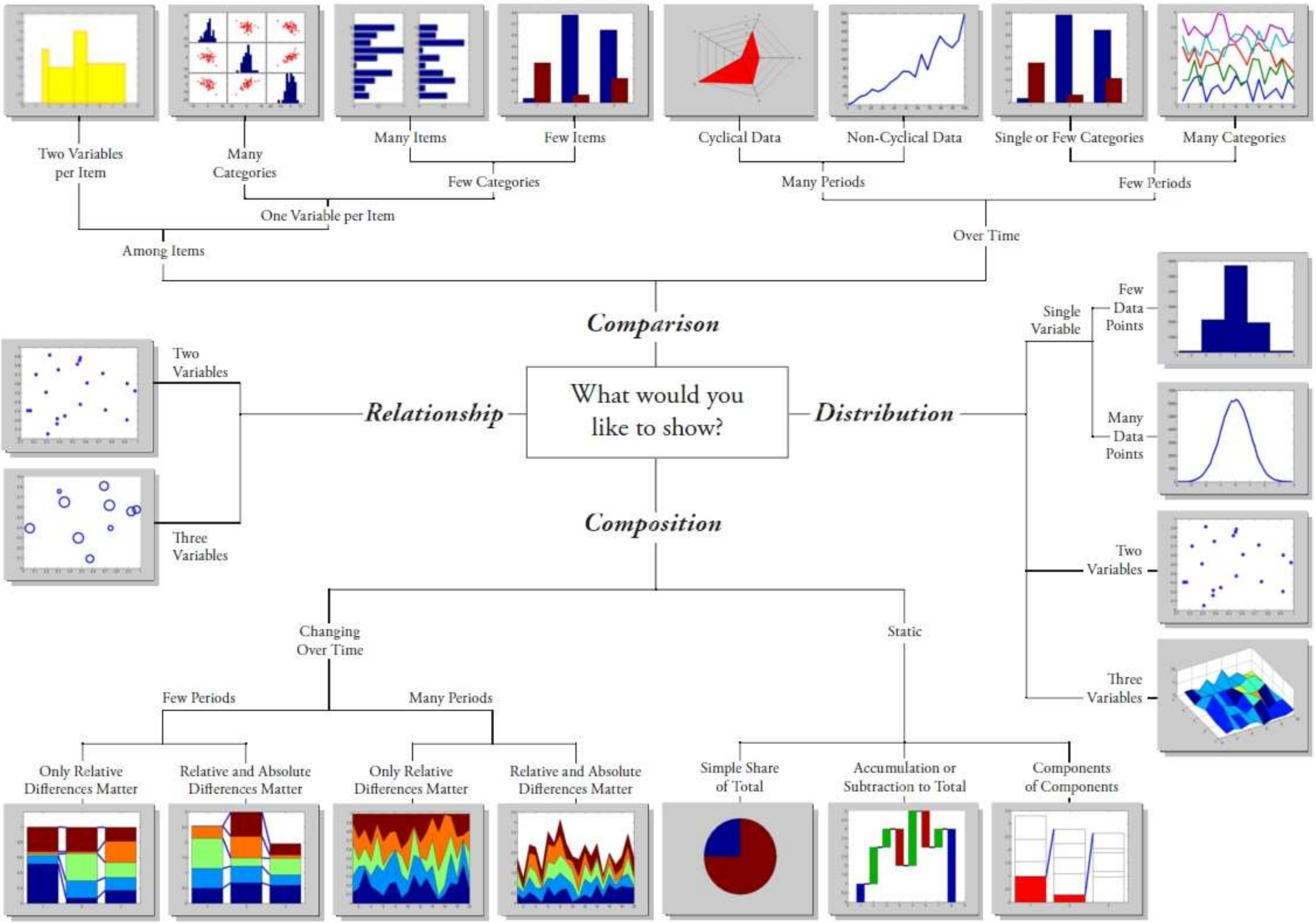
APPLICATIONS OF MATLAB IN ENGINEERING

Yan-Fu Kuo
Dept. of Bio-industrial Mechatronics Engineering
National Taiwan University

Fall 2015

Today:

- Advanced 2D plots
- Color space
- 3D plots

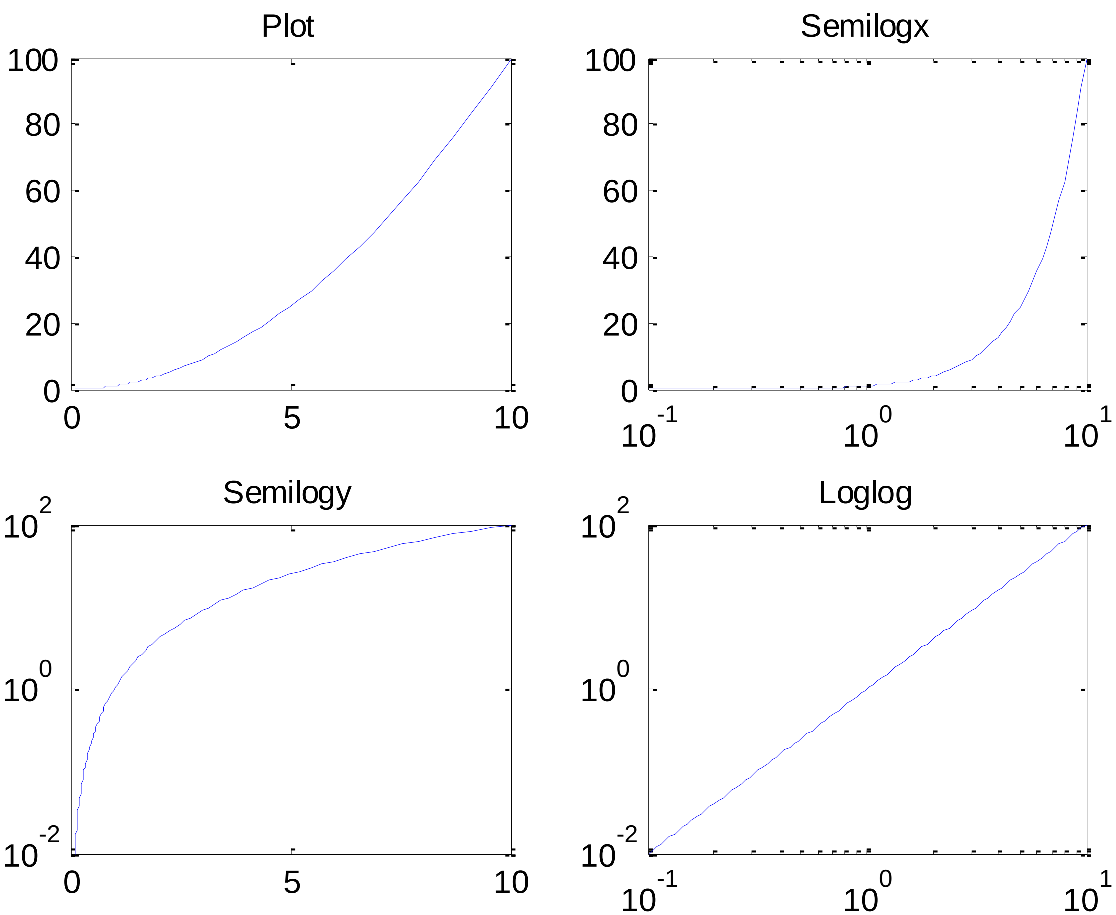

What would you like to show?
Comparison
Relationship
Distribution
Composition
Among Items
Over Time
Two Variables per Item
One Variable per Item
Many Categories
Few Categories
Many Items
Few Items
Many Periods
Few Periods
Cyclical Data
Non-Cyclical Data
Single or Few Categories
Many Categories
Two Variables
Three Variables
Single Variable
Few Data Points
Many Data Points
Two Variables
Three Variables
Changing Over Time
Static
Few Periods
Many Periods
Only Relative Differences Matter
Relative and Absolute Differences Matter
Only Relative Differences Matter
Relative and Absolute Differences Matter
Simple Share of Total
Accumulation or Subtraction to Total
Components of Components

Special Plots

Function	Description
`loglog`	Graph with logarithmic scales for both axes
`semilogx`	Graph with a logarithmic scale for the x-axis and a linear scale for the y-axis
`semilogy`	Graph with a logarithmic scale for the y-axis and a linear scale for the x-axis
`plotyy`	Graph with y-tick labels on the left and right side

Function	Description
`hist`	Histogram plot
`bar`	Bar graph
`pie`	Pie chart
`polar`	Polar coordinate plot

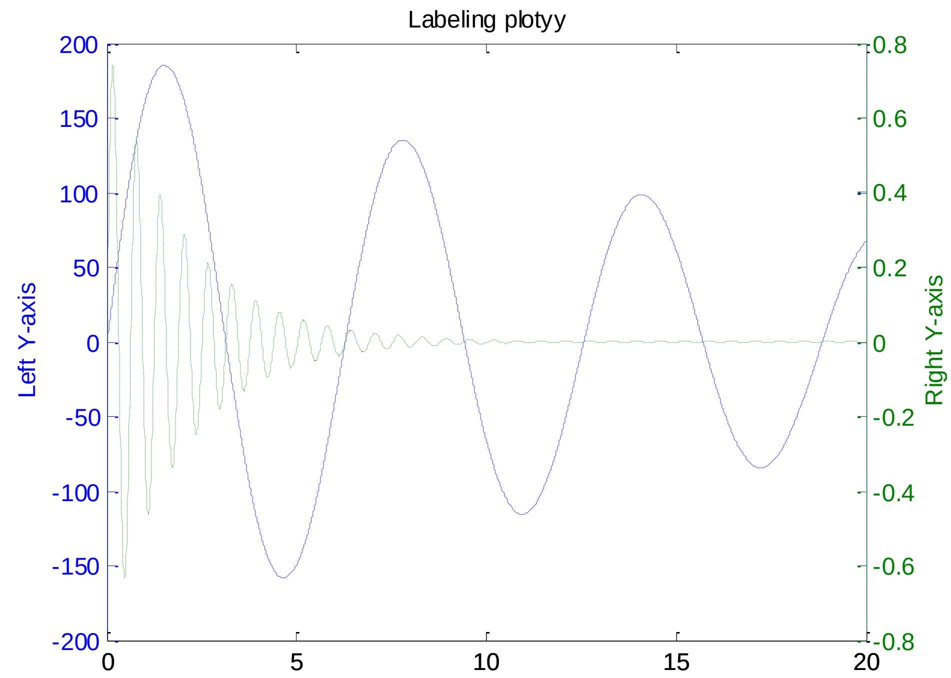
Logarithm Plots




```
x = logspace(-1,1,100);
y = x.^2;
subplot(2,2,1);
plot(x,y);
title('Plot');
subplot(2,2,2);
semilogx(x,y);
title('Semilogx');
subplot(2,2,3);
semilogy(x,y);
title('Semilogy');
subplot(2,2,4);
loglog(x, y);
title('Loglog');
```

```
set(gca,'XGrid','on');
```

