An introduction to ggplot2

Getting Started:

We'll need to start by loading the library ggplot2, as well as some datasets to play around with:

```
library(ggplot2)
library(datasets)
data(iris)
```

Exploring shapes and colours:

Let's take a look at the Iris dataset using our formula:

```
ggplot(dataset, aes(aesthetics of plot))+
geom_we'dliketoplot(aesthetics of geom)+
additional_aesthetics()
```

Lets look at a scatterplot between sepal length (on the X) and petal length (on the y):

```
ggplot(iris, aes(x = Sepal.Length,
  y = Petal.Length))+
  geom_point()
```

We can change the shape or color of the point by simply adding a command in the ggplot aes() parameter.

Change color:

```
ggplot(iris, aes(x = Sepal.Length,
  y = Petal.Length, col="red"))+
  geom_point()
```

or shape:

```
ggplot(iris, aes(x = Sepal.Length,
  y = Petal.Length, shape="square"))+
  geom_point()
```

We can also code the points on the graph by another variable (continuous or discrete) in the dataframe.

By discrete character:

```
ggplot(iris, aes(x = Sepal.Length,
  y = Petal.Length, col=Species))+
  geom_point()
```

Or continuous character

```
ggplot(iris, aes(x = Sepal.Length,
  y = Petal.Length, col=Sepal.Width))+
```

```
  geom_point()
```

Exploring different geoms:

'geoms' are simply the type of plot you wish to produce. There are lots of different types of geoms one can use to plot. Here are just a few:

Scatterplots (geom_point, geom_line)
This is what we've been plotting so far: requires continuous variables in the X and Y.

Boxplots (geom_boxplot)
*Plots a number of summary statistic for each character (**XXX**). Requires discrete character for X and continuous variable for Y.*
For example, this will plot sepal length for each species:

```
ggplot(iris, aes(x=Species, y=Sepal.Length))
    +geom_boxplot()
```

Histograms (geom_histogram)
Plots the distribution of a character (i.e. how many counts there are of each numerical value). Uses only one continuous character.

We can plot all species together:

```
ggplot(iris, aes(x=Sepal.Length))
    +geom_histogram()
```

Or, we can plot the histogram for each species separately:

```
ggplot(iris, aes(x=Sepal.Length,
fill=as.factor(Species)))
    +geom_histogram()
```

Plotting means for discrete factors (geom_barplot, geom_point)
Plots a single summary statistic for each character (usually the mean). Requires discrete character for X and continuous (or at least numerical) variable for Y.

Lets try to plot the data as it is:

```
ggplot(iris, aes(x=Species, y=Sepal.Length))
    +geom_point()
```

Well, that doesn't look very good! We only want to plot the mean! In order to do so, we must first calculate the mean for each group. Luckily, there are some nice libraries in R that can help us summarize data in the fashion that we want. Let's use one called 'plyr'.

```
library(plyr)
```
#here we are summarizing the variables Sepal.Length by the factor Species. Our summary stats are: the # of observations (N), the mean, standard deviation, and standard error.
```
v<-ddply(iris, c("Species"), summarise,
N=length(Sepal.Length),mean=mean(Sepal.Length),
sd=sd(Sepal.Length), se=sd/sqrt(N))
```

Now, let's plot it:

```
ggplot(v, aes(x=Species, y=mean))
     +geom_point()
     +geom_errorbar(aes(ymin=mean-se, max=mean+se))
```

You'll notice we've added another aesthetic to the graph- errorbars! There are many circumstances in which a boxplot can obscure differences between groups, because of how much data it plots all at once. Mean + errorbars can be a nice, simple solution to showing differences between groups, if they exist.

Reaction Norms (geom_line)

Plots the change in a value over two (or more) observations (i.e. over a treatment, or an environment). Requires Y be continuous, but X can be discrete or continuous.

One nice way to use geom_line() is to look at changes over time. Let's use this gene expression dataset from Colin Maxwell, a former Duke Biology grad student, who was interested in. Colin measured gene expression using RNAseq over a developmental time course in C. elegans *larvae. He's parsed the dataset to include only a handful of genes from the original 18000.*

```
isoforms <- read.csv("isoforms.csv")
```

We'll get into more complicated graphs shortly, but for now, let's focus on 1 gene and plot its expression over time.

```
col <- isoforms[which(isoforms$gene=="col-107"),]
ggplot(col, aes(x=time, y=mean, col=isoform))
     +geom_line()
```

We could add errorbars to this graph by using the values of 'lo' and 'high' as a range:

```
ggplot(col, aes(x=time, y=mean, col=isoform))
     +geom_line()
     +geom_errorbar(aes(ymin=lo, ymax=hi))
```

Additional Aesthetics:

Change labels:

Try adding the `xlab("title"), ylab("title"),` or `ggtitle("title")` *command to change labels*

Change colors of points:

You can specify specific colors that you want for points by using

```
scale_color_manual(values=c("col1", "col2", "col3", etc)
```

Here's a useful list of a bunch of colors available (and their respective names) in R:
http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf

