Introduction to Mathematica

The ENTER key above the SHIFT key is simply a carriage return. The ENTER key on the numeric keypad executes the actual commands.

Commands that you type in appear in boldface, while Mathematica output appears in normal type.

Mathematica is case-sensitive. All Mathematica built-in commands begin with capital letters. If your variable names all begin with lower-case letters, there will be no conflict.

If you forget what a command like "Plot" does, type:

?Plot (and then press ENTER)

```
Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x
   from xmin to xmax. Plot[{f1, f2, ...}, {x, xmin, xmax}] plots
   several functions fi.
```

There are three types of grouping symbols:

Parentheses are used to group mathematical operations

2 * (a+b)

Curly braces are used to group lists

{a,b,c}

Square brackets hold the arguments of Mathematica functions

Sqrt[2] * Sin[z]

Mathematica can be used as a calculator, but in an effort to be precise the program does not give the decimal expansion of the answer unless you specifically request it.

349/544

```
349
---
544
```

The way to request decimal output is with the "N" command (N for numerical)

N[349/544]

```
0.641544
```

The default number of decimal places is 6, but you may request more. For example, 40 places.

```
N[349/544,40]
```

```
0.6415441176470588235294117647058823529412
```

The constant pi to 100 decimal places.

```
N[Pi,100]
```

```
3.141592653589793238462643383279502884197169399375105820974944592307\
  816406286208998628034825342117068
```

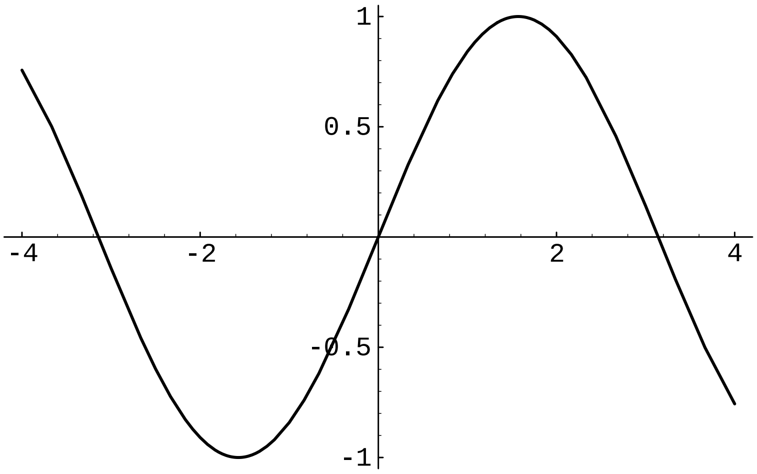
Powers are obtained using the "caret" (found over the "6" key)

```
3^2
```

```
9
```

Mathematica can plot functions. For example, the function sin[z] from z=-4 radians to z=+4 radians.

```
Plot[Sin[z],{z,-4,4}]
```



The variable "z" above is a "dummy variable". The following commands will produce exactly the same picture.

```
Plot[Sin[y],{y,-4,4}]
```

```
Plot[Sin[dummy],{dummy,-4,4}]
```

Mathematica assumes that the arguments of trigonometric functions like Sin, Cos, Tan are in RADIANS. If you wish to work in DEGREES, then you must type the word "Degree" after the angle, as follows

```
Plot[Cos[theta],{theta,20 Degree,80 Degree}]
```

Best fit straight line

One common use for Mathematica in this lab is fitting the best straight line through some data points. mydata is a list of the form {{x1,y1},{x2,y2},{x3,y3},...,{xn,yn}}

```
mydata={{1.1,2.1},{1.9,4.2},{3.1,5.8},{4.2,7.9},{4.9,10.0}};
```

The following command fits the best straight line through the data. The "1" in the curly brackets is a request to fit the best y-intercept. The "x" in the curly brackets is a request to fit the best slope.

```
bestline[x_] = Fit[mydata,{1,x},x]
```

