STAT 133, SPRING 2015
HOMEWORK 8: RESAMPLING

Due Thursday, April 23 at 11:59pm.

The bootstrap is a method for approximating the variability of a statistic. You have seen in lectures that it works by drawing new data, not from the population, but from our original dataset or some transformation of our original dataset. The method then examines how the statistic varies when computed for each resampled dataset.

For this homework, you will use the bootstrap to estimate the amount of variability in the coefficient estimates from a linear and quadratic models, and to get a sense of potential bias. The `getData` function used to generate the data is given to you with a fixed seed—please leave the seed unchanged so that we can confirm your results.

You will implement two varieties of the bootstrap to generate replicates of the (x, y) pairs.

(1) For each unique x the `getData` function constructs 10 y values (i.e. there are ten observations at each x-value). For each x value, sample **with** replacement 10 times from these 10 y values. This will usually give you some duplicate points. This is called "case resampling."
(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 a randomly chosen residual. Sample the errors **without** replacement (i.e. permute them). This is called "residual resampling."

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 we are fitting the wrong model.

We have broken the simulation study into 5 steps consisting of a total of 6 functions. 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. The `genBootY` function generates y values using case resampling. The `genBootR` generates y values using residual resampling.

PART 2. Fit a model. Write the `fitModel` function that fits a line or a quadratic in x to y. Use the `lm` function in R to do this. This function takes a bootstrap sample in its `x` and `y` arguments and returns the coefficients of the fit. For each bootstrap sample, we will fit either the linear or quadratic to the bootstrapped (x, y) pairs. If the `degree` argument is 1 fit a linear model and if the `degree` argument is 2 fit a quadratic model.

PART 3. Generate a bootstrap sample and fit a "curve" to it. Write the `oneBoot` function. This function takes the arguments `data` and `fit`. The `data` argument is a matrix where the first column is the xs and the second column is the ys from `getData`. The `fit` argument is also a matrix, the first column of which is the fitted values (i.e. the estimated ys from the model constructed with `fitModel`) and the second column is the residuals. This is a wrapper function that calls `genBootY` if the `fit` argument is not given and calls `genBootX` if the fit argument is given.

PART 4. Replicate. For each of the different variations of the bootstrap procedure, replicate the results 1000 times.

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.

We have provided you with a seventh function `runSim`, which runs your experiment by calling the the replicate function and then the plotting function.

When you have completed your project review it and ponder these questions:

- 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?

Note, you do not need to provide an answer, just think about them. We will include a brief discussion in the solutions.