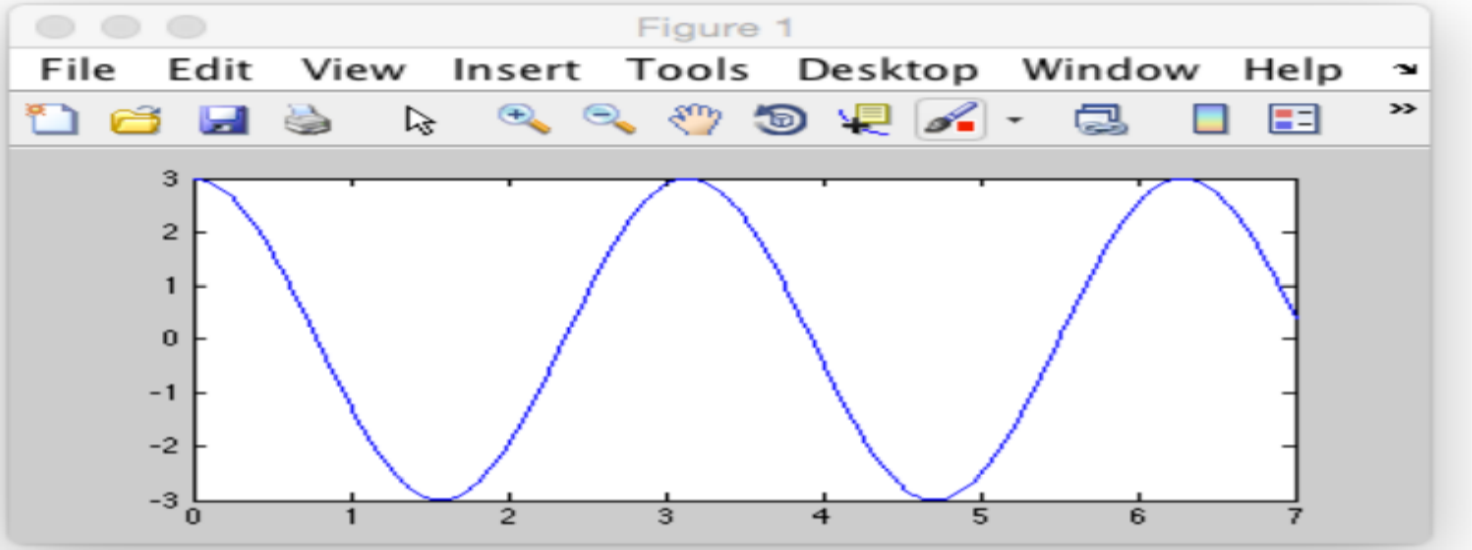
2D Plotting

Example :

$y=3\cos 2x$ for $0 \leq x \leq 7$

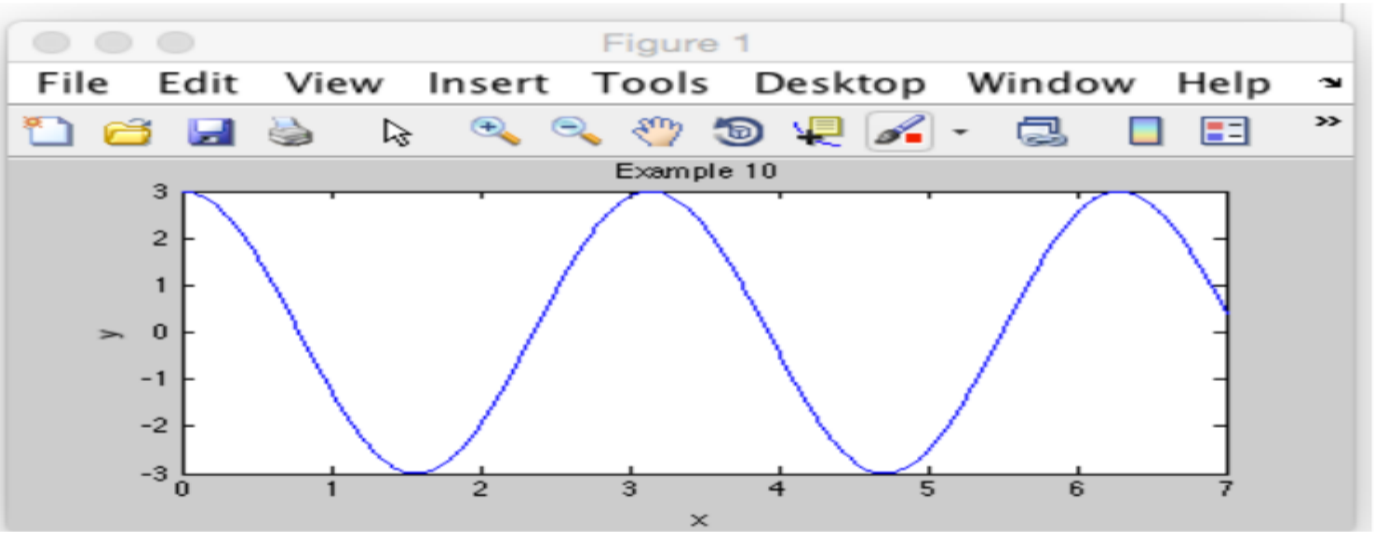
We choose to use an increment of 0.01 to generate a large number of x values in order to produce a smooth curve. The function plot(x,y) generates a plot with the x values on the horizontal axis and the y on the vertical access.

```
1 -     x=0:0.01:7;
2 -     y=3*cos(2*x);
3 -     plot(x,y)
```



