# Homework 2, part A

## Simulation Exercise

We will start with a simulation. Simulations are an incredibly important part of a statistician's job because it helps us to understand the performance of algorithms under various conditions. In simulations, we have access to the ground truth- something we never get with real data. In this exercise, we are going to gain experience with the bootstrap. In this exercise, you MAY NOT use the function "boot." Instead, we will write our own bootstrap function

1. Write a function called myboot. myboot will take three arguments: data, statistc, and R. Data is your dataset, statistic is the statistic you are calculating (for instance, mean, median, etc), and R is the number of bootstrap datasets you want to run (I'm naming these to line up with the boot function, by the way). We will assume that the data is univariate, passed in as a vector, and that the statistic being calculated is also univariate. Also, note that this is not the most efficient way to code up the bootstrap, necessarily, but it is the most illustrative. Here is some pseudocode:

myboot <- function(data, , statistic=mean, R){ #what is the "=" sign doing here?

# initialize a vector that will store the values from your boostrap. This will be of length R.

# write a for loop that, over R times

# 1. Samples from the data WITH REPLACEMENT (there are several ways to do this), and call this sample sample.boot.

# 2. Calls statistic(sample.boot) to calculate your statistic on this bootstrapped dataset

# 3. Record your statistic in the vector you initialized.

# After the forloop, calculate the standard deviation of the vector of bootstrapped statistic.

# return the standard deviation. Note here that it is often interesting to see the entire distribution, in which case you would return the vector, not the standard deviation of the elements in the vector.

}

2. In question 1, we wrote a function that calculates the bootstrap standard deviation from a dataset. Now we are going to see how the bootstrap performs. We are going to draw data from a normal distribution. We will consider three different sample sizes- 30 data points, 100 data points, and 200 data points.
    a. Do the following:
        i. Draw 30 data points from a normal distribution with mean=0 and variance = 10. If you were running an experiment in the wild, this would be your dataset.
            1. Calculate the mean of your sample
            2. Calculate the standard error of your sample using the myboot function you created.

        a. How many bootstrap samples are you running? What is the effect of running more bootstrap samples?
    ii. Repeat a. 100 times
        1. Store the means in a vector
        2. Store the bootstrapped standard errors in a vector
    iii. What is the distribution of the standard errors? (Note, you do not get to see multiple realizations of the standard errors on real datasets. This is why simulations are so helpful). Discuss, and compare this distribution of errors to the theoretical standard error.
    iv. Construct 95% confidence intervals for all of your means (a *vectorized* solution is the best way to approach this!). Of the 100 confidence intervals, how many contain 0?

b. Repeat a. using 100 data points and 200 data points (one way to avoid writing a bunch of code over is to avoid *hard coding* either by writing functions or by writing your code so that you only need to change a few values at the beginning of the code.
c. What happens to the accuracy of the bootstrapped standard error as your sample size grows? (you should be looking at this distributions of standard errors for this)
d. In your own terms, summarize the bootstrap.