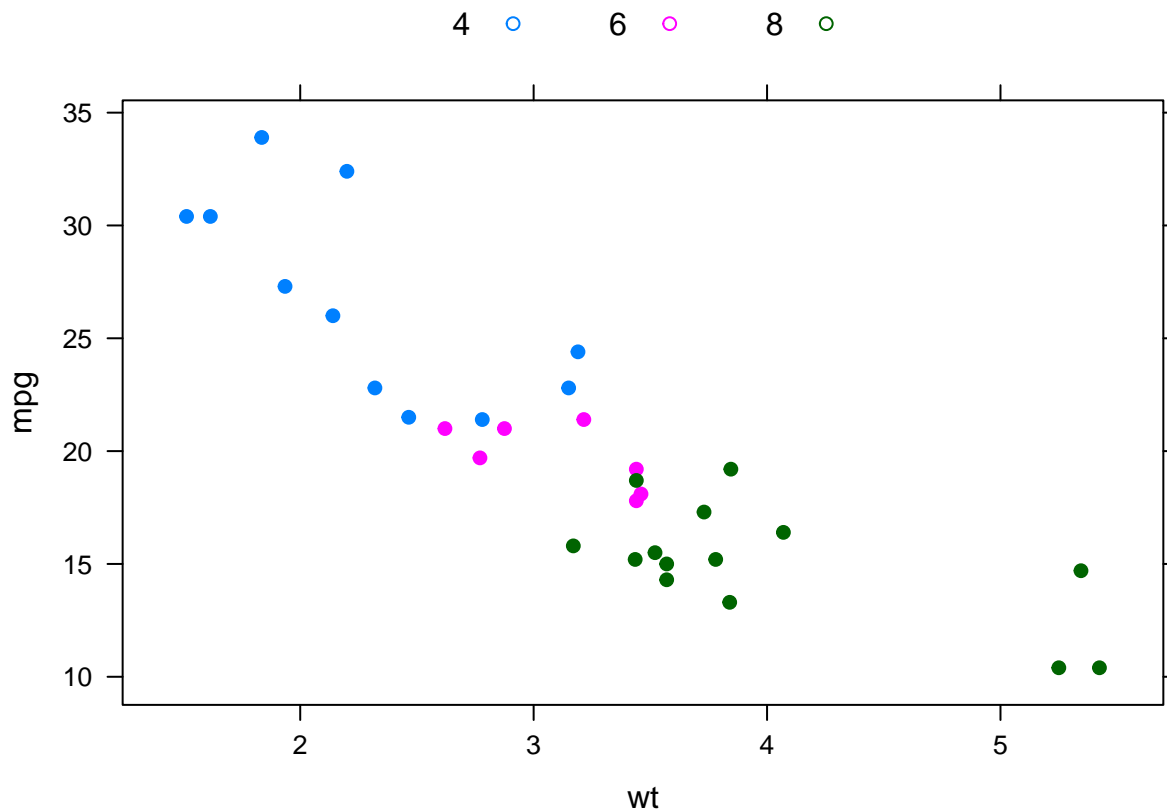
In class on Wed I told you about a little problem with lattice graphics and how it creates an automatic legend that can have different plot characters than what you had specified in the graph. As an example let's begin with the following:

```
library(lattice)
xyplot(mpg~wt,data=mtcars,
       group=cyl,
       auto.key=list(columns=3))
```