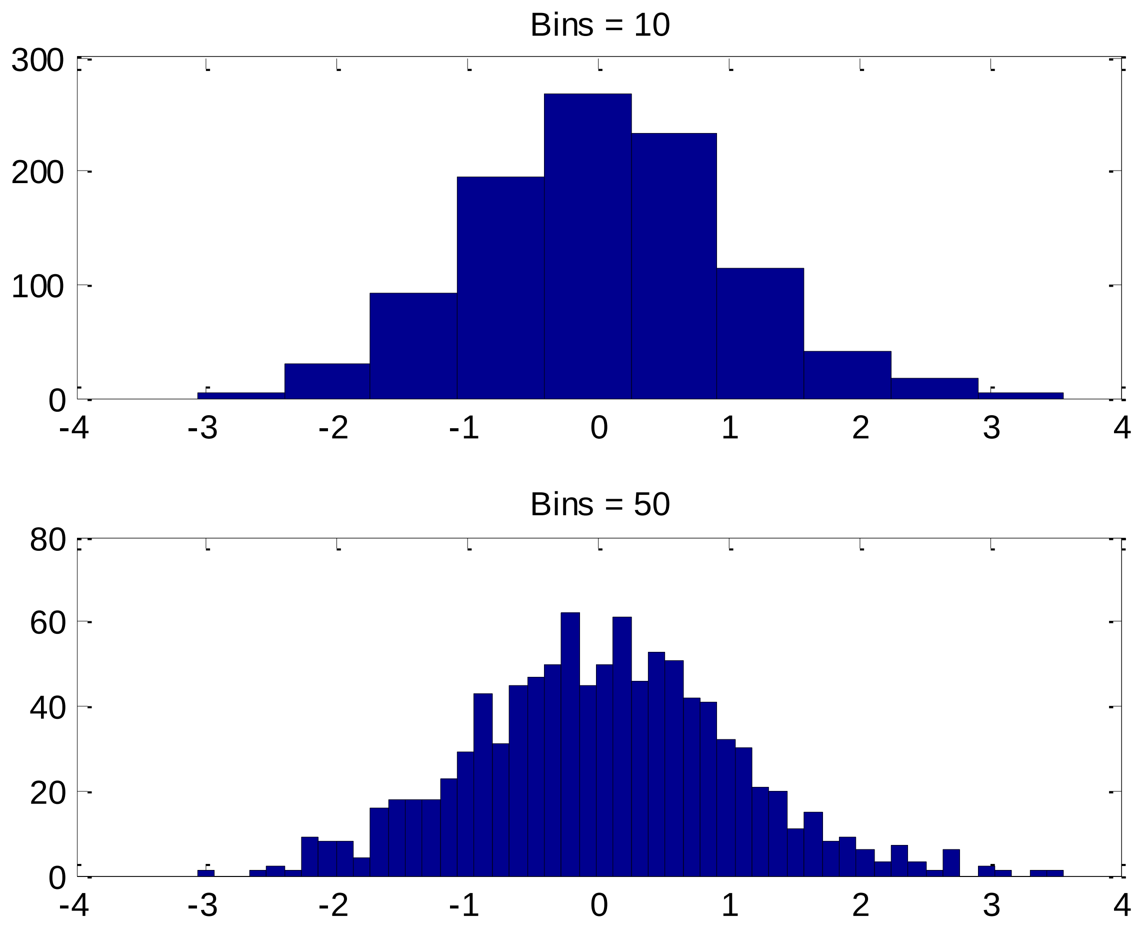
plotyy()




```
x = 0:0.01:20;
y1 = 200*exp(-0.05*x).*sin(x);
y2 = 0.8*exp(-0.5*x).*sin(10*x);
[AX,H1,H2] = plotyy(x,y1,x,y2);
set(get(AX(1),'Ylabel'),'String','Left Y-axis')
set(get(AX(2),'Ylabel'),'String','Right Y-axis')
title('Labeling plotyy');
set(H1,'LineStyle','--');  set(H2,'LineStyle',':');
```

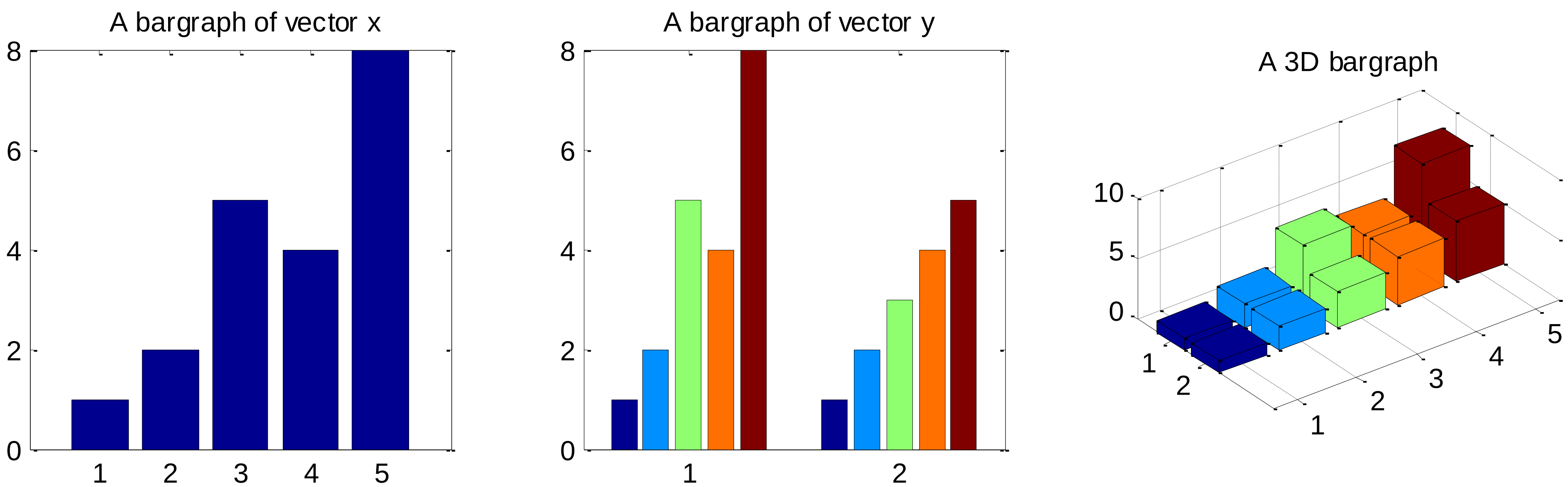
Histogram

```
y = randn(1,1000);
subplot(2,1,1);
hist(y,10);
title('Bins = 10');
subplot(2,1,2);
hist(y,50);
title('Bins = 50');
```

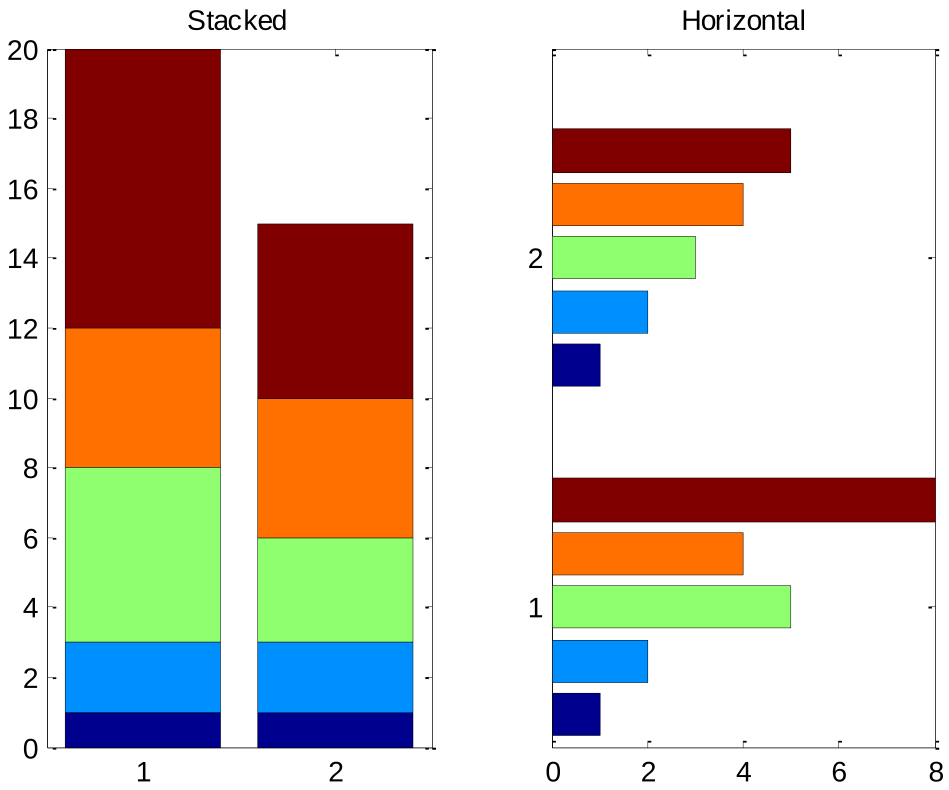


Bar Charts

```
x = [1 2 5 4 8];  y = [x;1:5];
subplot(1,3,1);  bar(x);  title('A bargraph of vector x');
subplot(1,3,2);  bar(y);  title('A bargraph of vector y');
subplot(1,3,3);  bar3(y);  title('A 3D bargraph');
```



Stacked and Horizontal Bar Charts

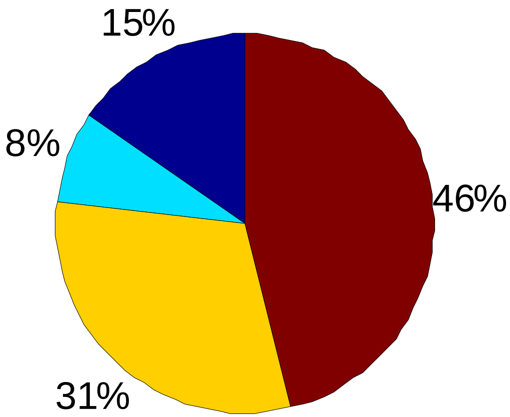


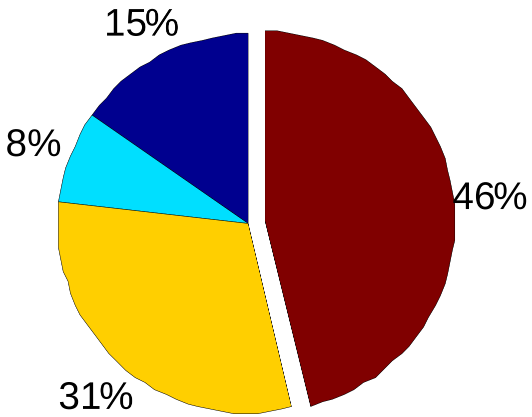

```
x = [1 2 5 4 8];
y = [x;1:5];
subplot(1,2,1);
bar(y,'stacked');
title('Stacked');

subplot(1,2,2);
barh(y);
title('Horizontal');
```

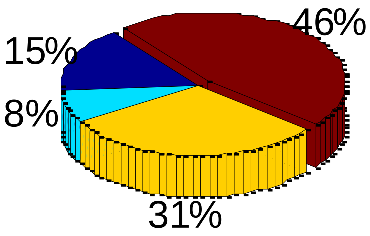
- Exercise: stack the horizontal bar chart

Pie Charts

```
a = [10 5 20 30];
subplot(1,3,1);  pie(a);
subplot(1,3,2);  pie(a, [0,0,0,1]);
subplot(1,3,3);  pie3(a, [0,0,0,1]);
```



