

DATA VISUALIZATION -- GGPLOT2

Created by: Megs Seeley

Section 0: Load Data

The first step in any data visualization process is acquiring, cleaning, and organizing the data. Your data can take many forms -- rasters, shapefiles, csv. The following is a brief introduction on how to import each data type.

Raster

```
library(raster)
data = raster('FILEPATH.tif')
```

Shapefile

```
library(rgdal)
data = readOGR('FILEPATH.shp')
```

CSV

```
data = read.csv('FILEPATH.csv')
```

You can explore your data by simply printing it:

```
data
```

Or using functions like:

```
head(data)
summary(data)
colnames(data)
```

If you do not have a dataset already, you can use the iris dataset and call it with the following commands:

```
library(datasets)
data(iris)
head(iris)
```

Section 1: Organizing your Data

Let's first call an entire column (Sepal.Width) in our iris dataset.

```
iris$Sepal.Width  
iris[2]  
iris[,2]
```

Both of the above commands perform the same function. The first uses the '\$' notation to call a column by name, and the second and third examples use indexing.

We can also call entire rows using indexing.

```
iris[2,]
```

When we include a comma after the two, we get the second row rather than a column. This is because we call data using [row, column]. If we exclude either the row or column value, we can call the entire row or column.

If we want to remove a row or column, we can use indexing with a '-'

```
iris[-2]
```

Remember that these examples simply print the dataset and do not save it. If we want to overwrite the dataset, we can use:

```
iris=iris[-2]
```

Next, we will go over dplyr. dplyr is a great package to know for organizing and cleaning your data.

```
library(dplyr)
```

We can use '%>% ' to run a dataset through multiple functions. For example

```
iris %>% group_by(Species) %>% print()
```

group_by is another dplyr function that allows us to group our data based on another column. Let's get the mean sepal.Length and sepal.Width for each species.

```
iris2 = iris %>% group_by(Species)
iris2 = summarise(iris2,
  sepalLength = mean(Sepal.Length),
  sepalWidth = mean(Sepal.Width))
```

You can find more information on the dplyr package here: <https://dplyr.tidyverse.org/>

Section 2: Plotting your data

ggplot requires us to build our graphs one piece at a time. First we ingest our data and identify our x and y values:

```
base = ggplot(data = iris2, aes(x=Species, y = sepalLength))
```

If we print our variable 'base' the above line, we get an empty plot. This is because we need to give the graph parameters. We can add parameters using a '+'

```
base +  
  geom_bar(stat = 'identity')
```

This prints a bar chart. We can change the chart type by using other geometry identities like `geom_boxplot` or `geom_point`. For more chart types, refer to this cheatsheet (<https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf>)

If we wanted to change the fill color of the bars to represent different species, we can add a fill parameter to our aes:

```
base = ggplot(data = iris2, aes(x=Species, y = sepalLength, fill = Species))
```

Now let's clean up our chart labels and remove the legend:

```
base +  
  geom_bar(stat = 'identity') +  
  ylab('Sepal Length') +  
  ggtitle('Average Sepal Length by Species') +  
  guides(fill = F) # Removes legend
```

The last visualization option we will cover is color. We can customize our chart using a color palette. For color codes, refer to the CSS color palette:

https://www.quackit.com/css/css_color_codes.cfm

```
palette = c("#000000", "#E69F00", "#56B4E9")
```

```
base +  
  geom_bar(stat = 'identity') +  
  ylab('Sepal Length') +  
  ggtitle('Average Sepal Length by Species') +  
  guides(fill = F) +  
  scale_fill_manual(values = palette)
```

Now explore the many options ggplot2 offers! You can include more parameters to each of the functions we already called or add new functions to the plot. For inspiration and the underlying code, check out these neat plots:

<http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html>