The exercises here all concern regression

I will illustrate possibilities using selected columns from the data set UScereals (MASS library), after having deleted the first and last variable, which leaves 9 variables; I call it **`cereal9`**. You are welcome to choose another data set that interests you, but be sure it is neither 'too small or too large'. I.e., if k = no. of variables and n = sample size, I suggest that you keep k between 3 and 6, n between 30 and 100. I call the data set **`xx`** below. Then decide what predictive issues seem to you to make sense for the data you choose; i.e. what is the criterion or response variable, and what are the predictors, at least 2 to be named?

You will see that the steps or parts move from the simple to the complex, so most of them should be done in the order I've specified. While numerical results will be essential, I'd like you to emphasize graphics (where I will supply various function names to help in the enterprise, but I hope you will find other functions that those I name; there will be several to choose from, and you have good sources to help.). Pay especially close attention to the interpretive parts of all of this, and ask questions if you have them. (I will ask for presentations from one or two of you on Thursday, where some of what you do can be open-ended, leading to discussion of your data.)

1. Given the initial (possibly larger than working) data set, summarize it.

   a. The **`my.summary`** function may work for you, but also consider **`describe`** (Hmisc library) or even **`apply(data,2,summary),`** where **`summary`** is a standard function in R.

   b. Show results for all variates graphically (recall **`datadensity, Hmisc`** lib.); but histograms might also be used (if there are 4 variables, then begin w/ **`par(mfrow=c(2,2));`** then execute **`histogram(xx[,j], nint= 10[intervals])`**

   c. Then summarize to show bivariate relationships, first numerically, then graphically:

      (i) **`cor(xx)`** and/or (ii) **`cov(xx)`**

      (iii) **`pairs`** and/or **`scatterplotMatrix`** [car library] and/or **`gpairs`** [YaletoolKit]

   d. Write several sentences describing what a, b and c results tell you. Details are important,

      so if, for example, you see 'pronounced' skew (for predictors or criteria) explain what this might mean for ensuing use of regression work.

2. Linear regression.

   a. Conduct at least two linear regressions using only one predictor for each. You may have gotten some results from c.(iii) that bear on these, but do separate regressions here and then focus on details. You might use functions **`lm`** or **`lsfit`** (see help); but again, there are other options.

      (i) *In each analysis provide* the linear regression equation; the *correlation* between x and y; the *squared correlation*; the *residual sum of squares*; and *any other summary statistics* you want to include (including t-statistics or c.i.s). (Note that if you create a '**`lm`** object' that the function **`confint`** can be used w/ this object to generate confidence intervals for the regression coefficients.) Remember, do all of these, i. and ii. Etc, for each regression run.

      (ii) Plot these data for regression. E.g., **`plot(x,y, pch=16)`** [16 makes points dark]. Then overlay the plot w/ the OLS regression line: **`abline(lm.object)`** will do the trick.

      (iii) Interpret what these (two) sets of regression results tell you. Elaborate as seems helpful.

3. Nonparametric regression.

a. Carry out loess regression for at least one of your x, y combinations (one you used in 2.i above). See **`?loess`** for details about how to set this up. Be sure to create a **`loess`** object, and use summary to see its contents. Try at least two or three values of the arg **`span`** to see what seems to you most sensible. Note that the **`predict`** function can be used to acquire predicted values of y for each choice of x (can also be used for lm objects, above). Plot the loess curve on top of the point plot and compare this non-parametric result with the linear version above. (Note that you can correlate **`predict`**(obj) values with y values for both lm and loess objects.) And you can find errors of prediction using y – predict(obj). (Plots of fitted values vs. errors is commonly recommended in regression. We shall discuss why in class.)

b. Compare the results from 3.a with the corresponding results from 2. What are the pros and cons of each, relative to the other?

c. Now try using two predictors w/ the **`loess`** function. Ask if you need help with this. How much does the second variable add to the first? Can you generate a 3-dimensional plot that shows graphically what the two-dimensional surface looks like. (Hint: John Fox has discussed how to do this for prestige data using his **`scatter3d`** function. See the help for it; it is in the **`car`** package.)

d. Summarize all main numerical results and interpret all findings. What have you learned?