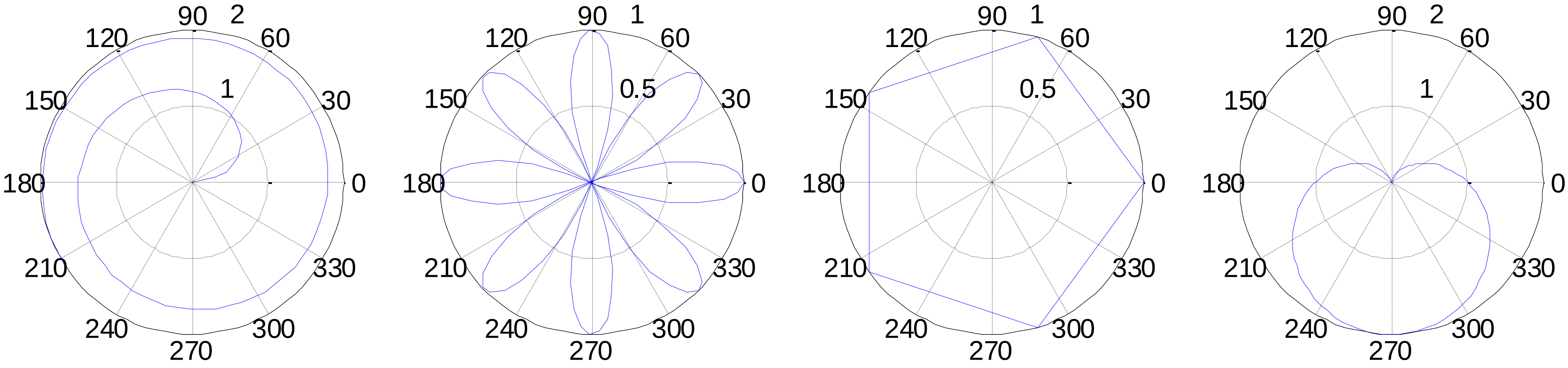




- Exercise: separate all the pieces in the pie chart

Polar Chart

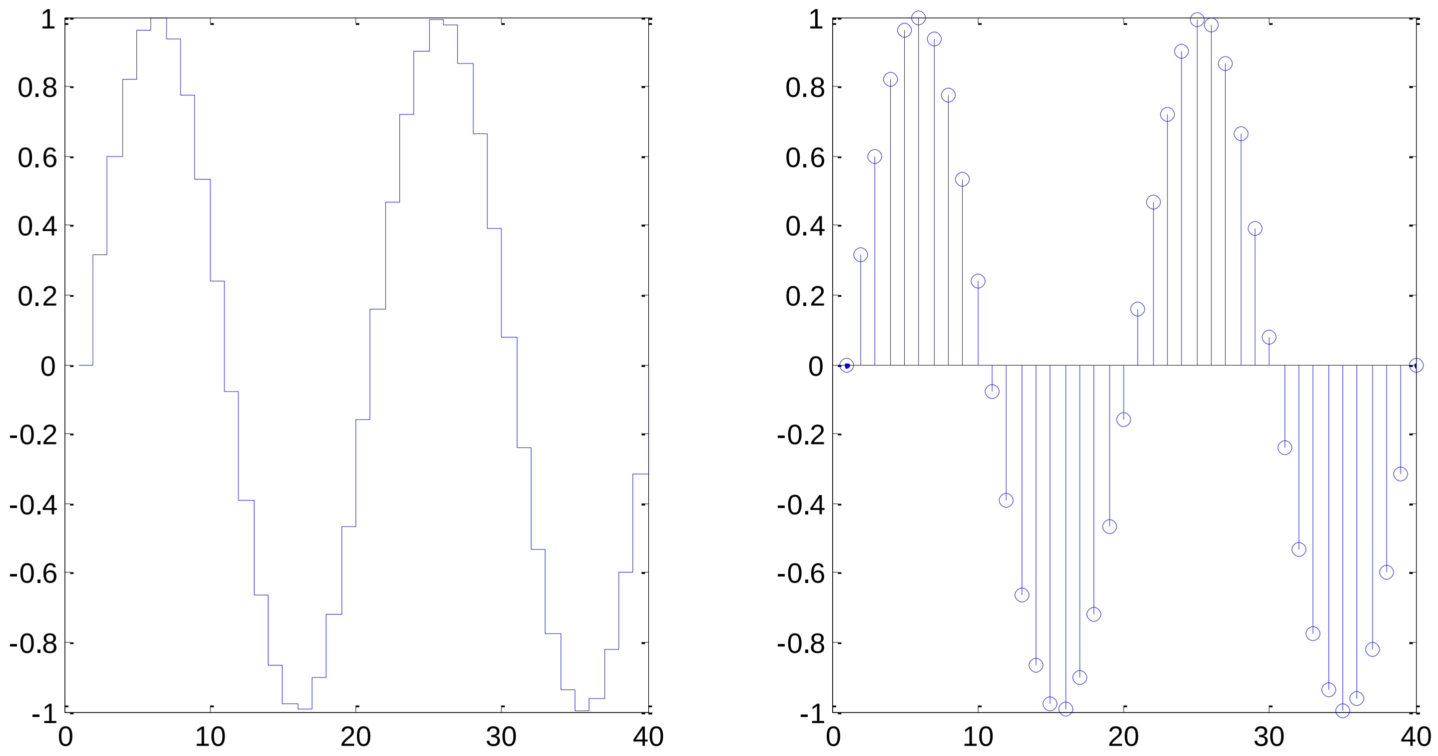
```
x = 1:100;  theta = x/10;  r = log10(x);
subplot(1,4,1);  polar(theta,r);
theta = linspace(0, 2*pi);  r = cos(4*theta);
subplot(1,4,2);  polar(theta, r);
theta = linspace(0, 2*pi, 6); r = ones(1,length(theta));
subplot(1,4,3);  polar(theta,r);
theta = linspace(0, 2*pi); r = 1-sin(theta);
subplot(1,4,4);  polar(theta , r);
```



- Exercise: plot a hexagon on a polar chart

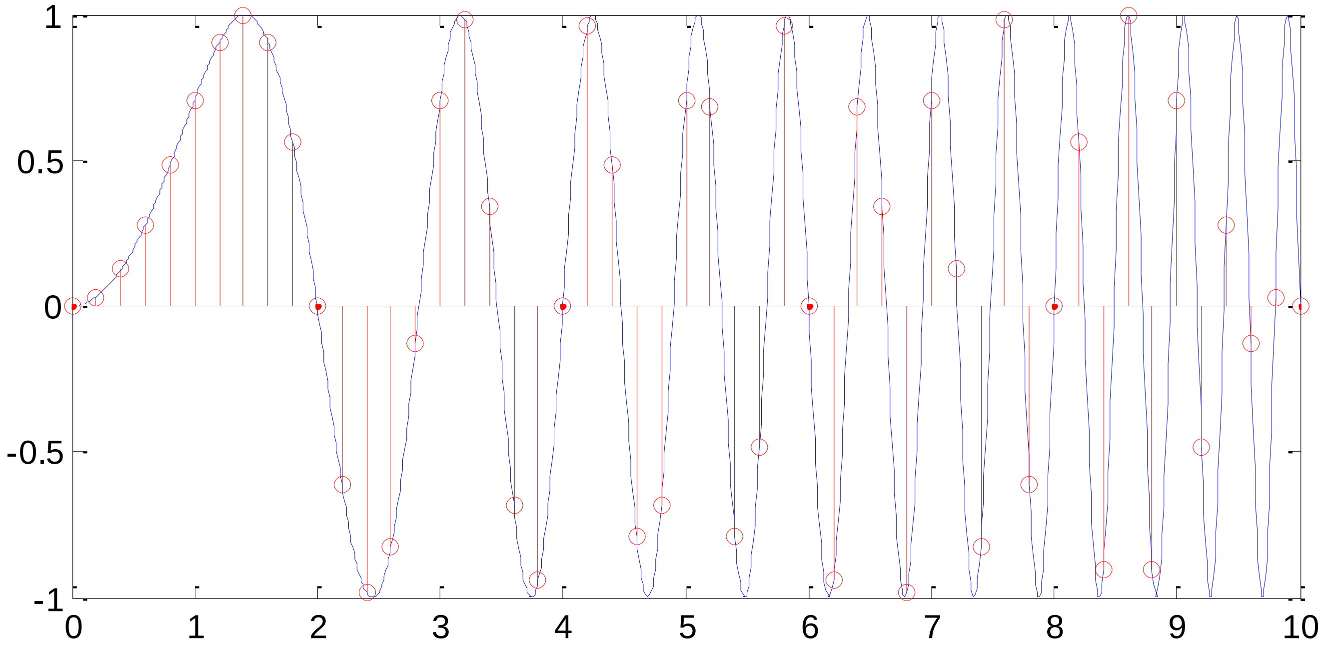
Stairs and Stem Charts

```
x = linspace(0, 4*pi, 40);  y = sin(x);
subplot(1,2,1);  stairs(y);
subplot(1,2,2);  stem(y);
```



Exercise

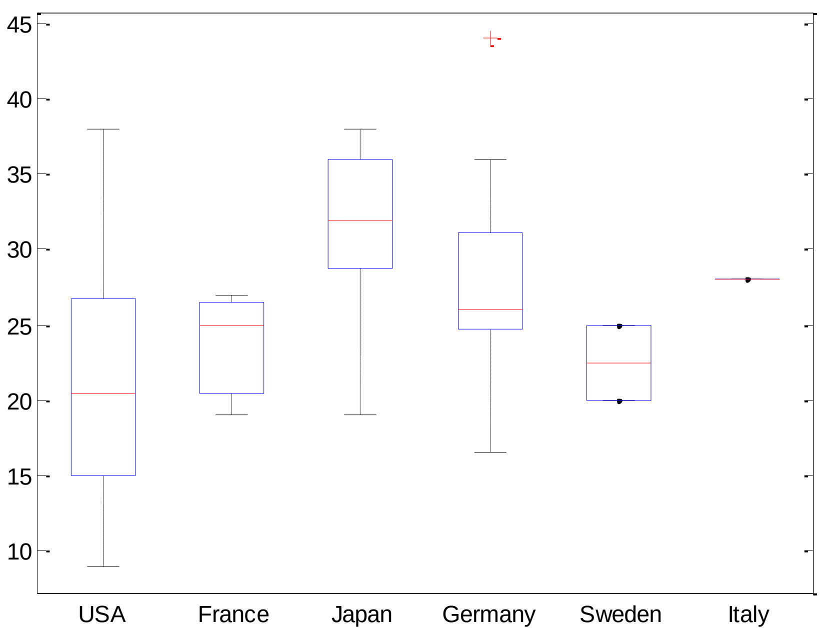
- Plot a function: $f(t) = \sin(\frac{\pi t^2}{4})$
- Add the points sampled at 5 Hz using `stem()`

