ggplot basics

A quick introduction to ggplot. Important terms you'll learn:

- **aesthetics**
- **mapping**
- **frame**
- **geometric objects ('geoms')**
- **scales**
- **facets**

The package we'll be using is called **ggplot2**, but is often referred to as simply 'ggplot.' Be sure you have installed ggplot2, and have it loaded in R Studio.

The main function in ggplot2 is called **`ggplot()`**. The first thing you'll need to tell **`ggplot()`** is the name of the data frame that holds your data. The first argument is
`data =` *[data frame name]*. For example, if my data frame is called 'survey,' then I would start with:

```
ggplot(data = survey, …..
```

Now, suppose you want to make a plot of the relationship between **height** and **weight** in UVM students. Further, you'd like to indicate the **gender** of each student, and whether or not they are a **smoker**. Perhaps you'll represent each student with a point on the plot, and height and weight will be represented using the point's position along the **x-axis** and **y-axis**. Further, you could let the **color** of the point represent the gender of the student, and the **shape** of the point represent whether they are a smoker or not. See the plot below on the right.

These ways of expressing the values (x-axis, y-axis, color, shape) are called **aesthetics**. The next thing you do in ggplot is to state how you will '**map**' ( ) different variables to different aesthetics. The schematic on the left below would 'map' the aesthetic of **height** to the x-axis and **weight** to the y-axis, and represent **gender** with different colors, and **smoker** with different shapes of. Here is the 'mapping' argument:

```
ggplot(data = survey, mapping = aes(x = Height, y = Weight,
color = Gender, shape = Smoker))
```

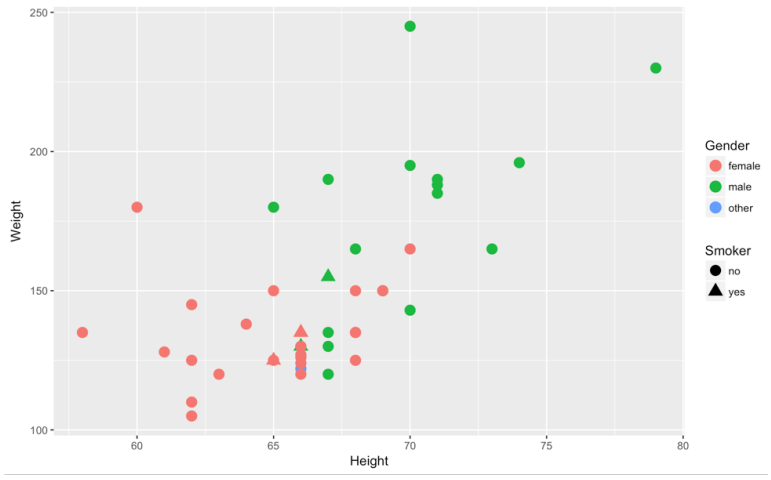
Variable → aesthetic
==========

Height → x-axis

Weight → y-axis

Gender → color

Smoker → shape
==========



Let's do an example using a data set that comes with the ggplot2 package. Access the data set **mpg**, by typing **`data(mpg)`**, then maybe **`View(mpg)`**. Suppose we plan to graph a car's highway mileage (variable **hwy**) and the engine size in liters (variable **displ**). A simple and traditional way to plot these two variables is a scatterplot. If we want displ as the 'x-variable' and hwy as the 'y-variable,' then we would tell R to **map** them to aesthetics like this:

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

If you run the two lines above, you'll get a nice pair of axes for your plot, but the plot will be empty. You've only created a **frame** for your plot. By identifying the x and y variables, you've indicated to R where your dots should go on the plot, but you haven't told R that dots, or points, are what you want. In fact, there are many geometric shapes or **geoms** available for plotting, and dots are just one type of geom. (Others include bars, lines, curves, and more.) To tell R which geometric shape you want to use, you'll need a '**geom_**' function. There are many, many geom_ functions, and we will explore more as we go along. To graph dots, you'll use **`geom_point()`**, like this:

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

Notice that you use a plus sign after your ggplot function, to tell R that you have more information coming, i.e., this is not just an empty frame!

Now suppose you want to map another aesthetic, this time to the variable **class**, which indicates type of car (compact, suv, 2seater, etc.). You might want to represent each class with a different color dot. You can simply add another aesthetic to the ggplot function:

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

You should see a legend which gives a **scale**, showing which colors go with which types of cars. The values on the x-axis give the scale for displ, and the values on the y-axis give the scale for hwy. The scale helps the viewer understand how each aesthetic is displaying the variable.

There are more useful aesthetics: try the code above, replacing color=class with each of these, one at a time:

- **`size = class`**
- **`alpha = class`**
- **`shape = class`**

The options (`shape = class` and `size = class`) will give you warning. What is the reason for each warning? Which of the aesthetics works the best for this data, in your opinion?

Facets are another useful aesthetic. They allow you to present your plot in separate panels, in order to plot "simple multiples." We map facets in a different way. Try this code:

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

You should see one facet for each class of car. Try changing the last line to each of these, to see some of the different facet formats that are possible:

```
facet_grid(class~.)
facet_grid(.~class)
```

Of course you can use multiple aesthetics in the same plot. For example:

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

There are many, many **geom_**'s possible. Here is one example: Suppose you wanted to plot hwy by displ with dots, but you wanted a smoothed line, also. You could simply add another geom as another layer to the plot:

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

In addition to plotting layers, there are many attributes that can be layered onto your graph in order to make it more attractive or more descriptive. For example, to add a title to your graph, simply add a label or 'labs' layer:

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
       geom_point() +
       geom_smooth() +
       labs(title = "Auto data with smoothed curve")
```

You may also use labs to add more descriptive titles to your x and y scales:

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
       geom_point() +
       geom_smooth() +
       labs(title = "Auto data",
            x = 'Engine displacement in liters',
            y = "Highway mileage")
```

You may add description to other scales, as well. For example, here we change the description of the color scale:

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color=class)) +
       geom_point() +
       labs(title = "Auto data ",
            x = 'Engine displacement in liters',
            y = "Highway mileage",
            color = "Type of car")
```