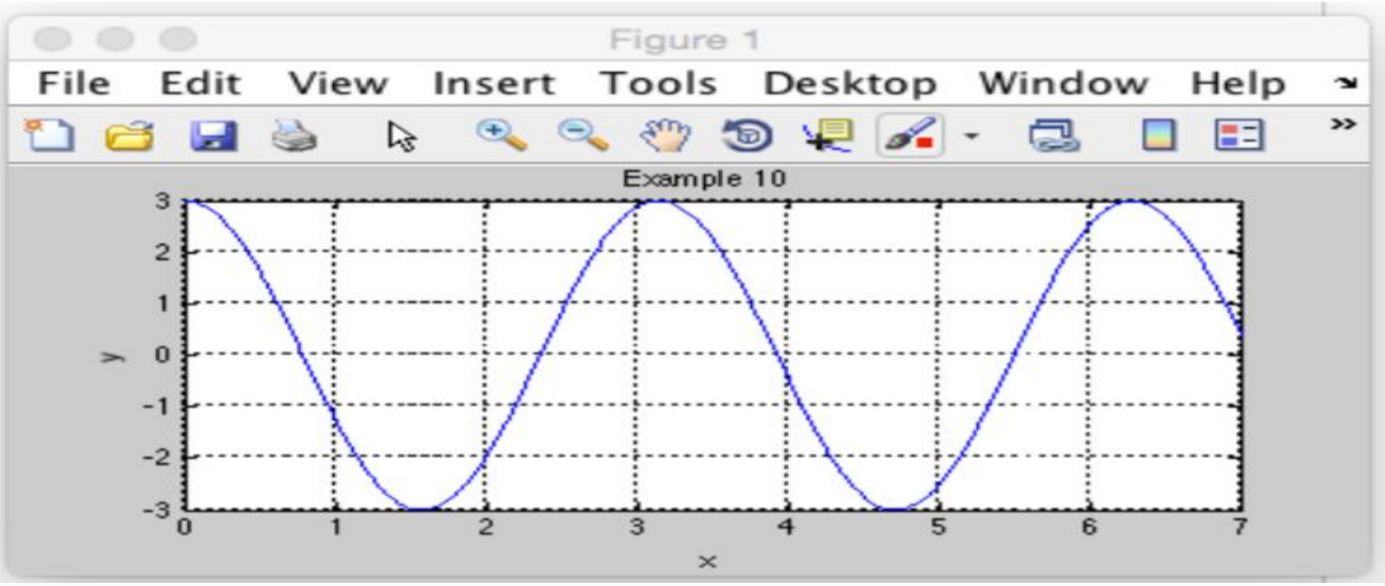
To add x and y labels and a title to the graph

```
x=0:0.01:7;
y=3*cos(2*x);
plot(x,y)
xlabel('x');
ylabel('y');
title('Example 10');
```



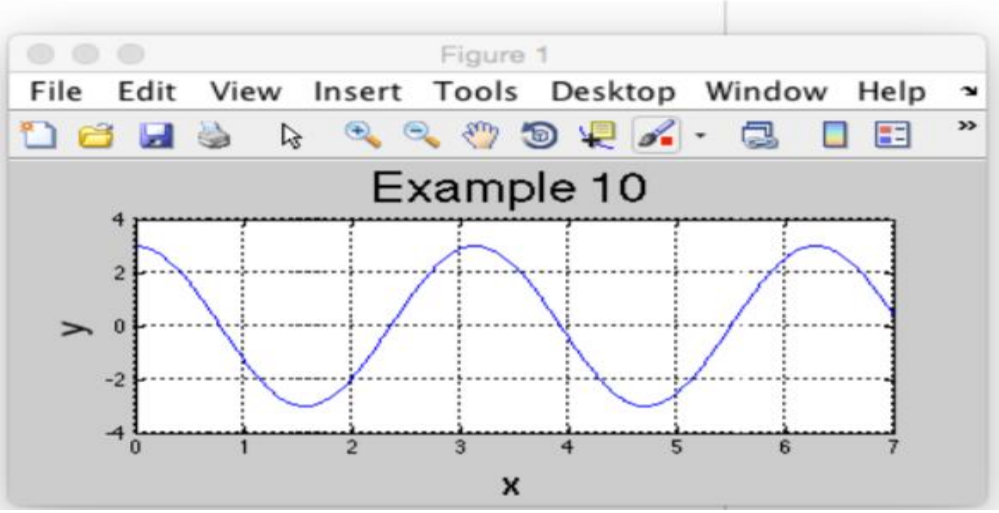
To add a grid to the graph

```
x=0:0.01:7;
y=3*cos(2*x);
plot(x,y)
xlabel('x');
ylabel('y');
title('Example 10');
grid
```



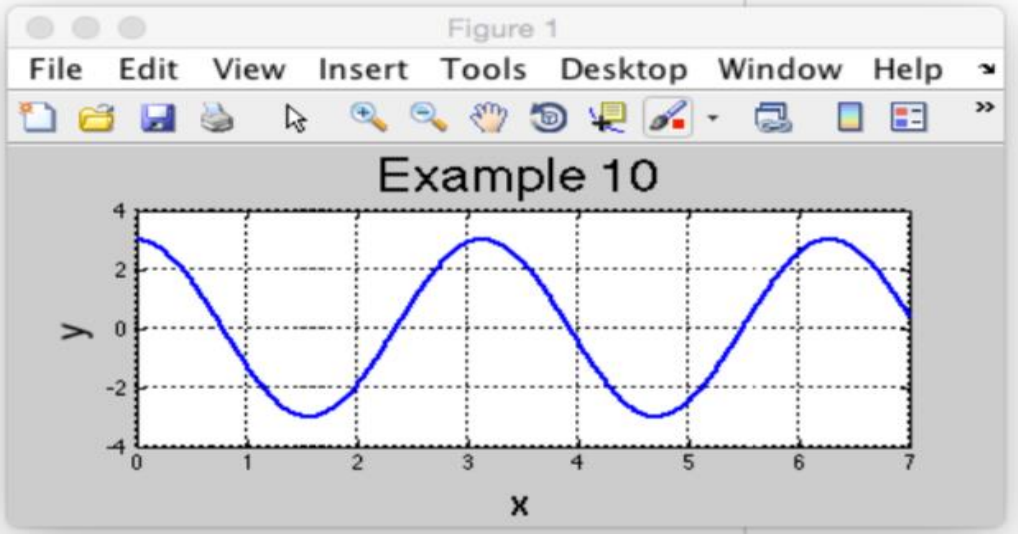
To increase the font size

```
1 -     x=0:0.01:7;
2 -     y=3*cos(2*x);
3 -     plot(x,y)
4 -     xlabel('x','FontSize',18);
5 -     ylabel('y','FontSize',18);
6 -     title('Example 10','FontSize',24);
7 -     grid
8
9
```



