A Brief Introduction to Graphics with ggplot2

November 6, 2017

Introduction

The ggplot2 package allows you to build very complex graphs layer by layer. Unlike graphs we construct using the base functions in R, ggplot2 takes care of details like legends and choice of plotting symbols automatically, although you can customize these choices if you wish. A handy cheatsheet which summarizes the commands available in ggplot2 can be downloaded here `www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf`.

ggplot()

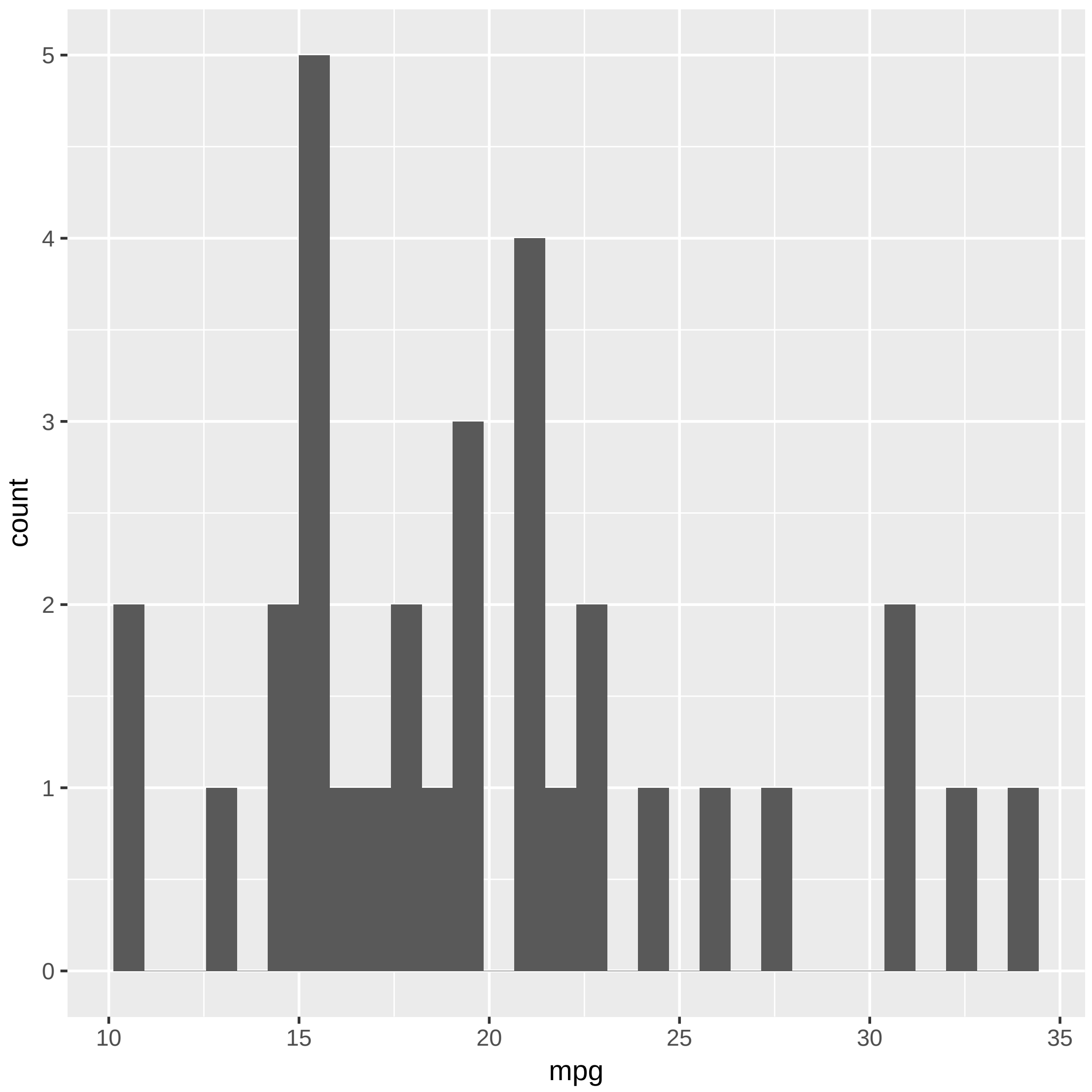
Using the ggplot2 suite of functions you start a graphic using the `ggplot()` command. This command does not display anything until you add a 'geom' command; it just sets up the scaffolding for the plot. The syntax of `ggplot()` is

```
ggplot(data = NULL, mapping = aes(x, y, <other aesthetics>))
```

