ACCESS - MATH

July 2004

Notes on Body Mass Index and actual national data

What is the Body Mass Index?

If you read newspapers and magazines it is likely that once or twice a year you run across an article about the body mass index (B.M.I.) , and its use in determining health risk factors for overweight and underweight people. If you search the internet for "body mass index" you will find many sites which let you compute your B.M.I., and which tell you a little bit about it.

A person's B.M.I. is computed by dividing their weight by the square of their height, and then multiplying by a universal constant. If you measure weight in kilograms, and height in meters, this constant is the number one. Thus, the proponants of the B.M.I index are claiming that for adults at equal risk levels (but different heights), weight should be proportional to the square of height. As we have discussed in class, if people were to scale equally in all directions ("self-similar") when they grew, volume and hence weight would scale as the cube of height.   That particular power law seems a little high, since adults don't look like uniformly expanded versions of babies; we seem to get relatively stretched out when we grow taller. One might expect the best predictive power for weight as a function of height to be somewhere between 2 and 3, if one expected a power law at all. If there is a predictive power, and if it is much larger than 2, then one could argue that the body mass index might need to be modified to reflect this fact. (In fact, when you find body mass index tables, they often explain that you should modifiy the acceptable BMI values for children.)

Most of us have collected several heights and weights, and hopefully in aggregate we will have a good number of representative measurements, from baby-sized to adult. (Actually, I added 21 extra data points for baby size through 49 inches in height - this data came from earlier ACCESS groups and helps give us a more balanced selection of heights.) Each group will use this data, see if it is consistent with a power law relating weights to heights, and decide whether the B.M.I. power of 2 is a good choice.

I often do this experiment with my linear algebra classes as well as with the Accessors, and we have gotten powers between 2.3 and 2.7. Also, several years ago I found a national data base at the U.S. Center for Disease Control web site. It contained a wide variety of body measurements collected between 1976 and 1980, including national median heights and weights for boys and girls, age 2-19. By using only the national medians a lot of the variance has been taken out of the data, compared to what yours will look like. The national data is very consistent with a power law, with power = 2.6. If any of you can figure out a valid mathematical model which explains this power law (or one close to it), you'll have a publishable paper.

How do you test for power laws?

When you studied logarithms in high school you might have wondered what they were good for. Well, as both Fred and Ken showed you this week, one important application is in looking for power laws. Remember how the discussion goes:

Suppose we have a set of "n" data points, which you can think of as your height-weight data, but which could really be any set of paired data:

$$[[x_1, y_1], [x_2, y_2], [x_3, y_3], ..., [x_n, y_n]]$$

We want to see if there is a power m and a proportionality constant b so that the formula

$$y = b\, x^m$$

effectively mirrors the real data. Taking (natural) logarithms of the proposed power law yields

$$\ln(y) = \ln(b) + m \ln(x)$$

So if we write $Y = \ln(y)$ and $X = \ln(x)$, $B = \ln(b)$, this becomes the equation of a line in the new variables X and Y:

$$Y = mX + B$$

Thus, in order for there to be a power law for the original data, the ln-ln data should (approximately) satisfy the equation of a line. Furthermore, this process is reversible; if the ln-ln data lies on a line with slope m and intercept B, then the original data satisfies a power law with power m and proportionality constant $b = \mathbf{e}^B$. That's because of the rules of exponents:

$$Y = B + m\,X$$
$$\mathbf{e}^Y = \mathbf{e}^{(B + m\,X)}$$
$$\mathbf{e}^Y = \mathbf{e}^B \,(\mathbf{e}^X)^m$$
$$y = b\,x^m$$

In real experiments, it is not too hard to see if data is well approximated by a line, so this trick with the logarithm is quite useful.

