This assignment is due by 4pm on Friday, September 16th. Minimal credit will be given for incomplete solutions or solutions that do not provide details on how the solution is found. You are encouraged to discuss the problems with your classmates, but all work (analysis and code) must be your own.

1. In this problem you will construct a space-efficient least-squares solver for solving $A\mathbf{x} = \mathbf{b}$ where A is full-rank and test your method on a data-fitting problem. Recall that for the full-rank case, the general procedure for finding the least-squares solution was

   - Compute reduced QR factorization $A = \hat{Q}\hat{R}$
   - Form $\hat{\mathbf{c}} = \hat{Q}^T\mathbf{b}$
   - Solve $\hat{R}\mathbf{x} = \hat{\mathbf{c}}$

   (a) Write a function `myQR()` that performs the space-efficient implementation of Householder QR that we discussed in class. Your function should take an $m \times n$ matrix A and return a vector γ of the multipliers involved in $Q = I - \gamma_k \mathbf{u}_k \mathbf{u}_k^T$ and the matrix A overwritten by the $\mathbf{u}_k$'s and the entries of $\hat{R}$. How you return the overwritten A will depend on the language you're using. Since Julia is pass-by-reference it's easy to explicitly overwrite A in memory. This is much more difficult to do with Matlab, so feel free to just return a matrix representing the overwritten A.

   (b) Write a function `myQb()` that takes the vector of γ's, your overwritten A matrix, and the right-hand side vector $\mathbf{b}$ and returns the product $\hat{Q}^T\mathbf{b}$. Your function should do the multiplication implicitly using the $\mathbf{u}_k$'s and γ_k's. Looking at Algorithm 10.2 in your textbook and the surrounding discussion is a good starting point, but you will have to modify it slightly to utilize the γ_k's.

   (c) Write a function `myLSsolve()` that takes A and $\mathbf{b}$ as arguments and returns the solution `x`. Your function should call `myQR()` to do the QR factorization, call `myQb()` to form $\hat{\mathbf{c}}$, then extract $\hat{R}$ and perform the required upper-triangular system solve for `x`. Don't worry about doing anything fancy for the solve. Just use the backslash command in Matlab or Julia.

   (d) Use your newly-built solver to fit linear, quadratic, and degree-5 polynomials to the data given in `auto.csv`. The file contains a list of points of the form (t_i, b_i) where t_i and b_i are the horsepower and gas mileage of particular cars, respectively. Consult your class notes or Examples 11.1 and 11.2 to see how to build the data matrix A. Make a plot of the data (with `horsepower` on the horizontal axis and `mpg` on the vertical axis) along with the three polynomial models you fit. Also, give the least-squares error ($E = \|\mathbf{b} - A\mathbf{x}\|_2^2$) for each of the three models.

2. (a) Let $\mathbf{u} \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, and consider the matrix $Q = I - \gamma \mathbf{u}\mathbf{u}^T$. Show that the product QA can be computed in approximately $4mn$ flops if the arithmetic is done in an efficient way.

   (b) Use the result of part (a) to estimate the leading order term (including any relevant coefficients) in the Householder QR algorithm you implement in Problem 1 above.