

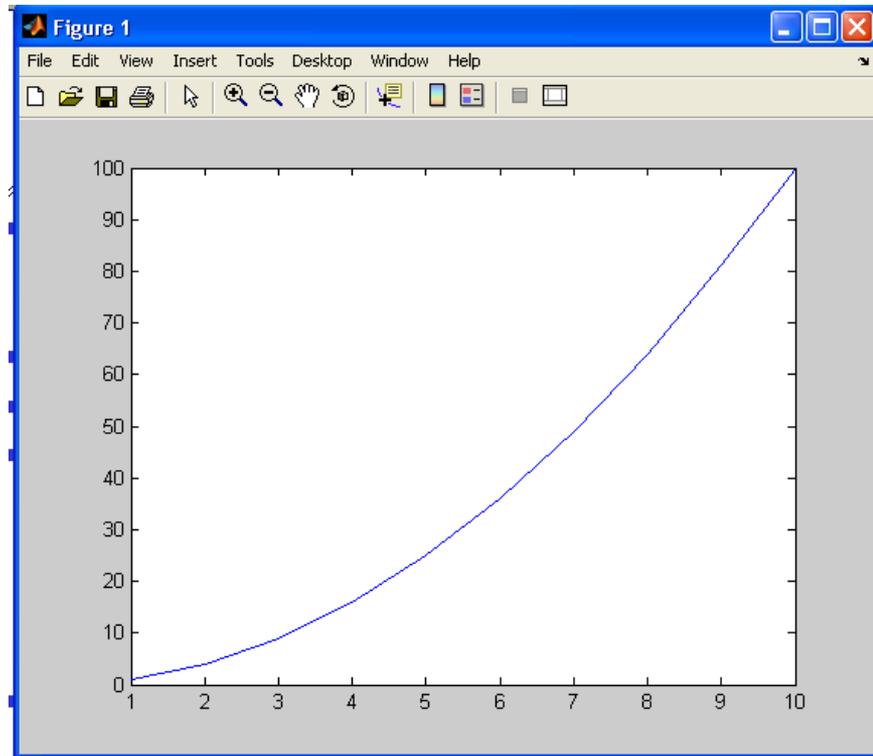
Plotting

Plotting

- To plot a data set, just create two vectors containing the x and y values to be plotted
- Use `plot` function
- When plot function is executed, MATLAB opens a Figure Window and displays the plot in that window.
- Any pair of vectors can be plotted versus each other as long as both vectors have the same length

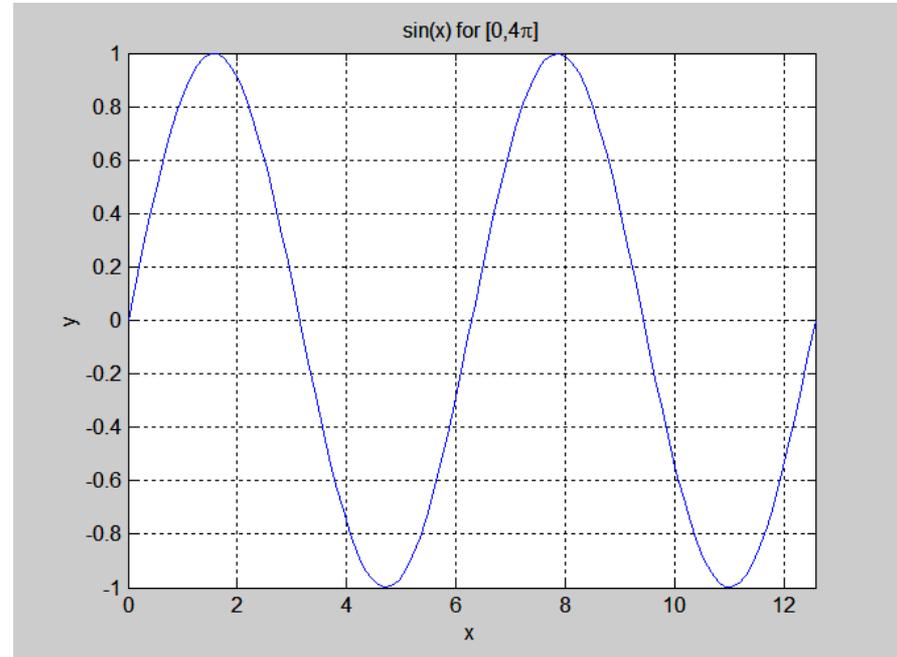
Plotting

- The result of a plot function call is not a finished product, since there are no titles, axis labels, or grid lines on the plot.
- `x=1:10;`
- `>> y=x.^2;`
- `>> plot(x,y)`
- The details can be added with the title, xlabel, ylabel, grid on/off and legend functions