National data example: I used colons after most of these commands to suppress the output. If you want to go back and see what each command is doing, replace the colons with semicolons.

```
> restart:
> with(plots):
 Warning, the name changecoords has been redefined

> boyhw:=[[35.9,29.8],[38.9,34.1],[41.9,38.8],[44.3,42.8],
         [47.2,48.6],[49.6,54.8],[51.4,60.8],[53.6,66.5],
         [55.7,76.8],[57.3,82.3],[59.8,93.8],[62.8,106.8],
         [66.0,124.3],[67.3,132.6],[68.4,142.4],[68.9,145.1],
         [69.6,155.3],[69.6,153.2]]:
   #boy heights (inches) weights (pounds): Ntl medians for ages 2-19
> girlhw:=[[35.4,28.0],[38.4,32.6],[41.1,36.8],[43.9,41.8],
         [46.6,47.0],[48.9,52.5],[51.4,60.8],[53.1,65.5],
         [55.7,76.1],[58.2,89.0],[61.0,100.1],[62.6,108.1],
         [63.3,117.1],[64.2,117.6],[64.3,122.6],[64.2,128.8],
         [64.1,124.5],[64.5,126]]:
   #girl heights(inches) weights (pounds): Ntl medians for ages 2-19
> boys:=pointplot(boyhw):
  girls:=pointplot(girlhw):
  display({boys,girls},
      title='plot of [height,weight], National medians ages 2-19');
```

plot of [height,weight], National medians ages 2–19
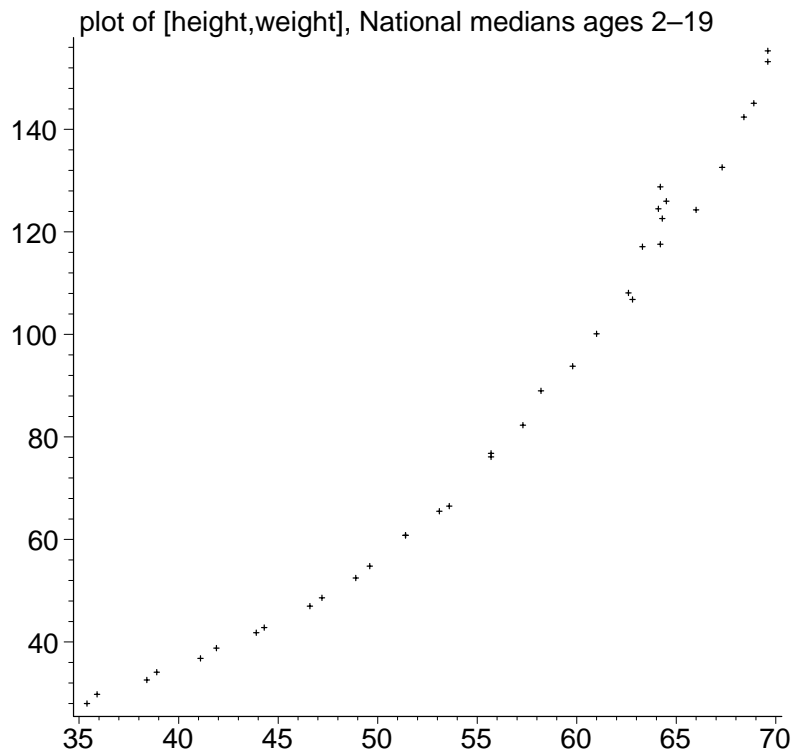
And now for the ln-ln data.

```
> with(linalg):   #linear algebra package

Warning, the protected names norm and trace have been redefined and unprotected

> B:=evalm(boyhw):   #evaluate matrix, this will
      #turn our list of points into a matrix, which will
      #be easiser to manipulate in Maple
  G:=evalm(girlhw):
  BG:=stackmatrix(B,G):   #stack the matrices on top of eachother.

> lnBGa:=map(ln,BG):   #take ln of the boy and girl height-weights
  lnBG:=map(evalf,lnBGa):   #Get decimal (floating point) values.
    #This speeds up computations later in the
    #least squares fit - otherwise Maple tries working
    #symbolically.
> lnlnplot:=pointplot(lnBG):
  display(lnlnplot,title=`ln-ln data`);
```