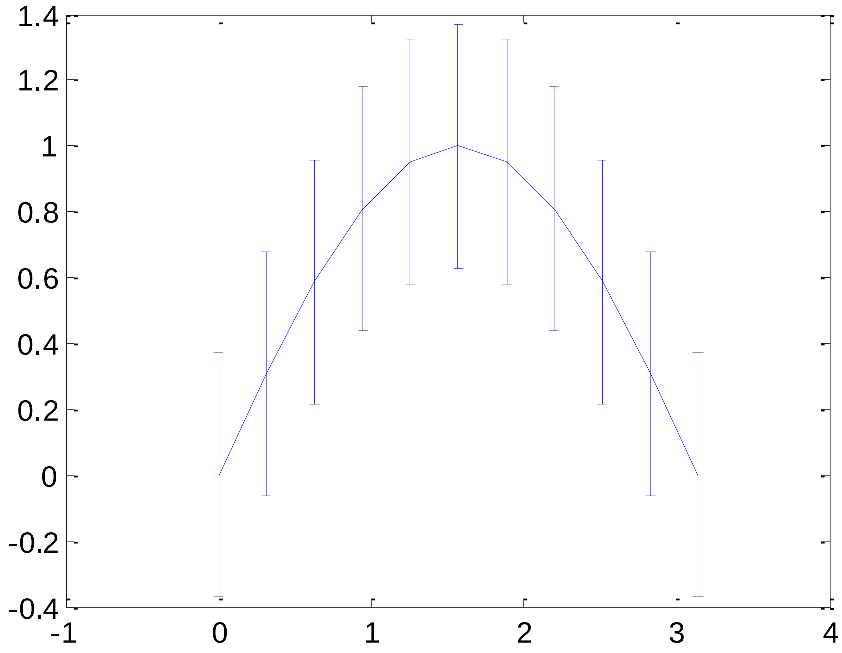


Boxplot and Error Bar

```
load carsmall
boxplot(MPG, Origin);
```

```
x=0:pi/10:pi; y=sin(x);
e=std(y)*ones(size(x));
errorbar(x,y,e)
```





fill()

- Stop sign

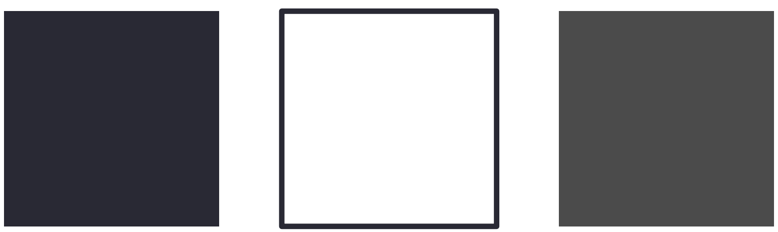
```
t =(1:2:15)'*pi/8;  x = sin(t);  y = cos(t);
fill(x,y,'r');  axis square off;
text(0,0,'STOP','Color', 'w', 'FontSize', 80, ...
    'FontWeight','bold', 'HorizontalAlignment', 'center');
```

Exercise

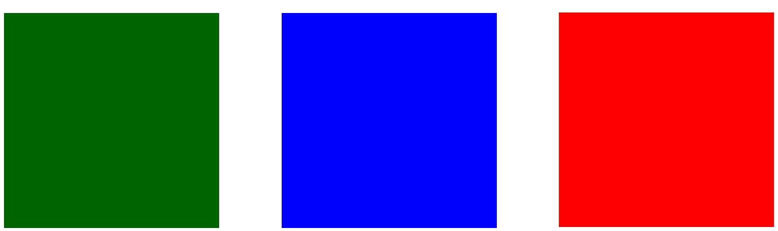
- Plot a wait sign

Color Space

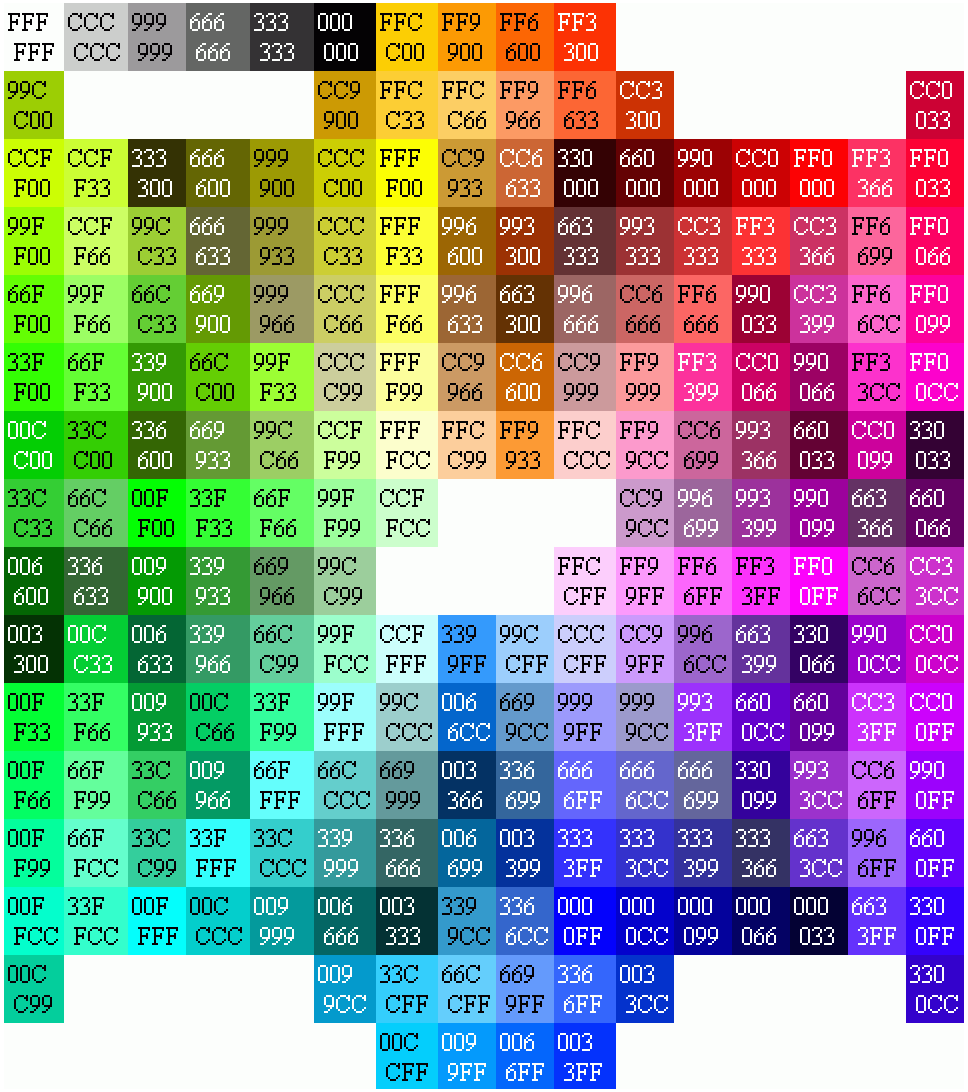
- [R G B]
  - 0 is minimum
  - 1 is maximum
  - 8-bit equivalence:



[0 0 0] [255 255 255] [75 75 75]

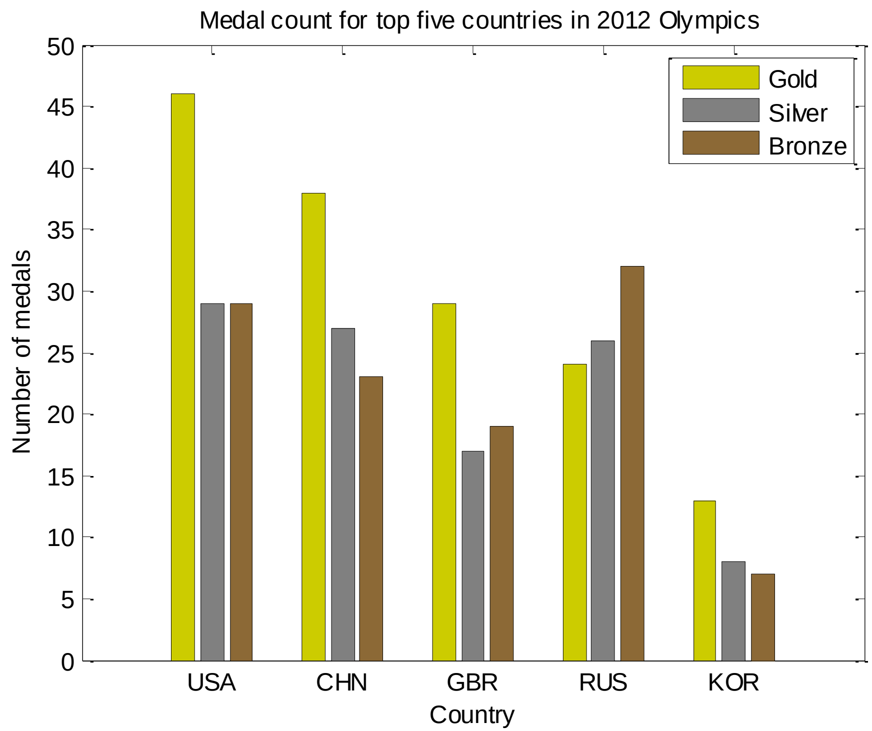


[0 255 0] [0 0 255] [255 0 0]



Exercise

```
G = [46 38 29 24 13];  S = [29 27 17 26 8];
B = [29 23 19 32 7];  h = bar(1:5, [G' S' B']);
title('Medal count for top 5 countries in 2012 Olympics');
ylabel('Number of medals');  xlabel('Country');
legend('Gold', 'Silver', 'Bronze')
```

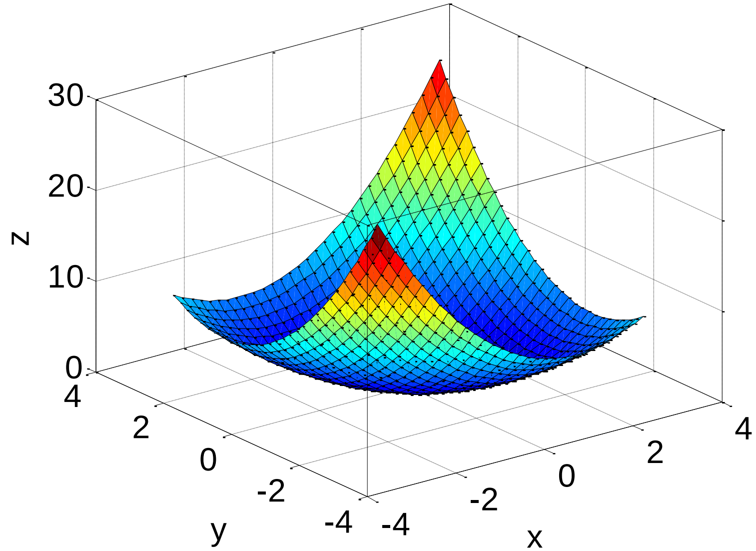


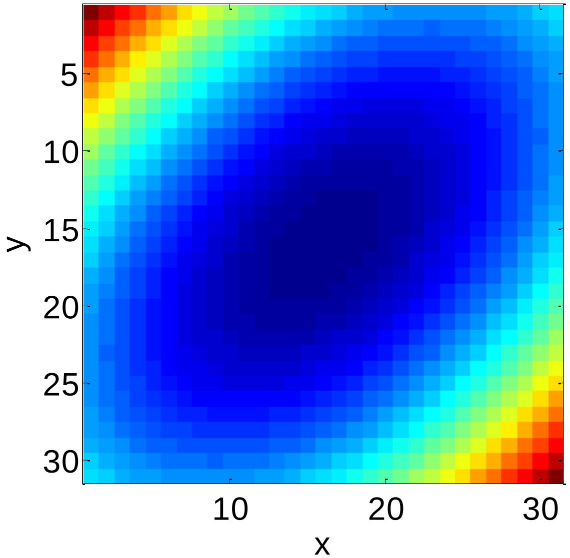
Visualizing Data as An Image: `imagesc()`

