Statistics 216
Homework 4, due Wednesday March 12, 2014.

1. **You may work in groups up to size 4 on this problem. If you do work in groups, write the names of all your group members on your problem set.**

   Recall the `body` dataset from problem 3 of Homework 3. In that problem we used PCR and PLSR to predict someone's weight. Here we will re-visit this objective, using bagging and random forests. Start by setting aside 200 observations from your dataset to act as a test set, using the remaining 307 as a training set. Ideally, you would be able to use your code from Homework 3 to select the same test set as you did on that problem.

   Using the `randomForest` package in `R` (hint: see section 8.3.3 in the textbook for guidance), use Bagging and Random Forests to predict the weights in the test set, so that you have two sets of predictions. Then answer the following questions:

   (a) Produce a plot of test MSE (as in Figure 8.8 in the text) as a function of number of trees for Bagging and Random Forests. You should produce one plot with two curves: one corresponding to Bagging and the other to Random Forests. *Hint: If you read the documentation for the* `randomForest()` *function, you can find a way to obtain the data for both curves with only one call each to the* `randomForest()` *function.*

   (b) Which variables does your random forest identify as most important? How do they compare with the most important variables as identified by Bagging?

   (c) Compare the test error of your random forest (with 500 trees) against the test errors of the three methods you evaluated in problem 3(f) on Homework 3. Does your random forest make better predictions than your predictions from Homework 3?

   If you did not successfully solve problem 3(f) on Homework 3, you may compare the test error of your random forest against the test errors in the Homework 3 solutions.

   (d) The `randomForest()` function uses 500 as the default number of trees. For this problem, is 500 enough trees? How can you tell?

2. Here we explore the maximal margin classifier on a toy data set.

   (a) We are given $n = 7$ observations in $p = 2$ dimensions. For each observation, there is an associated class label. Sketch the observations.

Obs.	X_1	X_2	Y
1	3	4	Red
2	2	2	Red
3	4	4	Red
4	1	4	Red
5	2	1	Blue
6	4	3	Blue
7	4	1	Blue

   (b) Sketch the optimal separating hyperplane, and provide the equation for this hyperplane (of the form $\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$).

   (c) Describe the classification rule for the maximal margin classifier. It should be something along the lines of "Classify to Red if $\beta_0 + \beta_1 X_1 + \beta_2 X_2 > 0$, and classify to Blue otherwise." Provide the values for β_0, β_1, and β_2.

   (d) On your sketch, indicate the margin for the maximal margin hyperplane. How wide is the margin?

   (e) Indicate the support vectors for the maximal margin classifier.

   (f) Argue that a slight movement of the seventh observation would not affect the maximal margin hyperplane.

(g) Sketch a hyperplane that is *not* the optimal separating hyperplane, and provide the equation for this hyperplane.

(h) Draw an additional observation on the plot so that the two classes are no longer separable by a hyperplane.

3. This problem involves the `OJ` data set which is part of the `ISLR` package.

(a) Create a training set containing a random sample of 800 observations, and a test set containing the remaining observations.

(b) Fit a support vector classifier to the training data using `cost=0.01`, with `Purchase` as the response and the other variables as predictors. Use the `summary()` function to produce summary statistics about the SVM, and describe the results obtained.

(c) What are the training and test error rates?

(d) Use the `tune()` function to select an optimal `cost`. Consider values in the range 0.01 to 10.

(e) Compute the training and test error rates using this new value for `cost`.

(f) Repeat parts (b) through (e) using a support vector machine with a radial kernel. Use the default value for `gamma`.

(g) Repeat parts (b) through (e) using a support vector machine with a polynomial kernel. Set `degree=2` for part (b).

(h) Overall, which approach seems to give the best results on this data?

4. Consider a dataset with n observations, $x_i \in \mathbb{R}^p$ for $i = 1, ..., n$. In this problem we show that the K-means algorithm is guaranteed to converge but not necessarily to the globally optimal solution.

(a) At the beginning of each iteration of the K-means algorithm, we have K clusters $C_1, ..., C_K \in \mathbb{R}^p$, and each data point is assigned to the cluster with the nearest centroid (at this point, the centroids are not necessarily equal to the mean of the data points assigned to the cluster).
Let $\bar{x}_{kj} = \frac{1}{|C_k|}\sum_{i \in C_k} x_{ij}$ denote the mean for feature j in cluster C_k. Show that

$$\frac{1}{|C_k|}\sum_{i,i' \in C_k}\sum_{j=1}^{p}(x_{ij} - x_{i'j})^2 = 2\sum_{i \in C_k}\sum_{j=1}^{p}(x_{ij} - \bar{x}_{kj})^2$$

(b) Define

$$W(C_k) \equiv \frac{1}{|C_k|}\sum_{i,i' \in C_k}\sum_{j=1}^{p}(x_{ij} - x_{i'j})^2$$

Show that $\sum_{k=1}^{K} W(C_k)$ decreases with each successive iteration of the K-means algorithm.

(c) Show that the K-means algorithm is guaranteed to converge.

(d) Give, as an example, a toy data set and a pair of initial centroids for which the 2-means algorithm does not converge to the globally optimal $\min_{C_1,C_2}\{W(C_1) + W(C_2)\}$.