<figure>
ln-ln data
</figure>
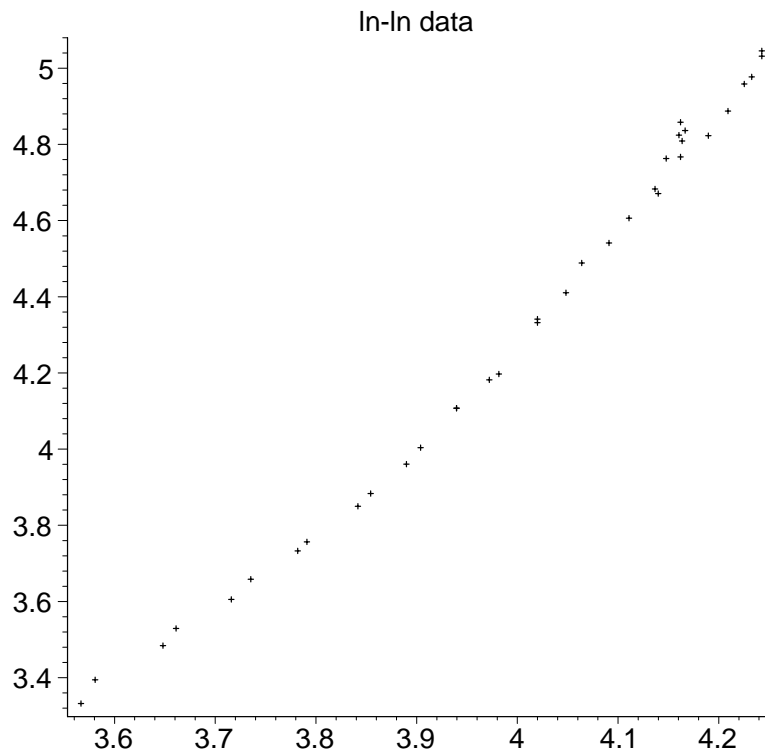
How do I find the best line fit to a collection of points?

From Calculus, there is a slope *m* and intercept *B* yielding a line which minimizes the sum of the squared vertical distances between your data points and the points on a line. If the n data points are

$$[[X_1, Y_1], [X_2, Y_2], [X_3, Y_3], ..., [X_n, Y_n]]$$

then *m* and *B* solve the system of equations

$$\begin{bmatrix} \sum_{i=1}^{n} X_i^2 & \sum_{i=1}^{n} X_i \\ \sum_{i=1}^{n} X_i & n \end{bmatrix} \begin{bmatrix} m \\ B \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} X_i Y_i \\ \sum_{i=1}^{n} Y_i \end{bmatrix}$$

You could solve this system with several "do-loops" to compute the matrix entries, followed by a "solve" command to solve the system, but the method is so common that every decent mathematical software or graphing calculator already has a command to do all of that work for you. In statistics this procedure is called linear regression as well as the method of least squares. If you looked through the help directory in your menu bar you would eventually find MAPLE's version of this command living in the stats library packag, and called "fit[leastsquare]". Here's how the command works. I've used it to check the example we worked by hand in class. You must be careful with brackets and parantheses.
```
> with(stats):
```

```
> fit[leastsquare[[X,Y]]]([[0,2,4],[1,2,1]]);#the syntax here is
  #to first name your variables, then give two lists, one of the
  first variable
  #values, and the second with the corresponding second variable
  values.  Thus
  #we are trying to find the best line fit for the points
  [0,1],[2,2],[4,1].
```

$$Y = \frac{4}{3}$$

Now that we've tested the command, we can use it on the national data:

```
> Xs:=convert(col(lnBG,1),list): #convert the first column of
     #the ln-ln data into a list of the "x's"  The least squares
     #command wants to have lists input, not matrix columns, even
     #though it's hard for us to see any difference
  Ys:=convert(col(lnBG,2),list):
> fit[leastsquare[[X,Y]]]([Xs,Ys]);
```