- The `data` argument is the dataframe containing the variables you want to graph. It must be an object of type dataframe.
- The x argument is set equal to the variable in your data you wish to be represented on the x-axis. The y argument will be the variable represented on the y-axis.
- You can map additional variables in your dataset to plot attributes like color or size of plotting symbol, by simply adding an argument like `color=gender` within aes().

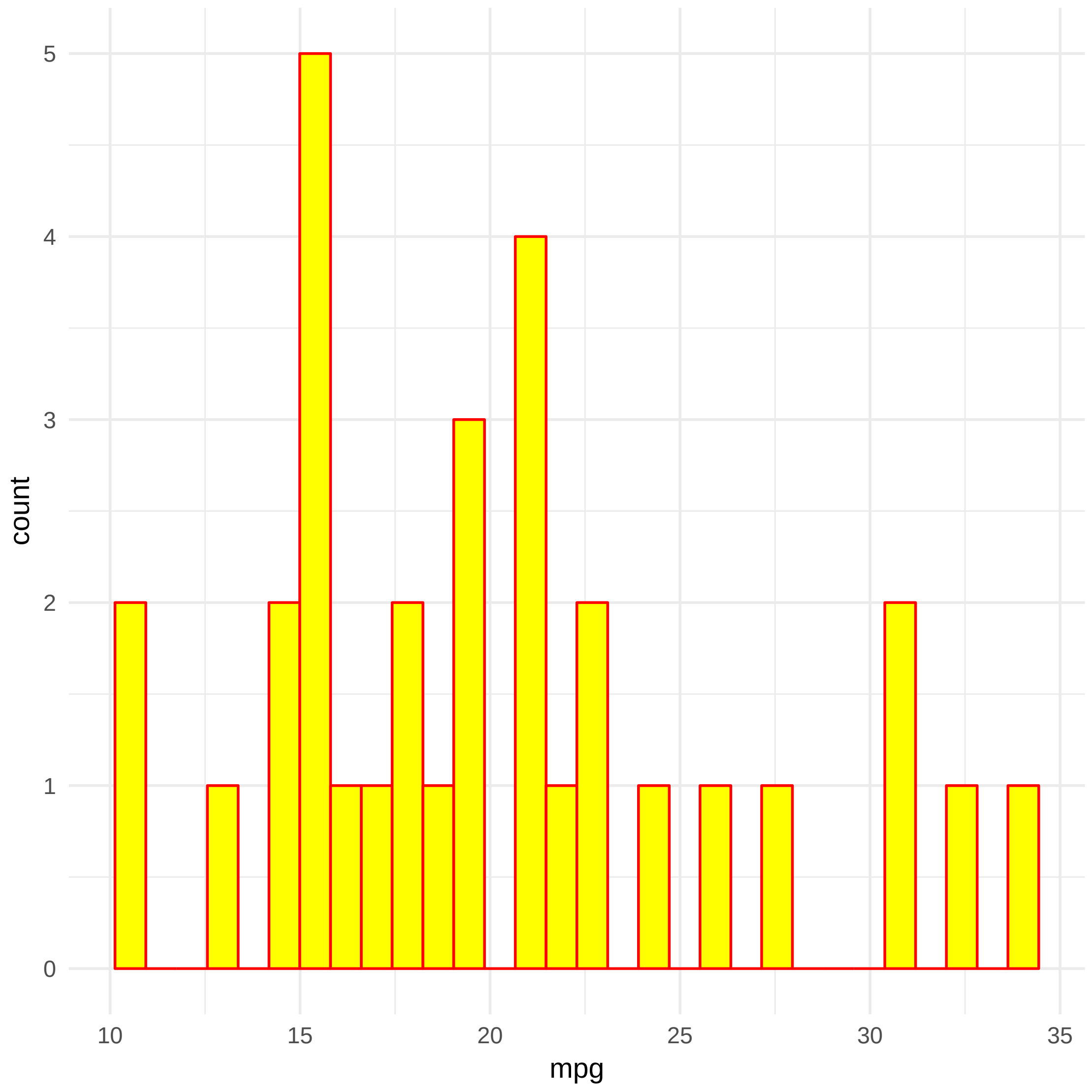
You add (literally, using a + sign) a 'geom' to `ggplot()` such as `geom_histogram()` or `geom_dotplot()` to choose the type of graph which will display the data. Let's obtain a histogram of mpg in the mtcars dataframe.

```
> library(ggplot2)
> class(mtcars)  #verify mtcars is a dataframe

[1] "data.frame"

> # sets up the plot, but does not produce a graph yet
> p0 <- ggplot(data=mtcars, aes(x=mpg))
> #now get the graph
> p0+geom_histogram()
```