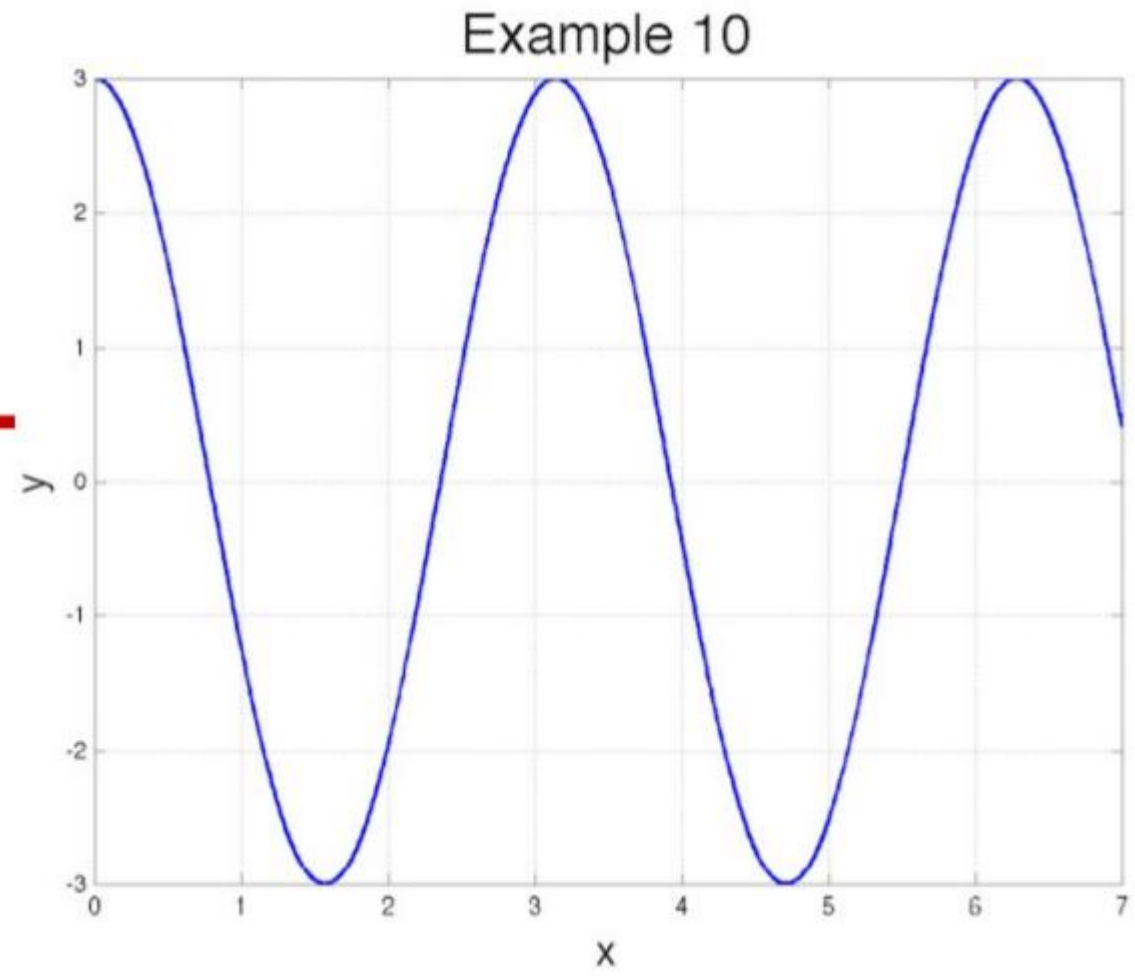
To increase the plotted line

```
1 -     x=0:0.01:7;
2 -     y=3*cos(2*x);
3 -     plot(x,y,'linewidth', 2)
4 -     xlabel('x','FontSize',18);
5 -     ylabel('y','FontSize',18);
6 -     title('Example 10','FontSize',24);
7 -     grid
8
9
```



To save the figure in Matlab

```
1 -     x=0:0.01:7;
2 -     y=3*cos(2*x);
3 -     plot(x,y,'linewidth', 2)
4 -     xlabel('x','FontSize',18);
5 -     ylabel('y','FontSize',18);
6 -     title('Example 10','FontSize',24);
7 -     grid
8 -     saveas(gcf, 'Example 10.tiff');
9
```




```
saveas(gcf, 'Example 1.fig');
saveas(gcf, 'Example 1.tiff');
```

Data Markers and Line Types In the graph

to choose different color or marker type or line type (solid, dashed, …..)

Choose from the following table :

Data markers		Line types		Colors	
Dot (.)	.	Solid line	—	Black	k
Asterisk (*)	*	Dashed line	– –	Blue	b
Cross (x)	x	Dash-dotted line	– .	Cyan	c
Circle (∘)	∘	Dotted line	:	Green	g
Plus sign (+)	+			Magenta	m
Square (▫)	s			Red	r
Diamond (◇)	d			White	w
Five-pointed star (w)	p			Yellow	y

Data Markers and Line Types

- A plotting command with a certain color:

```
plot(x,y, 'r')
```

- A combination of line type & color :

```
plot(x,y, '--r')
```

- A combination of mark & color :