- Display values of a matrix as an "image"

```
[x, y] = meshgrid(-3:.2:3,-3:.2:3);
z = x.^2 + x.*y + y.^2;  surf( x, y, z);  box on;
set(gca,'FontSize', 16);  zlabel('z');
xlim([-4 4]);  xlabel('x');  ylim([-4 4]);  ylabel('y');
```

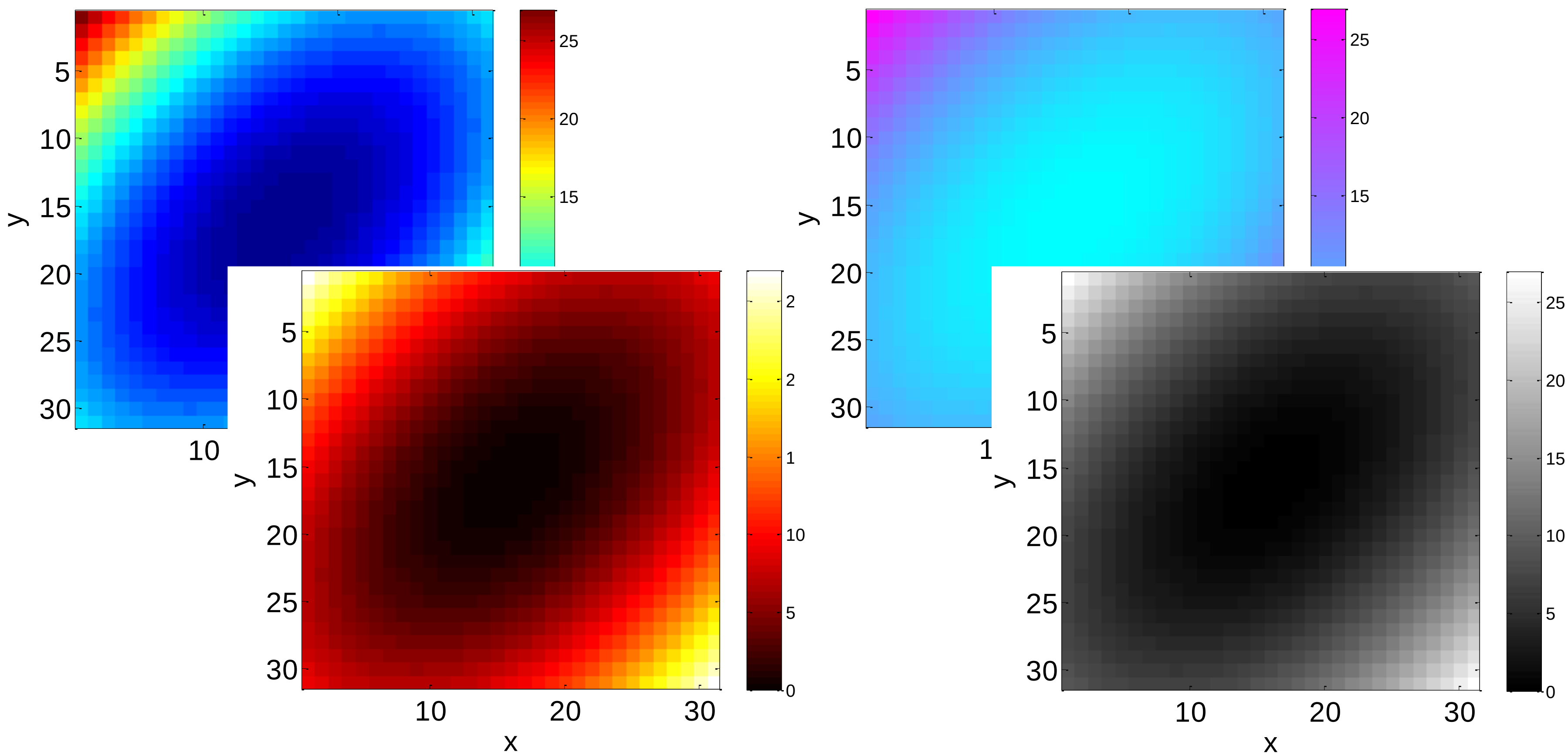
```
imagesc(z);  axis square;  xlabel('x');  ylabel('y');
```





Color Bar and Scheme

`colorbar;`	`colormap(cool);`
`colormap(hot);`	`colormap(gray);`



Built-in Colormaps

- Use built-in color maps:

```
colormap([Name])
```

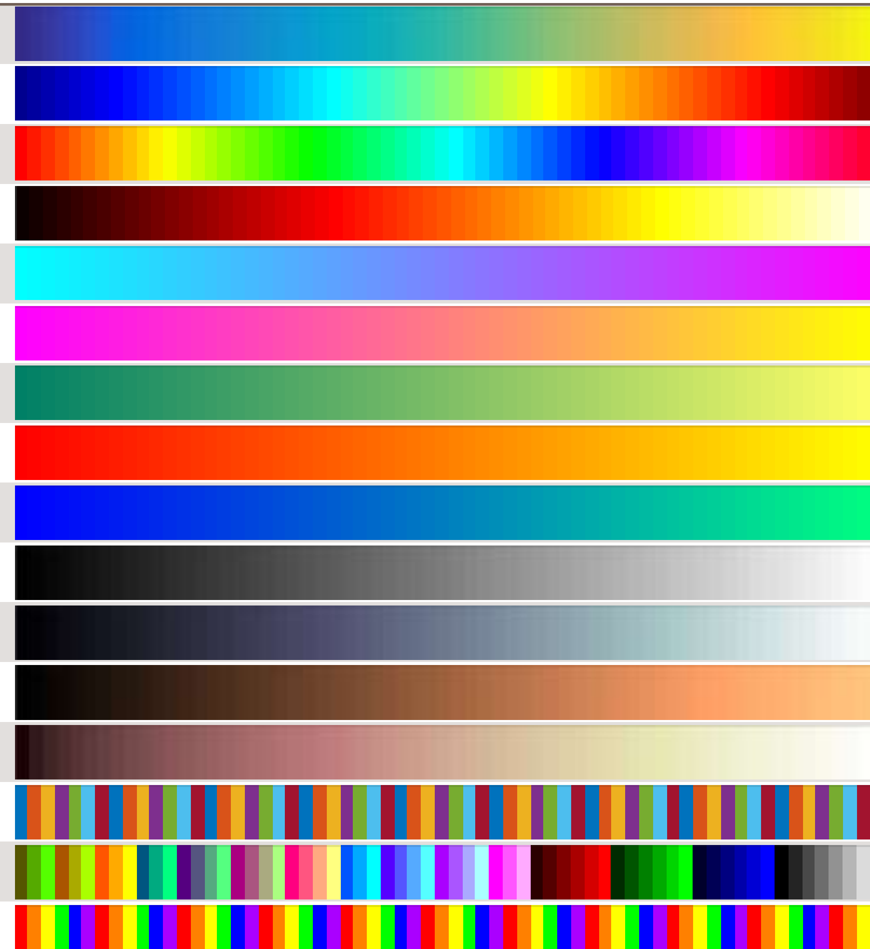
- A color map is a matrix of 256X3

```
a = colormap(prism)
```

- Use a customized color map:

```
a = ones(256,3);
colormap(a);
```

Name	Color Scale
parula	
jet	
hsv	
hot	
cool	
spring	
summer	
autumn	
winter	
gray	
bone	
copper	
pink	
lines	
colorcube	
prism	

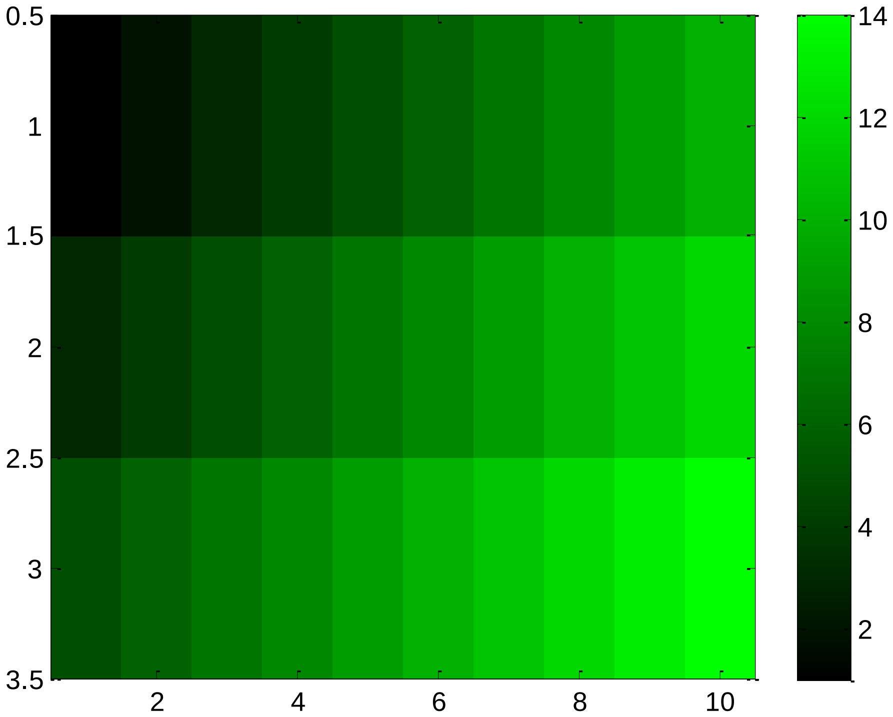


Exercise

- Create a custom green color map such that the output of the script below looks like:

```
x = [1:10; 3:12; 5:14];
imagesc(x);
colorbar;
```

```
map = zeros(256,3);
map(:,2) = (0:255)/255;
colormap(map);
```