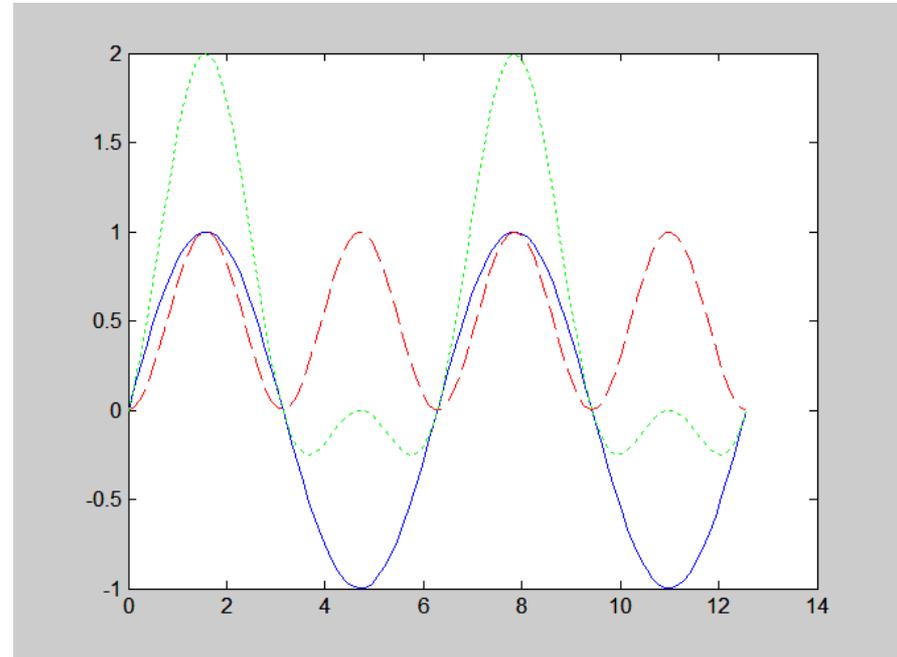
Plotting

```
x = linspace(0, 4*pi);  
y = sin(x);  
plot(x,y);  
title( 'sin(x) for [0,4\pi]' );  
xlabel( 'x' );  
ylabel( 'y' );  
grid on;  
axis( [ 0 4*pi -1 1 ] );
```



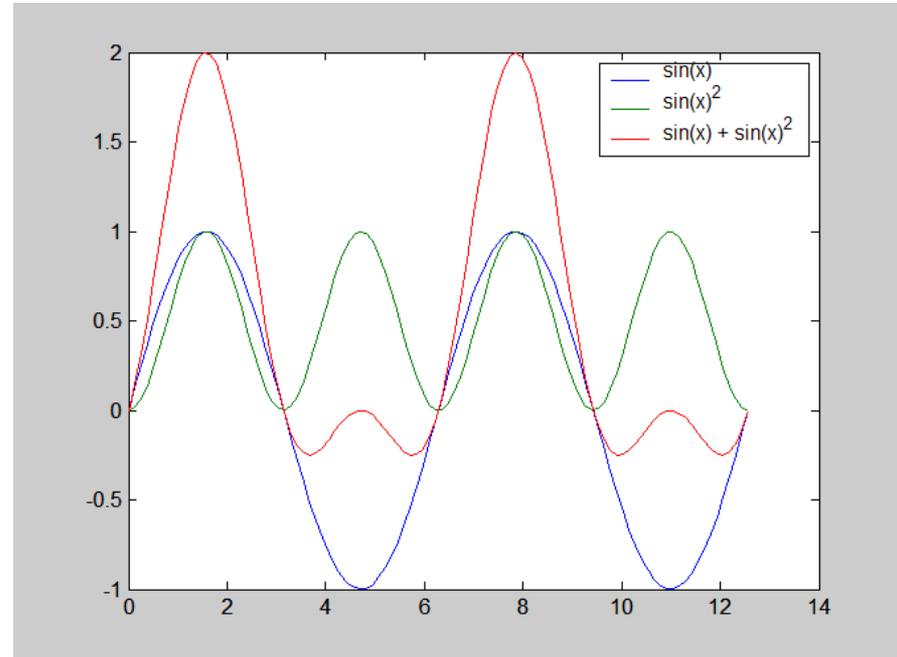
Plotting: Multiple Graphs

```
x = linspace(0, 4*pi);  
y1 = sin(x);  
y2 = sin(x) .^ 2;  
y3 = y1 + y2;  
plot(x,y1,'b-');  
hold on;  
plot(x,y2,'r--');  
plot(x,y3,'g:');  
hold off;
```



