CogSci 109 Fall 2007 Assignment 4 : Basic Statistics, Least Squares, Correlation, and Data Fits

C. Alex Simpkins

November 5, 2007

1 Description

1.1 The Basics

Please read this entire document before attempting the homework...

This assignment is worth 100 points, and a a 10 point extra credit will be possible!

In this assignment you will apply concepts of data approximation and fitting to some real data generated from your class surveys (and last year's). Each modeling tool gives you another way to represent, simplify, quantify and make decisions about the real system you are dealing with. Understanding the measure of goodness of fit is important, so you will also not only fit data, but judge the goodness of fit and make decisions based on those fits.

You will compute basic statistics of the data, and make some statements about basic relationships between variables (ie could height be related to weight?).

You will be required as usual to turn in a well-commented listing of either your matlab scripts or the commands you used to perform your calculations.

1.2 The formatting details (fulfill all of these, and get your 10 bonus points!!! A midterm present)

To avoid unnecessary complexity, we will be turning in the assignment in the same way as before: Printed out. But to avoid a Karate chop to the wallet (this joke was from a student in last year's class) you are NOT required to turn in any color prints, just black and white (Yay!).

The midterm covers all of the assignment material up to section 2.7. Doing the homework will help you study for the midterm, so you are encouraged to do so.

Formatting Requirements:

- Cover page with your name, the date, class, quarter, your section, and the homework number/title (2 pts.)
- Pages must be numbered (1pt.)
- No plots should be JPEGs. When using the *save-as* command in Matlab, use some vector-based graphics format (or one of more appropriate compression strategies) such as PDF, EPS, etc when exporting figures from matlab to your document file (if you have any issues with this, email us or otherwise let us know). The plots must be clear and not blurry. (2 pts.)
- You may recycle printer paper which has been printed on only one side IF what is on the back does not bleed through to the front or in any way interfere with your assignment. Also the material on the reverse side should be clearly unrelated to this assignment. (1 pt.)
- You may NOT use lined paper (such as from a binder or college ruled paper) (1 pt.)
- All figures must have axis labels and a figure title. If there is more than one data line or there is a scatterplot of *'s and lines, you must use a legend to clarify which line corresponds to what information source. (1 pt.)
- If you have plots with more than one line, and since you are printing in black and white, you should use more than one linetype to set the lines apart. (1

pt.)

- Turn in a well commented listing of your matlab code in an Appendix at the end of your assignment paper, or alternatively, (1 pt.)

2 Instructions

2.1 Download the data

Download the data files for this assignment on the handouts section of the page, or the assignments section. Both are the same data. The file is called **hw4data.zip**. Unzip the file, and you should have a single data file called **hw4data.mat**. You will also have a brief readme which describes the data file.

2.2 Load the data

The data file is a binary mat file. Load the file into matlab using the load command, by double clicking it, or the import wizard:

```
load hw4data.mat
```

You should now have a single variable with several rows and columns. The rows correspond (to keep in standard format) to observations, the columns to variables. The order of the columns are as follows:

Height	*Weight*	*#Hrs_TV*	*#Hrs_Hmwk.*	*Pairs_shoes*	*Sleep_time*	*Wake_time*	*Shoe_size*	*gender*
0	0	0	0	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0	0	0	0	0	0	0	0

The name of the variable is `hw4data`

2.3 Compute the basic statistics (20 points)

The first thing we will do is to compute some basic statistics and tabulate them, as well as plot the distributions of scores of each variable. This will allow us to make some intuitive decisions about which more complex statistical analyses and data fits may elucidate interesting relationships.

In addition, these quantities can be used to provide support for logical arguments and hypotheses.

2.3.1 The equations (4 points each)

Write down the equation for the mean and standard deviation, and define each variable in the equation (for example, *N represents the number of samples*). Hint: see matlab help for std.

2.3.2 The Computations (12 points for completed table)

(4 points per row, 4/9 pt. per value) For each column of the data(i.e. each variable) compute the following:

- **mean - use the matlab command** `mean()`
- **median - use the matlab command** `median()`
- **standard deviation - use the matlab command** `std()`

And tabulate the results (ie put the results in a table which has along one axis the variable names, and along another axis the statistic. The table may look something like the following:

	Height	Weight	...
Mean	0	0	...
Median	0	0	...
StDev	0	0	...