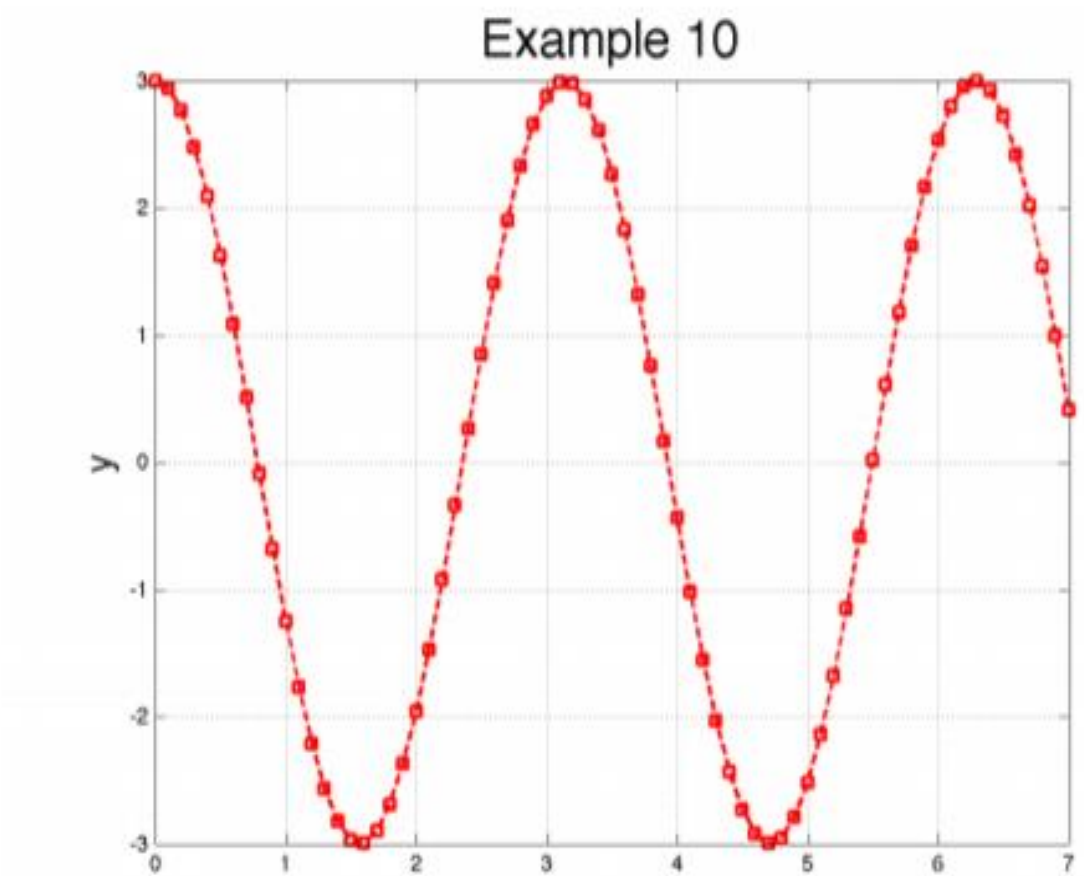
```
plot(x,y, '+r')
```

- A combination of line type, mark & color :

```
plot(x,y, '--+r')
```

4 -

```
plot(x,y,'--rs','LineWidth',2)
```

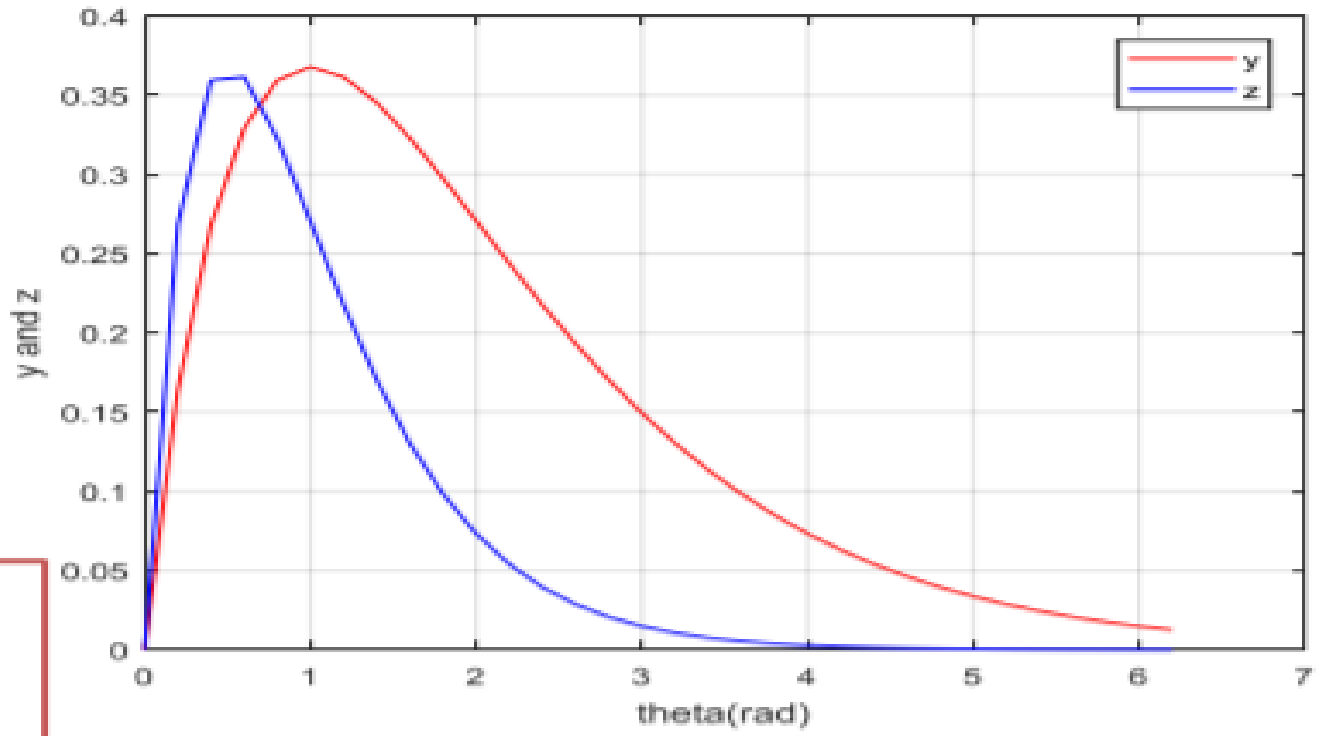


Multiple Curve Plots

```
clc; clear; close all
theta = 0:0.2:2*pi ;
y = theta.*exp(-theta) ;
z = 2*theta.*exp(-2*theta) ;
plot(theta , y , 'r', theta , z , 'b')
xlabel('theta(rad)') ;
ylabel('y and z')
grid on
```

