How To Use FitNonlinSR.m

1. Verify that FitNonlinSR.m is the appropriate template file to use
   a. The data points must be of the form $\left(x_1, x_2, \cdots, x_{n_s}, \hat{y}\right)$
   b. The model being fit to those data must be of the form $y = f\left(x_1, x_2, \cdots, x_{n_s}, \theta_1, \theta_2, \cdots, \theta_{n_p}\right)$
2. Save a copy of FitNonlinSR.m as *newname*.m in the current MATLAB working directory or in a directory that is in the MATLAB search path ("*newname*" should be some meaningful file name)
3. Change the function declaration statement to match the filename from step 2
   a. from: `function FitNonlinSR(p_guess)`
   b. to: `function newname(p_guess)`
4. Find the comment indicating the location of the first required file modification and replace it with statements define a variable for each constant provided in the problem statement and setting its value. Universal constants like the ideal gas constant should be defined here, as well, and all the values should have consistent units.
5. Find the comment indicating the location of the second required modification and replace it with a statement defining a matrix named `x`
   a. There should be one row in the matrix `x` for each data point in the data set being fit
   b. There should be one column in the matrix `x` for each set variable in the data set
   c. The matrix `x` should contain the values of the corresponding set variables and data points
6. Find the comment indicating the location of the third required modification and replace it with a statement defining a column vector named `y_hat`
   a. There should be one row in the `y_hat` for each data point in the data set being fit and it should contain the corresponding value of the measured response for that data point
7. Find the comment indicating the location of the fourth required modification and change the line that follows the comment
   a. from: `y(i) = ; % insert statement(s) to calculate y for data point i`
   b. so that it evaluates the function f in step 1b and sets `y(i)` equal to the result
   c. If the parameters or set variables are needed in order to evaluate the function, f, they are available in the column vector `p` and the matrix `x`, respectively (i. e. for θ_1 use `p(1)`, for θ_2 use `p(2)`, etc., and for x_1 use `x(i,1)`, for x_2 use `x(i,2)`, etc.)
8. Save the modified version of newname.m (where newname is the filename chosen in step 2)
9. Create a column vector named `p_guess` in the MATLAB workspace; it should contain guesses for the values of the parameters, $\underline{\theta}$, one per row
10. Execute the file by typing the following at the MATLAB command prompt (again using "newname" to represent the filename chosen in step 2): `newname(p_guess)`

Copyright © 2013, Carl R. F. Lund

11. The following quantity will be listed in the MATLAB command window
    a. `r_squared` - the correlation coefficient for the fit
12. The following quantities will be returned
    a. `pf` - a column vector containing the fitted parameters
    b. `pf_u` - a column vector containing the ± 95% confidence limits for the fitted parameters
    c. `y` - a column vector containing the values of the response variable predicted by the fitted model
13. The following figures will be displayed
    a. If there is one set variable per data point
        i. A model plot
    b. If there are two or more set variables per data point
        i. A parity plot
        ii. A set of residuals plots with each of the set variables as the abscissa
14. Copy the values of `pf` to `p_guess` and repeat step 10
    a. Repeat this step until the values returned as `pf` equal the values in `p_guess` and none of the other returned quantities have changed indicating a converged minimization
15. To search for a different minimum of the objective function, repeat steps 9 through 14 using a significantly different `p_guess` in step 9