Table 1: Example of a results table (this does not include all the variables)

2.3.3 Hints for how to compute the statistics

Recall that the matlab functions will each operate on a matrix with rows being the observations and columns the variables. So for example if you had a variable called

COOLDATA which, when at the command prompt in matlab you type

```
size(COOLDATA),
```

you would get matlab's response as (keep in mind that your variables may NOT be this size, or this name, this is a demonstration only)

```
 ans =

 81 6
```

This means you have COOLDATA as a variable with 81 rows (which correspond to observations) and 6 columns (which correspond to different variables such as height, weight, etc). To compute the mean of a matrix of variables and observations (assuming the variable is in the form of the rows being observations, columns being individual variables) type something like

```
MatrixMean = mean(COOLDATA);
```

and matlab will compute the mean of each column of data and return it as an array of size 1x6 (ie one row, 6 columns of variables, or one mean per variable). Matlab works similarly for `median()` and `std()`

2.4 Plot the distributions (20 points)

2.4.1 Plot the distributions of each variable using the function `cogs109hist()` in matlab. (9 points, 1 point per histogram)

You should get a result for each variable that looks something like Figure 1 (not necessarily in shape, just overall appearance of the plot).

We use the function cogs109hist(), rather than the built-in matlab function hist() because there is a second function installed in the CSB115 computers which has the same name. So we wrote our own version. This is available from the 109 website handouts page. You must download that and place the cogs109hist.m file in your homework directory or in a folder for custom functions (such as 'myfunctions') which you have added to matlab's path.

I suggest that you create a single figure with 9 subplots, each one being a histogram (i.e. `subplot(3,3,1); cogs109hist(hw4data(:,1))` and so on up to 9 - you could even use a for loop - for n=1:9 in order to avoid repeating the command with only the number changing. Just be sure to label each subplot with what the data represents, as we have done here on the x-axis). Each histogram can be small, as long as the information is conveyed by the figure. If you desire to create multiple figure windows that is acceptable as well.

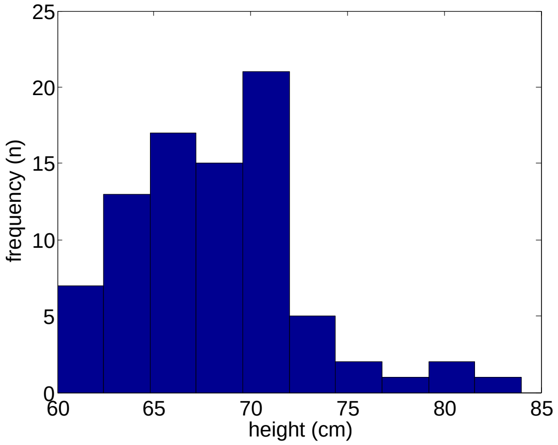

Figure 1: Distribution of heights of students in CogSci 109, the coolest class around!

If you have any problems with matlab crashing remember to type our old friend at the command prompt:

```
opengl neverselect
```

If you still have problems let Alex, Nick, Slavik or Leo know with an email or other communication, and we will help you resolve the issues.

2.4.2 Question (5 points, 1 point each)

Pick 5 of the histograms, then for each of the five, is there a standard distribution you can see the variables behaving according to? Each variable might have a different distribution. Don't try to find one for all of them, just one for each variable. For example, it may be unimodal, bimodal, normal, positive skew or negative skew, etc.

2.4.3 Question (6 points)

Briefly, pick one distribution and make an observation about the characteristic of the distribution - ie, if the mean for sleep time was 2am, and there was a very narrow distribution, what would that suggest (brief answer, maybe one or two sentences - not intended to be complex)?

2.5 Scatterplots and linear fits (20 pts.)

2.5.1 First create the scatterplot (10 points)

Choose one pair of variables, and create a scatterplot. Do this by simply plotting one variable as the x-axis, and the other as the y-axis, and format the plot to use '*' instead of lines. Please also include units with the x- and y-axis labels, such as *(lbs.)*. For example:

```
plot(height, weight, 'r*')
```

We will later comment on the relationship after fitting a simple curve, so you can be creative with the pairing. Your scatterplot will look something like the following:

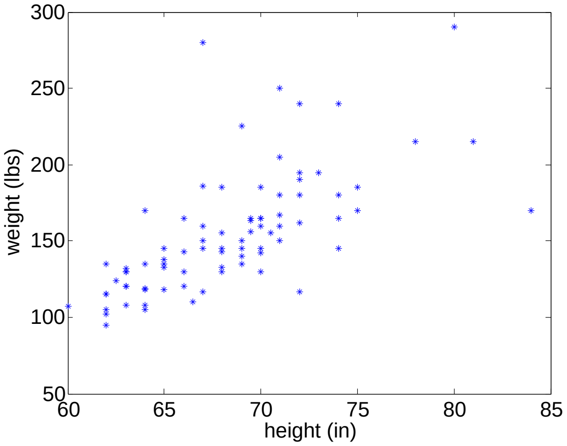

Figure 2: Height vs. Weight for the class

I suggest that you find two variables which appear to have some relationship, when you create the scatterplot. This will make the fits more meaningful.