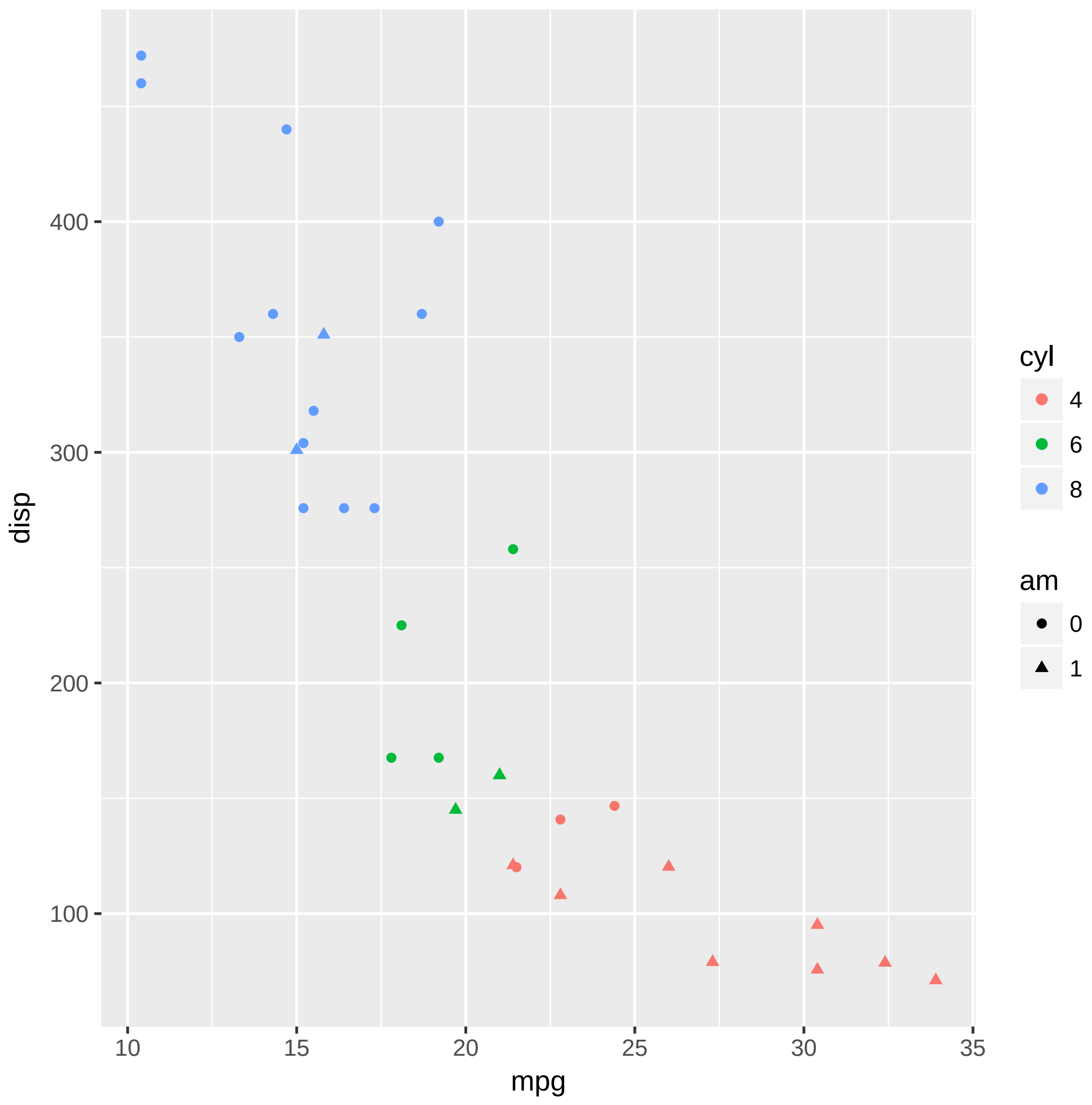
Let's customize the graph. There are different 'themes' which determine the setup of the plot area, i.e. whether gridlines are shown and the colors of gridlines and the background. See the cheatsheet, page 2, bottom right for a few choices. You can choose the fill and outline colors.

```
> p0 <- ggplot(data=mtcars, aes(x=mpg))
> p0+geom_histogram(fill="yellow", color="red")+theme_minimal()  #ugliest graph ever!
```