OR

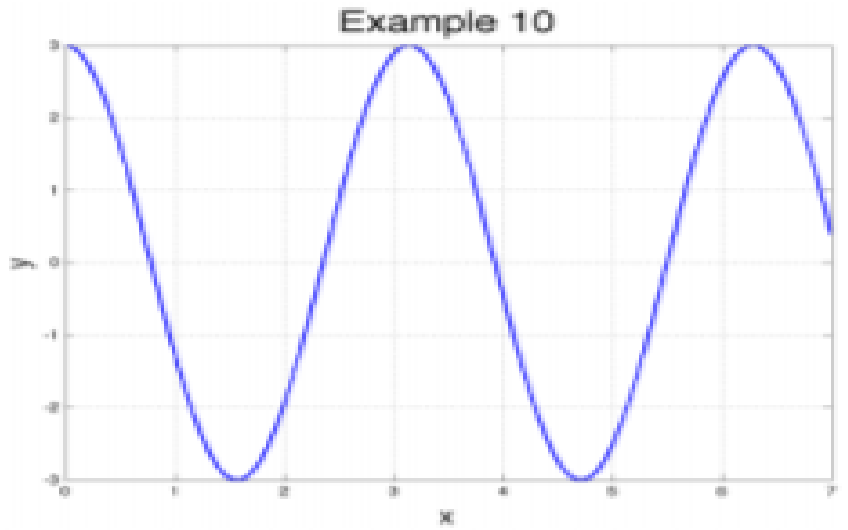
```
clc; clear; close all
theta = 0:0.2:2*pi ;
y = theta.*exp(-theta) ;
z = 2*theta.*exp(-2*theta) ;
plot(theta , y , 'r')
hold on
plot(theta , z , 'b')
legend('y','z')
xlabel('theta(rad)') ; ylabel('y and z')
grid on
```

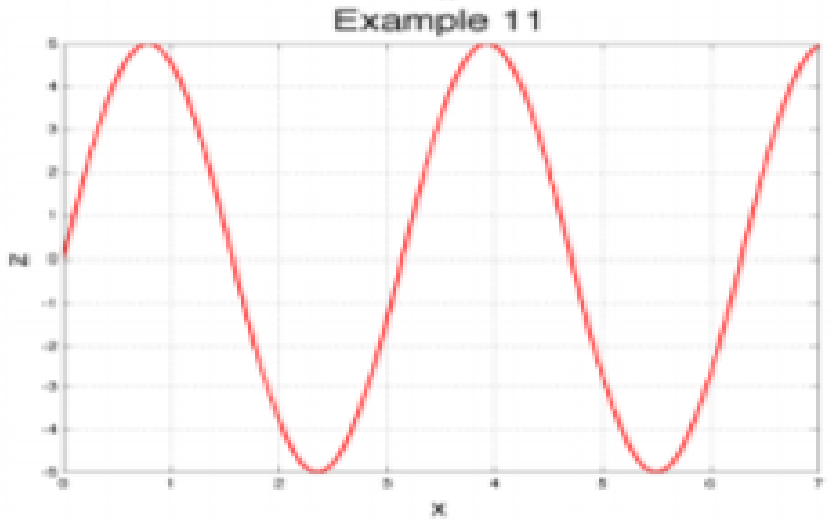


grid off : To delete the grids
hold off : To release the figure

If we want to plot two figures, we use the following commands:

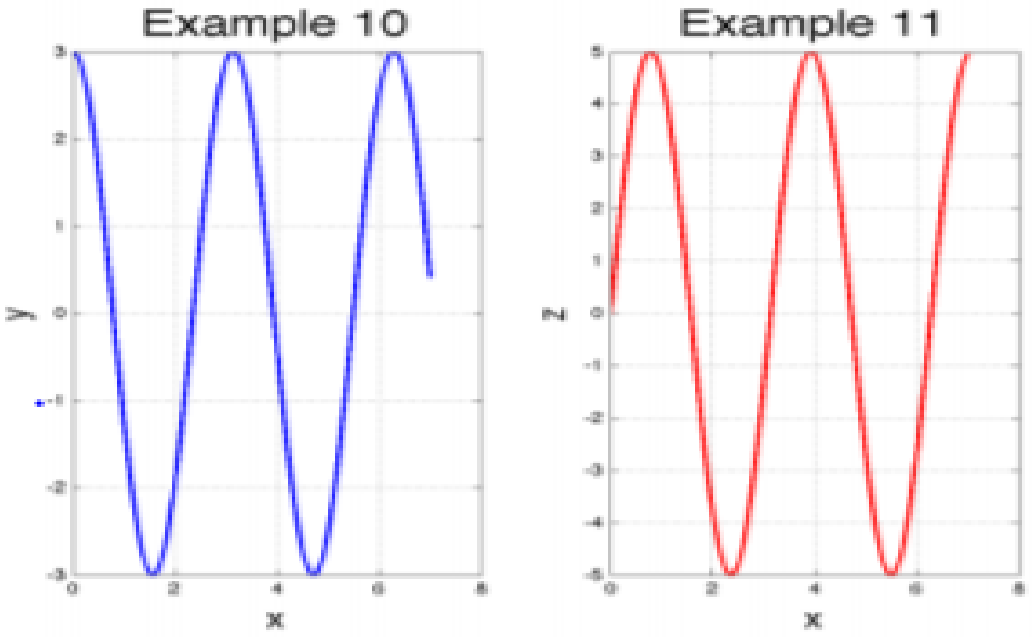
```
x=0:0.01:7;
y=3*cos(2*x);
z=5*sin(2*x);
figure(1)
plot(x,y,'linewidth', 2)
xlabel('x','FontSize',18);
ylabel('y','FontSize',18);
title('Example 10','FontSize',24);
grid
saveas(gcf, 'Example 10.tiff');
figure(2)
plot(x,z,'r','linewidth', 2)
xlabel('x','FontSize',18);
ylabel('z','FontSize',18);
title('Example 11','FontSize',24);
grid
saveas(gcf, 'Example 10_2.tiff');
```





If we want to plot two plots in the same window, we use the following commands:

