Tutorial 11 - Plotting in R

Which plot when?

What plot type you want to make is completely dictated by the type of data you want to visualize. When considering plot type ask two questions:

1) How many variables (measurement types) do I want to visualize on my plot?
2) Is each variable to be shown on my plot continuous or categorical?

- continuous data: can have any value between a theoretical minimum and maximum (e.g. temperature, height, concentration)
- categorical (discrete) data: can only take on one of a fixed number of values (e.g. species, site, country)

Which plot when?

Common answers to those two questions in Biology are:

- one categorical variable: bar plot
- one continuous variable: histogram, density plot
- one categorical and one continuous variable: bar, box, & violin plots
- two continuous variables: scatter plots
- two continuous variables and one or more categorical variables: scatter plots with point colors or shapes

Less often, you might have two categorical values or three continuous variables

Look at the ggplot cheatsheet to see what simple examples of some of these plots look like

Challenge - imagine and hand draw some plots

Think about what information from your daily life you could depict with the following plots:

1) histogram (single continuous variable)
2) violin plot (one categorical and one continuous variable)
3) scatter plot (two continuous variables)

Now sketch what you think those plots would look like

A ggplot checklist

At a minimum, making plots with ggplot in R involves the following steps and ggplot functions:

- linking data (in a dataframe) to a ggplot object: `ggplot()`
- defining which variables you would like displayed where: `aes()`
- creating one or more layers that display the data: `geom` and `stat` functions

You can customize the appearance of your plot by:

- specifying the display of axes, legends, etc.: `scale` and `label` functions
- specifying other plot details (font size, background color, etc.): `theme` functions or arguments to `geom` calls

The minimum ggplot call

At a minimum, code to generate a plot using ggplot would be:

```
ggplot(data=**1**,aes(**2**)) + geom_**3**()
```

1 is the name of a data frame that exists in your current R environment

2 is one or more column names in the data frame

3 is the suffix of a geom_ function from ggplot

The geom function you choose depends on what you specify for **2** (e.g. x, y, color) and the particular plot type you want to create.

Worked example #1

Let's make a histogram of engine size from the mpg data set.

Worked example #1

Let's make a histogram of engine size from the mpg data set.

```
ggplot(data=mpg,aes(x=displ)) + geom_histogram()
```

This is a minimum call. What would we add to make the x axis label more informative, make the bars red, and get rid of the grey gridded background?

Worked example #1

Let's make a histogram of engine size from the mpg data set.

```
ggplot(data=mpg,aes(x=displ)) + geom_histogram()
```

This is a minimum call. What would we add to make the x axis label more informative, make the bars red, and get rid of the grey gridded background?

```
ggplot(data=mpg,aes(x=displ)) +
geom_histogram(fill='red') + theme_classic() +
xlab("engine displacement (l)")
```

Worked example #2

Let's make a scatter plot of displ and cty from the mpg data set.

Worked example #2

Let's make a scatter plot of displ and cty from the mpg data set.

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

This is the minimum call. What would we add to change the point shape to squares, color the points by cyl, and get rid of the grey gridded background?

Worked example #2

Let's make a scatter plot of displ and cty from the mpg data set.

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

This is the minimum call. What would we add to change the point shape to squares, color the points by cyl, and get rid of the grey gridded background?

```
ggplot(data=mpg,aes(x=displ,y=cty,fill=cyl)) +
geom_point(shape=22) +  theme_classic()
```

Worked example #2

```
ggplot(data=mpg,aes(x=displ,y=cty,fill=cyl)) +
geom_point(shape=22) +  theme_classic()
```

The code above creates a continuous gradient of color for the number of cylinders a car model has. There are a limited number of numbers of cylinders a car engine has and so we might want to treat this as categorical data instead. We can force this with `as.factor()`.

```
ggplot(data=mpg,aes(x=displ,y=cty,fill=as.factor(cyl)))
+ geom_point(shape=22) +  theme_classic()
```

Challenge - practice plotting

1) Using the mpg dataset, generate a bar plot of mean engine dissplacement (displ) for engines with different numbers of cylinders (cyl). Include error bars representing standard error around each mean.

2) Using the iris dataset, generate a density plot of sepal width, but include a density line for each Iris species.

Looking ahead to next week

As always, Exercise 9 is due next Friday (11/18). Be sure to fork and clone this from your TA's github repository.

We will be working on more sophisticated coding in R next week. Please complete a Software Carpentry activity on loops in R before lecture on Monday (11/14).