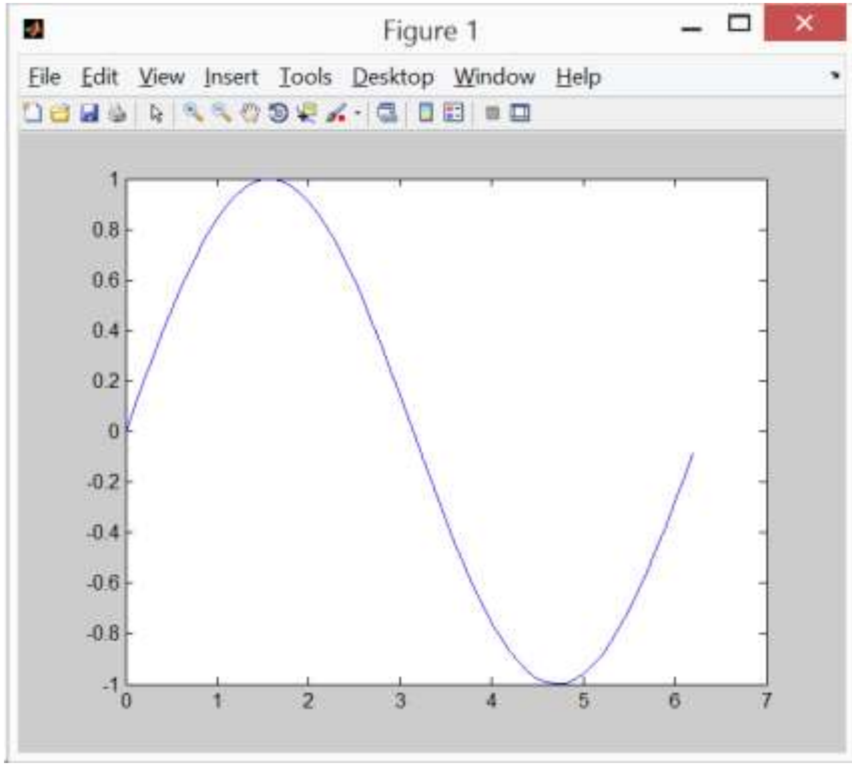


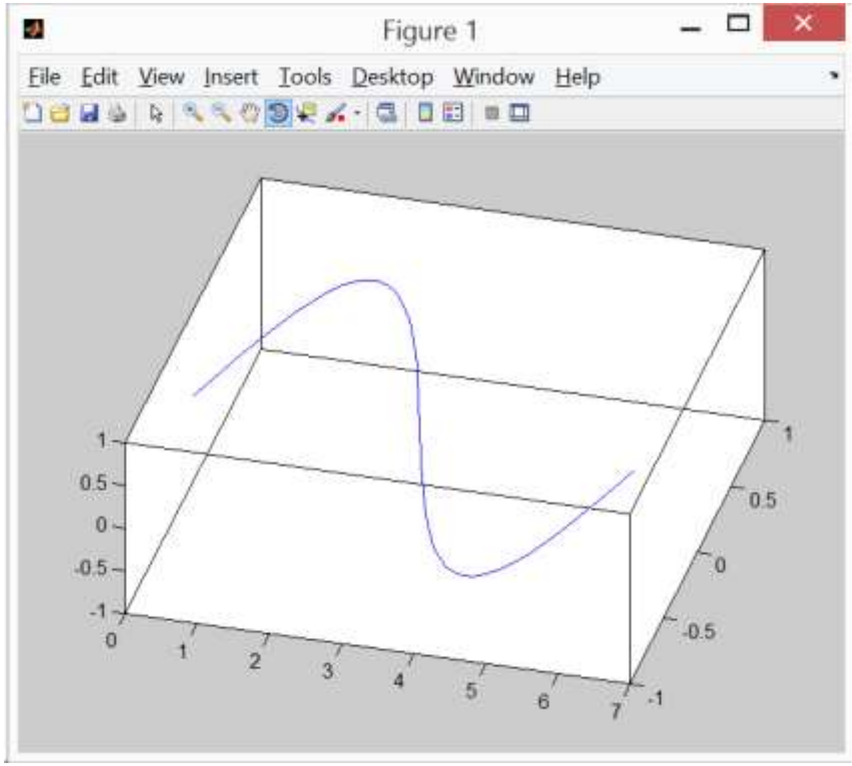
3D Plots

Function	Description
`plot3`	3-D line plot
`surf`	3-D shaded surface plot
`surfc`	Contour plot under a 3-D shaded surface plot
`surface`	Create surface object
`meshc`	Plot a contour graph under mesh graph
`contour`	Contour plot of matrix
`contourf`	Filled 2-D contour plot

2D vs. 3D

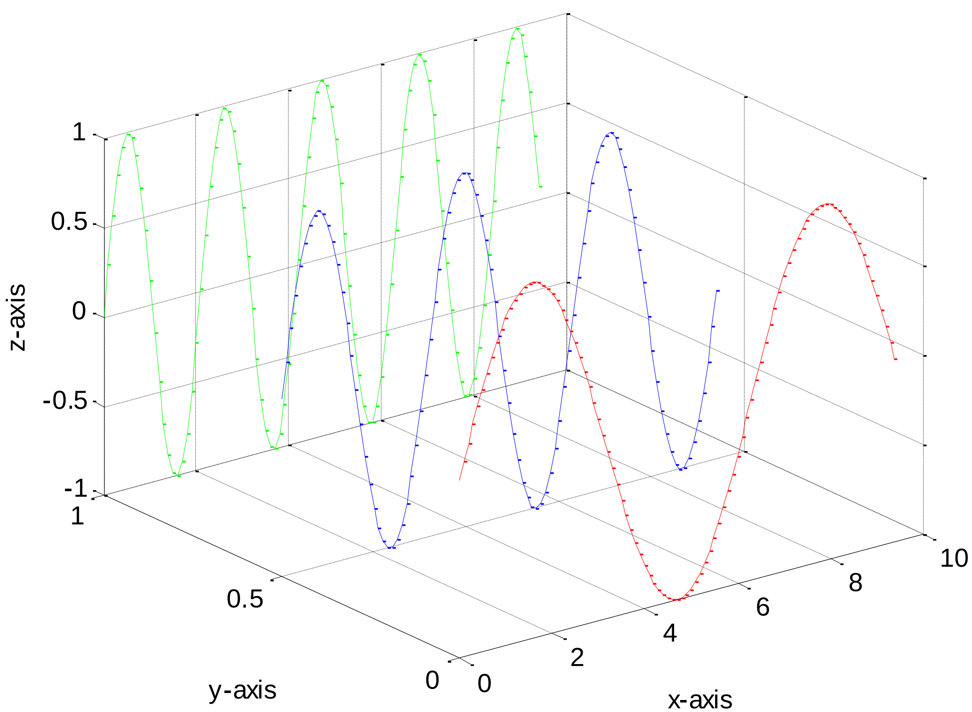
```
x=0:0.1:2*pi;
plot(x,sin(x));
```





plot3()

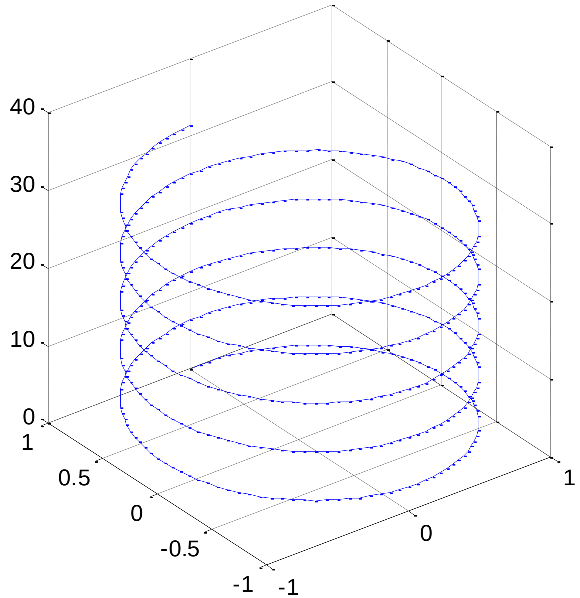
```
x=0:0.1:3*pi;  z1=sin(x);  z2=sin(2.*x);  z3=sin(3.*x);
y1=zeros(size(x));  y3=ones(size(x));  y2=y3./2;
plot3(x,y1,z1,'r',x,y2,z2,'b',x,y3,z3,'g');  grid on;
xlabel('x-axis');  ylabel('y-axis');  zlabel('z-axis');
```

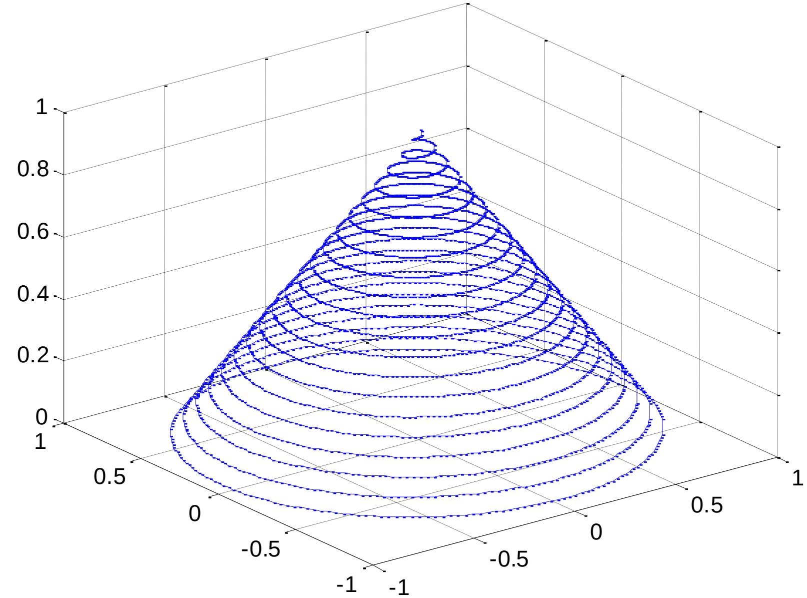


More 3D Line Plots

```
t = 0:pi/50:10*pi;
plot3(sin(t),cos(t),t)
grid on;  axis square;
```

```
turns = 40*pi;
t = linspace(0,turns,4000);
x = cos(t).*(turns-t)./turns;
y = sin(t).*(turns-t)./turns;
z = t./turns;
plot3(x,y,z);  grid on;
```





Principles for 3D Surface Plots

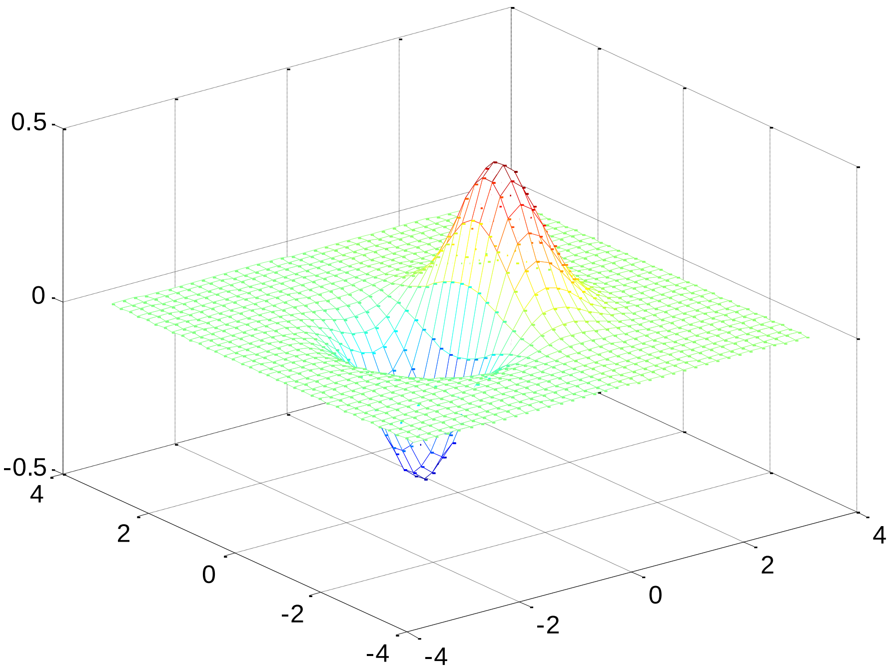
- Usually for plotting functions: $z = f(x, y)$
- Need to provide MATLAB a set of (x, y, z) points
- Use `meshgrid` to create matrices X and Y for a given range