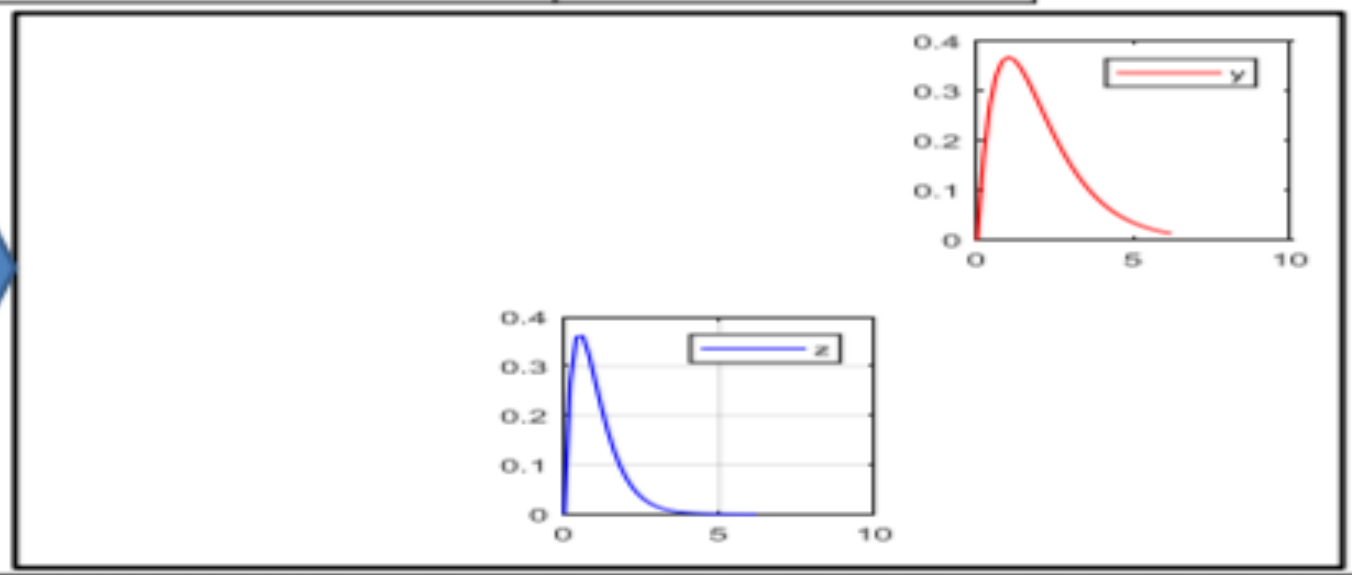
```
 1 -     x=0:0.01:7;
 2 -     y=3*cos(2*x);
 3 -     z=5*sin(2*x);
 4 -     subplot(1,2,1)
 5 -     plot(x,y,'linewidth', 2)
 6 -     xlabel('x','FontSize',18);
 7 -     ylabel('y','FontSize',18);
 8 -     title('Example 10','FontSize',24);
 9 -     grid
10 -     subplot(1,2,2)
11 -     plot(x,z,'r','linewidth', 2)
12 -     xlabel('x','FontSize',18);
13 -     ylabel('z','FontSize',18);
14 -     title('Example 11','FontSize',24);
15 -     grid
16 -     saveas(gcf, 'Example 10_subplot.tiff');
17
```



Subplot Commands
Example:

Subplot (2, 3, 1)	***Subplot (2, 3, 2)***	***Subplot (2, 3, 3)***
Subplot (2, 3, 4)	***Subplot (2, 3, 5)***	***Subplot (2, 3, 6)***

```
clc; clear; close all
theta = 0:0.2:2*pi ;
y = theta.*exp(-theta) ;
z = 2*theta.*exp(-2*theta) ;
subplot(2,3,3)
plot(theta , y , 'r')
legend('y')
subplot(2,3,5)
plot(theta , z , 'b')
legend('z')
grid on
```



Axis Command

The maximum and minimum of the coordinates on the graph may be specified by the command:

```
axis([xmin,xmax,ymin,ymax])
```

Example: `axis([0 , 5 , -0.5 , 1.5])`

A figure can be reshaped by the command:

```
axis('square')
```

- Figure clearing The command clf clears everything inside the graphic window.

Text in graph

Text can be written in a graph by the following command:

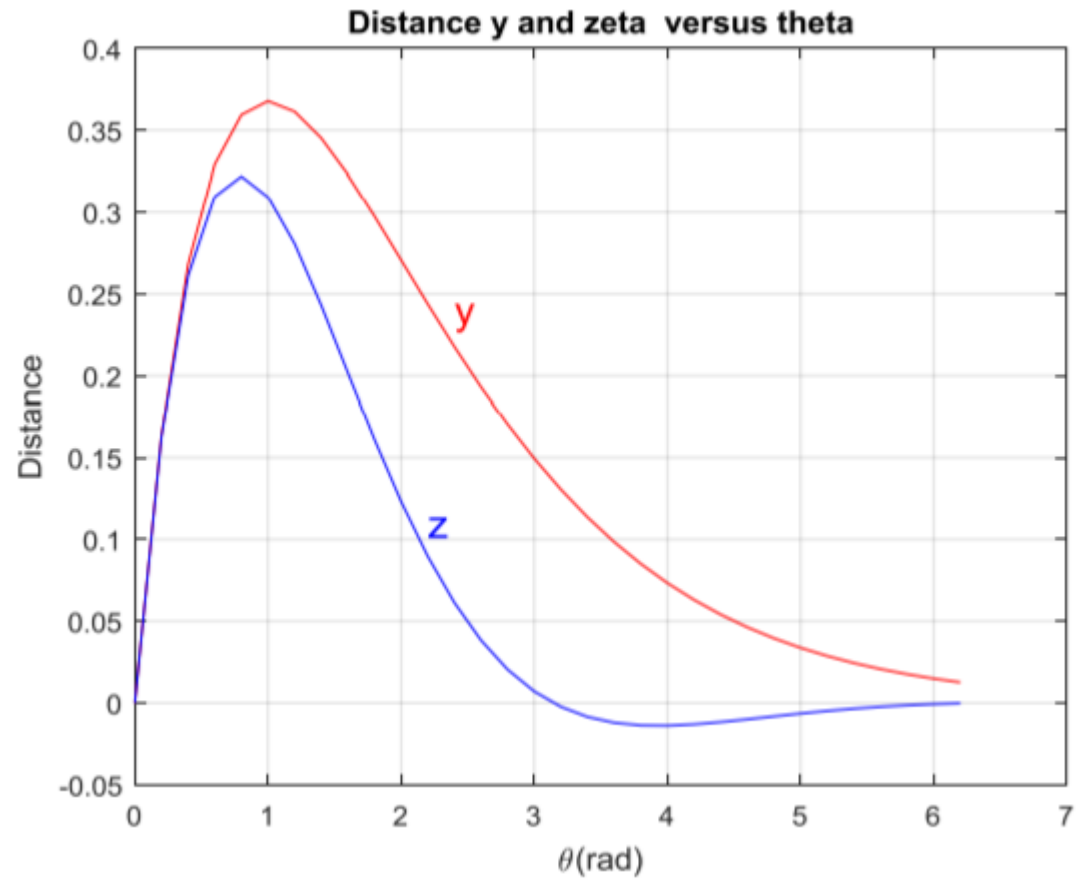
```
text
```

Example

```
text(x , y , ' string ')
```

where `x,y` are the coordinates where the string starts
`string` is the required string