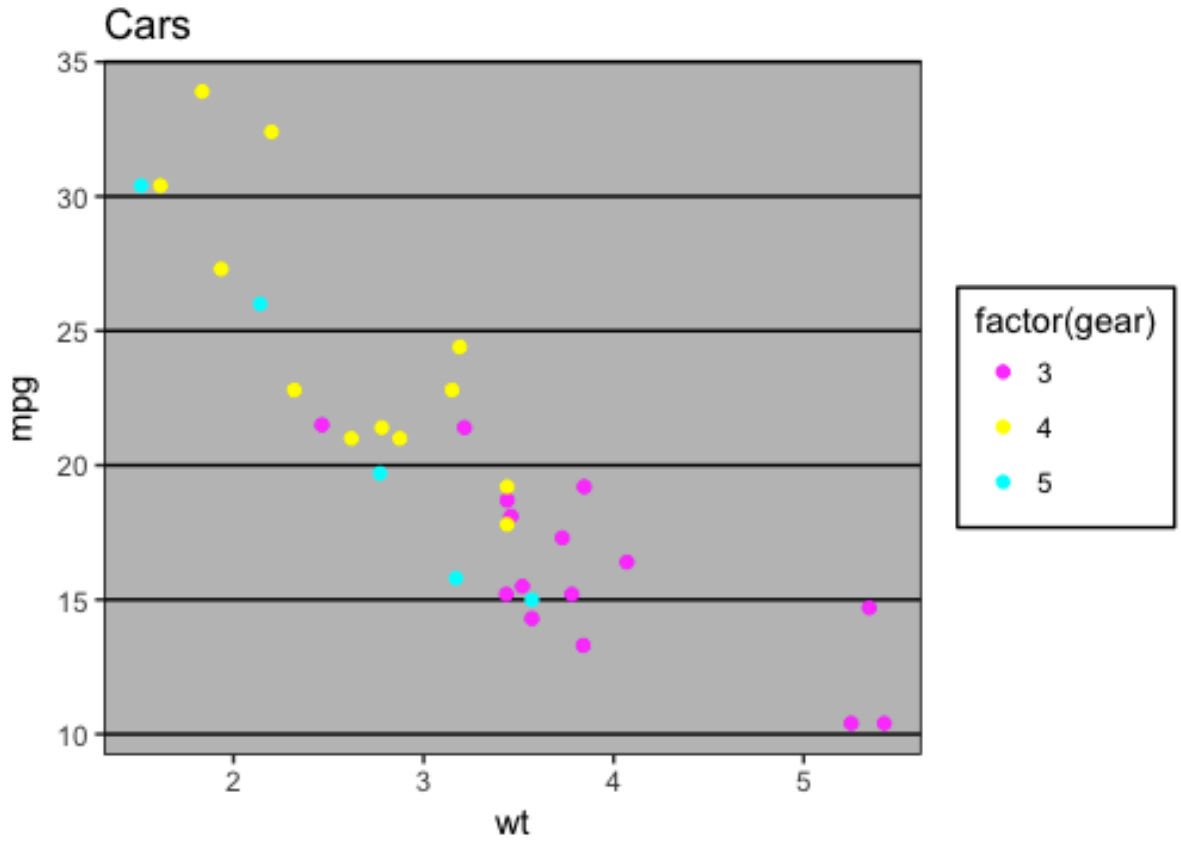
Add lines (to show a cut off, or a mean, or a slope):
Use `geom_vline(), geom_hline,` *and* `geom_abline` *to show specified vertical and horizontal lines, and a line with a specified slope, respectively*

Themes:
Don't like that grey background on the default ggplot? Change it!

```
Theme_"themeyouwant"()
```

I like:
```
theme_bw()
theme_classic()
```

but there are some real doozies that you have to download certain palettes for, for example, this old beauty uses `theme_excel()` *to mimic excel graphs circa early 2000s.*
Check out some more here:
https://cran.r-project.org/web/packages/ggthemes/vignettes/ggthemes.html



Facets:

So far we have been plotting 1 plot at a time. But what if you have more complicated data? Lets go back to isoforms.

Try to plot all genes at once using this code:

```
ggplot(isoforms, aes(x=time, y=mean, col=isoform,
shape=gene))+geom_point()
```

This is pretty illegible, to me. Let's clean it up using a facet_wrap().
Let's make each panel a different gene:

```
ggplot(isoforms, aes(x=time, y=mean, col=isoform))
     +geom_line()
     +facet_wrap(~gene, scale="free_y")
     +scale_y_log10()
```

We've also added a log10 scale, but allowed each panel to vary in their y scale using the `scale_y_log()` *and* `scale="free_y"` *aesthetic in facet_wrap(), respectively.*

Let's try to integrate all of what we've learned to make this graph a little nicer looking. We can add errorbars, change the colors to be more appealing, change labels and colors, and tidy up that background.

```
ggplot(isoforms, aes(x=time, y=mean, color=isoform))
     +geom_line()
     +geom_point()
     +geom_errorbar(aes(ymin=lo, ymax=hi))
     +facet_wrap(~gene, scale="free_y")
     +scale_y_log10()
     +ylab("Mean isoform expression, +/- 95% CI")
     +xlab("Time(hours)")
     +scale_color_manual(values=c("dodgerblue", "goldenrod",
     "firebrick","mediumseagreen","lightcoral","darkorange","d
     arkmagenta","aquamarine"))
     +theme_bw()
```

Other Great Resources:
http://ggplot2.tidyverse.org/index.html