You can use color and plotting character shape to represent variables in your datasete.

```
> mtcars$cyl=factor(mtcars$cyl)  #factor type required for representing cylinder categories with different co
> mtcars$am =factor(mtcars$am)
> p1 <- ggplot(data=mtcars, aes(x=mpg,y=disp,color=cyl, shape=am))
> p1+geom_point()
```



Exercise:

1. Using the ChickWeight (note W is capitalized) data, obtain the subset for observations at time 21, using the code below:

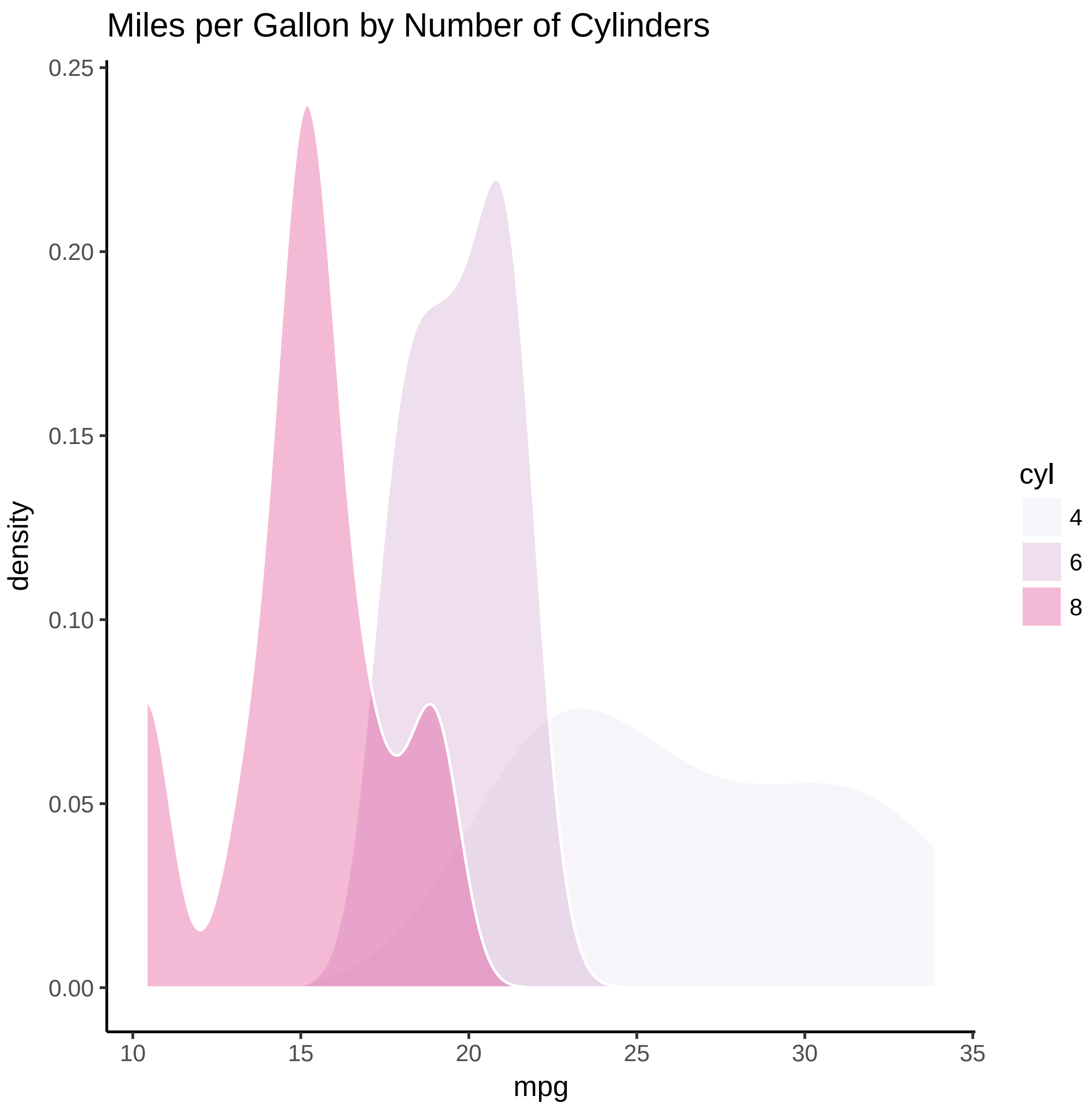
   ```
   > chick.sub <- subset(ChickWeight, Time==21)
   ```

