Some More Matlab Pointers:

The following are suggestions, not absolute instructions, on commands you might use to perform certain functions. For each you should consult the online help. help x if you haven't figured out already, provides you with the online help for command x.

Matlab Input/Output:

1. To write a matrix to a text file, use dlmwrite:
   dlmwrite(filename,M,dlm) writes a matrix M into filename using the character dlm as the delimiter. Filename has to be surrounded by single quotes e.g. 'test' or 'input'. Dlm can be things like tab or space: '\t' denotes tab and ' ' denotes space. dlmwrite('test',M,' ') would write matrix M in to filename test delimited by spaces.

2. To read a matrix from a text file, you could use dlmread or load:
   M=dlmread(filename,dlm) puts the numbers in filename(which are delimited by dlm) in to matrix M; it puts each line of the file in to a different row of the matrix. Load has several functionalities, but the simplest use of it is simply: load filename. In this case the resulting matrix is named 'filename'.

3. When accessing functions or scripts that are stored in .m files you simply enter the filename at the matlab command line. HOWEVER, Matlab has to know what directory to look in to find the .m file. The command addpath can be used to specify additional directories where matlab should look:

   addpath ../temp looks in the temp directory which is located in a directory one level up from the one from which you executed matlab.

Functions:

Functions are similar to scripts in that they are both a collection of matlab commands that can be executed simply by entering the name of the .m file at the command line. A script is really just a shortcut to retyping all the commands in its .m file. This means that all the variables defined in the script are accessible and can be manipulated once you've run the script. With a function, the variables defined within the function disappear once the function has executed – you can never access them from the command line (they are called "local variables"). The purpose of a function is simply to transform some input variables in to an output variable. In our homework, for example, the MAP_normalize function takes a matrix of unnormalized values and transforms it into a vector of normalization factors.

When you create a new function in a .m file, the format is:

```
function c = addition(a,b)          % function declaration:
                                    % c is the output variable, a, b are the input
                                    % variables. There can be as many input
                                    % variables as you want, but only one output
                                    % variable. The output variable can be a vector
                                    % or a matrix, or any data structure in matlab.
                                    % The name of the function must be the same
                                    % as the .m file name

c = a + b;                          % function body: as many commands, loops, etc
                                    % as you want

return                              % the function ends when it sees a return; if you
                                    % don't use return, it ends when all the
                                    % commands have been executed
```

When you call 'addition' from the command line, you use the variables you are interested in; they don't have to be named a,b,c but their type(vector,matrix etc.) has to be compatible with the operations you've used in the body of the function :

- x = addition(y,z);

Distribution Functions:

To compute p-values you need to compute the "area under the tail(s)" of the density function. Remember the probability density function,pdf, is the function whose area integrates to 1, and whose *integral* between the values a and b represents the probability of getting any value between a and b . The "distribution" function (also called a cumulative distribution function, cdf, and often denoted by a capital letter) is a different function with the following definition: G(d) = P(x <= d). It measures the area to the left of d under the *density* function, i.e. the integral of the density from negative infinity to d. Another way to think about it is that G(d) or 1-G(d) correspond to the one tailed p-value associated with the value d in the underlying density function. In problem set 2 you are asked to compute significance for an "F" density function. To access the distribution function of this density use:

```
fcdf(d,v1,v2)       computes probability of seeing any value equal to or less than d
                    under an F density with v1 and v2 degrees of freedom.
```

Matlab has access to many different cdfs. Look in the statistics toolbox.