Matlab Tips

Useful features

Explore the Matlab Start menu (button in the bottom left corner). Look at Workspace explorer (**Desktop Tools/Workspace**) where you can see the variables, click on the variables and see what you can do in the **Array Editor** that opens. Learn the operations of Import Wizard (**MATLAB/Import Wizard**), Plot tools (**MATLAB/Plot Tools**), and statistical distribution tools (**Toolbox/Statistics**). Play with it and use MatLab Help extensively to learn about the commands and their syntaxis.

MatLab is essentially a programming language, and most things can be done from the command line, or by running program files. But graphical interface is extremely convenient for standard tasks. Most things (arrays, matrices, plots, etc.) can be made into a variable by a simple assignment operator, which greatly simplifies their handling.

Working with variables and arrays

All variables in Matlab are treated as matrices, or arrays. The dataset 'normtemp' in your exercise, for example, is represented by a 130x3 matrix, because it has 130 rows and 3 columns (temperature, gender, and heart rate). To use only the first column, you can use the variable *normtemp* with an explicit specification of the range of its elements:

```
normtemp(:,1)
```

The colon ':' is a "wildcard". It tells Matlab to take *all* rows. The second index tells it to take the first column. It is a common practice to isolate the required elements of an array into a separate variable, for convenience:

```
temp = normtemp(:,1)
```

This creates a variable *temp*, which contains the first column of normtemp and is therefore a 130x1 matrix.

To refer to specific elements, you can specify those elements or their range:

```
temp(65)          % refers to the 65th element of array temp
normtemp(65,1)    % the same number but in array normtemp
temp(1:65)        % refers to a 1-dimensional array that contains elements of
                  % temp from 1 to 65
```

Note that, in the latter case, where the colon ':' is sandwiched between two numbers, it means 'from .. to", and not the dimension of the array.

A matrix that consists of only one row or only one column is often called a *vector*. This is

because that's what a vector is: an array of numbers, e.g. space coordinates. Sometimes your vectors are columns, sometimes they are represented as rows. To switch between these two representations, you can use

```
transpose(temp)
```

where instead of 'temp' of course you can use the name of any variable. This operator transposes any matrices, not just vectors.

Matlab review of probability distributions.

Matlab Help/Statistics Toolbox/Probability Distributions.

Each distribution in Matlab has a name, for example, normal distribution has a name 'norm'. Adding a suffix defines a function associated with this distribution. For example, 'normrnd' generates random numbers from distribution 'norm', 'normpdf' gives p.d.f., 'normcdf' gives cumulative distribution function, 'normfit' fits the normal distribution for a given dataset. Please, look at each function for its syntax, input, output, etc. Type 'help normrnd' to quickly see how the normal random number generator works. Also, there is a graphic user interface tools like 'disttool' (to run it just type disttool in the main Matlab window) that allows you to play with different distributions, or 'randtool' that generates and visualizes random samples from different distributions.

Useful tools for working with distributions

```
>> disttool        % Excellent tool for investigating the shapes of PDF and CDF of
                   % standard distributions. You can also use it for reading out the
                   % probabilities and confidence intervals!

>> dfittool        % Distribution fitting tool. Generates a fit line, calculates mean,
                   % variance, and error on the mean. Also allows you to exclude
                   % points and investigate probability intervals.

>> chi2gof(x)      % Uses chi-square to test the variable x if it is normally distributed,
                   % with mean  and variance calculated from the array in x.
                   % Returns '1' if x is not normally distributed with 5% confidence.
                   % For additional options and testing for other distributions, see Help.

>> [h,p, stats]=chi2gof(x,'nbins',nbins)
                   % explicitly specifies the number of histogram bins in the variable
                   % nbins. The structure variable 'stats' contains the calculated value
                   % of chi-square and the number of degrees of freedom that was
                   % used in the calculation. The variable 'p' contains the probability
                   % of obtaining this or more extreme value of chi-square.
```

Plotting

Most plots can be made using the 'plot(X)' command, or with the corresponding button in the Workspace browser. Consult Help for the numerous options of the 'plot' command. Histograms can be made with a 'hist(X)' command.

```
>> hist(X,nbins)     % plots a histogram of X with the number of bins given by nbins
```

To plot several datasets on the same graph, use either multiple specifications 'plot(x,A,x,B)' (see Help) or -- very useful – the 'hold on' command. Until you enter 'hold off', all plots will be made on the same graph. Click on the "cursor arrow" icon in the toolbar at the top of the figure window; click on the graph and explore the graphical plotting options, like color, line style, etc.

Instead of a default line plot, a scatter plot can be produced by specifying *plot(x,y,'o')*. That's small letter 'o', nor zero.

For visualizing scatter in statistical data, a *boxplot* can be useful

```
>> boxplot(X)       %vector X contains a set of data
>> boxplot(X,Y)    % vector Y contains descriptions of groups, corresponding to X
```

The boxplot generates a figure that shows the median, the maximum and minimum values in the dataset, and the upper and lower quartiles. The boxplot is also used in the ANOVA analysis.

Correlations and fitting

Matlab includes a number of convenient and sophisticated tools for analyzing correlations in the data and fitting linear, polynomial, or other curves. Run these tools and explore their capabilities.

```
>> cftool              % curve fitting tool
>> polytool(x,y)    % fits the y(x) data and shows the bounds for possible fits
```

The following commands are useful and can be used directly from the command line:

```
>> fit(x,y,'poly1')    % fits y(x) and shows the statistics of the fit
>> corrcoef(x,y)       % calculates correlation coefficient
>> corrcoef([X Y Z])   % generates a matrix of correlation coefficients
```

Miscelaneous useful features

```
>> format long     % allows more decimal places to be displayed, avoiding round off error
```

```
>> y=x          % copies variable x into y and prints its content
>> y=x;         % copies variable x into y without printing its content
```

Putting semicolon ‘;’ at the end of a MatLab command in general suppresses the output to screen.