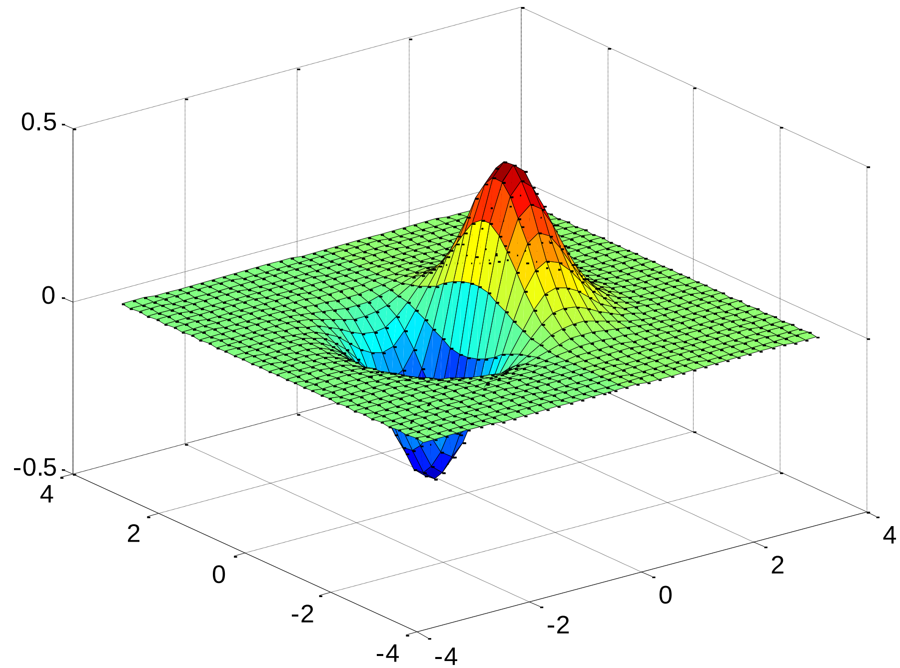
```
x = -2:1:2;
y = -2:1:2;
[X,Y] = meshgrid(x,y)
```

```
X =
    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2
    -2    -1     0     1     2
Y =
    -2    -2    -2    -2    -2
    -1    -1    -1    -1    -1
     0     0     0     0     0
     1     1     1     1     1
     2     2     2     2     2
```

Surface Plots: `mesh()` and `surf()`

```
x = -3.5:0.2:3.5;  y = -3.5:0.2:3.5;
[X,Y] = meshgrid(x,y);
Z = X.*exp(-X.^2-Y.^2);
subplot(1,2,1);  mesh(X,Y,Z);
subplot(1,2,2);  surf(X,Y,Z);
```

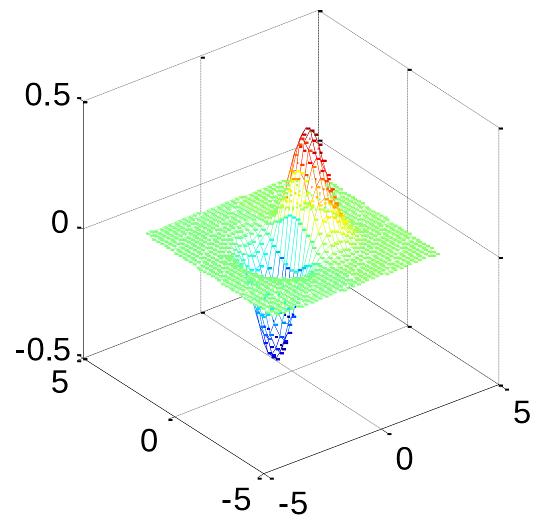


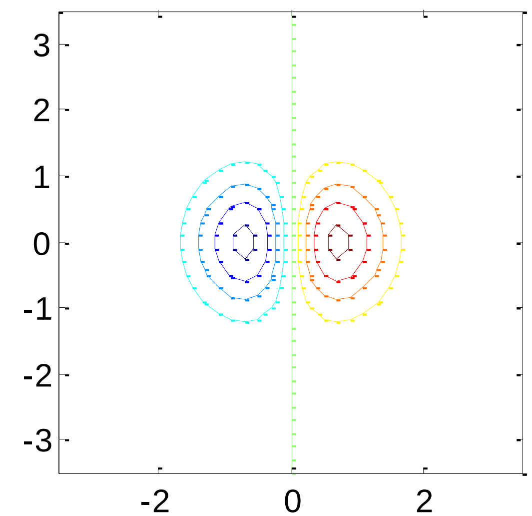


contour()

- Projection of equal heights of 3D plot onto a 2D plane

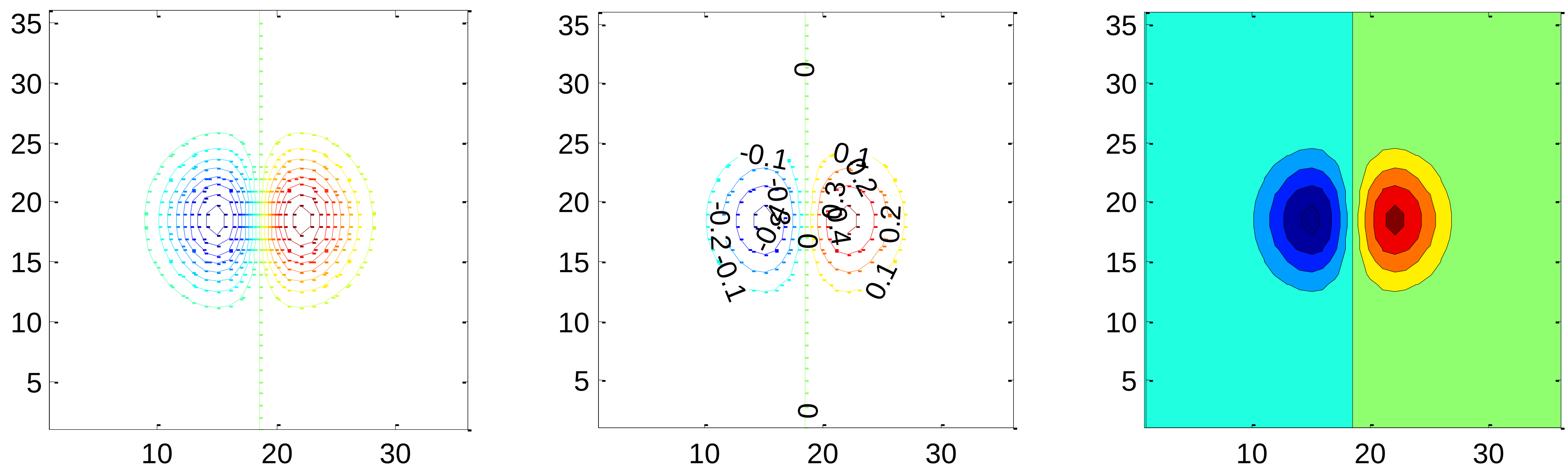
```
x = -3.5:0.2:3.5;
y = -3.5:0.2:3.5;
[X,Y] = meshgrid(x,y);
Z = X.*exp(-X.^2-Y.^2);
subplot(2,1,1);
mesh(X,Y,Z);
axis square;
subplot(2,1,2);
contour(X,Y,Z);
axis square;
```





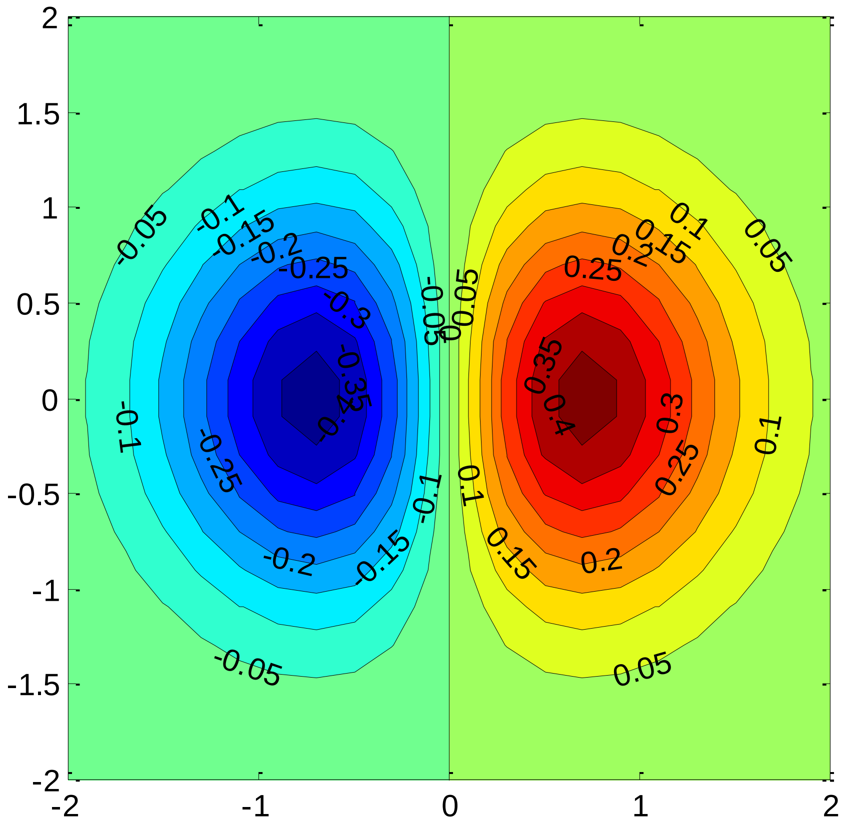
Various Contour Plots

```
x = -3.5:0.2:3.5;  y = -3.5:0.2:3.5;
[X,Y] = meshgrid(x,y);  Z = X.*exp(-X.^2-Y.^2);
subplot(1,3,1);  contour(Z,[-.45:.05:.45]);  axis square;
subplot(1,3,2);  [C,h] = contour(Z);
clabel(C,h);  axis square;
subplot(1,3,3);  contourf(Z);  axis square;
```



Exercise

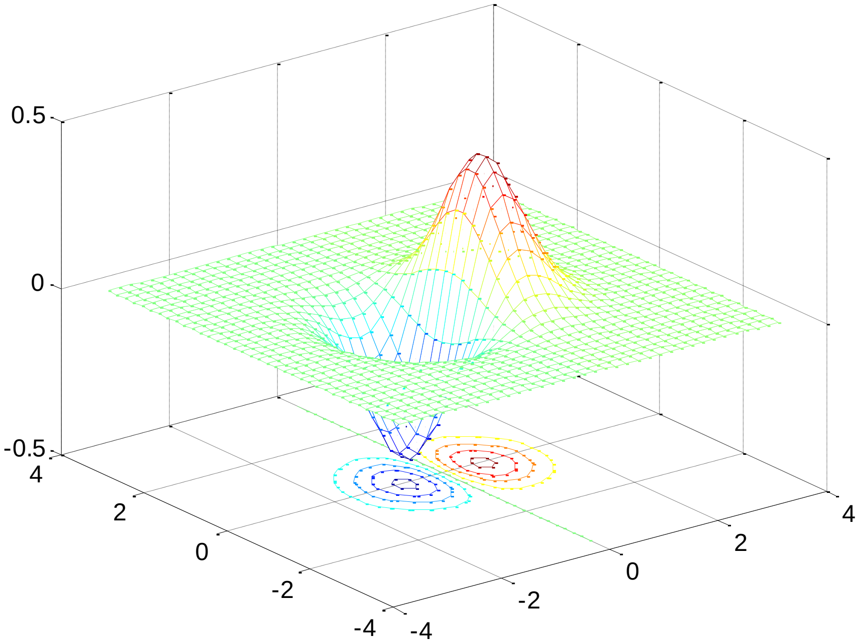
- Combine the contour techniques to generate a figure as shown below