2. Load the ggplot2 package and construct a histogram of the Weights variable with fill and outline colors of your choice.

3. Use the `iris` data for this part. Plot Petal.Length on the x-axis and Petal.Width on the y-axis. Use different colors or plot symbols to represent the Species variable.

Graph a smoothed density of the mpg for each 'cyl' category and choose a specific color palette. To see color palette choices, load the RColorBrewer package and type `display.brewer.all()`.

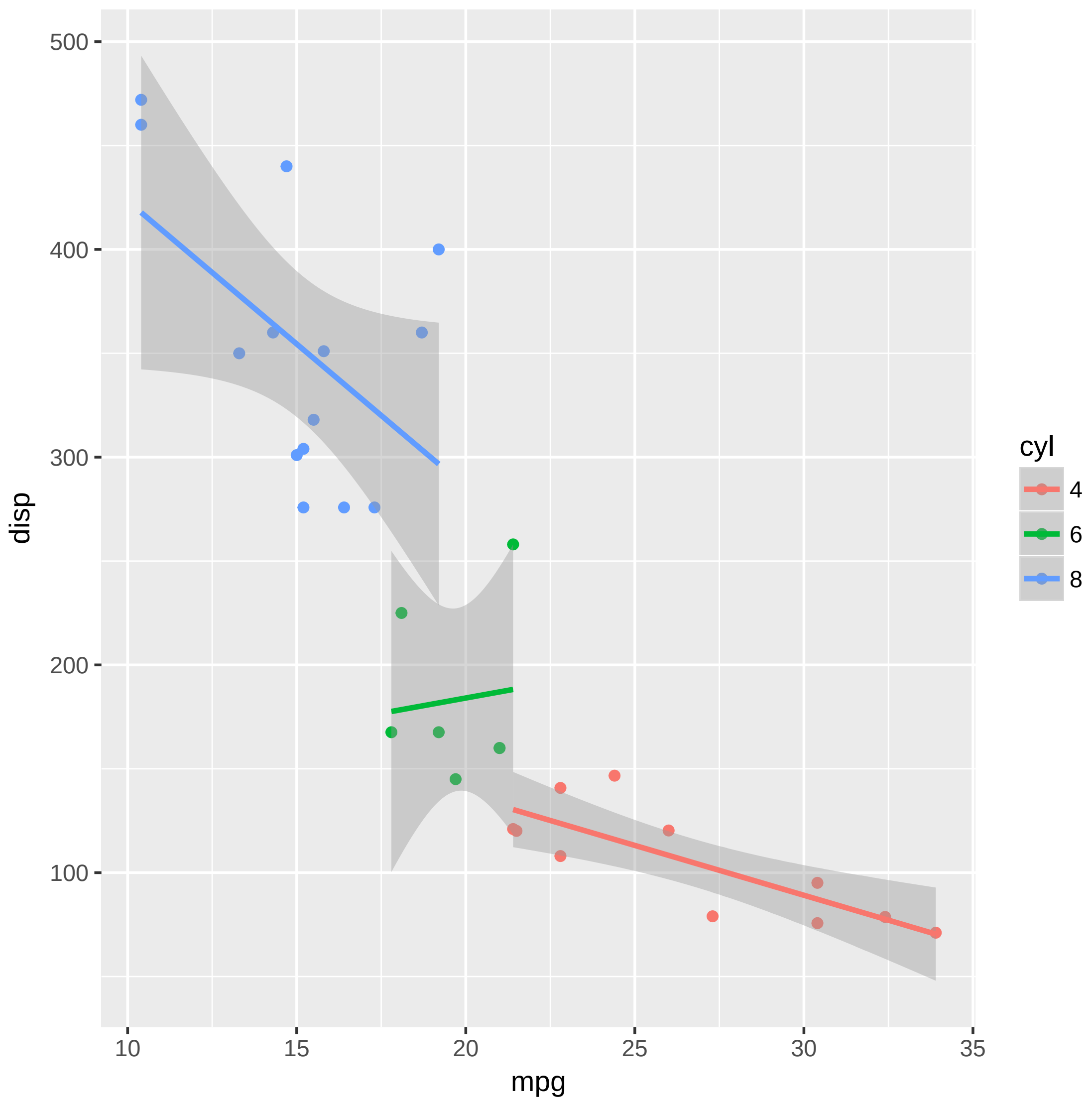
```
> p1 <- ggplot(data=mtcars,aes(x=mpg))
> p2 = p1+ggtitle("Miles per Gallon by Number of Cylinders")
> p3 = p2+geom_density(aes(group=cyl,fill=cyl),color='white', alpha=0.3)
> p4 = p3+theme_classic()+scale_fill_brewer(palette="PuRd")
> p4
> #alpha sets the transparency of the fill color
```



Exercise: Using the ChickWeight data, produce separate smoothed density graphs of weights at time 21 by Diet.