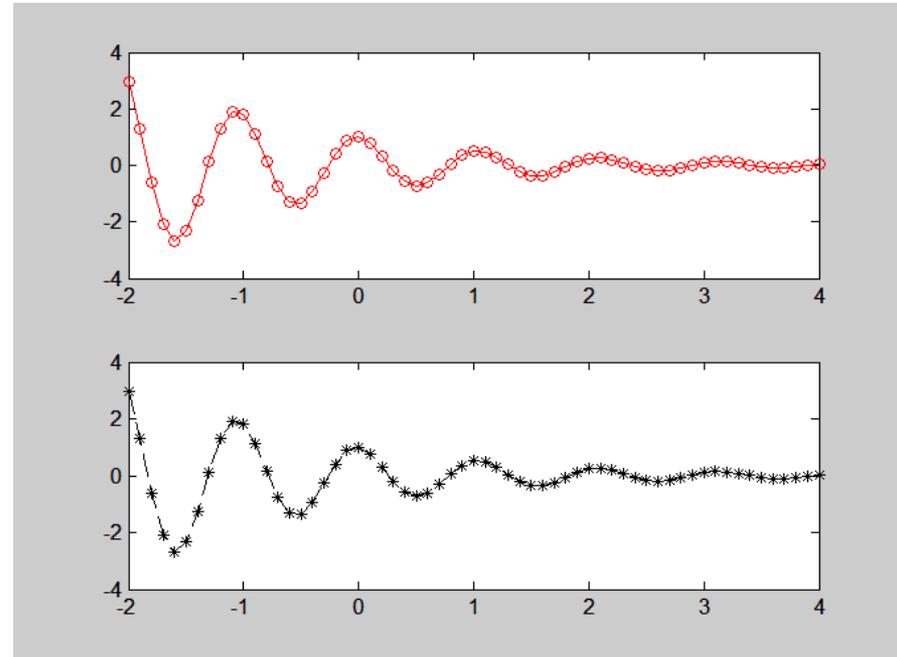
Plotting: Multiple Graphs

```
x = linspace(0, 4*pi);  
y1 = sin(x);  
y2 = sin(x) .^ 2;  
y3 = y1 + y2;  
plot(x,y1,x,y2,x,y3);  
legend( 'sin(x)', ...  
        'sin(x)^2', ...  
        'sin(x) + sin(x)^2' );
```



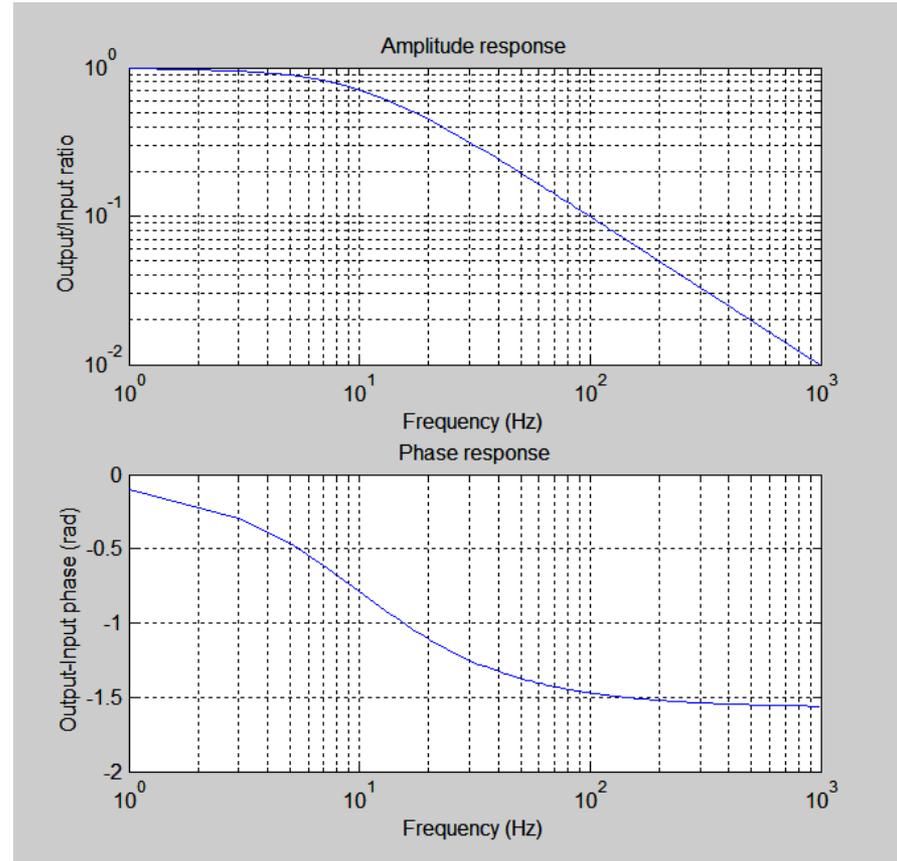
Plotting: Subplots

```
x = -2:0.1:4;  
y = 3.5 .^ (-0.5*x) .* ...  
    cos(6*x);  
figure(1); % opens empty figure  
           window  
subplot(2,1,1);  
plot(x,y,'r-o');  
subplot(2,1,2);  
plot(x,y,'k--*');
```