$$Y = -6.038703303 + 2.593828004\,X$$

We can paste in the equation of the line and see how well we did.

```
> line:=plot(-6.038703303+2.593828004*X, X=3.4..4.5,Y=2.7..5.7,
     color=black):
  display({line,lnlnplot}, title='least squares fit');
```
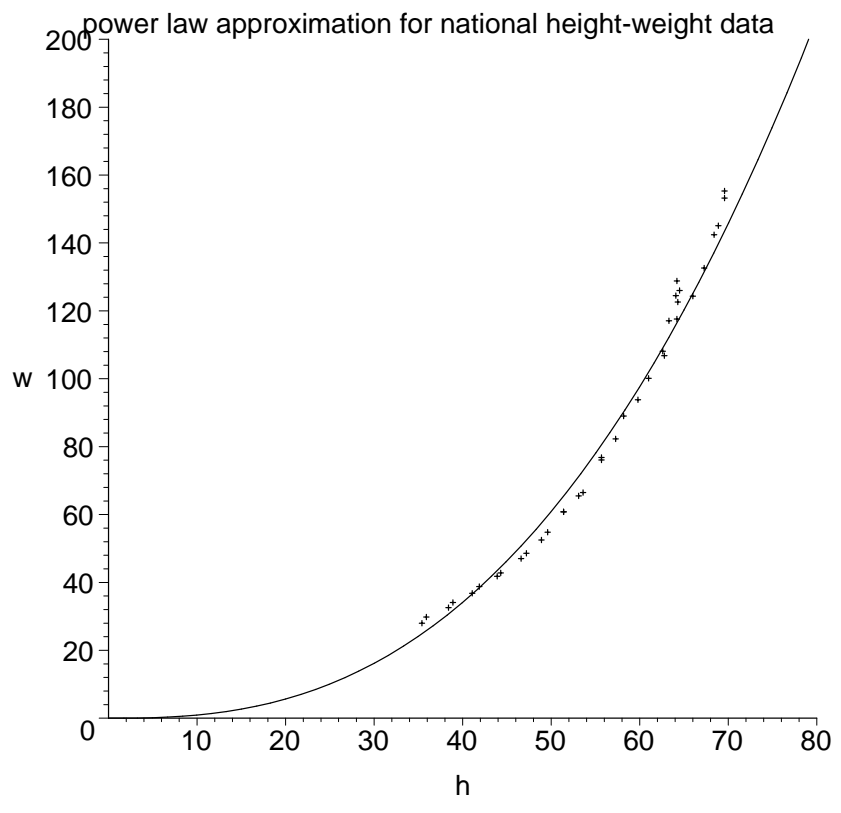
least squares fit

>
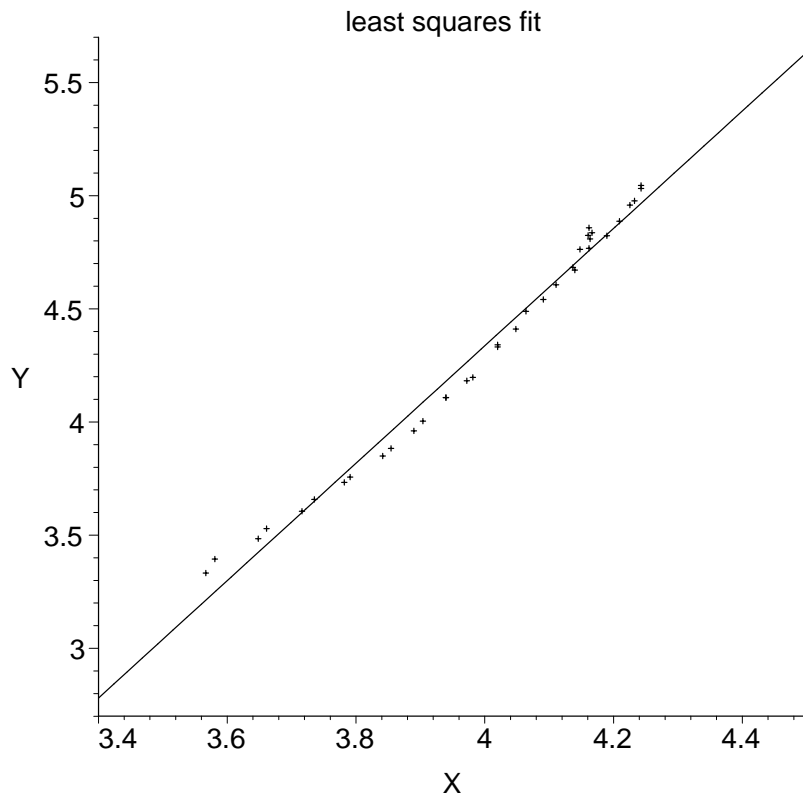
Finally, we can go back from the least squares line fit to a power law

```
> m:=2.593828004; #power
  b:=exp(-6.038703303);   #proportionality constant
```

$$m := 2.593828004$$

$$b := 0.002384649077$$

```
> powerplot:=plot(b*h^m,h=0..80,w=0..200,color=black):
  display({powerplot,boys,girls},title=
    `power law approximation for national height-weight data`);
  #by calling the variables h and w, and giving their ranges, I
  #get Maple to label the axes as I want
```

power law approximation for national height-weight data

>
>