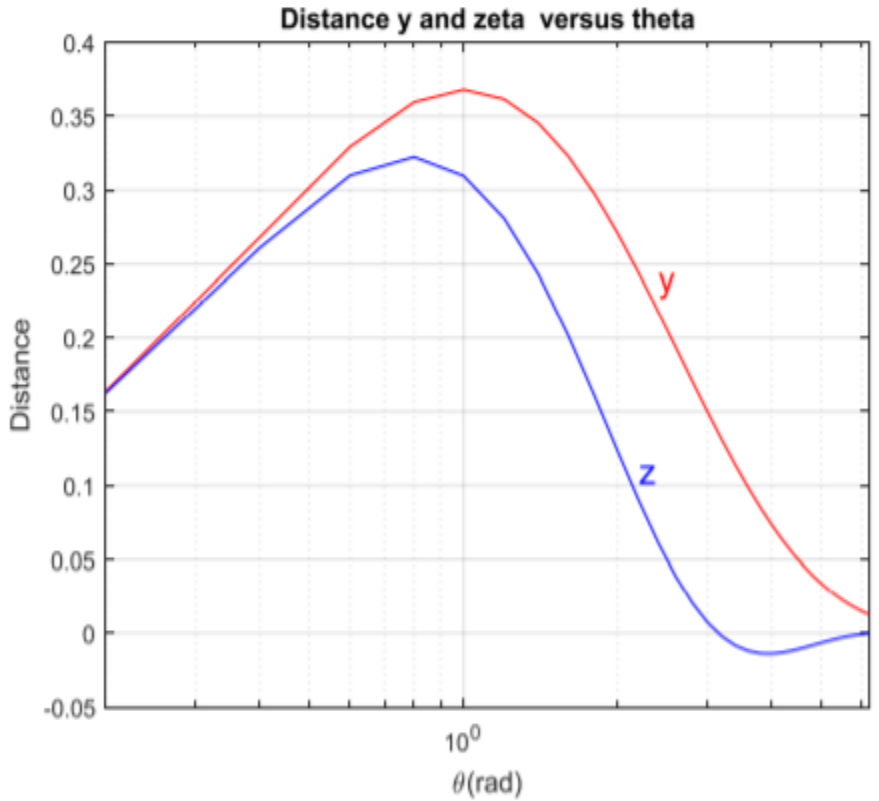
```
clear ; clf ;
theta = 0:0.2:2*pi ;
y = theta.*exp(-theta) ;
z = sin(theta).*exp(-theta) ;
plot(theta , y , 'r' , theta , z , 'b')
xlabel('\theta(rad)') ; ylabel('Distance')
title('Distance y and zeta  versus theta')
text(2.4 , 0.24 , 'y' , 'fontname' ,
'arial','fontsize' , 14 , 'color', 'r')
text(2.2 , 0.11 , 'z' , 'fontname' ,
'symbol','fontsize' , 14 , 'color' , 'b')
grid
```



