DUE: Februrary 4, 2016

1. For $(y_1, y_2, y_3) = (0.001, 2.6, 14.6)$ and $(x_1, x_2, x_3) = (0.05, 1.1, 2.2)$, we wish to fit the model:
$$Y_i = \frac{\theta_1}{\theta_2} x_i^{\theta_2} + \epsilon_i, \quad i = 1, 2, 3,$$
where $\epsilon_i \sim iid N(0, \sigma^2)$.

   (a) Estimate the parameters θ_1, θ_2 by a grid search using the candidate values:
$$\theta_1 = \{4.9, 4.95, 5.0, 5.05, 5.1, 5.15\}$$
$$\theta_2 = \{2.3, 2.4, 2.5, 2.6, 2.7\}$$
(NOTE: You should use a computer program such as R, MATLAB, or a spreadsheet to do the calculations.) Report your estimate of $SSE(\theta_1, \theta_2)$ for each candidate set and note your "best" estimates.

   (b) Obtain reasonable starting values for the above model by assuming multiplicative errors and taking logs of both sides.

   (c) Using starting values of $\hat{\theta}_1^{(0)} = 4.9$ and $\hat{\theta}_2^{(0)} = 2.4$, estimate $\hat{\theta}_1^{(1)}$ and $\hat{\theta}_2^{(1)}$ by using the Gauss-Newton procedure. Do not use an automated statistics computer package to do this (you can use a program such as R, MATLAB, or Excel to do the intermediate matrix calculations). Show your work (specifically, show your values for $\mathbf{F}(\hat{\boldsymbol{\theta}}^{(0)})$, $\mathbf{f}(\hat{\boldsymbol{\theta}}^{(0)})$, $[\mathbf{F}'(\hat{\boldsymbol{\theta}}^{(0)})\mathbf{F}(\hat{\boldsymbol{\theta}}^{(0)})]^{-1}$, $\hat{\boldsymbol{\delta}}^{(1)}$, and $\hat{\boldsymbol{\theta}}^{(1)}$.)

   (d) Using the same starting values as in (c), use the steepest descent algorithm (with $\alpha_j = 1$) to get $\hat{\theta}_1^{(1)}$ and $\hat{\theta}_2^{(1)}$.

   (e) Using the same starting values as in (c), use the Newton-Raphson algorithm (with $\alpha_j = 1$) to get $\hat{\theta}_1^{(1)}$ and $\hat{\theta}_2^{(1)}$.

   (f) Assuming that after convergence you obtain estimates $(\hat{\theta}_1, \hat{\theta}_2) = (5.0, 2.5)$ and
$$[\mathbf{F}'(\hat{\boldsymbol{\theta}})\mathbf{F}(\hat{\boldsymbol{\theta}})]^{-1} = \begin{pmatrix} 1.2423 & -0.6122 \\ -0.6122 & 0.3332 \end{pmatrix}.$$
Now, further assume that you have sufficient number of observations so that the asymptotic assumptions hold.

      i. Find an approximate 95% C.I. for θ_1.
      ii. Find an approximate 95% C.I. and 95% P.I. for Y_i given $x_i = 1.0$.
      iii. Test the null hypothesis that $\theta_1/(\theta_2)^2 = 0.9$ versus the two sided alternative at $\alpha = 0.05$.
      iv. What is the value of R^2_{pseudo}?
      v. Do any of the observations exhibit substantial leverage?

2. Consider the nonliinear regression model,

$$y_i = \frac{\theta_0 + \theta_1 x_i}{1 + \theta_2 \exp(0.4x_i)} + \epsilon_i, \quad i = 1, \ldots, n,$$

where $\epsilon_i \sim N(0, \sigma^2)$. The data set `S16hw1pr2.dat` contains 250 observations (the first column is x and the second column is y) that we wish to fit to this model to do estimation and inference on the parameters as described below. You are to do this problem by writing your own modified Gauss-Newton algorithm! [You are NOT allowed to use SAS other than to check your results if you like.] You should attach your R program as an Appendix to your HW writeup. Note, you are required to do your own work on this question - your code should not look exactly that from one of your classmates! Note that don't need to use R's ability to do automatic derivatives, but if you want to, the following commands at the end of this homework helpful.

(a) Give the starting values you select and describe how you obtained them.

(b) Construct your own modified **Gauss Newton** algorithm in R. NOTE: this means you are programming the algorithm yourself, not using a predefined function in R!! Find the parameter estimates for the θs and their asymptotic variance covariance matrix, as well as σ^2. Report these values along with a summary of how many iterations it took to converge, as well as the objective function value at convergence. Discuss any particular issues you encountered in solving this problem and what you did to solve those issues. Include your R code as an Appendix to your homework.

(c) Extra Credit (optional): Implement your own **Newton-Raphson algorithm** and report the same things as for the modified Gauss Newton algorithm (and, include your R code in the Appendix).

R commands to do derivatives of a defined function: (thanks to John Snyder!)

Taking derivatives in R involves using the ?expression,? ?D,? and ?eval? functions. You wrap the function you want to take the derivative of in expression(), then use D, then eval as follows.

```
> f=expression(sqrt(x))
> f
expression(sqrt(x))
> df.dx=D(f,'x')
> df.dx
0.5 * x^-0.5
> d2f.dx2=D(D(f,'x'),'x')
> d2f.dx2
0.5 * (-0.5 * x^-1.5)

> eval(f,list(x=3))
[1] 1.732051
> eval(df.dx,list(x=3))
[1] 0.2886751
> eval(d2f.dx2,list(x=3))
[1] -0.04811252
```

The first argument passed to eval is the expression you want to evaluate, and the second is a list containing the values of all quantities that are not defined elsewhere.