MATH 282B – Homework 4
Due Monday, 02/29/2016, by 11:59 PM

Send your code to `math282ucsd@gmail.com`*. Follow the following format exactly. For Homework 1, in subject line write "MATH 282B (HW 1)" and nothing else in the body. There should only be one file attached, named* `hw1-lastname-firstname.R`*. Make sure your code is clean, commented and running. Keep your code simple, using packages only if really necessary. If your code does not run, include an explanation of what is going on.*

Problem 1. (Comparing L_2 and L_1 regression via cross-validation) Consider the `04cars` dataset focusing on variables `mpg` (highway MPG) and `hp`. (Remove the incomplete cases.) Consider the following two predictors: the first one fits a degree 2 polynomial explaining `mpg` as a function of `hp` using least squares regression; the second one uses least absolute regression. Perform K-fold CV to compare these models with $K = 5$, $K = 10$, and then $K = n$ (n being the sample size). The latter is leave-one out CV. Make sure to randomly permute the observations beforehand. What model would you choose based on that?

Problem 2. (Backward selection) As we saw in lecture, the function `mle.stepwise` with the backward option seems flawed. Write your own function for performing backward selection. Call your function `backwardSelect(X, y)`, with `X` being the predictor matrix and `y` the response vector. To avoid dealing with an intercept, in the function, start by centering all the variables. Try your function on the `Boston` dataset in the package `MASS`. Let the full model be the simple linear model explaining `medv` as a function of the other *numerical* variables. (Make sure that the categorical variables are considered as such.)

(It would be better that the function takes in a formula, as `lm` does, and treats each variable according to its type. If you can pull this off, great!)

Problem 3. (Estimating the prediction error by subsampling) This variant was mentioned in lecture. It's somewhere between cross-validation and the leave-one-out bootstrap. The method proceeds as follows. Sample a proportion p of observations without replacement. This plays the role of training set and the remaining observations play the role of validation set, and the result is an estimate of the prediction error. The whole process is repeated B times and the final estimate is the average of the B estimates. Write a function `subsampleSelect(X, y, Fit, p = 0.5, B = 99)` where `X` and `y` are as before and `Fit` is a function that takes observations and returns a function that estimates the regression function.[1] Test your function `subsampleSelect` in the same way that you tested your `backwardSelect` function.

Note. This problem is harder and you are allowed to form teams of size up to 3. The names of the team members need to appear in your code at the beginning of your solution to the problem.

[1] In more detail, `Fit` is of the form `Fit(X.train, y.train, X.new)` where `X.train` and `y.train` are paired and represent a training set, and `X.new` is matrix of new predictor observations with the same characteristics as `X.train`. The function trains a model on (`X.train`, `y.train`) and computes a prediction on `X.new`. Even better, you can have `Fit` is of the form `Fit(X.train, y.train)` return a function (the trained predictor) that will then be applied to the part of the data playing the role of validation.