```
clear;clc;close all;

% Just a quick sidenote...
% In MatLab we can have a 1-D array (singular value), 2-D (multiple
% values), and 3-D dimensional arrays (many 2-D arrays under a single
% variable) such as:

u = zeros(2,2,3);
u(:,:,2) = u(:,:,2) +1;
u(:,:,3) = 2*ones(2,2);

% What would 3-D arrays be used for? What if I had a problem that is time
% dependent and I need to store my solution at every time step. Well, I
% would store the values at every time-step in a 2-D array, but assign all
% the 2-D arrays to one variable by using the third dimension. Let's
% demonstrate this:

F = @(x,y,t) (x.^2)*t + y*t; % Why do you think *t?
x = [0:5]';
y = [0:5]';
t = [0:.25:.75]';

for i = 1:numel(t)
    f(:,:,i) = F(x,y,t(i));
end

% What if I had two functions, dependent on one another and wanted to plot
% them on the same graph, but with different y-axis scales?

F = @(x) x.^2;
x = [0:10]';
f1 = F(x);
f2 = f1.^2;

figure1 = figure;
plot(x,f1,'--k')
hold on
plot(x,f2,'-b')
hold off

% The above figure is difficult to decipher. So, I would need two y-axes. 1
% for the first function and 1 for the second function.

figure2 = figure;
plot(x,f1,'--k')
yyaxis right
```

```
hold on
plot(x,f2,'-b')
hold off

% A few spot the error problems

% x = linspace(0:.25:10);
% y = sind(x);
% figure = figure1;
% plot(x,y,'-b')
% title('sin(x)')
% xtitle('x')
% ytitle('sin(x)')
% xlimit([0 10])

% f = @(x) exp(x./5);
% x = [0:.25:2.5]';
% figure2 = figure;
% plot(x,f,'.-.0r') % 3 errors on this line alone
% hold on
% plot(x,x,'-k')
% hold off
% legend('\Omega','\tao','Location','Best')

% x = [0:10];
% y = exp(x);
% z = ln(y);
% figure3 = figure;
% log(x,y,z,'-b','-k','markersize',12) % 4 errors on this line alone
% hold on
% hold off
```