Plotting: Logarithmic Plots

```
r = 16000; c = 1.0e-6;
f = 1:2:1000;
res = 1 ./ ( 1 + j*2*pi*f*r*c );
amp = abs(res);
phase = angle(res);
subplot(2,1,1);
loglog(f,amp);
title( 'Amplitude response' );
xlabel( 'Frequency (Hz)' );
ylabel( 'Output/Input ratio' );
grid on;
subplot(2,1,2);
semilogx(f,phase);
title( 'Phase response' );
xlabel( 'Frequency (Hz)' );
ylabel( 'Output-Input phase (rad)' );
grid on;
```



Plotting Summary

- `plot(x,y)`
linear plot of vector y vs. vector x
- `title('text'), xlabel('text'), ylabel('text')`
labels the figure, x-axis and y-axis
- `grid on/off`
adds/removes grid lines
- `hold on/off`
allows/disallows adding subsequent graphs to the current graph

Plotting Summary

- `legend('string1', 'string2', 'string3', ...)`
adds a legend using the specified strings
- `v = axis`
returns a row vector containing the scaling for the current plot
- `axis([xmin xmax ymin ymax])`
sets axes' limits

Plotting Summary

line color		line marker		line style	
b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star		
y	yellow	s	square		
k	black	d	diamond		
		v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

Plotting Summary

- `semilogy(x,y)`, `semilogx(x,y)`, `loglog(x,y)`
logarithmic plots of vector y vs. vector x
- `figure(k)`
makes figure k the current figure
- `subplot(m,n,p)`
breaks the figure window into an m -by- n matrix of small axes and selects the p^{th} axes for the current plot
- `clf`
clears current figure

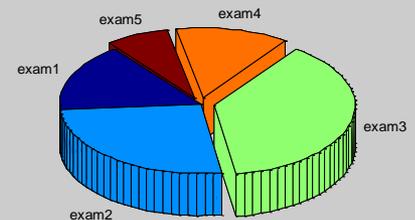
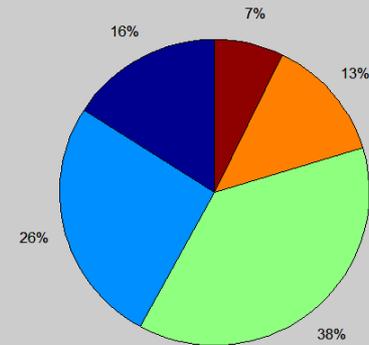
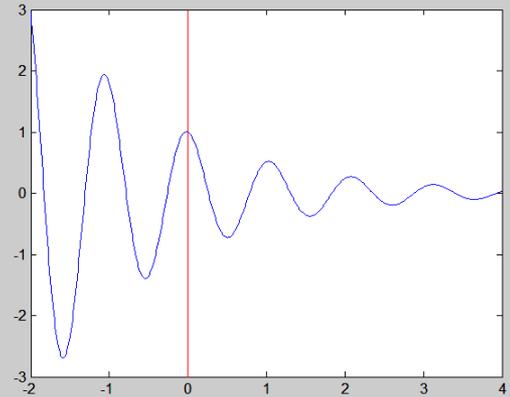
Plotting Examples

■ Line plot

```
>> x = -2:0.01:4;
>> y = 3.5.^...
(-0.5*x) .* cos(6*x);
>> plot(x,y);
>> line([0 0], [-3 3], 'color', 'r');
```