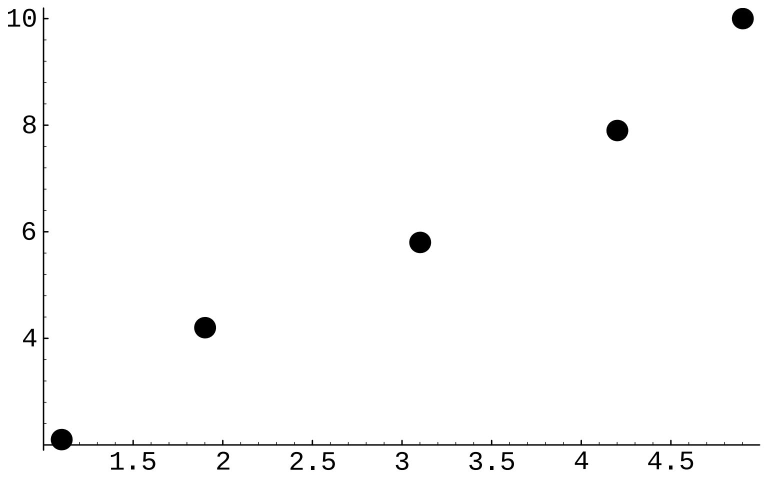
```
0.0721232 + 1.94996 x
```

So the best y-intercept is 0.0721232 and the best slope is 1.94996. By the way, "bestline" is an arbitrary variable name; any other name like "myline" will work equally well. You are free to replace any lower-case name with one of your own choosing.

```
myline[x_] = Fit[mydata,{1,x},x]
```

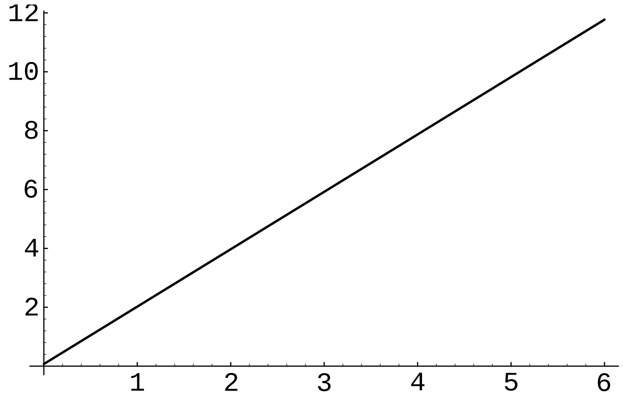
Now we can plot the data points.

```
p1 = ListPlot[mydata,PlotStyle->{PointSize[0.03]}];
```



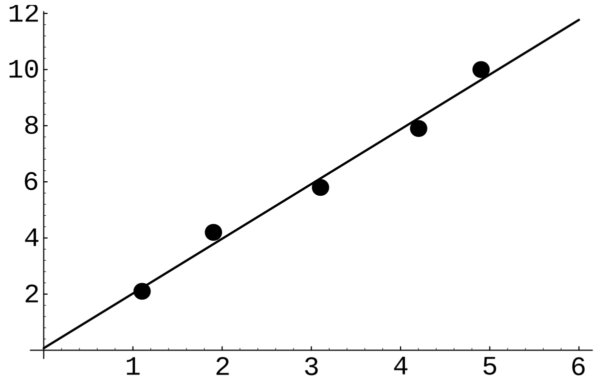
And plot our best fit line.

```
p2 = Plot[bestline[x],{x,0,6}];
```



And finally overlay the two plots to check the fit.

```
Show[p1,p2];
```



To fit the best line through the origin, that is, with a constrained y-intercept of zero, one simply replaces the "1" in the curly brackets of the Fit command with a "0".

```
bestlinethroughorigin[x_] = Fit[mydata,{0,x},x]
```

```
1.96951 x
```

Assignment

```
1) Plot a function of your choice.  Choose something other than sine.

2) Fit the best straight line through the following data:
   x1 = 11,  y1 = 23
   x2 = 15,  y2 = 33
   x3 = 19,  y3 = 45
   x4 = 24,  y4 = 51
   x5 = 30,  y5 = 62

3) Print out your results.
```