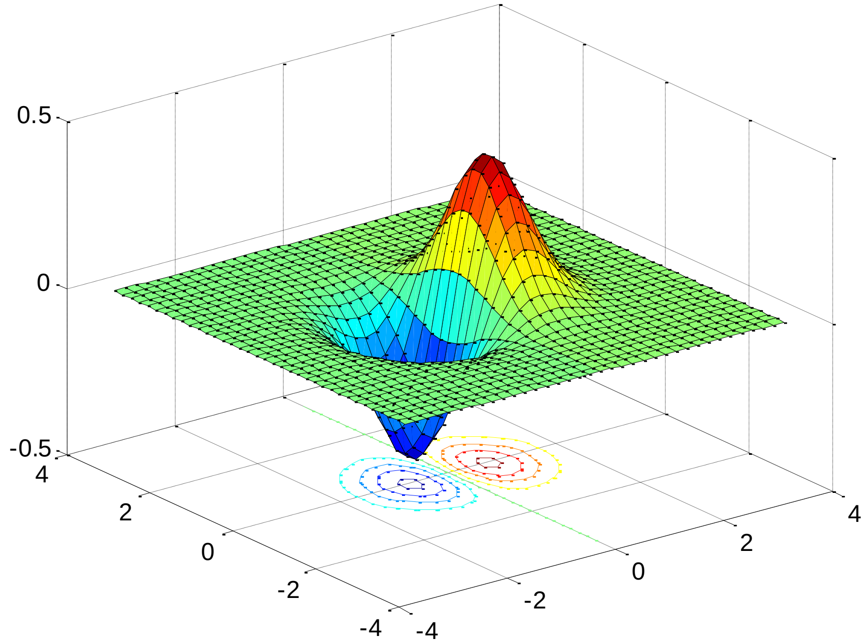


meshc() and surfc()

- Combination of surface/mesh and contours

```
x = -3.5:0.2:3.5;  y = -3.5:0.2:3.5;
[X,Y] = meshgrid(x,y);  Z = X.*exp(-X.^2-Y.^2);
subplot(1,2,1);  meshc(X,Y,Z);
subplot(1,2,2);  surfc(X,Y,Z);
```



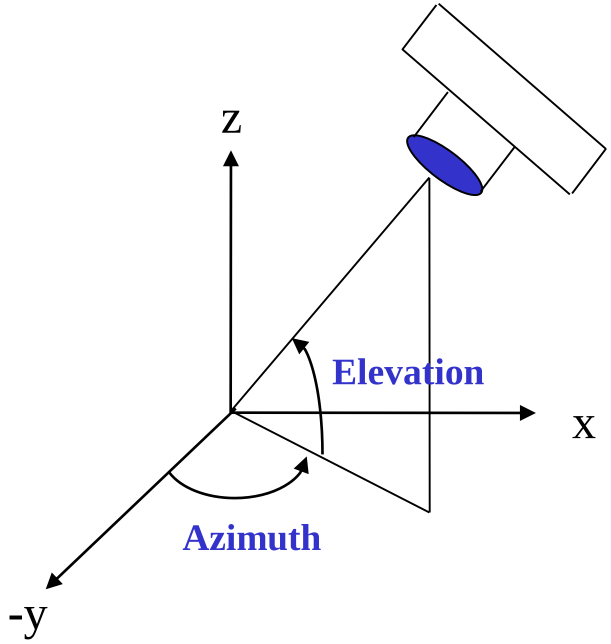


View Angle: `view()`

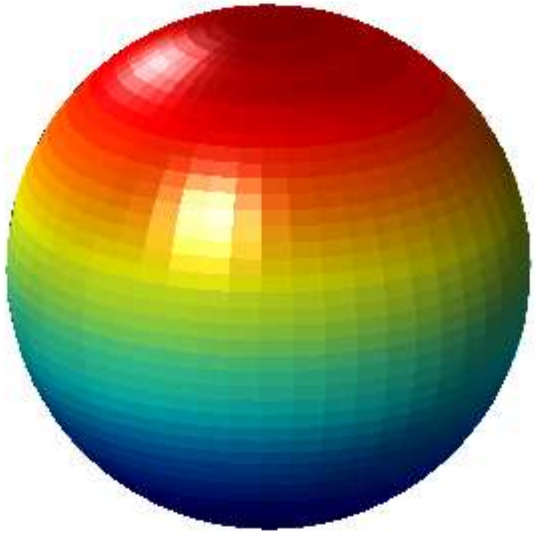
- Vary the view angle

```
view(-45,20);
```

in the script below




```
sphere(50);  shading flat;
light('Position',[1 3 2]);
light('Position',[-3 -1 3]);
material shiny;
axis vis3d off;
set(gcf,'Color',[1 1 1]);
view(-45,20);
```

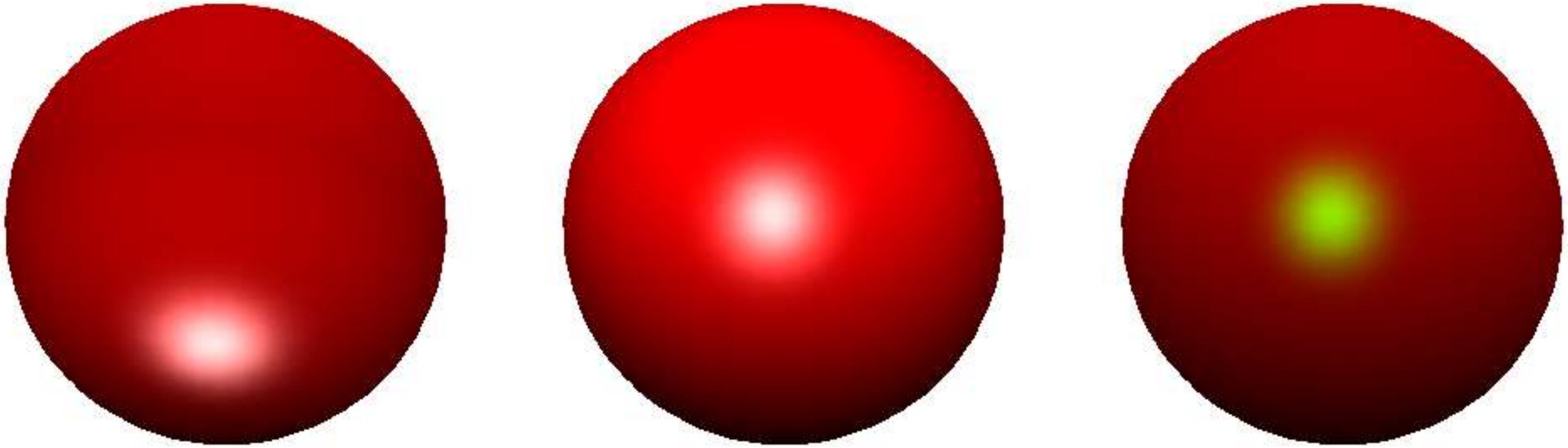


Light: `light()`

```
[X, Y, Z] = sphere(64);  h = surf(X, Y, Z);
axis square vis3d off;
reds = zeros(256, 3);  reds(:, 1) = (0:256.-1)/255;
colormap(reds);  shading interp;  lighting phong;
set(h, 'AmbientStrength', 0.75, 'DiffuseStrength', 0.5);
L1 = light('Position', [-1, -1, -1]);
```

```
set(L1, 'Position', [-1, -1, 1]);
```

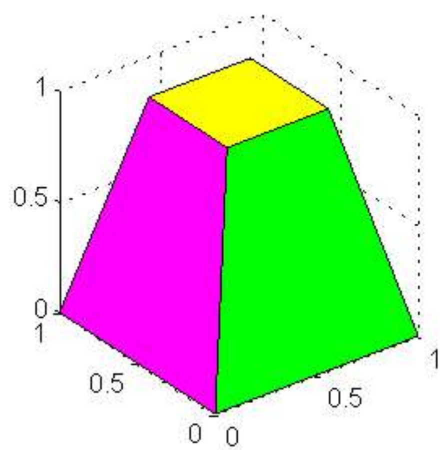
```
set(L1, 'Color', 'g');
```

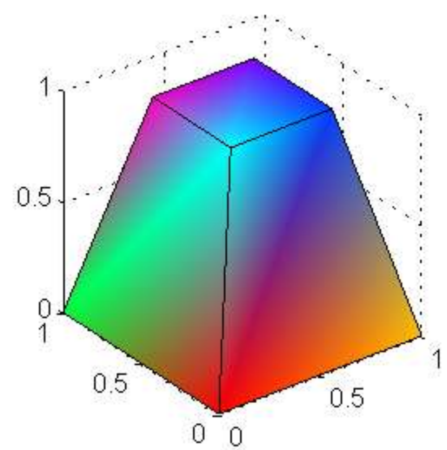


patch()

- A graphical object containing polygons

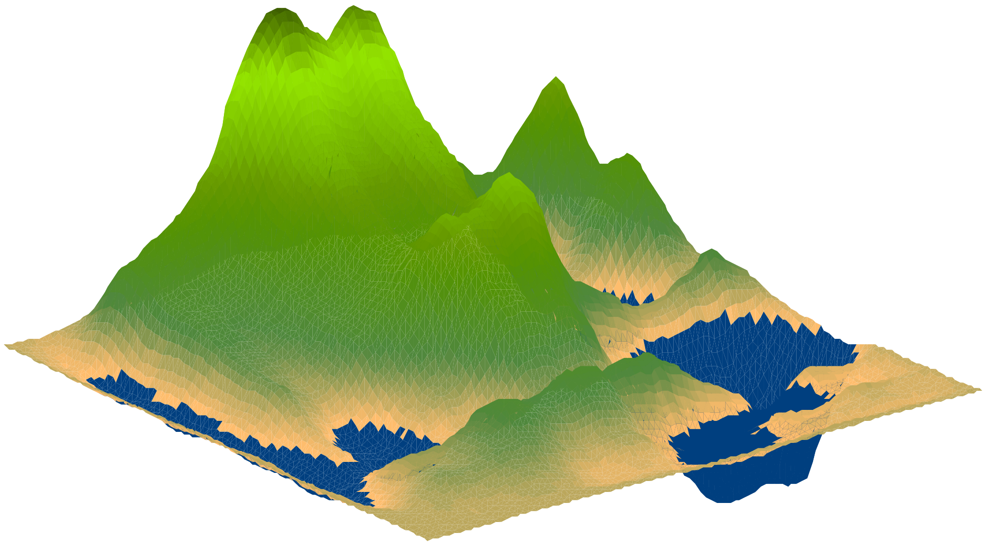
```
v = [0 0 0; 1 0 0 ; 1 1 0; 0 1 0; 0.25 0.25 1; ...
    0.75 0.25 1; 0.75 0.75 1; 0.25 0.75 1];
f = [1 2 3 4; 5 6 7 8; 1 2 6 5; 2 3 7 6; 3 4 8 7; 4 1 5 8];
subplot(1,2,1);  patch('Vertices', v, 'Faces', f, ...
    'FaceVertexCData', hsv(6), 'FaceColor', 'flat');
view(3);  axis square tight;  grid on;
subplot(1,2,2);  patch('Vertices', v, 'Faces', f, ...
    'FaceVertexCData', hsv(8), 'FaceColor', 'interp');
view(3);  axis square tight;  grid on;
```





Exercise

- MATLAB plots can be very professional!



```
load cape
X=conv2(ones(9,9)/81,cumsum(cumsum(randn(100,100)),2));
surf(X,'EdgeColor','none','EdgeLighting','Phong',...
'FaceColor','interp');
colormap(map); caxis([-10,300]);
grid off; axis off;
```

End of Class