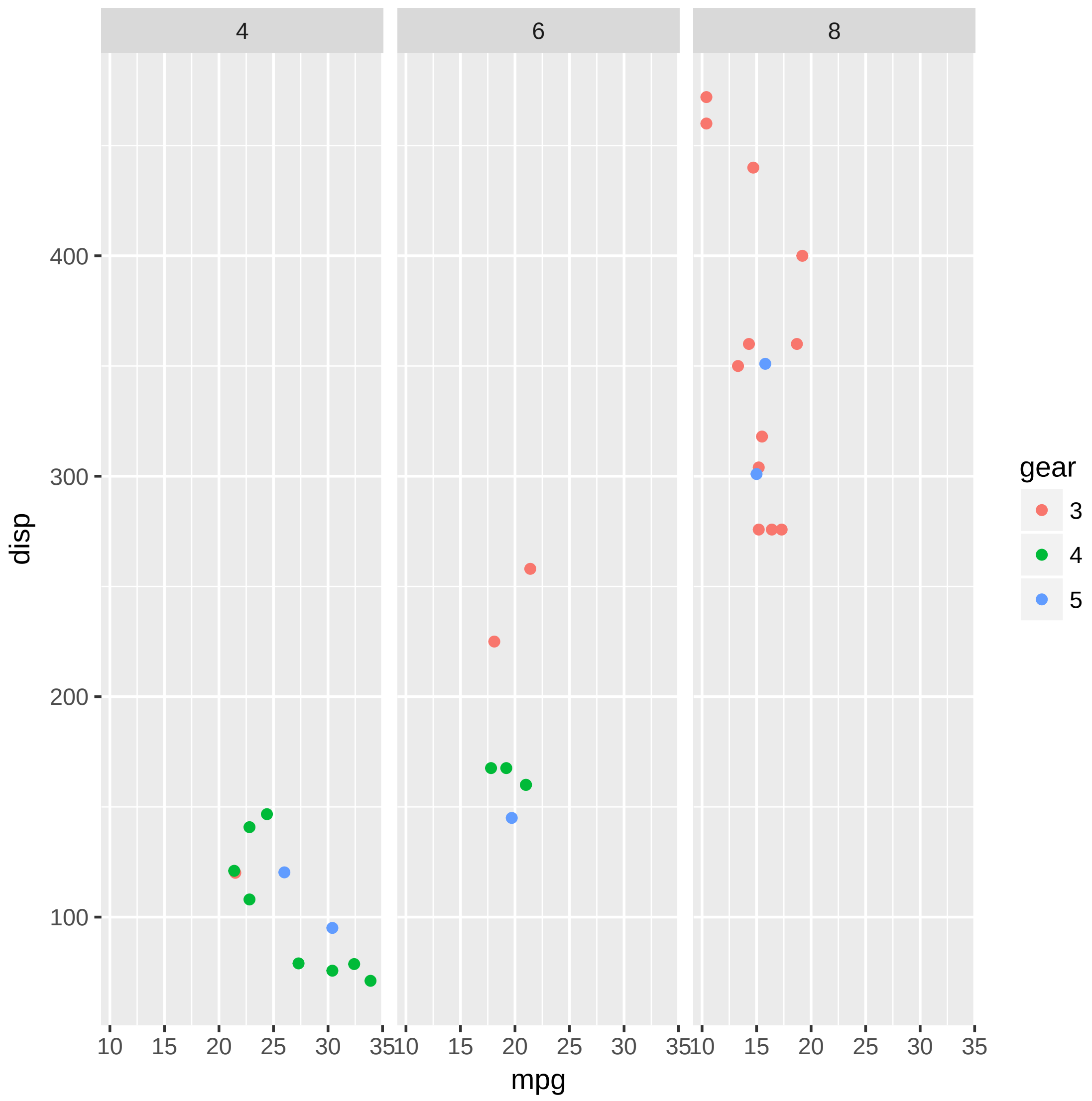
Adding Layers

```
> p1 <- ggplot(data=mtcars,aes(x=mpg,y=disp,color=cyl))
> p1+geom_point()+geom_smooth(method='lm')
>
> # the default method in geom_smooth overfits the data, IMO
> # p1+ geom_point()+geom_smooth()
```



- If you want a multi-pane graph with the same graph repeated on subsets of you data, you can use the facets argument. You must input it as a formula. For example, in the mtcars dataset if you want to graph mpg vs. disp in separate columns for 4,6, and 8 cylinder cars, you'd use `facets=.~cyl`. Separate graphs for each cylinder and gear combination would use `facets = cyl~gear`; cylinders vary by row and gear varies by column.

