

```
% BASIC COMMANDS FOR PLOTTING
```

```
%1
```

```
disp("Q1");  
help plot;  
help stem;  
n=0:0.01:10; %this is the array of the domain of the function  
x=cos(2*pi*n);  
plot(x); %stem( Y ) plots the data sequence, Y , as stems that extend from a  
baseline along the x-axis.  
stem(x);
```

```
%2
```

```
disp("Q2");  
plot(n,x); %plots x for all the points in array n
```

```
%3
```

```
disp("Q3");  
n=0:0.01:1; % domain for a single cycle  
x=cos(2*pi*n);  
plot(n,x);
```

```
%4
```

```
disp("Q4");  
y=sin(2*pi*n);  
plot(n,x,n,y); % plot for graphs x and y on the same plot having domain n.
```

```
%5
```

```
disp("Q5");  
x=2:20;  
y=x.*x; % array y having sq of elements of x  
z=x+1i*y;  
plot(z);% plot the graph y=f(x)
```

```
%6
```

```
disp("Q6");  
n=0:0.01:pi/2;  
x=cos(2*pi*n);  
y=sin(2*pi*n);  
plot(n,y,"--",n,x,"Diamond",'Color','r','LineWidth',2);  
legend('Cosine Signal','Sine Signal'); %signyfing legends for the x and y  
respectively  
xlabel("N"); %Labelling x-axis as N  
ylabel("Two functions of n"); % labelling y-axis  
plot(n,y,"--",n,x,"Diamond",'Color','b','LineWidth',2);  
%Lindewidth- specififes width of line  
% "Diamond" or "--" signifies type of line  
% Color denoted color of plotted graph
```

```
%6
```

```
disp("Q6");  
help hold;  
n=0:0.01:pi/2;  
x=cos(2*pi*n);  
y=sin(2*pi*n);  
plot(n,x,"Diamond",'Color','r','LineWidth',2); hold ON;  
plot(n,y,"--",'Color','b');hold OFF;  
% Hold saves the current plot so that it can plotted wiht the next graphs  
% until hold is turned oFF.
```

```

%7
disp("Q7");
help axis;
axis([0 pi/2 -0.5 0.5]);
% specifies range of x and y to be 0 to pi/2 and -0.5 to 0.5 respectively.

%8
disp("Q8");
help subplot;
subplot(211);
plot(n,x);
subplot(212);
plot(n,y);
%subplot divides the graph into r by c matrix and the plot individual graph
%on the specified number/index

%9
disp("Q9");
x=0.01:0.01:1;
y=x.^(-10);
subplot(221);
plot(x,y);
subplot(222);
semilogx(x,y);
subplot(223);
semilogy(x,y);
subplot(224);
loglog(x,y);
subplot(111); %To restore graph size to original
% plots the 4 function on the 4 sections of graph.
% semilogx(The x axis has a log scale)
% semilogy - The y axis has a log scale
% loglog - both axis have a log scale

%10
disp("Q10");
semilogy(x,y);
grid ON;
% turns ON the grid

%11
disp("Q11");
semilogy(x,y);
grid ON;
title("Example of a Semilog Plot");
% Give a title to the graph

% PLOTTING OF CONTINUOUS TIME SIGNALS

%1
x=-5:0.001:5;
grid ON;
%a) Unit step function
subplot(251);
u=[zeros(1,5000) 1 ones(1,5000)];
plot(x,u,'LineWidth',3);
title('A');

```

```

%b) Signum function
subplot(252);
u=[-1*ones(1,5000) 0 ones(1,5000)];
plot(x,u,'LineWidth',3);
title('B');

%c) Rectnagular pulse with duration 2 sec
subplot(253);
u=[zeros(1,5000) ones(1,2000) zeros(1,3001)];
plot(x,u,'LineWidth',3);
title('C');

%d) Unit ramp function
subplot(254);
u=x;
u=u.*(u>0);
plot(x,u,'LineWidth',3);
title('D');

%e) Square wave of period 2
subplot(255);
help square;
u=square(x*pi/2);
plot(x,u,'LineWidth',3);
title('E');

%f) A sawtooth wave of period 3 sec
subplot(256);
u=sawtooth(x);
plot(x,u,'LineWidth',3);
title('F');

%g) Triangular wave if period 3
subplot(257);
u=sawtooth(x,0.5);
plot(x,u,'LineWidth',3);
title('G');

%h) sinc() funtion
subplot(258);
u=sinc(x);
plot(x,u,'LineWidth',3);
title('H');

%i) Timescaling sinc function
subplot(259);
u1=sinc(x);
u2=sinc(2*x);
u3=sinc(x/2);
plot(x,u1,x,u2,x,u3);
title('I');

%j) Timeshifting sinc function
subplot(2,5,10);
u1=sinc(x);
u2=sinc(x+2);
u3=sinc(x-2);
plot(x,u1,x,u2,x,u3);
title('J');

```

```
subplot(111); %Restore plot size to 1:1
```

```
%2) Dirac-Delta Function
```

```
u=[zeros(1,5000) 1 zeros(1,5000)];  
plot(x,u,'LineWidth',3);  
title('Dirac-Delta Function');
```

```
% HISTOGRAMS, SKEWNESS AND KURTOSIS
```

```
%1)
```

```
help histogram;  
x=rand(1,10000);  
plot(x);  
u=histogram(x); %makes an histogram of x and stores its values in the variable u
```

```
%2)
```

```
disp(u); %prints the properties of the histogram.  
% 20 bins are used here.
```

```
%3)
```

```
histogram(x,100); %makes an histogram using 100 bins
```

```
%4)
```

```
x=2*rand(1,10000)-1; %This generates random numbers between -1 to 1  
histogram(x,20) % makes an histogram with 20 bins on the above data
```

```
%5)
```

```
help skewness;  
help kurtosis;  
s=skewness(x); % Skewness is the third central moment of X, divided by the cube of  
its standard deviation.  
k=kurtosis(x); % Kurtosis is the fourth central moment of X, divided by fourth  
power of its standard deviation.  
fprintf("Skewness is %d\n\n",s);  
fprintf("Kurtosis is %d\n\n",k);
```

```
% SAVING AND LOADING
```

```
%1)
```

```
savefig("Histogram_x.fig"); % saves the last output image as a file in .fig format
```

```
%2)
```

```
p=openfig("Histogram_x.fig"); % open a file in .fig format and saves it in a  
variable p.
```

```
%3)
```

```
saveas(p,"Histogram_xx","jpg"); % saves the image p in the desired format  
eg.jpg,jpeg from .fig
```