■ Pie plot

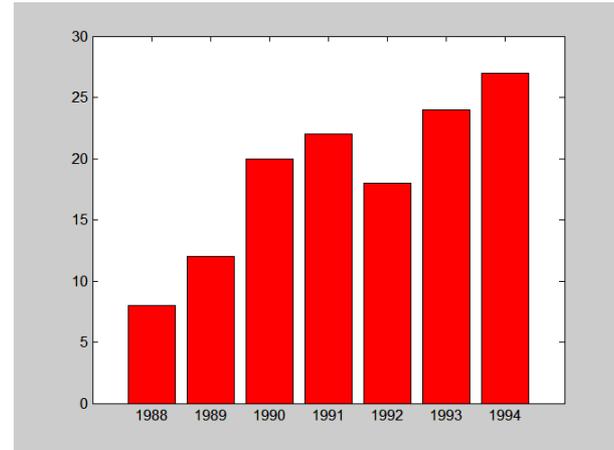
```
>> grades = [ 11 18 26 9 5 ];
>> pie(grades); Specify slices moved
out from pie
>> pie3(grades, [0 0 0 1 0]
, { 'exam1', 'exam2', 'exam3', 'exam4',
exam5' });
```



Plotting Examples

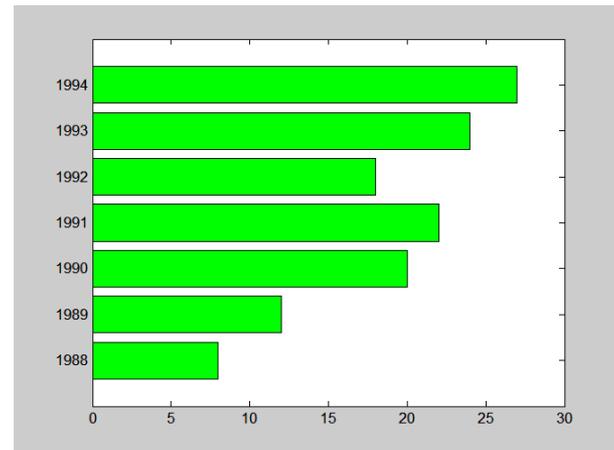
- Vertical bar plot

```
y = 1988:1994;  
s = [ 8 12 20 22 18 24 27 ];  
bar(y,s,'r');
```



- Horizontal bar plot

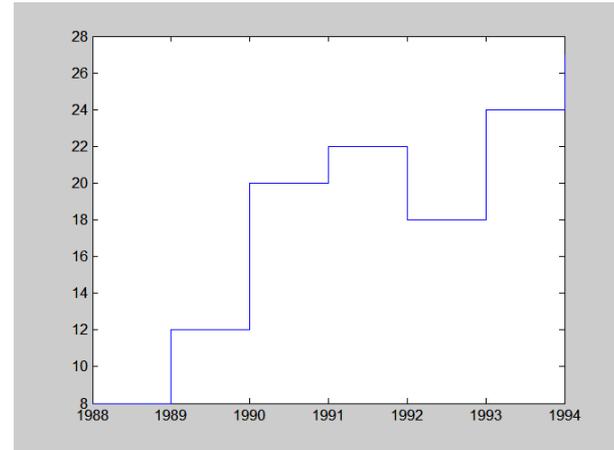
```
y = 1988:1994;  
s = [ 8 12 20 22 18 24 27 ];  
barh(y,s,'g');
```



Plotting Examples

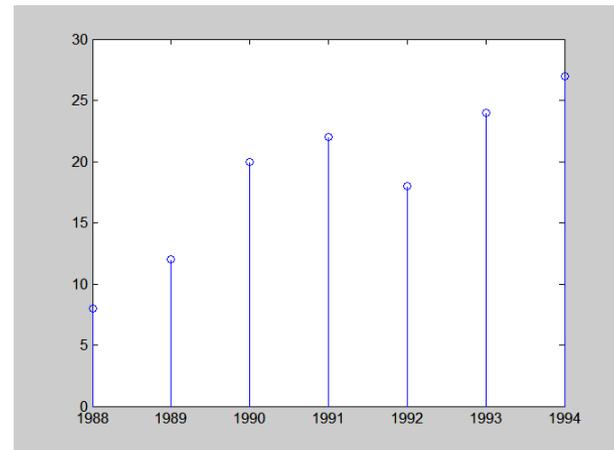
- Stairs plot

```
y = 1988:1994;  
s = [ 8 12 20 22 18 24 27 ];  
stairs(y,s);
```



- Stem plot

```
y = 1988:1994;  
s = [ 8 12 20 22 18 24 27 ];  
stem(y,s);
```



Plotting Examples

- Histogram-
- $N = \text{hist}(Y)$ bins the elements of Y into 10 equally spaced containers and returns the number of elements in each container.

```
x = randn(1,100);  
hist(x,10);
```

```
hist(x,20);
```

