

1. Sweep

Write and test a subroutine to sweep the columns of a square matrix. The subroutine should be written so that the calling routine can specify which columns of the matrix are to be swept. This would be necessary, for example, in stepwise regression. Test it on an a matrix of your choice, and try to verify the results through some other means. Make sure you read and understand part 2 of the assignment before writing the subroutine, and you can save yourself some work.

Hint: One way to test the function is to sweep *all* the columns of the matrix and compare the result to the inverse of the matrix obtained from R, matlab, etc. You can also form a test matrix as an augmented $X'X$ matrix, and use your subroutine to perform a series of regressions, checking the results with a package or subroutine library, or the Gram-Schmidt regression program you wrote for assignment 3.

2. Writing Functions in R

Write a function in R which will call the sweep subroutine from part 1 of this assignment. Name the R function “`bsweep`” to avoid conflict with the R function “`sweep`” which performs a different operation on a matrix. There should be two arguments to the R function; the matrix to be swept, and a vector of indices of the columns which are to be swept. Verify that the function works by performing a regression with it. Don’t forget that R stores it’s matrices by columns, not by rows.

3. Non-linear regression

Write a program in C to perform non-linear regression. The core of your program should be a function which is passed the data for the regression and the name of functions which will evaluate the non-linear function being fit and its gradient. (Make sure to write your function so you can change the routines which will be evaluating the regression function and the gradient through the argument list.) Use the Gauss-Newton algorithm with capability for step halving when necessary, and notice that there are a variety of ways to determine the iterative step directions: Cholesky decomposition of $J(\theta)'J(\theta)$ followed by solution of a triangular system of equations or regression of $f(\theta)$ on $J(\theta)$ to name two. Test your program on the two data sets contained in the files `prob1` and `prob2` in the `s243` samples directory. A description of the content of these files is in the files `prob1.des` and `prob2.des` in the same directory (`~s243/samples`). If you have the time and inclination, you can verify your results using R, SAS, or any other method you know of.

4. EXTRA CREDIT I: R Function for Stepwise Regression

Write an R function to perform backward deletion using the `bsweep` function which you wrote in part 1. The function should have an option for inclusion of an intercept, and should return a logical matrix which has one row for each model investigated, and a column for each variable containing either `TRUE` or `FALSE`, depending on whether or not each variable is involved in that regression. How would you change the design of the function in order to make it run more quickly?

5. EXTRA CREDIT II: R Function for Non-Linear Regression

Write an R interface to the non-linear regression program written in part 3. The interface should allow you to define the non-linear function and its gradient as functions in R which

will be passed to your program using the `call_R` function. (If you choose to do this part of the assignment and run into problems, don't hesitate to ask questions!)