IOM 530: Applied Modern Statistical Learning Methods
Assignment 2 (Due 9/19/2008)

Guidelines for assignment submission:

1. Type each question before you answer it, and provide a clear separation between each part.
2. All relevant computer output should be provided unless noted otherwise.
3. Print your homework, and submit it at the beginning of the class. Make sure that it is stapled, and your name is typed on it.
4. Attach your R code as an Appendix. Make sure to provide comments on what your code is doing. Keep it clean and clear!

5. Note the main aim of this homework is to get practice doing linear regression in R. Hence you shouldn't restrict yourself to only doing specifically what is asked. Anything else you might want to do to build a better linear regression model would be welcome. Questions 1 and 2 should either be answered using the auto data set from lab 1 or, if you are feeling bored with this data, using a data set you have gathered. Some good sources for interesting data sets are
   http://www.econ-datalinks.org/search.html
   http://fisher.osu.edu/cgi-bin/DB_Search/db_search.cgi?setup_file=finance.setup.cgi
   http://fisher.osu.edu/fin/fdf/osudata.htm
   http://www.census.gov/epcd/www/recent.htm
   http://www.bized.ac.uk/dataserv/freedata.htm

   Warning: These websites are fascinating, with thousands of possible data sets to explore. You may find yourself ignoring your loved ones and your other classes just so that you can sneak back and spend more time exploring. This is unfortunately one of the dangers of taking a statistics class and is unavoidable! Seriously, real data sets often have real problems (that have nothing to do with R) associated with them. If you run into problems you may find it easier to work with the auto data.

6. You can refer to Section 3.6 (Lab: Linear Regression) in the course book and the facebook group for this class if you need some help with the R code necessary to finish this assignment.

IOM 530: Applied Modern Statistical Learning Methods
Assignment 2 (Due 9/19/2008)

1. **Simple Linear Regression**

    a. Use the *lm* function to perform a simple linear regression between the response and one of the continuous predictors. Use the *summary* function to print the results. Comment on what the output tells you e.g. is there a relationship, how strong is it etc.
    b. Plot the response and predictor. Include the estimated regression line from a. (use *abline* to do this).
    c. Plot the residuals vs. the predictor. Comment on any problems you see with the fit. Hint: Try *plot(lm.fit)*.
    d. Produce a normal quantile plot using the standardized residuals. Are there any indications of non-normality? Try *plot(lm.fit)*.

2. **Multiple Linear Regression**

    a. Produce a scatterplot matrix, which includes the response and the predictors. If there are too many predictors to plot just select a reasonable subset of them.
    b. Compute the matrix of correlations between the variables. Again, if there are too many predictors just select a reasonable subset of them. Note you will need to exclude any categorical variables. Hint: Try: *cor(matrix_name)*
    c. Use the *lm* function to perform a multiple linear regression (if you have a large number of predictors (i.e. dozens) try to remove the ones that don't look important and just report the results on the rest). Use the *summary* function to print the results. Comment on what the output tells you e.g. is there a relationship, which variables appear to be important, how strong is the relationship etc.
    d. Use the * and : options to produce some interaction effects. Can you see any interactions that appear to be statistically significant?
    e. Try a few different transformations of some of the variables e.g. I(x^2), log(x), sqrt(x) etc. Make sure to hand in the output so we can see which ones you tried. (Hint: Check section 3.6.5 in the course book, page 115).

3. **R Functions**

    Write an R function (call it print_seq) that prints a sequence of numbers from 1 up to n. This function will take "n" as its input and produces the sequence 1, 2, 3, 4, …, n.

    Input:

    print_seq(6)

    Output:

    The sequence is: 1,2,3,4,5,6