2.5.2 Linear and nonlinear least squares fits (10 points)

Now before saving the figure let us create a few data fits to model the trend. Then we will see what kind of correlation coefficient we get. You can save one image file which includes the scatterplot and the two fits we're going to perform.

Set up a linear least squares problem just like the handout given on least squares. Follow that example to fit a two parameter system ($y = a_0 + a_1 x$, with unknown parameters a_0 and a_1) and then plot the fit in the same plot as the scatterplot.

Do the same thing, but create a quadratic fit (fit the equation $y = a_0 + a_1 x + a_2 x^2$ by finding the best fit for a_0, a_1, and a_2)

Plot lines for each fit in the same figure window as the scatterplot figure. (hint: use the hold on command in matlab

Use the legend command in matlab to create a legend for the plot, and label each line or the data. See the examples for plotting on the course web page for examples. Remember that you have to call the legend command AFTER you have plotted all your data and fits.

2.6 Computing the correlation coefficient (20 points)

2.6.1 Compute the correlation coefficient (using `corrcoef()`) and tabulate it (8 points)

Use the matlab command `corrcoef()` to compute the correlation coefficient for the pair of variables you chose. List the correlation coefficient matrix in a table such as the following:

0	height	weight
height	1	.2
weight	.2	1

In addition, use the built-in hypothesis test to compute the probability that the correlation is significant, and include the results in a table as above, but for probabilities as each number, rather than correlations (here is matlab's section of the help about doing that - it is a simple matter of changing the r=corrcoef(...) command to [r, p]=corrcoef(...)):

[R,P]=CORRCOEF(...) also returns P, a matrix of p-values for testing the hypothesis of no correlation. Each p-value is the probability of getting a correlation as large as the observed value by random chance, when the true correlation is zero. If P(i,j) is small, say less than 0.05, then the correlation R(i,j) is significant.

2.6.2 Question (3 points)

Is the relationship between the pair of variables you chose a positive correlation, or a negative correlation? Explain briefly.

2.6.3 Question (3 points)

Is the correlation strong or weak (see the lecture slides for a table of what correlation numbers typically refer to a strong/weak relationship)? Explain briefly.

2.6.4 Question (3 points)

Briefly explain why the coefficients along the diagonal (i.e. the diagonal of the correlation matrix resulting from `corrcoef`) are always one.

2.6.5 Question (3 points)

What might you postulate from these results (ie what can you suggest from your data as a question? We performed a simple hypothesis test by computing the probability that the correlation result is due to randomness, and not a relationship between the variables. Considering the resulting probability (if the p is less than 0.05, or 0.01, the correlation is considered significant here), what questions might this data lead you to ask? Briefly state your response.)

2.7 Very Basic Model Error Analysis (20 points)

2.7.1 Norm-based error (10 points)

When modeling it is very important to create a quantitative description of modeling errors so one can evaluate and possibly improve the models generated if need be. One simple way to do this is the following.

Compute the 2-norm of the error for each of your fits (using the matlab command `norm()`). Let us define

$$\epsilon = \sqrt{\sum_{i=1}^{n} \left\{ y_i - P(x_i) \right\}^2} \tag{1}$$

This provides a basic piece of information in a single number about how good your fit to the data is. Use the matlab command `norm()` to compute the norm of the difference between your actual y-variable and the predicted y-variable from your fits. Do this by taking the linear and nonlinear fits performed above, one at a time (you have an equation from when you created the plots) and using them to compute predicted y-values given the x-values at each data point. Then make a new variable such as NBE by subtracting actual y minus predicted y. Finally compute the norm of those quantities with the norm command in matlab, and include the resulting number.

Here is a place to be careful: when you plotted your fit, you may have used different x-axis points (i.e. the data may be at 0, 0.5, 1, ..., and you may have plotted your fit from 0:0.2:1, which would not input the x-value 0.5 - this makes it impossible to compare the error exactly at 0.5).

2.7.2 Question (5 points)

Is the linear or nonlinear fit better?

2.7.3 Question (5 points)

Why might one fit be better than another (i.e. can you justify your statement numerically with some quantity you calculated here)?

END OF HOMEWORK