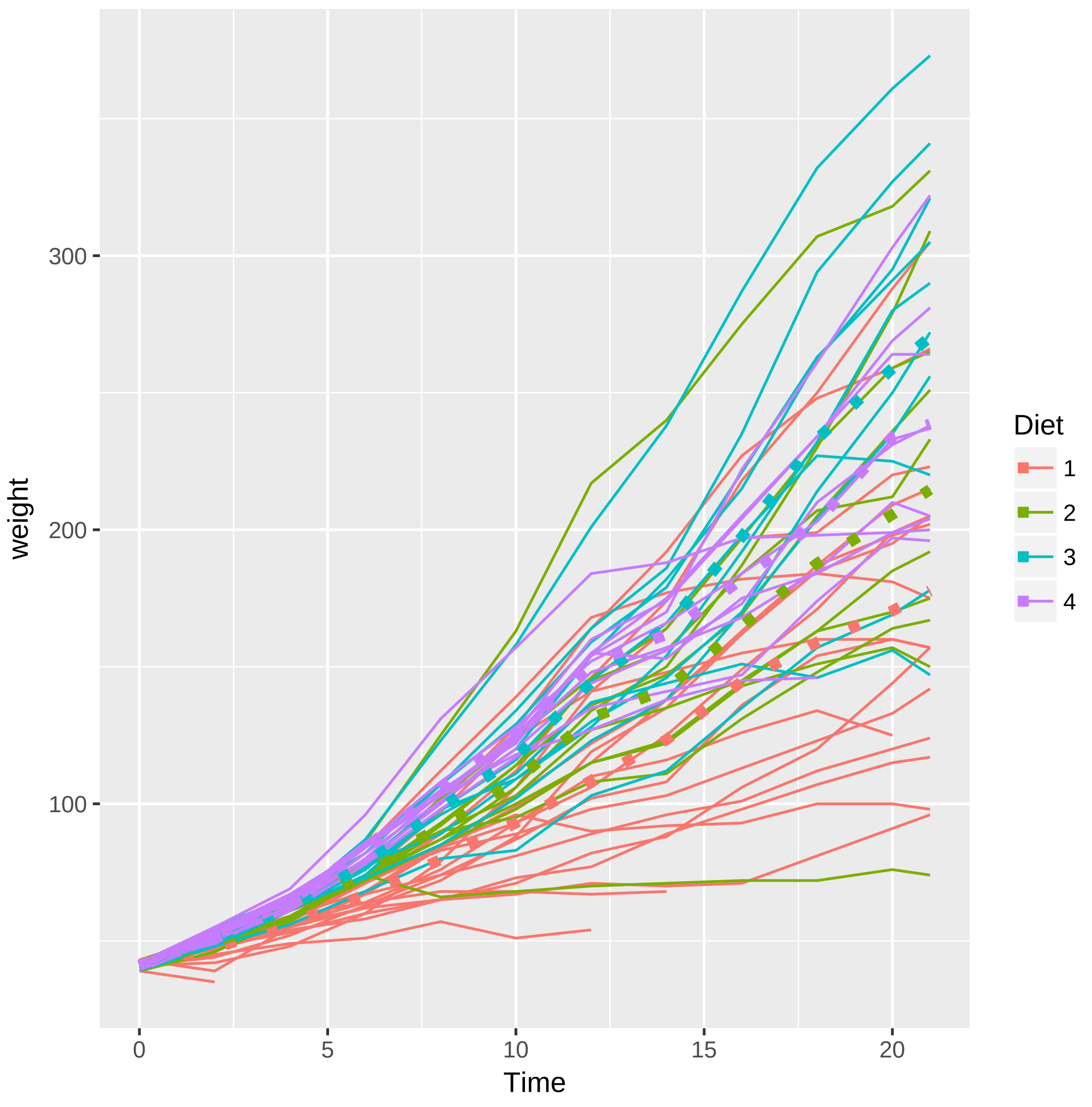
```
> mtcars$gear <- factor(mtcars$gear)
> p2 <- ggplot(data=mtcars,aes(x=mpg,y=disp, color=gear))
>              p2+facet_grid(.~cyl)+geom_point()
>
```



Exercise: Using the iris data, obtain a scatterplot of x=Petal.Length vs. y=Petal.Width. Facet by Species.

A graph of longitudinal data (each subject is observed repeatedly over time) for the ChickWeight dataset.

```
> my.data.summary <- plyr::ddply(ChickWeight, c('Time', 'Diet') ,
+                               plyr::summarise, mean = mean(weight), sd = sd(weight))
> p1 = ggplot(data=ChickWeight, aes(x=Time,y=weight,color=Diet))
> p2=p1+geom_line(aes(group=Chick))
> p3=p2+geom_line(data=my.data.summary, aes(x=Time, y=mean, color=Diet), linetype=3, size=2)
> p3
> #add mean line by group
>
```



How to use the ggplot2 cheatsheet examples

The examples use datasets included with the ggplot2 package. Worth through the geom_line() example.