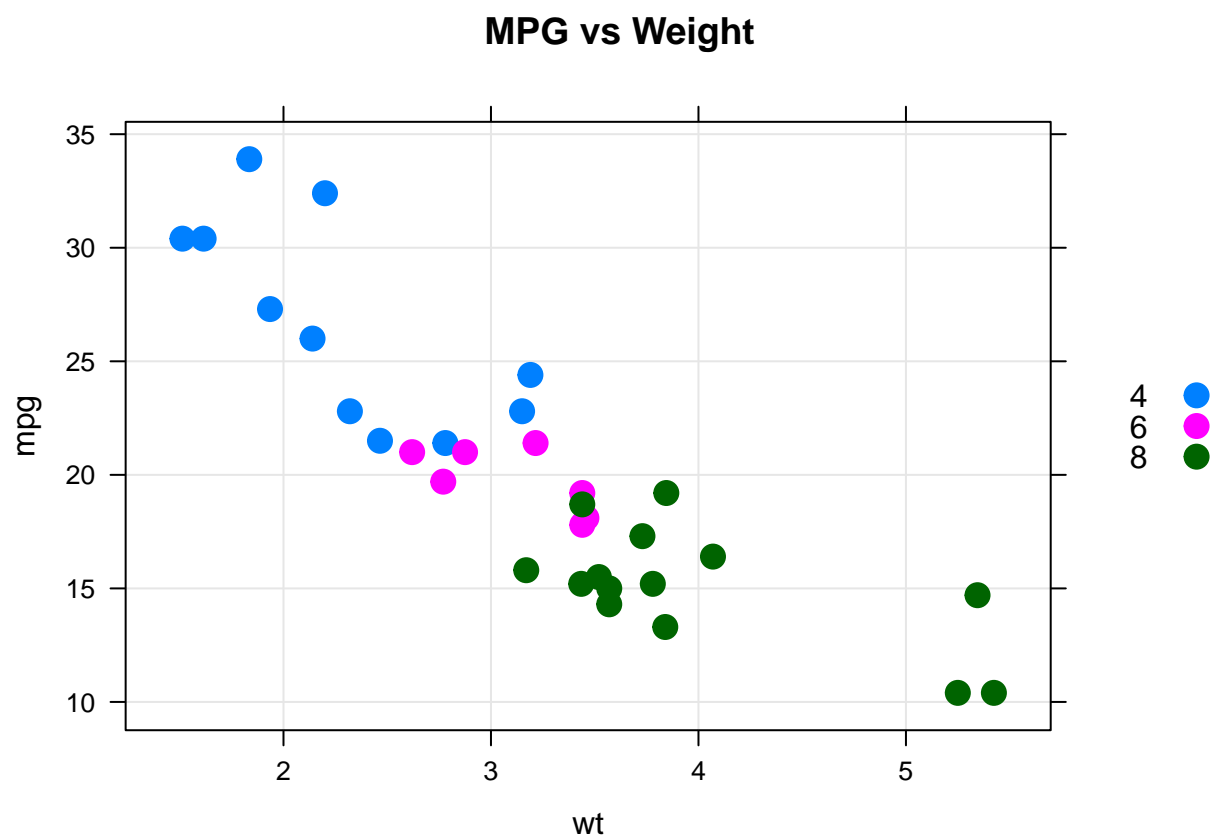
Okay next we'll pick a different plot character such as 19 which is a filled point. However, the resulting legend doesn't have the filled points.

```
library(lattice)
xyplot(mpg~wt,data=mtcars,
       group=cyl,
       auto.key=list(columns=3),
       pch=19)
```

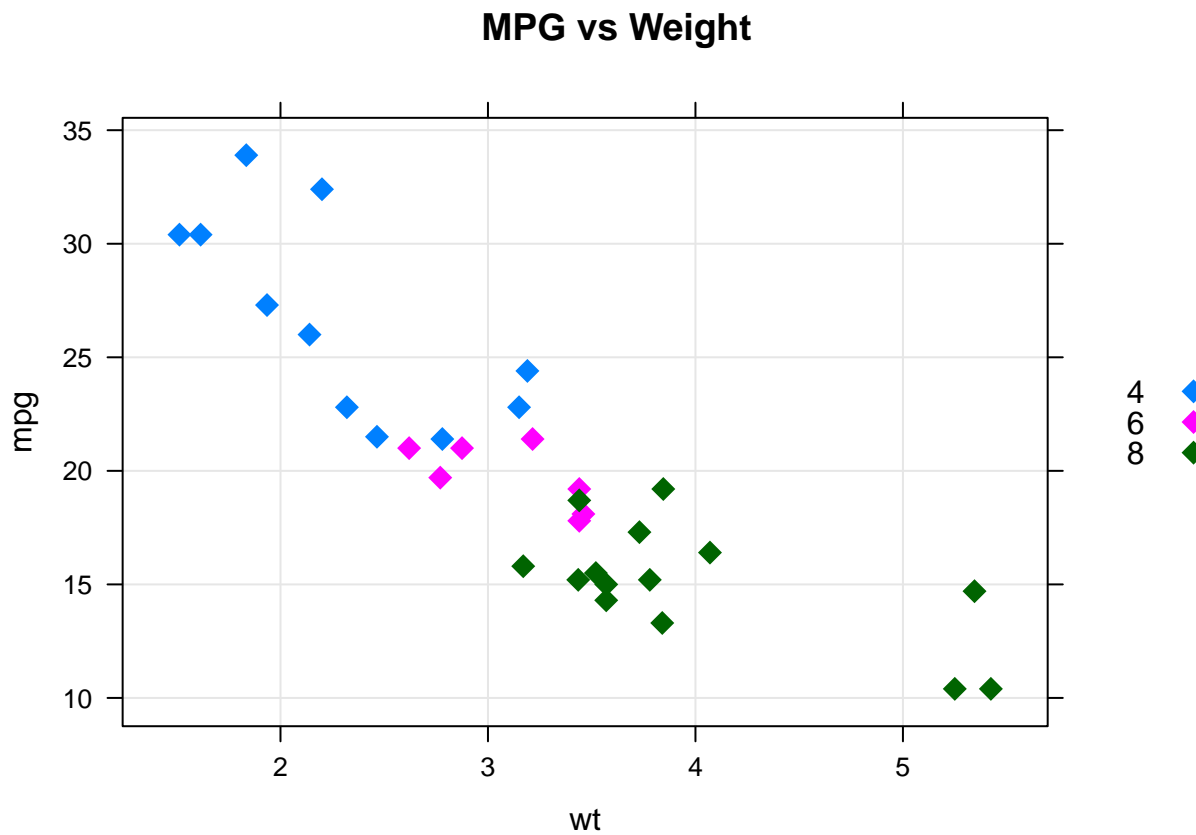


In class I showed you a couple of workarounds but I thought I would show you another that might be easier although it is more verbose - that is there is more to type. In this case we include the **par.settings** argument which takes a list as an argument. Within that list we can specify any number of parameters that impact the look of the legend such as the plot character (pch), size (cex), and the auto.key

```
library(lattice)
xyplot(mpg~wt,data=mtcars,
       group=cyl,
       type=c("p","g"),
       main="MPG vs Weight",
       par.settings=list(superpose.symbol=list(pch=c(19), cex=1.5)),
                          auto.key=list(space="right"))
```