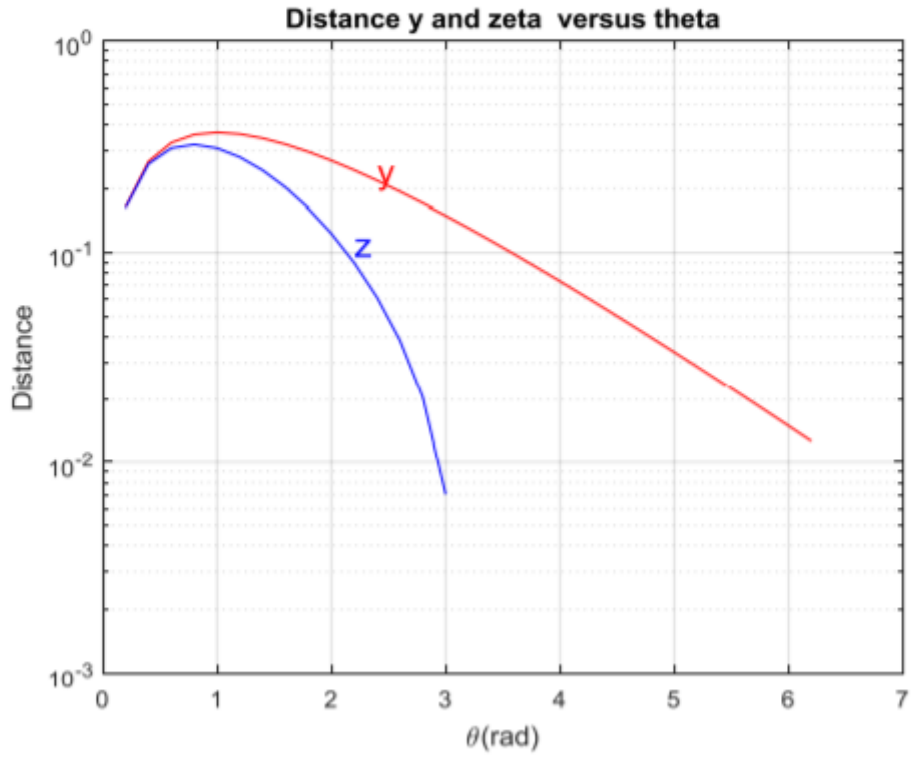
Logarithmic Plots

Replace *plot* command with *semilogx* or *semilogy* or *loglog*

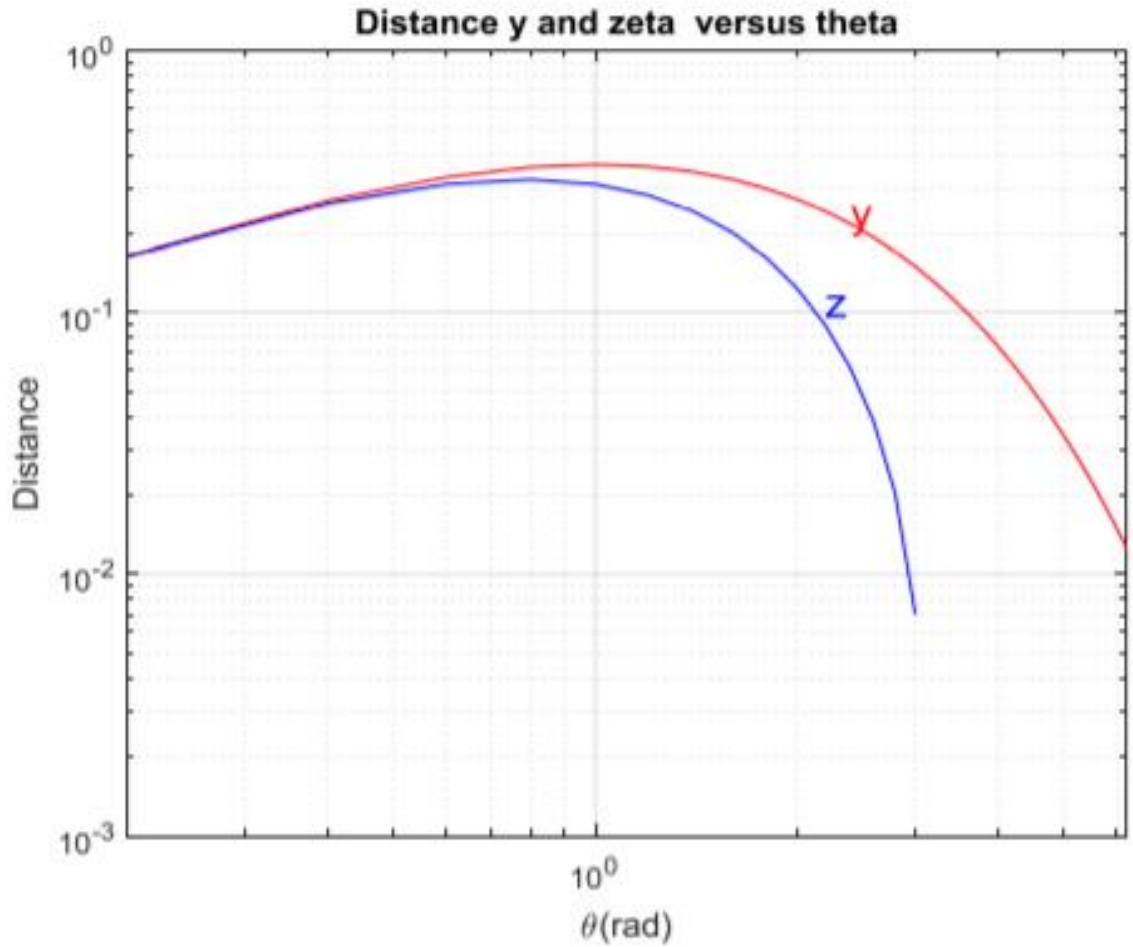
```
semilogx(theta , y , 'r' , theta , z , 'b')
```




```
semilogy(theta , y , 'r' , theta , z , 'b')
```



```
loglog(theta , y , 'r' , theta , z , 'b')
```

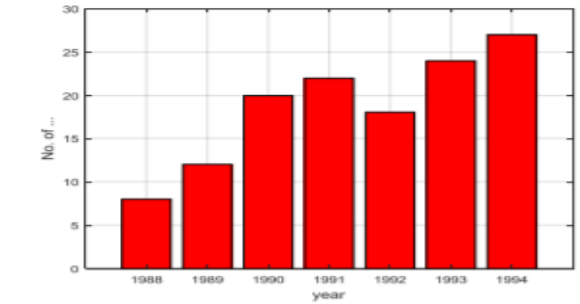


Specialized plot commands

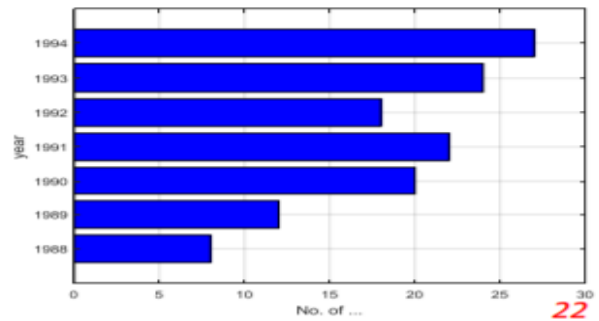
Command	Description
`bar(x,y)`	Creates a bar chart of `y` versus `x`.
`plotyy(x1,y1,x2,y2)`	Produces a plot with two *y*-axes, `y1` on the left and `y2` on the right.
`polar(theta,r,'type')`	Produces a polar plot from the polar coordinates `theta` and `r`, using the line type, data marker, and colors specified in the string `type`.
`stairs(x,y)`	Produces a stairs plot of `y` versus `x`.
`stem(x,y)`	Produces a stem plot of `y` versus `x`.

Bar Plots

```
clc; clear, close all
x = 1988:1994;
y =[ 8 12 20 22 18 24 27];
bar(x, y, 'r')
xlabel('year')
ylabel('No. of ...')
grid
```



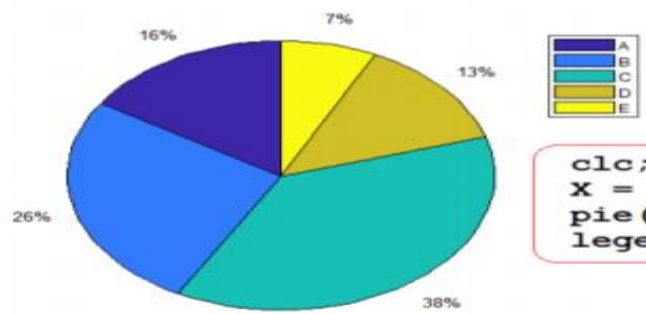

```
clc; clear, close all
x = 1988:1994;
y=[ 8 12 20 22 18 24 27];
barh(x, y , 'b')
xlabel('No. of ...')
ylabel('year')
grid
```



Pie Plot

Pie charts are useful for visualizing the relative sizes of different but related quantities.
For example, the table below shows the grades that were assigned to a class.
The data is used to create the pie chart that follows.

Grade	A	B	C	D	E
Number of students	11	18	26	9	5




```
clc; clear, close all
X = [11 18 26 9 5];
pie(X)
legen('A','B','C','D','E')
```