Stat 230, Spring 2016
Homework 8: Bootstrap
Read and understand all problems in Ch 8 (there aren't a lot and they are mostly pretty quick given what we've done in lecture).

Due Thursday 3/31/15 at 11:59pm on bspace.
In R part of this HW, you will bootstrap in order to estimate the SE for the coefficient estimates from a linear and quadratic model, and to get a sense of potential bias. The function used to generate the data is given to you with a seed so that everyone is using the same data.

You will perform the bootstrap with two methods to generate a bootstrap sample of (x, y) pairs.

1. For each unique x value in the data there are 10 y values. For each x value, sample WITH replacement 10 times from these 10 y values. This will generally give you some duplicate points.

2. Use the predicted/fitted values and residuals from the fit of your data to a line (or a quadratic). For each fitted value, construct a y by adding to it a randomly chosen error. Sample from the residuals WITHOUT replacement (i.e. permute them) to generate the bootstrap errors.

For each of these methods, you will perform the bootstrap using two different models, linear and quadratic. The goal is to compare the two bootstrap approaches under the two models. The data are being generated from a quadratic model so in one case you are fitting the wrong model.

The simulation study is broken into 5 steps consisting of a total of 6 functions. (PART 6 is from the book.) Some of these functions are only 1 or 2 lines long.

PART 1. Generate a bootstrap sample
Write the *genBootY* and *genBootR* functions, to generate a bootstrap sample of (x, y) pairs. The xs will be the same as your data. To generate the ys to go with the xs, you will use one of the two methods described above. The *genBootY* function generates y values according to the first method. The *genBootR* generates them according to the second method.

PART 2. Fit a model
Write the $fitModel$ function that fits a line or a quadratic in x to y. Use the `lm()` function do get the coefficients. For each bootstrap sample, we will fit either the linear or quadratic to the bootstrapped (x, y) pairs.

PART 3. Generate a bootstrap sample and fit a "curve" to it.
Write the *oneBoot* function that considers the combinations of the two bootstrap approaches and the two models. That is, this function generates a bootstrap sample according to one of the two methods and fits either a line or a quadratic to the bootstrap sample x to y.

PART 4. Replicate
For each of the four different variations of the bootstrap procedure, replicate the results 1000 times. For each of the four variations, get a bootstrap 95% confidence interval for the linear coefficient for x.

PART 5. Visualize
This function makes a plot showing the data and the 1000 lines/curves (use transparency). In addition, you should also plot the true model in a contrasting color so that you can see how closely the bootstrapped models follow the true model.

You are given a seventh function, which runs your experiment by calling the the replicate function and then the plotting function. Include in your .R file a few comments that describe what you see. When is the variability smallest? Is there a problem with bias when the model being fitted is wrong? Is there a problem with basing the bootstrap on the data, i.e. are the data close to the truth? Are any problems you find worse for one method than the other?

PART 6. Ch 8, Example 4.
Do the bootstrap procedure following Example 4 from Ch 8. Make a bootstrap 95% confidence interval for c.

W is generated for you in the .R file. You generate the rest, with $a_1 = 1, a_2 = 2, b = 0.3, c = 4$, and ϵ generated according to G and K with $K_{11} = 1, K_{22} = 2, K_{12} = K_{21} = 0.5$. Let $Y_{0,1} = Y_{0,2} = 0$. Keep in mind as you do the bootstrap that as a data analyst you don't get to know the parameters or if you have the right model, but it's useful to do this kind of simulation to get a look at how good the bootstrap is at figuring out what the bias is when the model is correct. You probably want to go back to HW4 and recycle some of your code.

Make a histogram of $\hat{c}^*_{FGLS} - \hat{c}_{FGLS}$. Estimate the bias in $\hat{c}_{FGLS}$. How did the bootstrap do? Comment briefly.