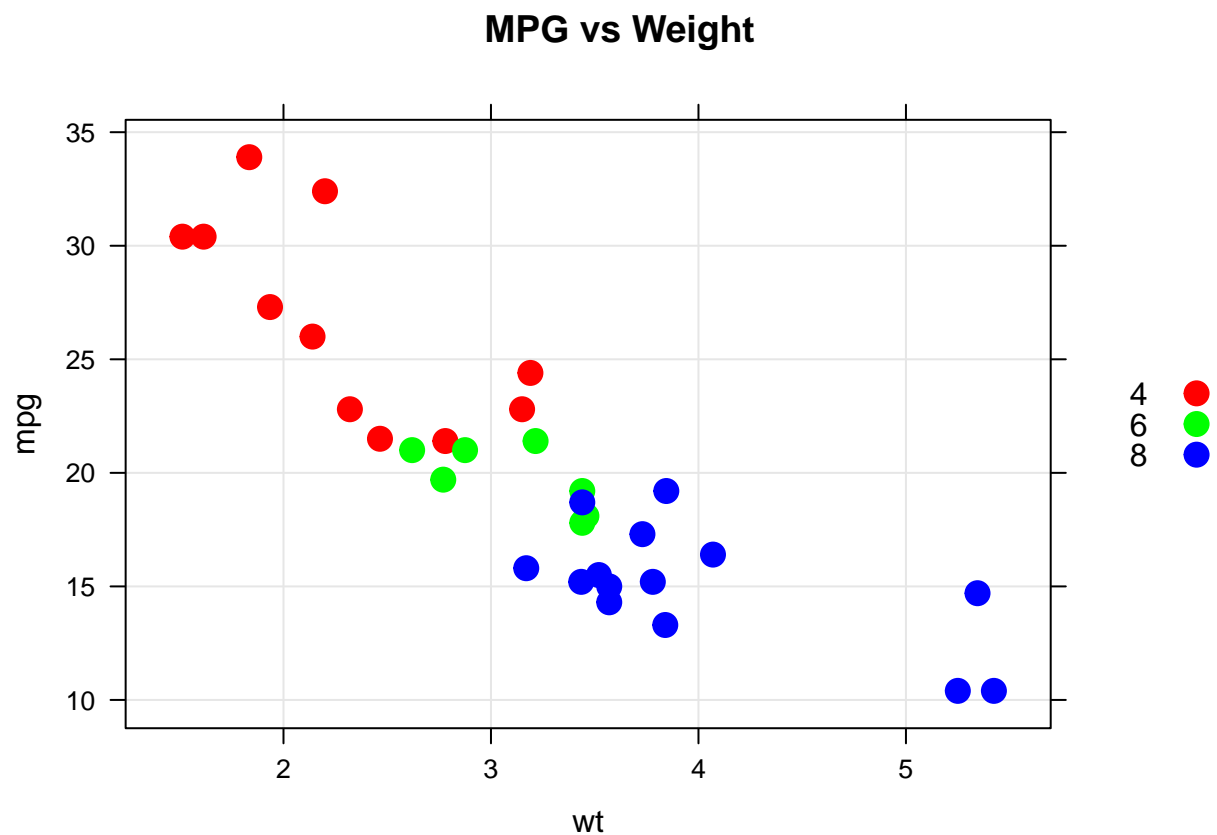
```
library(lattice)
xyplot(mpg~wt,data=mtcars,
       group=cyl,
       type=c("p","g"),
       main="MPG vs Weight",
       par.settings=list(superpose.symbol=list(pch=c(18), cex=1.5)),
       auto.key=list(space="right"))
```



You could even pass in some colors if you wanted something other than the default colors provided by lattice.

```
library(lattice)
colors <- rainbow(3)

xyplot(mpg~wt,data=mtcars,
       group=cyl,
       type=c("p","g"),
       main="MPG vs Weight",
       col=colors,
       par.settings=list(superpose.symbol=list(col=colors,pch=c(19), cex=1.5)),
                        auto.key=list(space="right"))
```



The only weakness of this approach is that we are passing in a list within a list so it is easy to type too many parentheses (or not enough). As I mentioned in class this kind of thing can become quite tedious. Now, to generalize this kind of thing you should probably create a function out of this and pass in arguments such as `pch` and `colors` to generalize this. Just substitute in the argument names into the `par.settings` list above.